

GISM-0: Beyond Context Windows

A Performance and Cost Benchmark of The GISM-0

Jeffrey Phillips Freeman

October 5, 2026

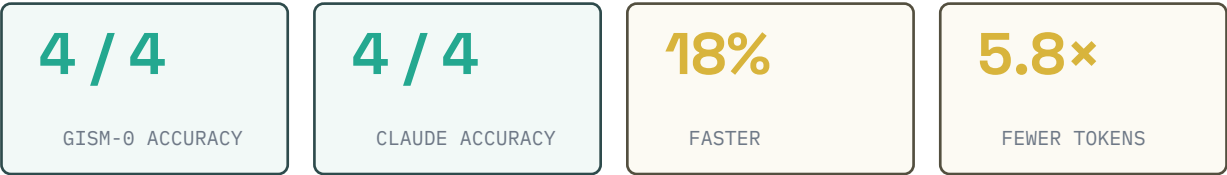
ABSTRACT

This report presents a direct benchmark of **GISM-0 (Graph-native Inferencing Semantic Model)** — the persistent inferenced-memory architecture powering the CleverThis platform — against a Claude Code CLI session on the same live workspace. Using the same underlying model (Claude Sonnet 4.6), both systems answered four factual questions requiring non-trivial multi-file retrieval. GISM-0 answered all four correctly in 18% less time while consuming **5.8× fewer tokens** — at an estimated **83% lower inference cost** per query. These results establish GISM-0 as a production-viable inferenced-memory substrate that reduces operational cost without sacrificing accuracy.

DATE	MODEL	METHODOLOGY	CLASSIFICATION
October 2, 2026	Claude Sonnet 4.6	Sequential · Single Run	Technical Whitepaper

1 Executive Summary

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



Both systems answered all four questions correctly (4/4), confirming accuracy parity. GISM-0 completed the sequence in 178.4 seconds versus 217.8 seconds for Claude Code CLI — an 18% wall-clock improvement. More significantly, GISM-0 consumed 83,749 tokens versus 486,650 for Claude Code CLI — a **5.8× token reduction**. At current Claude Sonnet 4.6 pricing, this represents a **83% cost reduction** per equivalent workload.

2 GISM-0 Architecture

GISM-0 (Graph-native Inferencing Semantic Model) replaces the LLM’s ephemeral context window with three persistent, independently queryable memory layers. As well as an aggregate query that automatically queries and merges all three layers.

2.1 L1: Time-Series Events

The L1 layer is a chronological event log indexed by source, day, and document ID, queried via the `gism_timeseries` tool. It supports temporal queries — “what happened on this date,” sibling event lookups, and date-range scans. At benchmark time: **537,000 timestamped events** drawn from emails, sessions, meeting transcripts, and system logs dating back over 15 years.

2.2 L2: Semantic Document Index

The L2 layer is a vector-indexed corpus queried via `memory_search`. Documents from the full workspace are embedded at **4,096 dimensions**. At benchmark time: **540,000 documents** across **7.8 million semantic chunks**. Natural-language queries resolve to ranked matches without full directory scans or file loading.

2.3 L3: Fact Graph

The L3 layer is a structured knowledge graph of entities, relationships, and assertions, queried via `gism_graph`. At benchmark time: **2.58 million triples**. Enables relational traversal — entity resolution, relationship queries, state lookups — that flat document retrieval cannot provide.

2.4 Aggregation

An advanced smart aggregation layer allows for automatically querying across all three layers and aggregating, including inferred knowledge, across all three layers.

3 Benchmark Methodology

3.1 System Configurations

Both systems operated on the same live workspace (`/home/openclaw/.openclaw/workspace`). The model was fixed to Claude Sonnet 4.6 in both cases. No prompting, coaching, or pre-staging of answers was performed.

CONFIGURATION	MEMORY ACCESS	RETRIEVAL METHOD	MCP TOOLS
GISM-0	L1 + L2 + L3	Semantic index · graph · time-series	Full GISM-0 toolset
Claude Code CLI	None — raw workspace only	Standard built-in	Standard built-in tools, no GISM-0 additions

3.2 Test Questions

Four questions were drawn from real workspace content — no synthetic or pre-staged data. Each required non-trivial retrieval across multiple files or knowledge domains.

Q#	QUESTION	DOMAIN	RETRIEVAL CHALLENGE
Q1	What are CleverThis's outstanding SAFE commitments and who are the participating investors?	Finance	Investment data spread across financial documents, investor communications, and board records — requires cross-source synthesis
Q2	What is CleverThis's current cash position and recent financing?	Finance	Multi-source: board packet files, bookkeeping archives
Q3	Describe the IBKR trading system setup and its key components.	Infrastructure	Distributed across skill definitions and archived configuration files
Q4	Who are the members of the CleverThis board and advisory team?	Organizational	Combines board, project, and meeting files

3.3 Measurement Approach

GISM-0: Wall-clock time for the agent session to answer all four questions sequentially. Token count from the session's aggregate usage counter (`subagent_tokens: 83,749`); the orchestrator's own overhead calls are not included in this figure. Prompt caching was disabled for both runs.

Claude Code CLI: Per-question timing via shell; token counts from `--output-format json`. Standard built-in tools were enabled; no GISM-0 memory tools were added.

4 Results

4.1 Per-Question Performance

Q#	TOPIC	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 aggregate token count (83,749) is captured from the session's `subagent_tokens` counter. The orchestrator agent's own calls are not included in this figure. Per-question GISM-0 token breakdown was not captured in this run — per-question values are shown as “—”; only the session aggregate is available. All Claude Code CLI token counts are exact per-question measurements.

GISM-0 was faster on three of the four questions (Q2, Q3, Q4). The Q1 exception: the agent selected `bash/grep` rather than a semantic search query — the worst-case retrieval path for this system. The largest Claude Code CLI token loads appeared on Q1 and Q4, where broad workspace scans forced full file loads.

4.2 Aggregate Results

178.4 s

GISM-0 TOTAL TIME

217.8 s

CLAUDE TOTAL TIME

39.4 s

WALL-CLOCK SAVED (18%)

83,749

GISM-0 TOTAL TOKENS

486,650

CLAUDE TOTAL TOKENS

82.8%

TOKEN REDUCTION

METRIC	GISM-0	CLAUDE CODE	RATIO / DELTA
Total time	178.4 s	217.8 s	GISM-0 18% faster
Total tokens	83,749	486,650	GISM-0 5.8× fewer
Accuracy	4 / 4	4 / 4	Parity
Estimated cost	~\$0.25	~\$1.46	GISM-0 83% cheaper

4.3 Cost Derivation

Cost estimates use Claude Sonnet 4.6 pricing (\$3/MTok input, \$15/MTok output). Claude Code CLI token counts are split 97.9% input / 2.1% output (from exact per-question `usage` fields). GISM-0 costs assume the same input/output ratio applied to the 83,749 aggregate total — GISM-0's actual I/O split was not captured in this run. A different output ratio would shift the GISM-0 cost estimate; this assumption is conservative if GISM-0 produces proportionally less output than Claude Code CLI.

5 Cost at Scale

The per-query token advantage compounds directly with query volume. The table below projects monthly inference cost at several query volumes using benchmark-observed ratios.

MONTHLY VOLUME	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

At 10,000 queries per month — a conservative enterprise deployment — GISM-0 saves approximately **\$3,025 per month** (\$36,300 annually). At 100,000 queries per month, annual savings exceed **\$363,000**.

Scaling note: Claude Code CLI token cost grows with the file count of the workspace — as the code-base, emails, and documents accumulate, each Claude Code CLI query loads more context. GISM-0's semantic retrieval cost is bounded by index precision, not corpus size. The efficiency gap is expected to widen as the workspace grows; this is an architectural property, not yet validated empirically across scale points.

Memory build cost: The one-time cost of constructing the GISM-0 memory substrate — ingesting 15+ years of workspace data, embedding 540K documents at 4,096 dimensions, and building 2.58M knowledge-graph triples — is not included in this per-query cost analysis. This cost is amortized across all future queries and varies significantly by data volume and ingestion cadence. A dedicated ingestion benchmark is planned for a subsequent study.

6 Strategic Implications

6.1 Inferred-memory Architecture as a Platform Capability

The benchmark demonstrates that GISM-0 is not a caching layer but a fundamentally different retrieval paradigm. Claude Code CLI rebuilds full context at each invocation; GISM-0 maintains persistent, structured representations across sessions and replaces an LLM’s typical paradigm for context management entirely.

This distinction has three compounding consequences:

Sub-linear scaling (architectural design). GISM-0’s retrieval cost is structurally bounded by index precision, not corpus size — the semantic index returns targeted matches regardless of whether the workspace contains 10,000 or 10,000,000 documents. Full validation at scale is planned for a subsequent benchmark.

Cross-session continuity. Facts, decisions, project states, full session transcripts, and preferences persist across agent invocations. Claude Code CLI sessions start cold on every query — rebuilding context from disk each time, and injecting the entire session (or a compaction) each time.

Multi-modal retrieval routing. L1 answers temporal queries; L2 answers semantic queries; L3 answers relational queries. No single retrieval modality covers the full query space — GISM-0 routes automatically based on the prompt type by calling a managed layer that will query all three layers and aggregate the results, including any inferencing done on L3.

6.2 Accuracy Trajectory

At benchmark scale (four questions, single workspace), accuracy was identical. As query complexity increases — multi-hop reasoning, temporal chains, entity disambiguation across documents — the structural capabilities of L2 semantic search and L3 graph traversal are expected to provide accuracy advantages that raw file reads cannot match.

The Q1 result is instructive: even when GISM-0 chose a suboptimal retrieval path (grep rather than semantic search), it produced a correct answer with fewer tokens. The system is robust to retrieval-strategy variance.

7 Conclusion

This preliminary benchmark — four questions, single run, internal workspace — demonstrates measurable performance advantages across speed, token efficiency, and estimated inference cost, with full accuracy parity. The efficiency gains stem from architectural design: GISM-0 replaces context-window loading with targeted, persistent retrieval. These advantages are expected to persist across model generations, as they do not depend on any model-specific behavior but rather on how context is assembled and passed to the model.

DIMENSION	RESULT	SIGNIFICANCE
Token efficiency	5.8× fewer tokens	Directly reduces inference cost at every query volume
Speed	18% faster wall-clock	Even with Q1 worst-case grep retrieval path
Accuracy	4 / 4 parity	No accuracy trade-off at benchmark scale

Cost reduction (est.)	83% per query	\$3,025/month savings at 10,000 queries (assumed I/O split)
Scalability	Architecturally sub-linear	Index-precision-bounded by design; full validation at scale is future work

GISM-0 is the inferenced-memory substrate for the CleverThis Platform and its GISM technology line. This benchmark represents single-run performance under normal operating conditions — no prompt coaching, no tuned evaluation setup. Future work includes multi-run averaging, adversarial question sets designed to probe the boundary between semantic and relational retrieval, and integration with the CleverThis agent orchestration framework.

Contact: To learn more about GISM-0 or the CleverThis Platform, contact cleverthis@cleverthis.com.