GDC

FinBench, It is time to evolve to its schema and approach to agent world

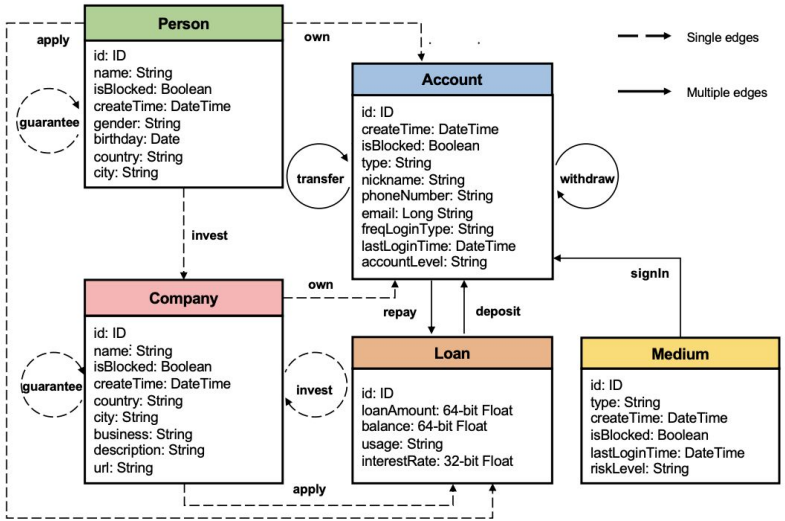
Yi Tae Jeong

XCENA **Field Application Engineer**

GUG **Organizer**

Dataset Schema

Surveyed on over 20 business clusters and take typical transaction scenario as design



Note: edge properties are omitted here.

5 vertex types and 9 edge types

- Entities as vertices while activities as edges
- Timestamp frequently used
- Various subtypes of same entities
- Rich properties
- Edge multiplicity

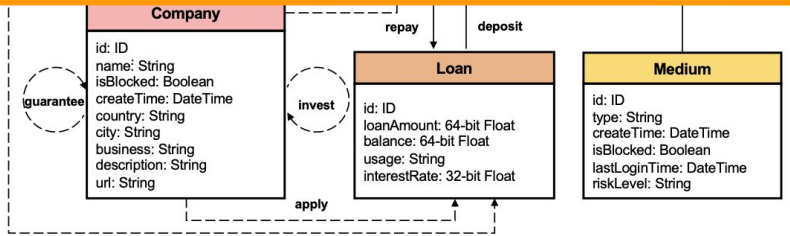
FinBench Recap

VLDB 2025 WELCOME TO LONDON

Dataset Schema

Surveyed on over 20 business clusters and take typical transaction scenario as design

The primary consumer of data is shifting from humans to AI agents.



- Various subtypes of same entities
- Rich properties
- Edge multiplicity

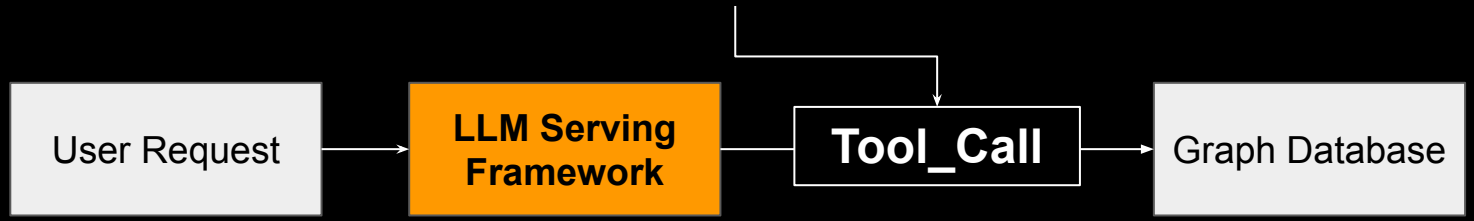
Note: edge properties are omitted here.

LDBC

The LDBC Financial Benchmark: Transaction Workload, VLDB2025, Shipeng Qi AntGroup

1.Query Routing & QueryIntent

2.Query validation & plan gate

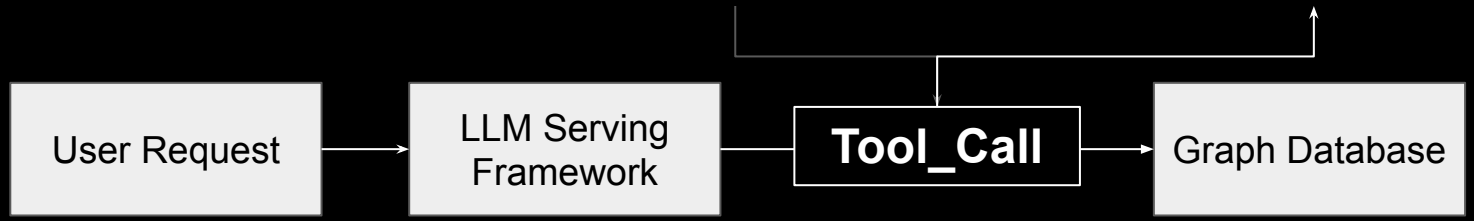


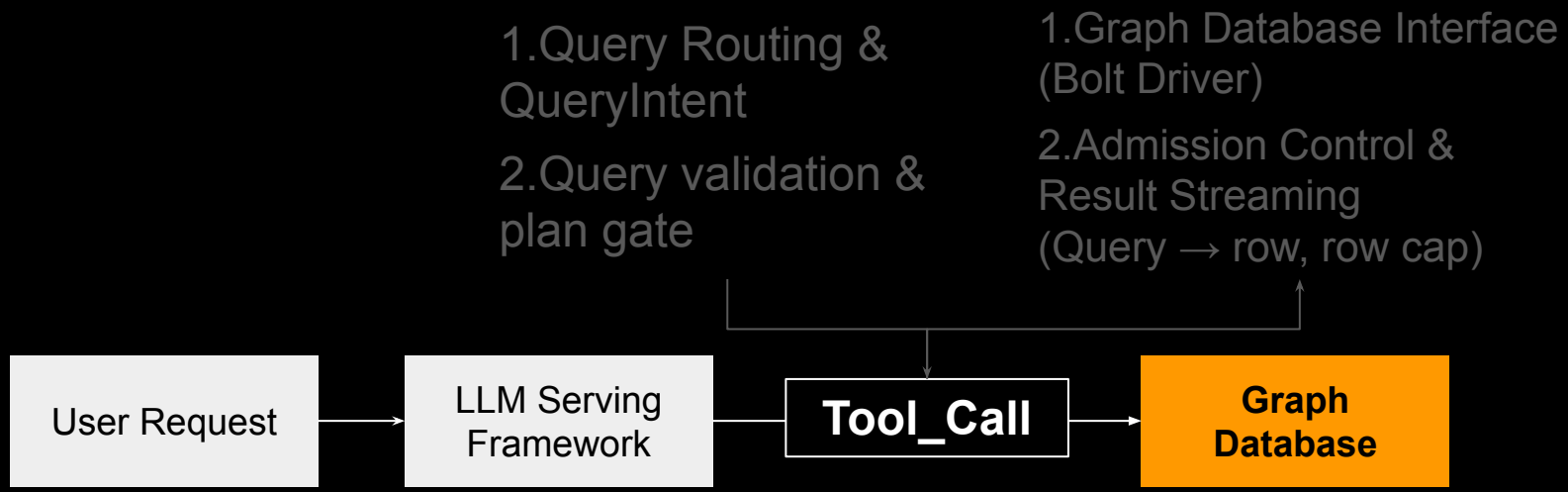
1.Query Routing & QueryIntent

2.Query validation & plan gate

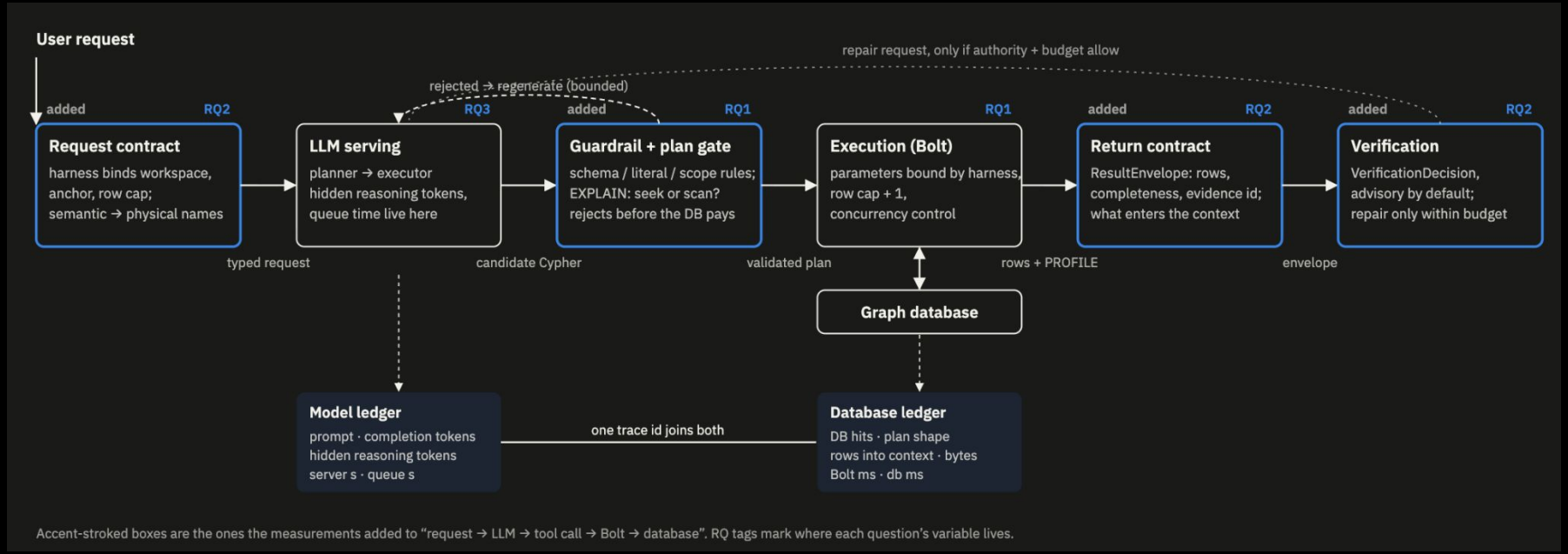
1.Graph Database Interface (Bolt Driver)

2.Admission Control & Result Streaming (Query → row, row cap)




1.Query Routing & QueryIntent
2.Query validation & plan gate
1.Graph Database Interface (Bolt Driver)
2.Admission Control & Result Streaming (Query → row, row cap)
User Request
LLM Serving Framework
Tool_Call
Graph Database

The Interface Between Agents and Graph Databases: Does the Interface Really Affect Graph Agentic RAG?



RQ1.
Data Scalability

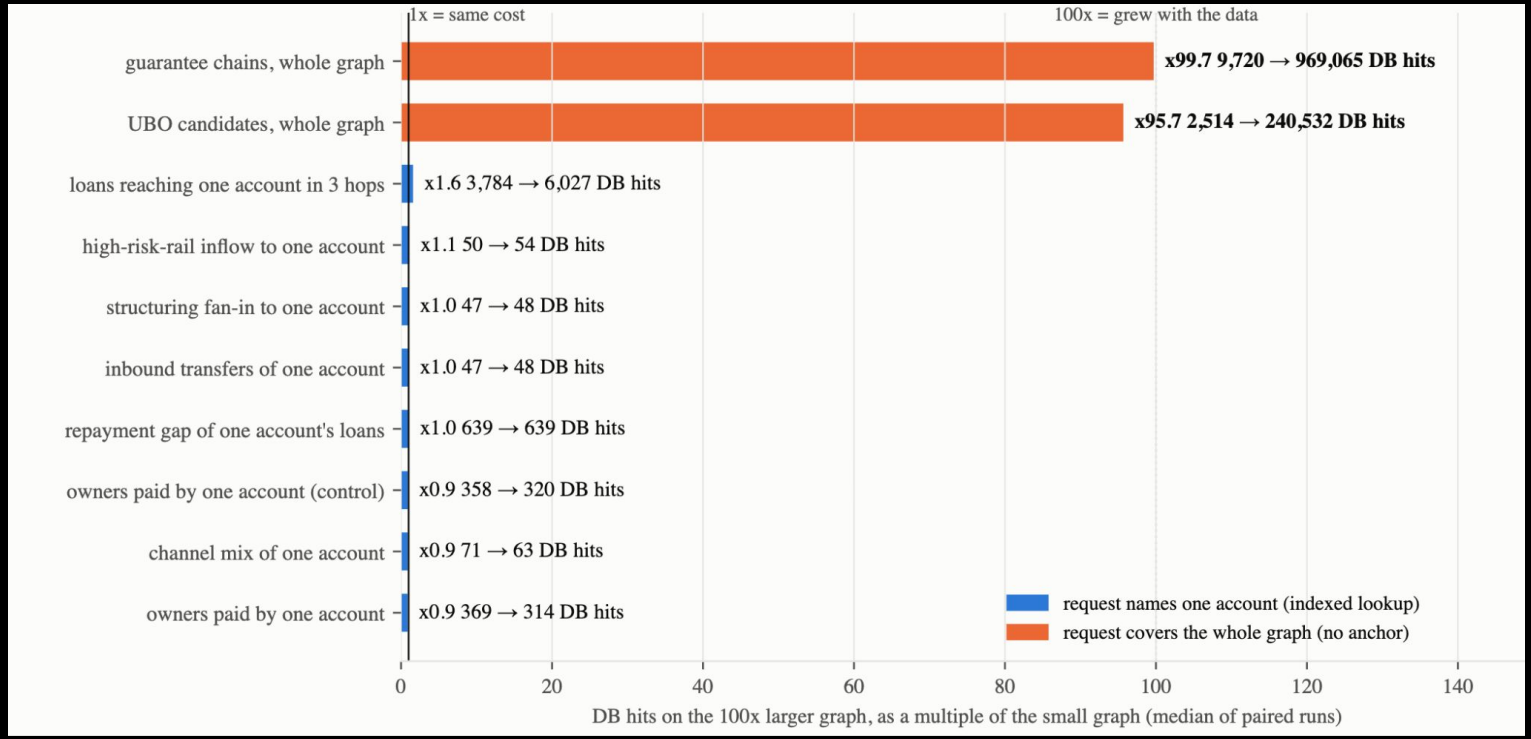
RQ2.
Schema-Contract Intelligence

RQ3.
Agent Capacity

RQ1 · data scalability

Does a 100x larger graph make the agent's queries 100x dearer?

Same request, same agent, same repeat; SF1 (4k nodes, 27k edges) versus SF100 (360k nodes, 2.6M edges). Cost = PROFILE DB hits of the executed Cypher.



All LLM tokens are supported by MARA Cloud.

RQ1 · data scalability

Does a 100x larger graph make the agent's queries 100x dearer?

Same request, same agent, same repeat; SF1 (4k nodes, 27k edges) versus SF100 (360k nodes, 2.6M edges). Cost = PROFILE DB hits of the executed Cypher.

CORRECT **Anchored request, SF1**

gemma-4-31B-it · effort default · SF1 · direct single

REQUEST IN	For account number 108: how many incoming transfers are there and what is their total amount?
EXECUTOR → CYPHER	`MATCH (:Account {_workspace_id: $workspace_id})-[t:TRANSFER]->(a:Account {_workspace_id: $workspace_id, acct_no: $acct_no}) RETURN count(t) AS incoming_count, sum(t.amount) AS total_amount LIMIT $limit`
HARNESS-BOUND PARAMETERS	{"acct_no": 108, "limit": 50}
DATABASE OUT	1 row(s) · 47 DB hits · 8.075 ms
GOLD	{"n": 5, "total": 163488}

1 model call(s) · 871 prompt · 72 completion (0 hidden reasoning) · server 1.3s · queue+network 0.7s

Index seek on one account: 47 DB hits.

CORRECT **Same request, SF100**

gemma-4-31B-it · effort default · SF100 · direct single

REQUEST IN	For account number 7: how many incoming transfers are there and what is their total amount?
EXECUTOR → CYPHER	`MATCH (:Account {_workspace_id: $workspace_id, acct_no: $acct_no})<-[r:TRANSFER]-(:Account {_workspace_id: $workspace_id}) RETURN count(r) AS incoming_count, sum(r.amount) AS total_amount LIMIT $limit`
HARNESS-BOUND PARAMETERS	{"acct_no": 7, "limit": 50}
DATABASE OUT	1 row(s) · 48 DB hits · 8.327 ms
GOLD	{"n": 6, "total": 108054}

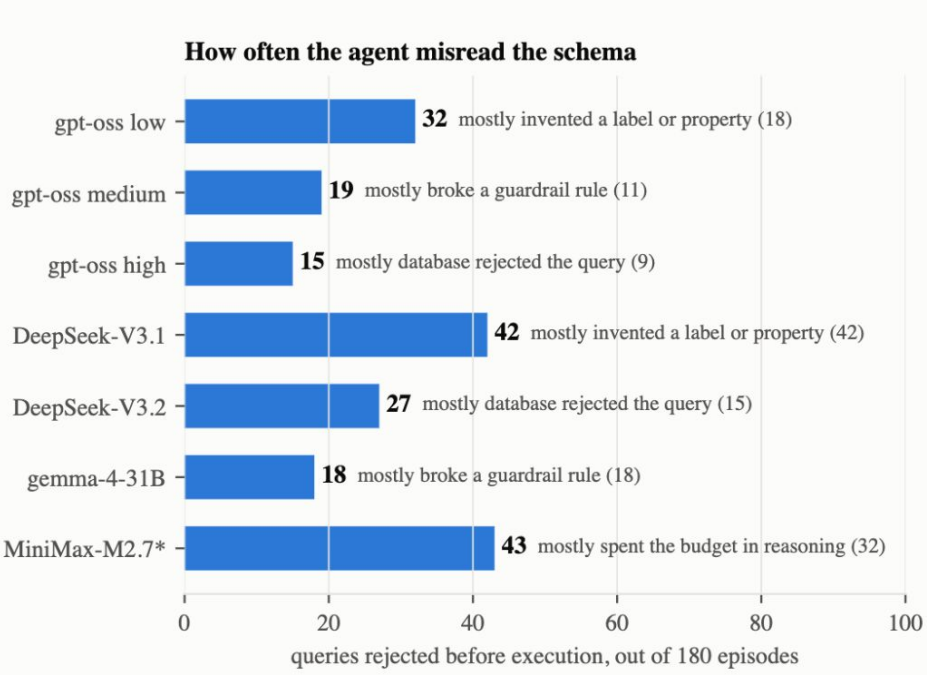
1 model call(s) · 869 prompt · 72 completion (0 hidden reasoning) · server 1.3s · queue+network 2.6s

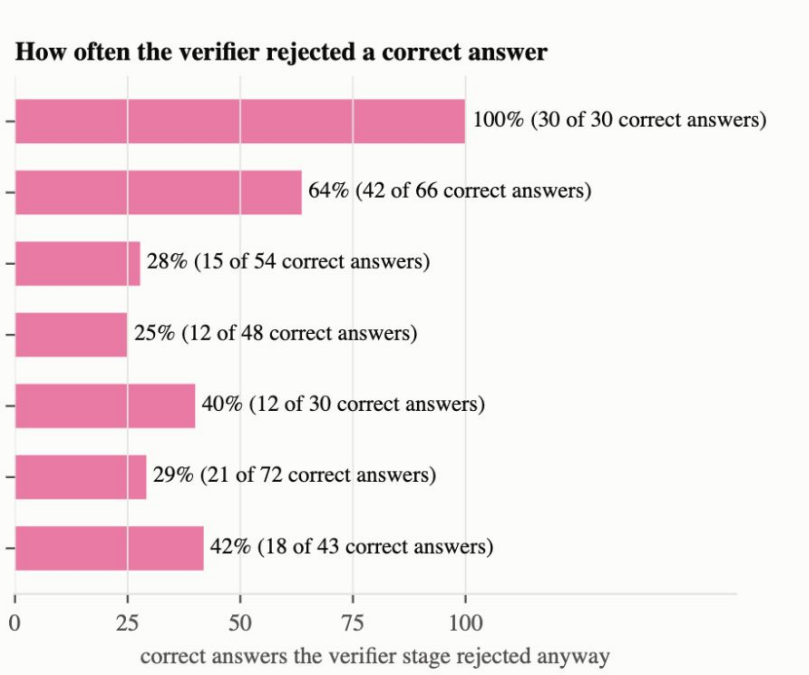
The graph is 100x larger; the seek still touches 48 records.

RQ2 · schema-contract intelligence

When the agent reads our schema, what does it get wrong — and does the contract catch it?

Same schema description for every model; the contract around it: typed Query Intent handoff, harness-owned ExecutionRequest, result envelope, an advisory verifier. Measured: rejected generations by cause, and how often the verifier rejects a correct answer.





All LLM tokens are supported by MARA Cloud.

RQ2 · schema-contract intelligence

When the agent reads our schema, what does it get wrong — and does the contract catch it?

Same schema description for every model; the contract around it: typed Query Intent handoff, harness-owned ExecutionRequest, result envelope, an advisory verifier. Measured: rejected generations by cause, and how often the verifier rejects a correct answer.

REJECTED **Guardrail rule broken**

gpt-oss-120b · effort low · SF1 · multi typed

REQUEST IN	How many persons stand behind company-owned accounts through an investment into the owning company?
PLANNER → QUERYINTENT (EXCERPT)	{ "entities": [{ "name": "Person", "type": "node", "attributes": ["person_id", "name"] }, { "name": "Company", "type": "node", "attributes": ["company_id", "name"] }, { "name": "Account", "…
EXECUTOR → CYPHER	(rejected before execution)
GUARDRAIL / ENDPOINT	ValueError: text2cypher rejected: unknown_properties:owner_type,inlined_literal:= 'Company'

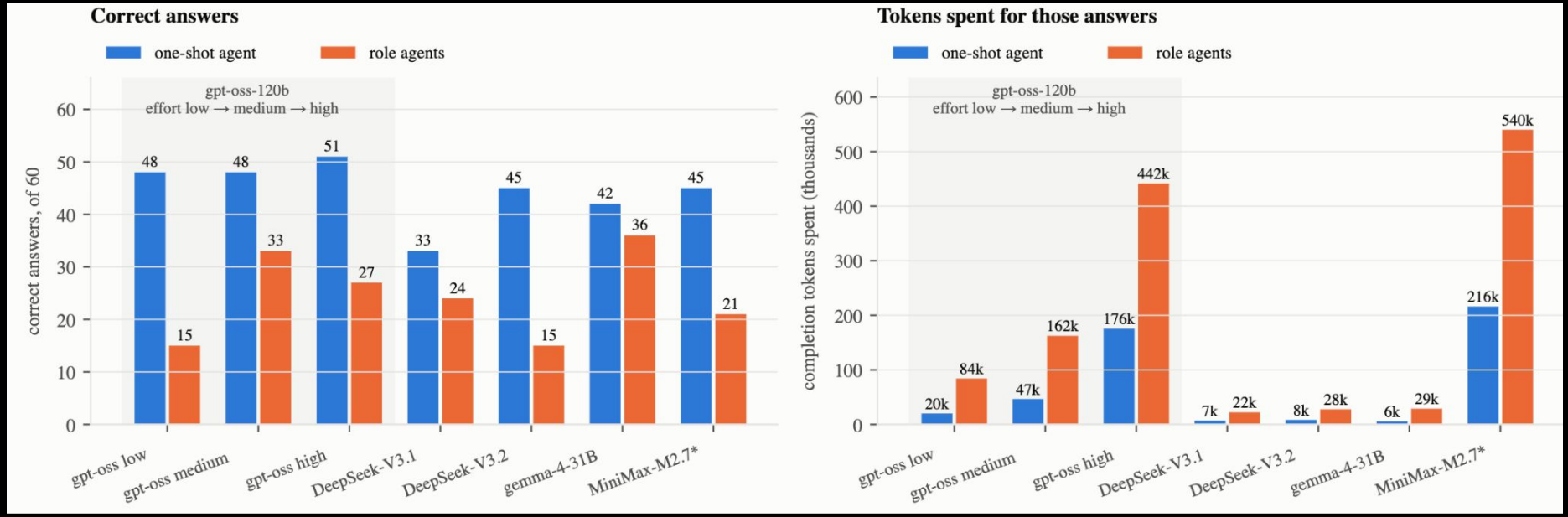
3 model call(s) · 2,888 prompt · 1,298 completion (523 hidden reasoning) · server 4.0s · queue+network 2.2s

At low effort the same request produced an inlined literal (= 'Company') and an unknown property: two contract rules the model was told and did not apply.

RQ3 · agent capacity

Does spending more model — more reasoning, a different model — buy more correct answers on the same schema?

reasoning_effort low / medium / high on gpt-oss-120b (the one model where the parameter changes behaviour), and four other models at their default. Measured: correct answers of 60 against completion tokens spent.



Conclusion

Where agents do well is where they cost the most

1. Cost attribution shifts from the data layer to the decision layer.
Cost used to track the scale factor of the data. With an agent in the loop, that link breaks — the agent decides how far to traverse, how many queries to issue, and how long to reason. Cost is now attributable to those decisions, which is exactly where we have to measure and control it.

2. Humans need explicit criteria to step in and verify.
We must be able to judge whether the agent's pipeline is actually working: how it interprets the natural-language question, and what information it uses to retrieve from the graph. The blurrier that threshold, the higher the operational cost.

3. Let go of the belief that more reasoning gets you closer to the right answer.
What matters is a well-defined schema, plus a clear signal to the agent of how relevant that schema is to the user's query. That relevance signal is the essential guardrail for a graph-based agent.

jeongiitae6@gmail.com

github.com/tteon/AIsummit26