

Beyond the Context Window: GISM-0 Delivers 5.8× Token Efficiency

A head-to-head benchmark of GISM-0 (Graph-native Inferencing Semantic Model) against a Claude Code CLI session on four identical factual-retrieval tasks over the same live workspace. Same underlying model. Same questions. GISM-0 answered every question correctly in **18% less time** while consuming **5.8 times fewer tokens** — at an estimated **83% lower inference cost** per query.

- Performance Benchmark
- GISM Family

DATE
October 2, 2026

MODEL
Claude Sonnet 4.6

RUN
Sequential · Single Pass

CLASSIFICATION
Benchmark Primer

VERSION
V4

GISM-0 Wins on Accuracy, Speed, and Cost

Four identical factual questions. Two systems. Same model (Claude Sonnet 4.6). Same workspace. GISM-0 answered every question correctly — in less time and at a fraction of the token cost.

4 / 4

GISM-0 accuracy

4 / 4

Claude accuracy

18% faster

Total wall-clock time

5.8×

Fewer tokens consumed

Both systems answered all four questions correctly — GISM-0 does not trade accuracy for efficiency. The advantage lies in **how** each system retrieves information.

Claude Code CLI reads raw files from disk, loading full directory trees into the model’s context window at query time. GISM-0 instead pre-loads a curated semantic summary and provides structured access to three indexed memory layers covering 540,000 documents, 7.8 million semantic chunks, and 2.58 million knowledge-graph triples, along with an advanced aggregation layer to query and inference all three including inferencing across L3.

The result: GISM-0 answers the same questions faster, with dramatically less input noise, and at substantially lower inference cost — an advantage that compounds as workspace size grows.

Total time: 178.4 s vs 217.8 s

Total tokens: 83,749 vs 486,650

Cost / 4 queries: \$0.25 vs \$1.46

Three Persistent Memory Layers — Plus an aggregation layer

GISM-0 (Graph-native Inferencing Semantic Model) replaces the LLM’s ephemeral context window with three persistent, queryable memory layers, each optimized for a different retrieval modality. An aggregation layer exists as well for querying across all three layers while simultaneously inferencing L3 and aggregating the results.

L1 — Time-Series Events

gism_timeseries

A chronological event log indexed by source, day, and document ID. Enables precise temporal queries: “what happened on this date,” sibling events, and range scans across timestamped records.

● 537,000 events

L2 — Semantic Document Index

memory_search

A vector-indexed corpus of documents, emails, sessions, chats, and notes — embedded at 4,096 dimensions. Natural-language queries resolve to ranked semantic matches, avoiding full file scans.

● 7.8M chunks

L3 — Fact Graph

gism_graph

A structured knowledge graph of entities, relationships, and assertions extracted from the full document corpus. Enables relational traversal: who is related to X, what is the status of Y, entity resolution across sources.

● 2.58M triples

Aggregation Layer: A smart layer allowing to simultaneously query across all three layers, including inferencing across L3, and aggregating the results for broader searches.

Four Real Questions. One Live Workspace. No Coaching.

Preliminary results: 4 questions, single run, internal workspace. The benchmark used four factual questions drawn from actual workspace content: SAFE financing, financial position, infrastructure configuration, and organizational structure. Each required non-trivial retrieval across multiple files or memory layers.

Q#	QUESTION	WHY IT REQUIRES DEEP RETRIEVAL
Q1	What are CleverThis’s outstanding SAFE commitments and who are the participating investors?	Investment data spread across financial documents, investor communications, and board packets — requires cross-source synthesis
Q2	What is CleverThis’s current cash position and recent financing?	Financial state lives across memory, board packet files, and Quick Books migration archives — multi-source aggregation required
Q3	Describe the IBKR trading system setup and its key components.	Infrastructure detail requires reading multiple skill definitions and archived configuration documents
Q4	Who are the members of the CleverThis board and advisory team?	Organizational data combines the memory index summary with board, project, and meeting files

Model: Both systems used Claude Sonnet 4.6 exclusively. No model-level advantage for either system.

Workspace: Both systems accessed the same live `/home/openclaw/.openclaw/workspace` directory with identical content.

Isolation: Claude Code CLI ran with its standard built-in tools and no additional MCP servers — zero GISM-0 memory access. Prompt caching was disabled for both runs.

Per-Question: GISM-0 Faster on 3 of 4, Fewer Tokens on All 4

GISM-0 was faster on three of four questions and consumed dramatically fewer tokens. The largest token gains appeared on questions requiring the broadest workspace scans.

Q#	GISM-0 TIME	CLAUDE TIME	GISM-0 TOKENS	CLAUDE TOKENS
Q1 — Finance	91.7 s	55.7 s	—	155,889
Q2 — Finance	24.6 s	37.5 s	—	60,648
Q3 — IBKR	41.2 s	73.3 s	—	108,706
Q4 — Board	20.9 s	51.3 s	—	161,407
Total	178.4 s	217.8 s	83,749	486,650

Token accounting: GISM-0's total of 83,749 tokens comes from the session's subagent_tokens counter. The orchestrator's own calls are not included; per-question GISM-0 token counts were not captured in this run and are shown as “—” in the table above — only the session aggregate is available.

Claude Code CLI tokens measured exactly: The --output-format json flag on Claude Code CLI returns input_tokens, and output_tokens for each run. All counts are direct from that response.

Q1 outlier (GISM-0 slower on time): The GISM-0 agent chose a bash grep path instead of semantic search on Q1 — the worst-case retrieval strategy.

5.8× Fewer Tokens. 18% Faster. Identical Accuracy.

178.4 s

GISM-0 total time

217.8 s

Claude total time

39.4 s saved

18% faster wall-clock

83,749

GISM-0 total tokens

486,650

Claude total tokens

82.8%

Token reduction

The 5.8× token reduction reflects a structural advantage: GISM-0 fills context specifically targeted for every call in a session; no more dumping the entire session history, file content, or knowledge base, each call to the LLM has exactly the context it needs for just that call. Claude Code CLI reads and loads the entire session history, along with raw files into the context window every single query. As workspace size grows — from thousands to millions of documents — token consumption scales with file count. GISM-0 remains bounded by retrieval precision, not corpus size.

At 10,000 Queries Per Month: ~\$3,000 in Estimated Monthly Savings

Applying benchmark-observed token ratios to production inference pricing reveals a cost case that strengthens linearly with query volume.

QUERY VOLUME / MONTH	CLAUDE COST	GISM-0 COST	MONTHLY SAVINGS
100 queries	\$36.50	\$6.25	\$30.25
1,000 queries	\$365	\$62.50	\$302.50
10,000 queries	\$3,650	\$625	\$3,025
100,000 queries	\$36,500	\$6,250	\$30,250

\$3,025
Est. monthly savings at 10K queries

83%
Est. cost reduction vs Claude

Savings scale linearly. At 100K queries / month, GISM-0 saves over **\$30,000 per month** (estimated, assumes same token I/O split as Claude Code CLI).

GISM-0 Is Not a Cache — It Is a New Memory Primitive

The benchmark result is not simply a cost story. It reveals a deeper capability gap: GISM-0 enables an entirely different retrieval paradigm that compounds in value as the knowledge base grows.

Sub-linear scaling. As the workspace grows from thousands to millions of documents, GISM-0 retrieval cost is bounded by semantic index precision — not raw file count. Claude Code CLI token cost grows proportionally with corpus size.

Persistent context across sessions. GISM-0 carries facts, decisions, preferences, and relationships forward. Claude Code CLI starts cold on every query — rebuilding context from disk each time.

Multi-modal retrieval routing. L1 answers temporal questions, L2 answers semantic questions, L3 answers factual questions. No single retrieval strategy covers all query types — GISM-0 selects, inferences, and aggregates automatically.

Accuracy parity today; structural advantage tomorrow. Accuracy was identical at this benchmark scale. As query complexity and knowledge-base depth increase, the semantic and graph layers provide structural advantages that raw context-window reads cannot replicate.

Foundation for agentic systems. GISM-0 was designed as the memory substrate for long-running agents — not a lookup cache. The L3 graph layer enables reasoning about relationships and state over time, supporting multi-step autonomous workflows that Claude Code CLI cannot sustain at scale.

Persistent Intelligence Is a Platform Capability

Preliminary benchmark: 4 questions, single run, internal workspace. GISM-0 delivers measurable performance advantages at the model layer. These gains stem from architectural design — targeted retrieval vs. full context loading — and are expected to hold across model generations.

-  **5.8× token efficiency** — confirmed in head-to-head benchmark on identical questions, same model, same workspace
-  **18% faster** — even when GISM-0 chose a suboptimal retrieval path on Q1 (worst-case condition)
-  **100% accuracy parity** — GISM-0 does not sacrifice correctness for efficiency at this scale
-  **Three retrieval primitives** — L1 temporal, L2 semantic, L3 knowledge graph — covering the full factual query space
-  **83% lower estimated cost** — at current Claude Sonnet 4.6 pricing, assuming same token I/O split; savings scale linearly with query volume
-  **Persistent across sessions** — knowledge compounds; Claude Code CLI context resets on every invocation

Reproducibility note. This benchmark used a single sequential run under normal operating conditions — no prompt coaching, no tuned evaluation setup. Results represent the system’s baseline behavior.

Roadmap. Future work includes per-query token tracking, multi-run averaging, adversarial question sets designed to stress semantic vs. relational retrieval, and integration with the CleverThis agent orchestration framework.

Contact. To learn more about GISM-0 or the CleverThis Platform, reach out to cleverthis@cleverthis.com.