

Repository-Level Vulnerability Detection via Graph-Based Compression and Assumption Validation

Thi-Hong-Cuc Le^{1,2} Hoang-Quoc-Bao Hua^{1,2} Xuan-Bach Le^{1,2}

¹ Faculty of Computer Science and Engineering, Ho Chi Minh City University of Technology (HCMUT) ² Vietnam National University Ho Chi Minh City, Vietnam

An LLM can **propose** a vulnerability in code. GraphLedger **reports it only** when repository evidence supports the source-sink path and every assumption behind it.

0.23

FDR

44% lower vs VulAgent

0.76

F1

best of six methods

71%

fewer tokens

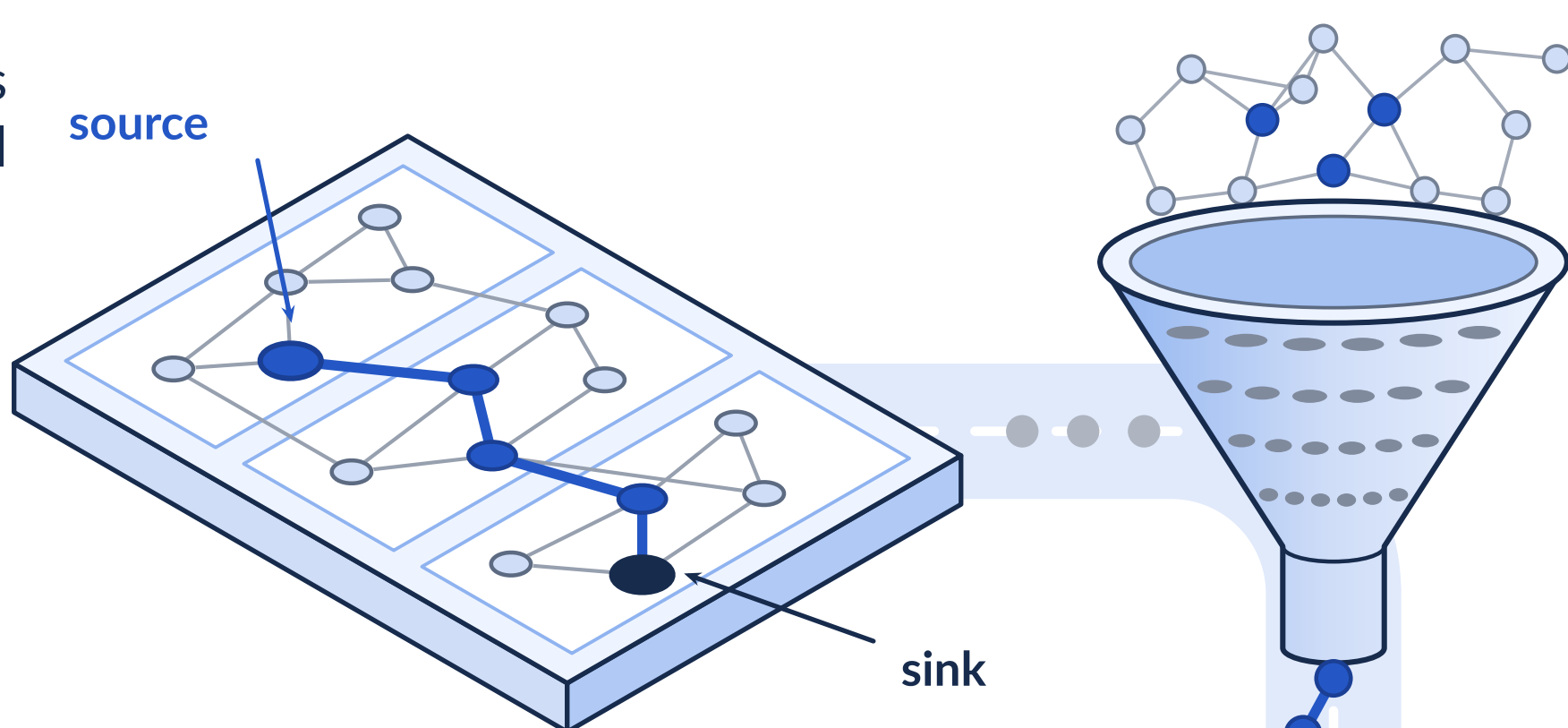
90% dependency edges kept

How GraphLedger turns a claim into a verdict

1 Repository graph

A code property graph spans the repository: syntax, control and data flow, **across files**.

The vulnerable path starts in one file and ends in another.



2 AACC compression

Adaptive Assumption-Corroborating Compression

AACC keeps the source-sink corridor and the key evidence nodes needed by the ledger.

71% graph tokens removed

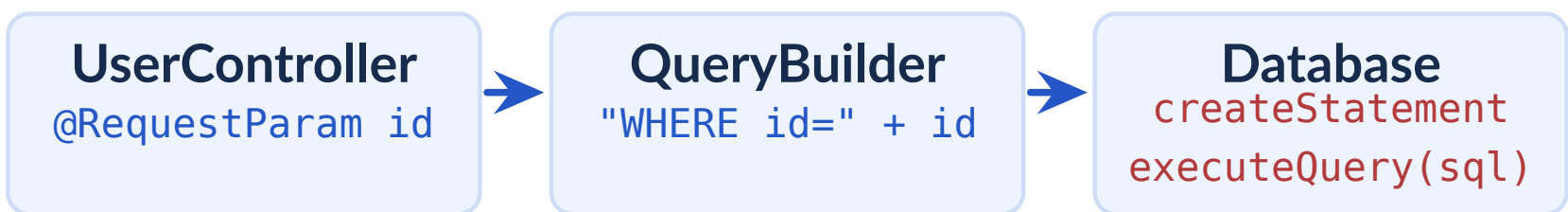
90% dependency edges kept

88% taint paths kept

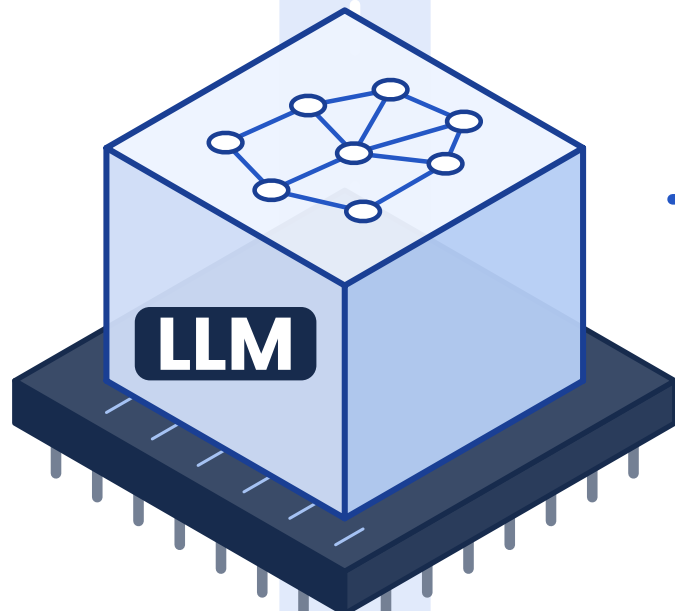
Measured over all 120 repositories; reduction uses serialised graph tokens.

3 Hypothesis generation

The LLM reads the compressed graph G' and states a **claim plus the assumptions** it depends on — not a bare yes/no.



repository evidence from G'



Claim: this path is SQL-injectable

CWE-89 · holds only if all three hold

A1 input is user-controlled

A2 no sanitiser on the path

A3 sink is not parameterised

4 Assumption Ledger

Assumption	Repository evidence	State
A1 user-controlled	@RequestParam id	✓ valid
A2 no sanitiser	flow, no sanitiser marker	✓ valid
A3 raw SQL sink	createStatement → executeQuery	✓ valid

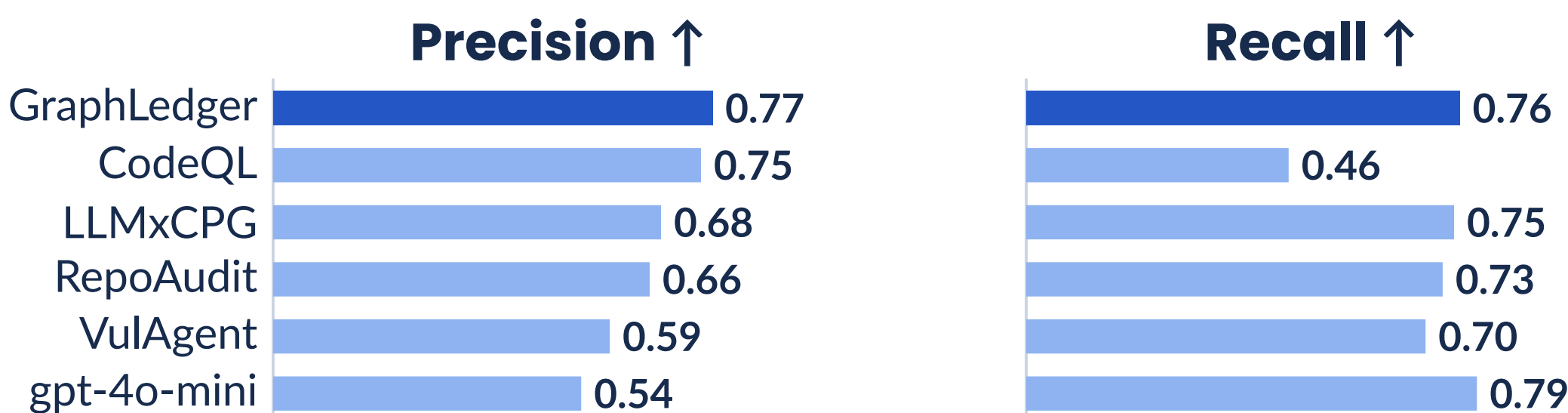
Monotonic rejection: once an assumption is contradicted, the claim cannot return to Verified.

5 Verdict

example above: 3/3 valid → Verified	
VERIFIED	witness path found and every assumption valid
REJECTED	any assumption contradicted by evidence
INCONCLUSIVE	unresolved within the bound → analyst review

Results on CWE-Bench-Java

Repository-disjoint 100-repository partition, $n = 200$ paired instances; same ground truth and scoring for every method; Inconclusive counted as negative.



Highest precision of the six; recall is second only to gpt-4o-mini, which has the lowest precision.

Ablation: each half needs the other

Configuration	F1 ↑	FDR ↓
Neither	0.66	0.40
Compression only	0.67	0.33
Ledger only	0.71	0.30
Full GraphLedger	0.76	0.23

Compression improves the evidence going in; the ledger filters what comes out.

Why it matters

- Verified findings carry auditable evidence
- Uncertainty is explicit, not a hidden label
- Fewer false alarms reach the triage queue

Scope & limits

- Taint-style source-sink flaws; not logic or config bugs
- Bounded search: $\ell = 20$ path length, 8 paths per pair
- Deep, framework-heavy paths may stay Inconclusive

Artifact

- Code + prompts
- Data splits
- Predictions
- Ledger logs



SCAN TO OPEN