The Cache Wars 2: An Empirical Study of Prompt-Cache Efficiency in Six LLM Agent Harnesses

Annie
AGNT Labs

Technical Report — September 8, 2026

Abstract

LLM agent harnesses re-transmit the full conversation context on every model invocation, making input cost quadratic in session length. Provider-side prompt caching discounts repeated prefix tokens by 90% at the tested model rates. Claude uses explicit cache policy; Codex uses automatic, provider-managed prefix reuse. Both require stable historical content, while retention controls and guarantees depend on the model and access path. We present a controlled measurement of prompt-cache efficiency across six production agent harnesses — AGNT v0.6.6, oh-my-pi (OMP) 18.1.14, Claude Code CLI 2.1.263, Codex CLI 0.153.4, OpenClaw 2026.9.2, and Hermes Agent 0.21.1 — using byte-identical workloads, version-pinned harnesses and an identified AGNT build, subscription-authenticated provider paths (Claude Sonnet 5 and GPT-6 Astra), and per-request billing counters obtained from provider telemetry with fail-loud validity guards. The central instrument is the pause test of our July 2026 study [1]: four rapid turns followed by a $\geq$390-second idle period that exceeds Anthropic's 5-minute cache TTL but not its 1-hour TTL. Thirteen configurations contribute 65 measured requests. We find a 5.2–6.6$\times$ divergence in post-pause non-read input volume: harnesses emitting 1-hour markers (AGNT, Claude Code) retained 84.9–90.5% of the request in cache, while harnesses emitting 5-minute markers (OMP, OpenClaw, Hermes on their shipping defaults) retained 0% and re-purchased their entire context at a 1.25$\times$ write premium. AGNT posts the highest post-pause cache-hit ratio on both provider paths — 90.45% on Claude and 88.42% on Codex — and maintains Claude cache writes equal to the measured per-turn input increment. The two patterns reported in [1] are not reproduced in these current configurations: Claude Code's per-turn scaffolding injection has fallen from 6,817 to 40 tokens per turn, and Hermes' intra-burst cache leak no longer occurs. What remains is the default-TTL trap: three of the five Anthropic-path harnesses still ship 5-minute markers and forfeit the pause unless reconfigured (OMP via PI_CACHE_RETENTION=long, OpenClaw via cacheRetention: long, Hermes via cache_ttl: 1h), each of which then survives at 80.9–84.9%. Extrapolated to a one-hour, 20-message session with four natural breaks over identical content, measured marker strategies yield totals of $0.85 (AGNT), $0.85 (Claude Code), $1.39 (OMP, shipping default), $1.39 (Hermes), and $1.40 (OpenClaw) against a $3.37 uncached ceiling. Composed over a working month (22 days, 8 h/day), the same measured parameters project $85/month (AGNT) versus $235–$236/month for the 5-minute-default harnesses and $592 uncached — a per-seat "cache tax" of $150/month, or $1,796–$1,805/year, attributable entirely to one client-side configuration bit. All artifacts, scripts, and raw counters are provided for independent replication. The Codex path is analyzed separately through measured whole-session costs, all five pause results, per-turn growth, and its own one-hour, multi-hour, monthly and annual model. At GPT-6 Astra rates the common-content one-hour input scenarios are $2.24 with retained reuse, $4.47 with assumed loss after each break, and $12.83 without reuse; these are conditional API equivalents, not subscription charges.

Keywords: prompt caching, LLM agents, agent harness, cost measurement, cache TTL, context management, subscription access, reproducibility.

1 Introduction

Autonomous and semi-autonomous LLM agents operate in a loop: the harness assembles a request containing a system prompt, tool schemas, the full conversation history, and the newest user message; the model responds; the cycle repeats. Because the entire context is re-transmitted on every call, cumulative input tokens grow quadratically with turn count. For multi-hour, tool-heavy sessions this term dominates total cost.

Anthropic and OpenAI both offer prompt caching: previously processed prefix tokens are re-served at a fraction of list price (10% on Anthropic [2]; 10% on OpenAI's published cached-input rate [3]). Realized reuse depends on the harness and provider routing/cache state. Three client-side decisions matter on Anthropic: (a) whether cache-control markers are emitted at all; (b) the TTL selected per marker (Anthropic: 5 minutes at a 1.25× one-time write premium, or 1 hour at 2× [2]); and (c) whether the serialized prefix remains byte-identical across turns, since provider caches are exact-prefix matches — any earlier-byte mutation invalidates everything after it.

Vendor documentation describes intended behavior; it does not establish what shipped binaries do under realistic use. Our July 2026 study [1] measured that directly for the releases then current. This paper is the second installment of The Cache Wars: it repeats the protocol of [1] on the current releases of the same six harnesses, on the subscription-authenticated provider paths that most users actually run, and on both provider families. Our contributions:

- A reproducible pause-test protocol that cleanly discriminates 5-minute from 1-hour cache retention using a $\geq$390 s idle gap, unchanged from [1].
- Per-turn billing telemetry for six production harnesses on byte-identical workloads across two subscription-authenticated provider paths (13 configurations, 65 requests), with instrumentation guards that abort on any validity violation.
- A re-examination of the two cost pathologies identified in [1]: Claude Code's per-turn scaffolding injection has fallen from 6,817 to 40 tokens, and Hermes' intra-burst cache leak is gone; the default-TTL trap persists in three harnesses, each of which survives once its documented retention option is set.
- A parametric cost model, fit to the measurements at current Sonnet 5 list prices, projecting one-hour and multi-hour session costs, and its composition into monthly and annual per-seat spend under three usage profiles.
- A complete artifact set enabling third-party replication.

2 Background: Prompt-Cache Semantics and Pricing

Anthropic. A request may carry up to four `cache_control` breakpoints. Marking a content block caches the serialized request prefix up to and including that block. Each marker carries a TTL: ephemeral 5 m (default) or 1 h. Billing for Claude Sonnet 5 (list, per 10^6 tokens): base input \$2.00; cache read \$0.20 (0.1×); 5 m cache write \$2.50 (1.25×); 1 h cache write \$4.00 (2.0×); output \$10.00 [2]. Cache hits refresh the TTL. Caching is exact-prefix: a single differing byte at position N invalidates all cached content at positions $\geq$ N.

OpenAI. Caching is automatic: requests sharing a sufficiently long prefix report `cached_input_tokens` at a discount. Since [1], OpenAI publishes explicit cache prices; for GPT-6 Astra (list, per 10^6 tokens, standard short-context): input \$10.00; cached input \$1.00 (0.1×); cache write \$12.50 (1.25×);

output $50.00 [3]. No Anthropic-style cache markers are used in this track. Current Platform model-specific retention settings and their distinction from the subscription path are described below [12]. Codex CLI relies exclusively on this mechanism [4], and every Codex-path receipt in this study reports zero cache-write tokens.

We use premium tokens to denote tokens billed at $\geq 1\times$ list (uncached input plus cache writes) and cached tokens for those billed at $0.1\times$. For a harness with zero overhead, steady-state premium per turn equals the size of the newest message.

2.1 Codex caching semantics and explicit price list

Codex caching is automatic prefix reuse, not Anthropic cache_control markers. OpenAI's Platform documentation for GPT-5.6 and later specifies prompt_cache_options.ttl = 30m as the default and only supported minimum lifetime; older model families expose different retention controls [12]. This is a Platform API contract, not proof that the subscription backend exposes every control. In these subscription runs, five paths retained cache across the measured pause; the OMP burst miss demonstrates that eligible retention and an actual hit are not the same thing.

Table C1. Complete standard list prices in USD per million tokens [2,3,13]. Sonnet writes are 5-minute / one-hour; Astra's write price is a separate accounting category, not an Anthropic TTL. The Astra threshold applies to the full request. The 65 measured requests are below it; the longer analytical sessions may cross it.

Model / context	Input	Cached	Write	Output
Sonnet 5	$2.00	$0.20	$2.50 / $4.00	$10.00
GPT-6 Astra $\leq$272k	$10.00	$1.00	$12.50	$50.00
GPT-6 Astra >272k	$20.00	$2.00	$25.00	$75.00

For Codex, total input I already contains cached input C; premium or non-read input P = I - C. The raw AGNT provider event uses input_tokens_details.cached_tokens, while Codex CLI reports cached_input_tokens. OMP, OpenClaw and Hermes expose normalized exclusive-input shapes that must be combined with their cache reads once, not twice. No accepted Codex request reports separately billed cache-write tokens. API-equivalent comparisons use $10/M uncached and $1/M cached input for these short-context measured requests. Subscription fees and quota meters are separate (§8).

3 Systems Under Test

Table 1: Harnesses, versions, installation channel, and measured provider path.

Harness	Version	Install	Provider path measured	Marker TTL (measured)
AGNT	v0.6.6	designated build (source fingerprints in artifacts)	Claude subscription; Codex subscription	1 h (Claude, all writes); auto (Codex)
oh-my-pi (OMP)	18.1.14	`bun i -g @oh-my-pi/pi-coding-agent@18.1.14`	Claude subscription; Codex subscription	5 m default; 1 h via `PI_CACHE_RETENTION=long`
Claude Code CLI	2.1.263	npm (official)	Anthropic (native, Claude subscription)	1 h (all writes)

Harness	Version	Install	Provider path measured	Marker TTL (measured)
Codex CLI	0.153.4	npm (official)	OpenAI (native, ChatGPT subscription)	n/a — automatic
OpenClaw	2026.9.2	`npm i -g openclaw@2026.9.2`	Claude subscription; Codex subscription (embedded runtime)	5 m default; 1 h via `cacheRetention: long`
Hermes Agent	0.21.1 (v2026.9.7)	editable install from tagged source (Py 3.13)	Claude subscription; Codex subscription	5 m default; 1 h via `prompt_caching.cache_ttl: 1h`

The tested versions were resolved on 2026-09-08; the AGNT build is identified by the source fingerprints in the artifact set (OMP under Bun 1.3.14; OpenClaw under Node 24.20.0; Hermes from its tagged source with Python 3.13 and its declared dependencies; AGNT's provider code under Node 22.16.0). Anthropic-path harnesses used model `claude-sonnet-5`; Codex-path harnesses used `gpt-6-astra`. Every arm authenticated with a subscription credential rather than a metered API key — the Claude subscription for Anthropic paths and the ChatGPT subscription for Codex paths (§8). OpenClaw and Hermes were driven through their public entry points (`openclaw agent --local --session-id ... --json` with all tools denied; Hermes' documented `AIAgent` API with `run_conversation(..., conversation_history=...)` and an empty toolset), each on the provider path where its caching demonstrably engages, so results reflect each system's best case, not a degraded default. OMP was driven through its own `omp -p --mode json --continue` print mode; its per-turn `agent_end` usage block reports a native per-TTL split (`cttl:{ephemeral5m, ephemeral1h}`), and it was measured on both its shipping default (5 m) and its `PI_CACHE_RETENTION=long` path (1 h). OpenClaw and Hermes were likewise measured on both their defaults and their documented 1-hour options (`cacheRetention`: long; `prompt_caching.cache_ttl: 1h`). Claude Code ran through `claude -p --output-format json --resume` with tools disabled; Codex CLI through `codex exec --json` / `exec resume`. AGNT was driven through its own orchestrator handler and provider adapters on an isolated loopback host, carrying its resident instruction context and tool schemas with tool execution disabled; on the Codex path OpenClaw was run on its embedded runtime rather than the separately available Codex app-server runtime, so that its cache behaviour is its own.

4 Experimental Design

4.1 Workload

Each turn t sends the message "CACHE-WARS RUN ⟨run id⟩ / TURN t: This is a fixed-output cache benchmark. Reply with exactly OKt, nothing else. Do not use tools. Ignore the synthetic data below." followed by the filler F(t), a deterministic function producing 260 rows and 7,878 UTF-8 bytes per turn (Listing 1) — ≈5,265 tokens under the Sonnet 5 tokenizer and ≈4,455 under GPT-6 Astra. The deterministic numeric-row structure follows [1]; byte identity is asserted only for this edition's shared within-track workload files, not the earlier edition's different runner inputs. Payloads are byte-identical across harnesses within a provider track (each track carries its own run identifier). Instructing fixed, short replies (OKt: five billed output tokens on Claude, six on Codex) minimizes output-side confounds; every accepted reply is exactly OKt.

Listing 1 — deterministic per-turn filler (JavaScript; Python port identical)

```
function filler(t) {
```

```
  let s = '';
  for (let i = 0; i < 260; i++)
    s += `row ${t}-${i} a=${(i*7919)%104729} b=${(i*104729)%7919} c=${i%13}; `;
  return s;
}
```

4.2 Protocol: the pause test

Four turns are issued back-to-back (inter-turn latency $\ll$ 5 m), followed by an idle period of at least 390 s, followed by turn 5. AGNT's measured gaps are 390.002 s (Claude) and 390.010 s (Codex); the other arms' gaps span 394.8–498.7 s, recorded from runner completion to next runner start (provider-visible inactivity is longer by client start-up). The idle length is chosen to strictly exceed the 5-minute TTL while remaining far below the 1-hour TTL, so post-pause telemetry classifies each harness's effective retention with no ambiguity: a 5 m cache must report `cache_read_input_tokens = 0` on turn 5; a 1 h cache must report a full-prefix read.

4.3 Instrumentation and validity guards

Ground truth is the provider's own per-request accounting: Anthropic's usage block (`input_tokens`, `cache_read_input_tokens`, `cache_creation_input_tokens`, and the per-TTL split `cache_creation.ephemeral_{5m,1h}_input_tokens`); OpenAI's `input_tokens` with `input_tokens_details.cached_tokens`. These reach the runner through each harness's native telemetry — Claude Code's `-output-format json` result blocks, Codex's `turn.completed` usage events, OMP's `agent_end` usage with its `cttl` split, OpenClaw's `lastCallUsage`, and Hermes' canonical session-usage deltas with exactly one API call per turn (Hermes does not expose the per-TTL split, so its write tier is taken from its configured policy and corroborated by the post-pause read). For the AGNT arm the provider's raw usage events (`message_start`/`message_delta`; `response.completed`) were captured on every request and reconciled with the harness's own counters: all ten agree exactly.

Driver scripts enforce three fail-loud invariants; violation aborts the run and discards its data: (G1) identity — every turn's reply must be exactly OKt and the reported model must be the requested model (defeats silent fallback to another model or a cached reply); (G2) payload — the SHA-256 of each outgoing prompt must equal the workload digest, byte-identical across every arm of a track (defeats drift in the controlled variable); (G3) continuity and timing — the session identifier is held constant across turns, provider-reported context must grow by the payload each turn (defeats non-accumulating history), inter-turn gaps inside the burst must be below 300 s and the gap before turn 5 at least 390 s. Every accepted record carries the raw usage object of its source so the transformations can be re-run offline (§10).

4.4 Cross-harness comparability

Harnesses ship system prompts and tool schemas of different sizes (6.6k–34.1k tokens at turn 1 on the Claude path; AGNT was measured with its full resident instruction context and tool schemas, the other harnesses in isolated, reduced-tool configurations), so absolute token counts are not comparable across systems. We therefore compare only: (i) the post-pause cache-hit ratio (dimensionless); (ii) steady-state premium tokens per turn against the known 5,265-token payload, which decomposes each bill into user payload + harness overhead; and (iii) dollar figures from applying each harness's measured marker strategy — its TTL policy and per-turn overhead — to fully identical content (§5.3, §6).

As an internal consistency check, per-turn context growth must equal the payload plus any harness self-injection. Measured deltas (turns 2–4, tokens/turn, Claude path): AGNT +5,265; OMP +5,265; Hermes +5,264; OpenClaw +5,285; Claude Code +5,305. The constant ≈5,265 component appearing in all five confirms workload identity to within tokenizer noise; the excess (+20 OpenClaw, +40 Claude Code) is measured harness overhead (§5.4). On the Codex path (GPT-6 Astra tokenizer) the payload is 4,455 tokens: AGNT, OMP and Hermes grow by 4,454–4,455 per turn, OpenClaw by 4,470, and Codex CLI by 4,455–6,344 (a mean +1,364 tokens/turn of its own turn scaffolding).

4.5 Codex-specific controls

The Codex track compares AGNT, Codex CLI, OMP, OpenClaw and Hermes on gpt-6-astra using one shared five-prompt file. Claude Code has no Codex arm in this study. The same 390-second minimum idle is retained for comparability, but it does not distinguish Codex TTL categories. The response is OKt (six billed output tokens); no tool execution is performed. The four measured growth increments, cache counters, prompt hashes and gaps are retained for every path. The native session footprints are reported separately from common-content projections, and a burst miss is never erased or converted into an estimated miss frequency.

5 Results

5.1 Per-turn billing telemetry

Table 2: Per-turn provider counters, Claude path, shipping defaults (tokens). C = cached (billed 0.1×); P = premium (billed ≥1×). Turn 5 follows the ≥390 s pause.

Turn	AGNT C	AGNT P	Claude Code C	Claude Code P	OpenClaw C	OpenClaw P	Hermes C	Hermes P
1	3,949	30,167	3,289	10,586	0	13,846	0	6,555
2	34,114	5,267	13,873	5,307	13,844	5,287	6,553	5,266
3	39,379	5,267	19,178	5,307	19,129	5,287	11,817	5,266
4	44,644	5,267	24,483	5,307	24,414	5,287	17,081	5,266
5 (post-pause)	49,909	5,267	29,788	5,307	0	34,986	0	27,611

Table 3: Per-turn provider counters, Codex path (GPT-6 Astra tokens). C = cached input; P = uncached input. Retention is provider-managed on this path; OMP's turn-4 miss is retained as observed.

Turn	AGNT C	AGNT P	Codex CLI C	Codex CLI P	OpenClaw C	OpenClaw P	OMP C	OMP P	Hermes C	Hermes P
1	0	21,845	12,288	8,787	0	9,816	0	6,663	0	5,278
2	21,760	4,540	20,864	6,555	9,600	4,686	6,528	4,590	5,120	4,612
3	26,112	4,643	27,264	6,287	14,080	4,676	10,880	4,693	9,600	4,586
4	30,592	4,618	33,408	4,598	18,560	4,666	0	20,028	13,952	4,688
5 (post-pause)	35,072	4,593	37,888	6,462	23,040	4,656	19,840	4,643	18,432	4,662

AGNT's per-TTL split reports `ephemeral_5m_input_tokens = 0` on every Claude turn (all writes 1 h); its turn-1 counters include a 3,949-token shared-prefix read the provider already held, as do Claude Code's (3,289). Every Anthropic-path harness bills a constant two uncached tokens per turn beside its writes. Claude Code's 1 h split is native (`ephemeral_1h_input_tokens`); OMP's is native (`cttl`); OpenClaw's is native (`cacheWrite1h`). OMP's Claude counters (both TTL paths) are reported

separately in §5.6 (Table 5), since its default and opt-in configurations must be distinguished; OpenClaw's and Hermes' 1-hour options are in §5.7 (Table 6). On the Codex path Codex CLI already read 12,288 tokens on its first request (a provider-side warm prefix), whereas AGNT read zero; OMP's turn 4 is a complete miss between two hits, retained as observed.

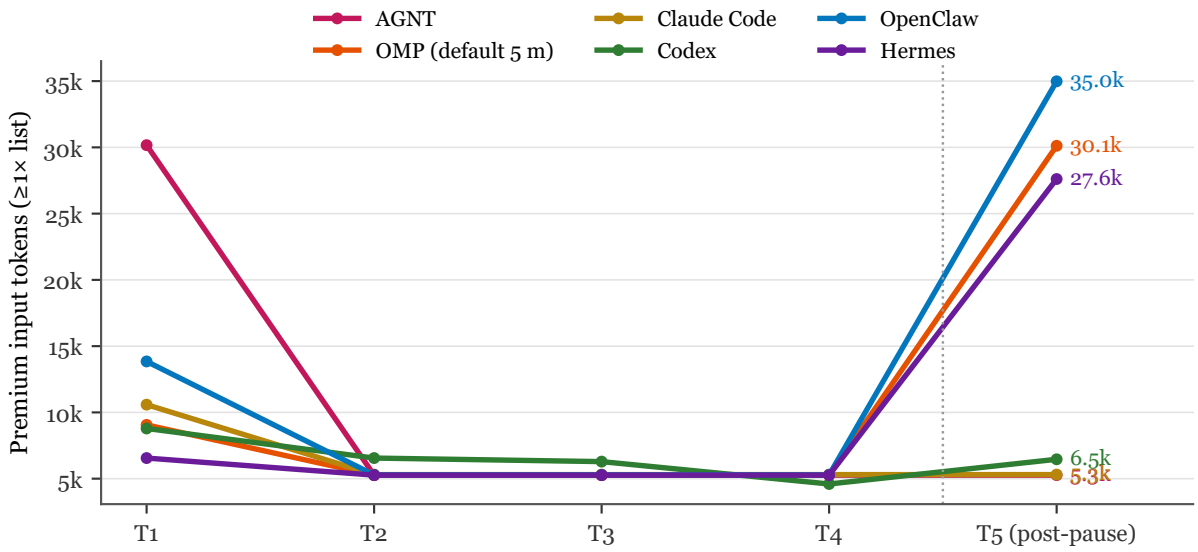

Figure 1: Premium input tokens billed per turn (rate ≥1× list). Turn 5 follows the ≥390 s idle: the 5-minute-TTL harnesses (OMP on its shipping default, OpenClaw, Hermes) re-purchase their full accumulated context at the 1.25× write premium, while the 1-hour harnesses (AGNT, Claude Code) resume at payload cost. Every Anthropic-path harness's premium is the payload alone through turn 4 — the per-turn scaffolding of [1] is gone (§5.4). Codex (GPT-6 Astra tokenizer) is shown in its own token units.

5.2 Pause survival

Table 4: Turn-5 cache-hit ratio h = $C_5/(C_5+P_5)$.

Harness	TTL	C_5	P_5	h	Classification
AGNT v0.6.6 (Claude)	1 h	49,909	5,267	90.5%	survived; premium = payload only
OMP 18.1.14 (=long)	1 h	24,851	5,267	82.5%	survived (opt-in); premium = payload only
OMP 18.1.14 (default)	5 m	0	30,119	0%	expired; full-context re-write (shipping default)
Claude Code	1 h	29,788	5,307	84.9%	survived; premium = payload + 40
OpenClaw (=long)	1 h	29,805	5,287	84.9%	survived (opt-in); premium = payload + 20
OpenClaw (default)	5 m	0	34,986	0%	expired; full-context re-write at 1.25×
Hermes (cache_ttl=1h)	1 h	22,347	5,266	80.9%	survived (opt-in); premium = payload only
Hermes (default)	5 m	0	27,611	0%	expired; full-context re-write at 1.25×
AGNT v0.6.6 (Codex)	auto	35,072	4,593	88.4%	survived this trial; retention unguaranteed [3]
Codex CLI	auto	37,888	6,462	85.4%	survived this trial; retention unguaranteed [3]

Harness	TTL	C_5	P_5	h	Classification
OpenClaw (Codex)	auto	23,040	4,656	83.2%	survived this trial
OMP (Codex)	auto	19,840	4,643	81.0%	survived this trial
Hermes (Codex)	auto	18,432	4,662	79.8%	survived this trial

The result partitions exactly along declared TTL, consistent with Anthropic's contractual expiry semantics: all three 5 m defaults lost 100% of cache across the pause, re-billing 27.6k–35.0k tokens at the write premium, while every 1 h configuration read its entire prior context at 0.1× — AGNT's 49,909-token read is 99.996% of its previous request. AGNT's ratio is the highest measured on both paths (90.45% Claude, 88.42% Codex), 5.52 points above the next Claude configuration (OpenClaw with its 1 h option) and 2.99 points above Codex CLI. The five Codex-path survivals are single observations of an explicitly unguaranteed mechanism and are not generalizable.

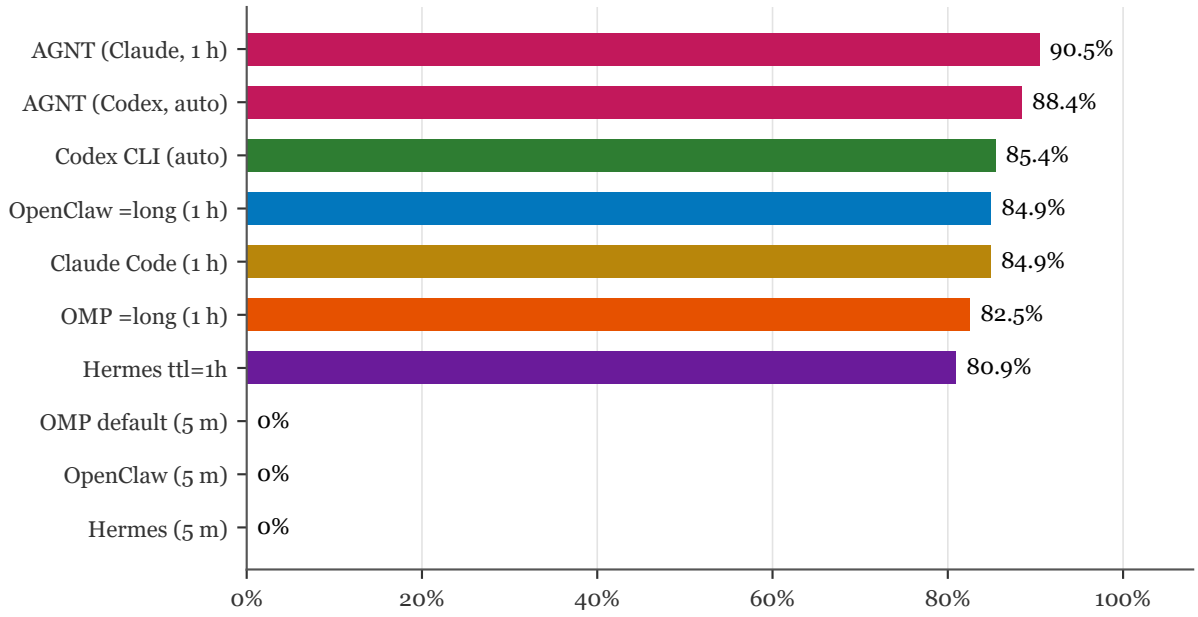

Figure 2: Turn-5 cache-hit ratio h = $C_5/(C_5+P_5)$ after the ≥390 s pause. The outcome partitions exactly along the effective TTL: 80.9–90.5% for 1-hour markers (including OMP, OpenClaw and Hermes with their retention options set), 0% for 5-minute markers (OMP default, OpenClaw, Hermes), with no intermediate value. AGNT posts the highest ratio on both provider paths (90.45% Claude, 88.42% Codex).

5.3 Controlled cost (identical content)

Applying the two marker strategies to identical content (§4.4: the 34,114-token first-turn prefix measured for AGNT — system prompt, tool schemas and the turn-1 payload — plus 5,265 tokens per subsequent turn) for the 5-turn protocol yields, at Sonnet 5 list prices: all-1 h strategy (AGNT, Claude Code layouts) \$0.254; 5 m strategies (OMP-default, OpenClaw, Hermes layouts) \$0.286 (+13%, driven by the post-pause re-write); no-cache control \$0.446 (+76%). Verification: all-1 h writes 34,114 + 4×5,265 = 55,174 tk × \$4/M = \$0.221; reads 168,046 tk × \$0.20/M = \$0.034; total \$0.254. ✓ 5 m writes 34,114 + 3×5,265 + (34,114 + 4×5,265) = 105,083 tk × \$2.50/M = \$0.263; reads 118,137 tk × \$0.20/M = \$0.024; total \$0.286. ✓

5.4 Pathology I revisited — per-turn scaffolding injection (Claude Code)

In [1], Claude Code 2.1.126 grew its context by 12,003 tokens/turn against a 5,186-token payload: 6,817 tokens/turn (2.31× payload) of self-injected system reminders and related scaffolding, re-billed at the 1 h write premium every turn. Claude Code 2.1.263 grows by 5,305 tokens/turn against the 5,265-token payload: 40 tokens/turn (0.8% of payload). The earlier large excess is not reproduced in this reduced-tool workload. Its steady-state premium (5,307 tokens/turn) is now within 40 tokens of AGNT's (5,267), and the perfect marker discipline it already showed in [1] now coexists with payload-clean spend. The residual is small enough that it is invisible in dollar terms (§6, $0.851 vs $0.846 per one-hour session).

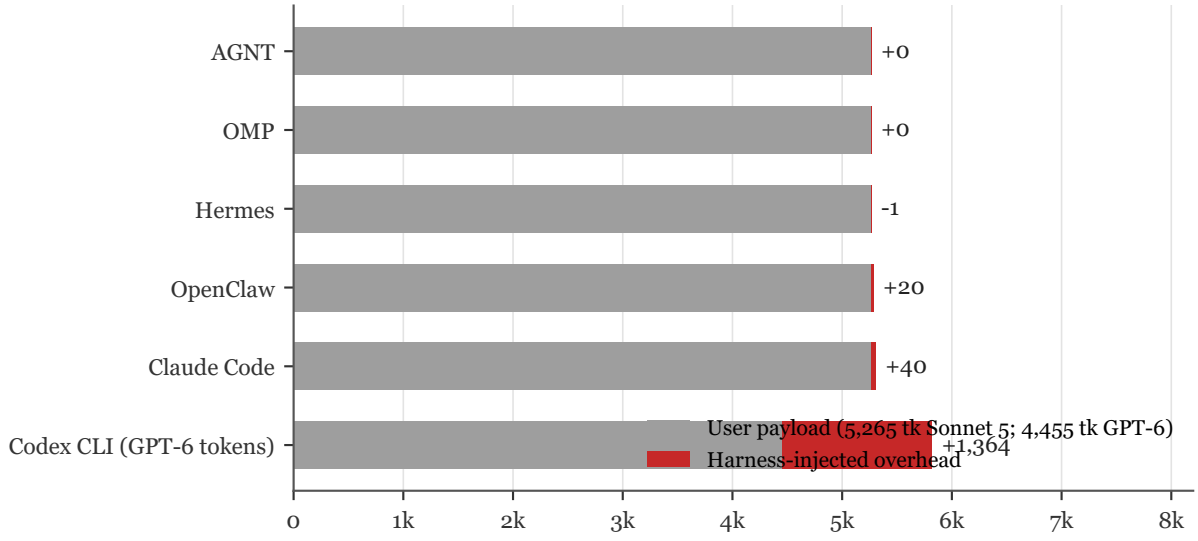

Figure 3: Per-turn conversation growth decomposed into the byte-identical user payload (gray) and harness-injected overhead (red). AGNT, OMP and Hermes inject nothing (Hermes -1 token is tokenizer noise); OpenClaw adds 20 tokens/turn and Claude Code 40 — down from 6,817 (2.31× the payload) in [1]. Codex CLI adds a mean 1,364 GPT-6 tokens/turn of its own turn scaffolding.

5.5 Pathology II revisited — intra-burst cache leak (Hermes)

In [1], Hermes 0.18.0 leaked within the rapid burst: cached reads decreased (22,837 → 17,339) while premium writes grew (5,186 → 15,869 → 21,054), because rolling message-marker placement never extended the cached prefix over accumulated history. Hermes 0.21.1 shows the opposite, correct behaviour: cached reads grow with the conversation (6,553 → 11,817 → 17,081) while premium writes stay pinned at the payload (5,266, 5,266, 5,266). Its read volume extends with the accumulated history; the earlier leak pattern is not present in these receipts. Hermes still loses the remainder at the pause on its 5 m default (h = 0%, 27,611-token re-write) and survives with `prompt_caching.cache_ttl: 1h` (C_5 = 22,347, h = 80.9%).

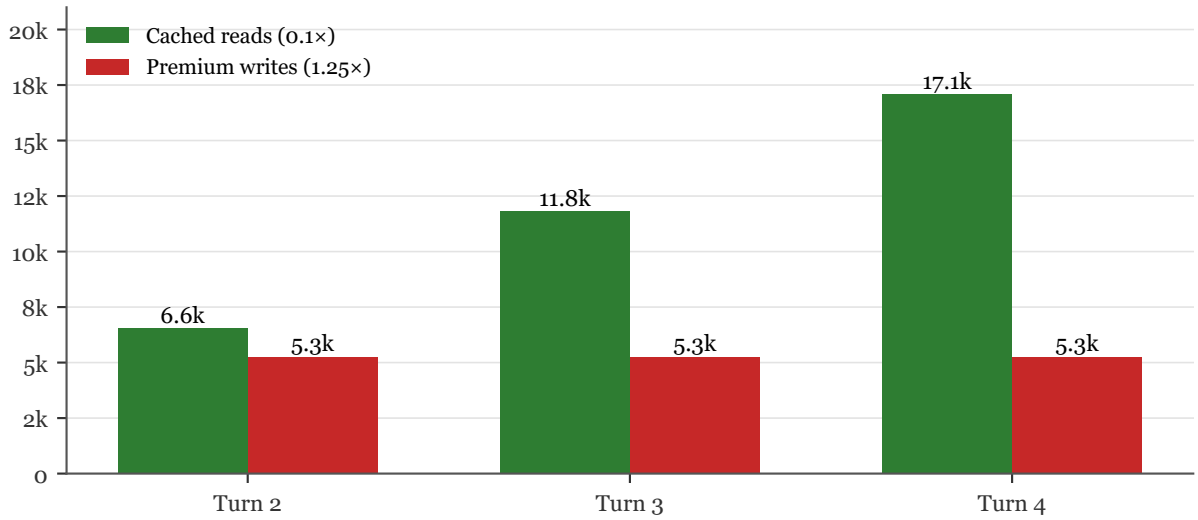

Figure 4: Hermes intra-burst behaviour (Table 2 rows, turns 2–4, before any TTL expiry). In 0.21.1 cached reads grow with the conversation (6.6k → 11.8k → 17.1k) while premium writes stay pinned at the payload (5,266 tokens): the rolling-marker leak of [1], in which reads collapsed to the static prefix while writes grew, no longer occurs.

5.6 OMP — clean spend, but a default-TTL trap

oh-my-pi bills premium tokens equal to the user payload with zero self-injection (5,265 tokens/turn, measured; Table 2-consistent delta +5,265), as it did in [1]. On the tested subscription path OMP 18.1.14 emits 5-minute markers by default and fails the pause test identically to OpenClaw and Hermes (h = 0%, full 30,119-token re-write; verified per-TTL as `ephemeral5m` in its `agent_end` usage). With `PI_CACHE_RETENTION=long` it emits 1-hour markers and survives (C_5 = 24,851, h = 82.5%; verified `ephemeral1h`). OMP demonstrates that payload-clean billing and the 5-minute default trap are independent axes: a harness can perfect the former and still forfeit the pause on the latter.

Table 5: OMP per-turn counters, both TTL paths (tokens). C = cached; P = premium. Premium is pinned at the payload every turn; only the default-vs-flag TTL differs.

Turn	Default (5 m) C	P	=long (1 h) C	P
1	0	9,059	0	9,058
2	9,057	5,267	9,056	5,267
3	14,322	5,267	14,321	5,267
4	19,587	5,267	19,586	5,267
5 (post-pause)	0	30,119	24,851	5,267

5.7 OpenClaw and Hermes — the same trap, the same cure

OpenClaw 2026.9.2 and Hermes 0.21.1 ship the same 5-minute default and fail the pause the same way (34,986 and 27,611 tokens re-written at 1.25×). Each documents a one-line retention option — OpenClaw's per-model `cacheRetention`: long [5], Hermes' `prompt_caching.cache_ttl: 1h` [6] — and each survives with it set (h = 84.9% and 80.9%), with first-turn writes within 106 (OpenClaw) and 2 (Hermes) tokens of the defaults'. The counters of both configurations are given in Table 6; premium is the payload (plus OpenClaw's constant 20 tokens) on every non-expiry turn.

Table 6: OpenClaw and Hermes per-turn counters, default vs 1-hour option (tokens). C = cached; P = premium.

Turn	OpenClaw 5 m C	P	OpenClaw 1 h C	P	Hermes 5 m C	P	Hermes 1 h C	P
1	0	13,846	0	13,952	0	6,555	0	6,557
2	13,844	5,287	13,950	5,287	6,553	5,266	6,555	5,266
3	19,129	5,287	19,235	5,287	11,817	5,266	11,819	5,266
4	24,414	5,287	24,520	5,287	17,081	5,266	17,083	5,266
5 (post-pause)	0	34,986	29,805	5,287	0	27,611	22,347	5,266

5.8 Codex — per-turn telemetry and complete session accounting

Table 3 contains the full cached/non-read pairs for all five Codex paths; Figure C1 gives the corresponding five-line trajectory. Tables C2 and C3 separately identify whole-sequence reuse, final-request reuse and actual context volume. Higher cache share is better for reuse; lower uncached tokens and API-equivalent dollars are better at the stated footprint. These are different rankings.

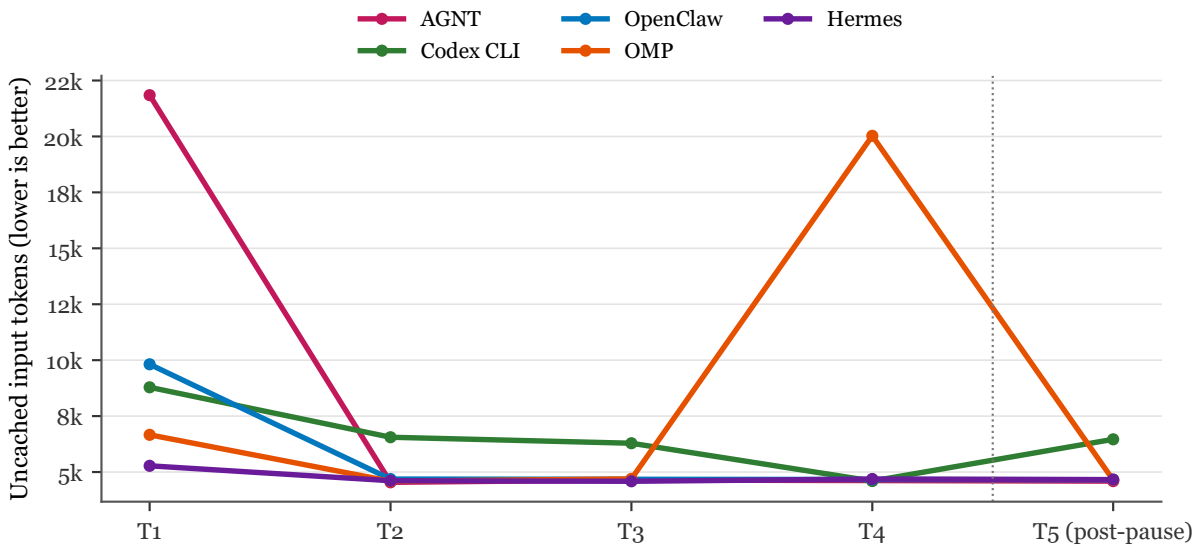

Figure C1. Codex per-request uncached input across all five harnesses, in GPT-6 Astra token units. All paths retain reads on request 5. OMP's request-4 miss is shown, not replaced by a favorable run. All reported cache-write counters are zero.

Table C2. Codex whole-sequence native-footprint accounting, including the cold/warm first requests and OMP's request-4 miss. USD is an API-rate equivalent including the observed 30 output tokens per sequence, not a subscription bill.

Harness	Input (5 turns)	Cached	Weighted h	5-turn USD
AGNT	153,775	113,536	73.83%	$0.517426
Codex CLI	164,401	131,712	80.12%	$0.460102
OpenClaw	93,780	65,280	69.61%	$0.351780
OMP	77,865	37,248	47.84%	$0.444918
Hermes	70,930	47,104	66.41%	$0.286864

Table C3. Codex post-pause request: all five harnesses, complete denominator and costs. AGNT leads h_5. The native-context USD column is not an equal-content harness ranking.

Harness	Pause s	$Input_5$	$Cached_5$	P_5	h_5	USD_5
AGNT	390.010	39,665	35,072	4,593	88.42%	$0.081302
Codex CLI	399.582	44,350	37,888	6,462	85.43%	$0.102808
OpenClaw	467.263	27,696	23,040	4,656	83.19%	$0.069900
OMP	458.198	24,483	19,840	4,643	81.04%	$0.066570
Hermes	409.639	23,094	18,432	4,662	79.81%	$0.065352

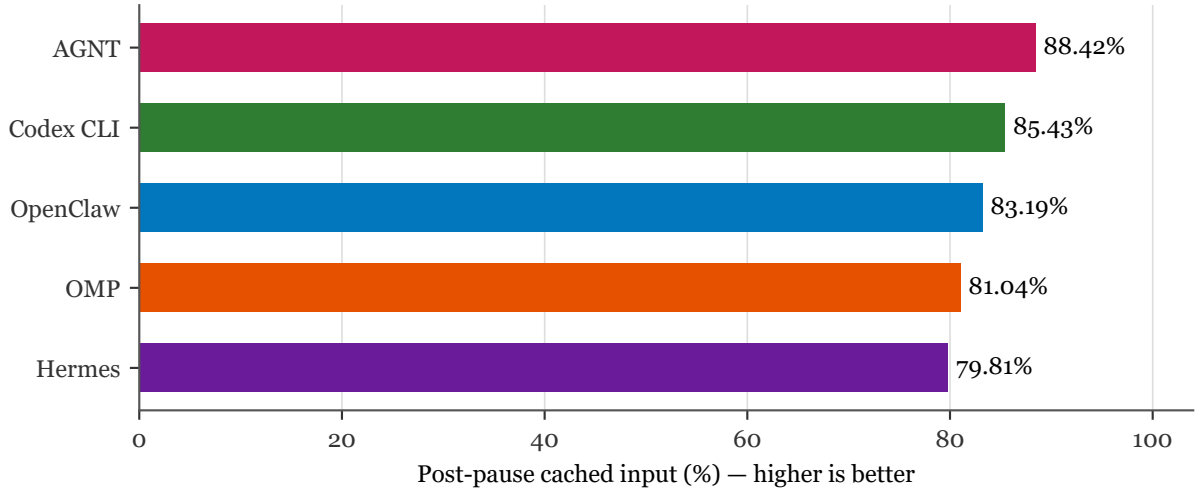

Figure C2. Codex request-5 cached-input share, with all five native paths. AGNT leads at 88.42%, followed by Codex CLI 85.43%, OpenClaw 83.19%, OMP 81.04%, and Hermes 79.81%. Total context is provided in Table C3: a larger reusable prefix can raise this ratio without minimizing total cost.

AGNT ranks first on the Codex pause metric at 88.42%, 2.99 percentage points ahead of Codex CLI. It uses 4,593 uncached tokens versus 6,462 for the CLI on that request. Across the full sequence, however, Codex CLI has the higher weighted cache share (80.12% versus 73.83%), in part because its first request reads 12,288 shared-prefix tokens while AGNT's reads zero. Hermes' smaller native context has the lowest full-sequence API equivalent. None of these results is hidden by the primary endpoint.

5.9 Codex — context growth and burst misses

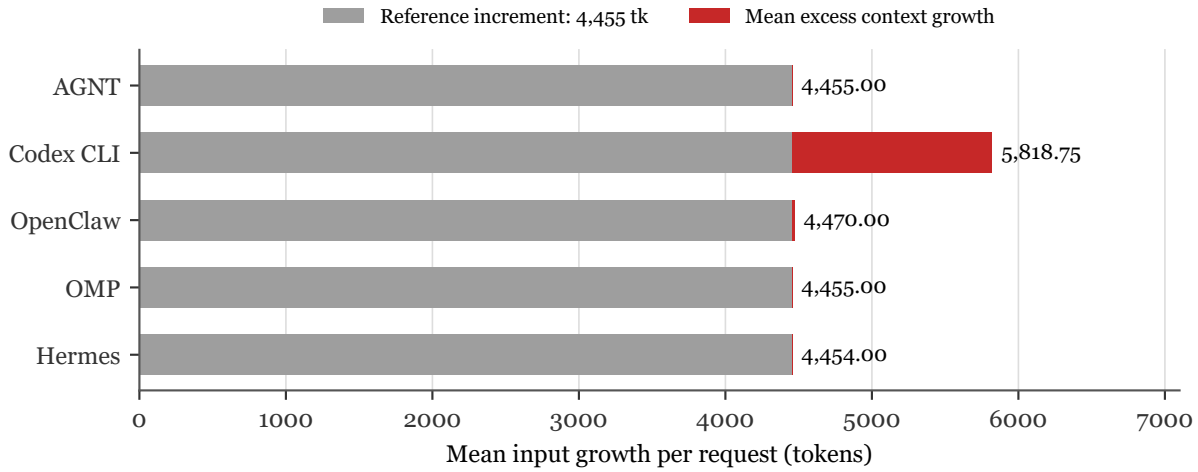

Figure C3. Codex context growth across the four request-to-request transitions. AGNT is 4,455 tokens per transition; Codex CLI averages 5,818.75. Growth is an observed difference, not proof of a particular internal injection mechanism. Small negative residuals for OMP/Hermes are not turned into negative overhead bars.

Table C4. Codex input increments. The reference 4,455-token increment is measured in AGNT; differences can include role/message framing and client context. They do not establish a specific source-code cause.

Harness	ΔI_2	ΔI_3	ΔI_4	ΔI_5	Mean
AGNT	4,455	4,455	4,455	4,455	4,455.00
Codex CLI	6,344	6,132	4,455	6,344	5,818.75
OpenClaw	4,470	4,470	4,470	4,470	4,470.00
OMP	4,455	4,455	4,455	4,455	4,455.00
Hermes	4,454	4,454	4,454	4,454	4,454.00

OMP's fourth request contains 20,028 total input tokens and zero reads; its fifth reads 19,840 tokens. The missed read increases the observed sequence cost. One miss in one sequence cannot estimate a reliable miss probability. We therefore use the same retained-prefix assumption for all harnesses in the growth-based projection and show loss after each break only as a separate sensitivity scenario (§6.2).

6 Cost Model and One-Hour Extrapolation

6.1 Claude cost model and one-hour projection

Let the session comprise n messages of payload m tokens, partitioned into bursts by breaks longer than the TTL. With prefix c_1 = first-turn write and per-turn overhead o, and prices r = \$0.20/M (read), w_5 = \$2.50/M, w_1 = \$4.00/M, u = \$2.00/M:

$$
\begin{aligned}
10^6 \mathrm{Cost}_{1h} &= w_1 \left[c_1 + (n-1)(m+o)\right] + r \sum_{t=2}^{n} \mathrm{prefix}(t) \\
10^6 \mathrm{Cost}_{5m} &= w_5 \Big[\sum_{\text{bursts}} \mathrm{rewrite}_b + \text{in-burst writes} \Big] + r\,(\text{in-burst reads})
\end{aligned}
\tag{1}
$$

where prefix(t) = c_1 + (t-2)(m+o) is the cached context preceding message t, and rewrite_b = c_1 + (s_b-1)(m+o) is the full context re-purchased at the first message s_b of each burst after the first. Cache hits refresh the TTL, so the 1 h prefix persists across every break shorter than one hour. Because absolute footprints are incomparable across harnesses (§4.4), the model is instantiated over identical content — c_1 = 34,114, m = 5,265 — with each harness's measured marker strategy: its shipping TTL and its per-turn overhead o. For n = 20 with four breaks (after messages 4, 8, 12, 16; breaks > 5 m, session $\leq$ 1 h):

Table 7: One-hour projection (measured marker strategies over identical content; Sonnet 5 list prices). M = uncached multiplier vs. $3.365 control.

Harness	Premium writes (tk)	@$	Cached reads (tk)	@$	Total	M
AGNT v0.6.6 (1 h, o=0)	134,149	$0.54	1,548,481	$0.31	$0.846	0.25×
Claude Code (1 h, o=40)	134,909	$0.54	1,555,321	$0.31	$0.851	0.25×
OMP (default 5 m, o=0)	460,145	$1.15	1,222,485	$0.24	$1.395	0.41×
OpenClaw (5 m, o=20)	461,245	$1.15	1,225,185	$0.25	$1.398	0.42×
Hermes (5 m, o=0)	460,145	$1.15	1,222,485	$0.24	$1.395	0.41×
No caching (control)	1,682,630 @ u	$3.37	—	—	$3.365	1.00×

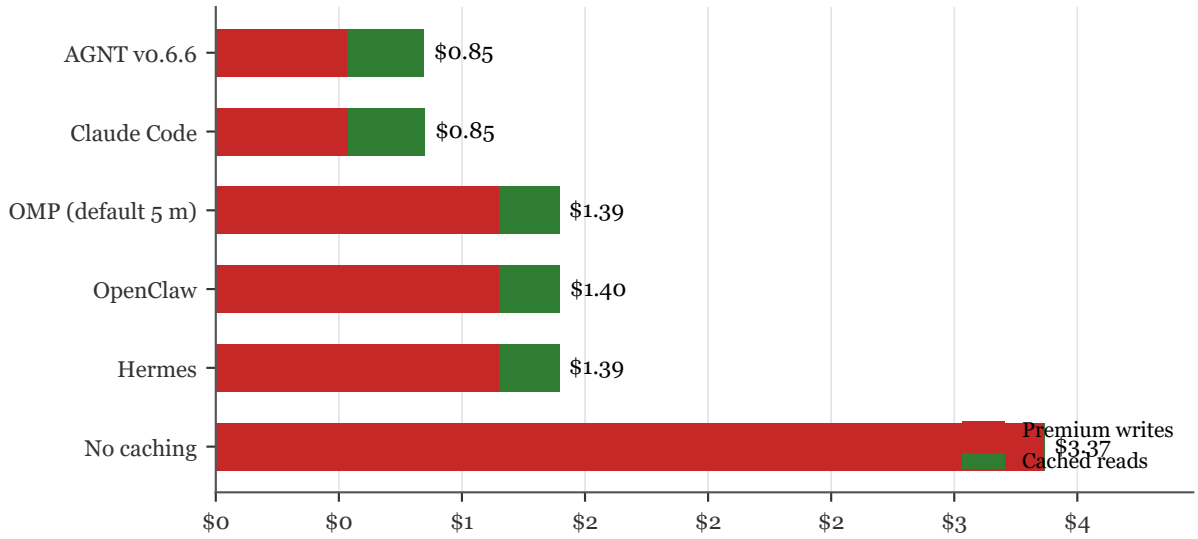

Figure 5: One-hour session projection (20 messages, 4 natural breaks; Table 7), decomposed into premium writes and cached reads over identical content. OMP (default), OpenClaw and Hermes spend on full-context re-writes at every break; AGNT and Claude Code resume at payload cost — the residual gap between them is Claude Code's 40-token per-turn overhead.

Extending the same model to longer sessions (a break after every fourth message, ≈12 min apart, at 20 messages/hour) yields the multipliers of Table 8. AGNT's and Claude Code's are monotone decreasing — the 2× build premium amortizes into 0.1× reads. OMP's, OpenClaw's and Hermes' flatten: every

break re-charges a full, ever-larger re-write. The model is conservative for 5 m harnesses: it assumes zero intra-burst expiry; the frequency of real-world pauses was not measured here, so the size of the modeled gap depends on the stated schedule. OMP, OpenClaw and Hermes are reported on their shipping 5-minute defaults; with their opt-in 1-hour options each collapses onto the AGNT curve ($0.846 at one hour for OMP =long, identical to AGNT to the token) — but that is not out-of-box behavior, so the default is used for the headline, consistent with the treatment of every other harness. Codex has its own price list, retention scenarios, one-hour projection and multi-hour multipliers in §6.2; its monthly, annual, team and tool-latency analyses follow in §7.4–7.6.

Table 8: Cost multiplier vs. the uncached baseline by session length (lower is better).

Harness	15 m	30 m	1 h	2 h	4 h	Shape
AGNT v0.6.6	0.57	0.37	0.25	0.18	0.14	monotone decreasing
Claude Code	0.57	0.37	0.25	0.18	0.15	monotone decreasing (offset by o)
OMP (default 5 m)	0.64	0.50	0.41	0.40	0.40	flattens (clean, but 5 m)
OpenClaw	0.64	0.50	0.42	0.41	0.40	flattens
Hermes	0.64	0.50	0.41	0.40	0.40	flattens

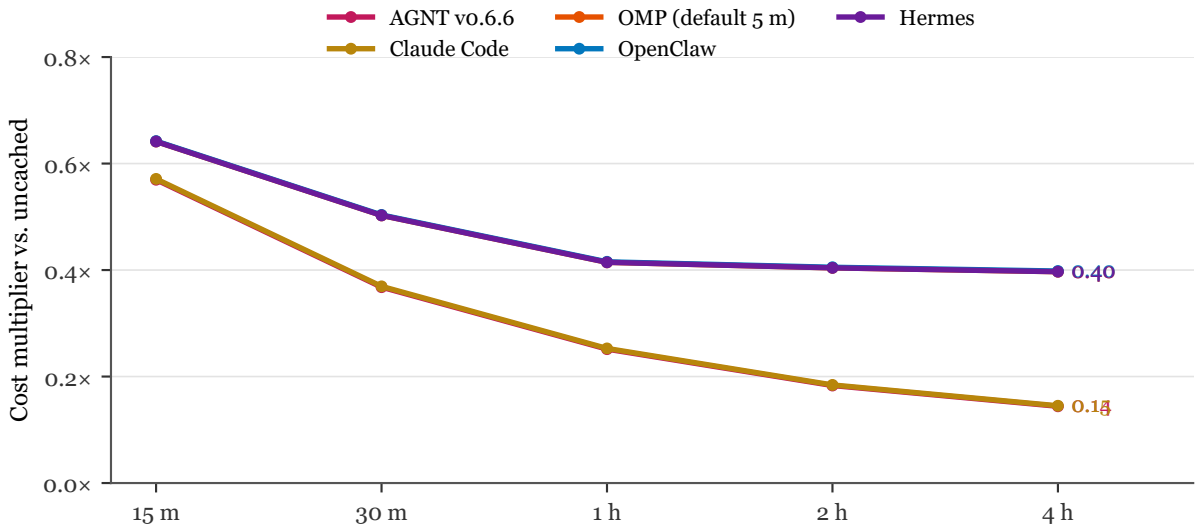

Figure 6: Cost multiplier vs. the uncached baseline by session length (Table 8; lower is better). AGNT's and Claude Code's curves are monotone decreasing — the 2× build premium amortizes into 0.1× reads; OMP (5 m default), OpenClaw and Hermes flatten because every break re-charges a full, ever-larger re-write. Without their retention options the 5-minute-default harnesses never reach the 1-hour trajectory.

6.2 Codex cost model and one-hour projection

The Codex model uses common initial context $c_1 = 21{,}845$ and increment $m = 4{,}455$, both from the measured AGNT sequence, to avoid comparing unequal native system prompts. Let $g = m + o$; o is the nonnegative mean excess input growth measured for a harness. Tiny negative residuals for OMP/Hermes are set to zero only in the model, never in the receipts. Every projected session starts cold. Let $q_t = 1$ when the whole previous prefix is retained and $q_t = 0$ when it is unavailable. Input $J_t = c_1 + (t\text{-}1)g$; reads $C_t = q_t[J_t\text{-}g]$ for $t > 1$, $C_1 = 0$; uncached input $U_t = J_t - C_t$.

$$
\begin{aligned}
K_X(n) &= 10^{-6} \sum_{t=1}^{n} \left[u(J_t)U_t + r(J_t)C_t + v(J_t)W_t + p(J_t)O_t \right], \\
(u, r, v, p) &= (10, 1, 12.5, 50) \quad (J_t \le 272000), \\
(u, r, v, p) &= (20, 2, 25, 75) \quad (J_t > 272000).
\end{aligned}
\tag{2}
$$

The input-only projection sets W = O = 0, matching zero separately reported writes and excluding identical fixed-output replies. The measured-dollar table includes output. The retained case sets q = 1 after the first request. The break-loss case sets q = 0 on requests 5,9,13,17 and so on, with retained prefixes between breaks. That is an assumed miss schedule, not a measured Codex expiry interval. No-cache sets q = 0 throughout. All five harnesses receive the retained assumption in the primary growth comparison.

Table C5. Codex one-hour, 20-request input-only API-equivalent projection. All initial prefixes and user payloads are equal; only mean observed excess growth differs in the first five rows. Identical prefix availability and zero excess growth yield identical costs.

Configuration	U tokens	C tokens	Input USD	Read USD	Total
AGNT / retained	106,490	1,176,860	$1.065	$1.177	$2.242
Codex CLI / retained	132,401	1,410,061	$1.324	$1.410	$2.734
OpenClaw / retained	106,775	1,179,425	$1.068	$1.179	$2.247
OMP / retained	106,490	1,176,860	$1.065	$1.177	$2.242
Hermes / retained	106,490	1,176,860	$1.065	$1.177	$2.242
Common / break loss	354,250	929,100	$3.543	$0.929	$4.472
Common / no cache	1,283,350	0	$12.834	$0.000	$12.834

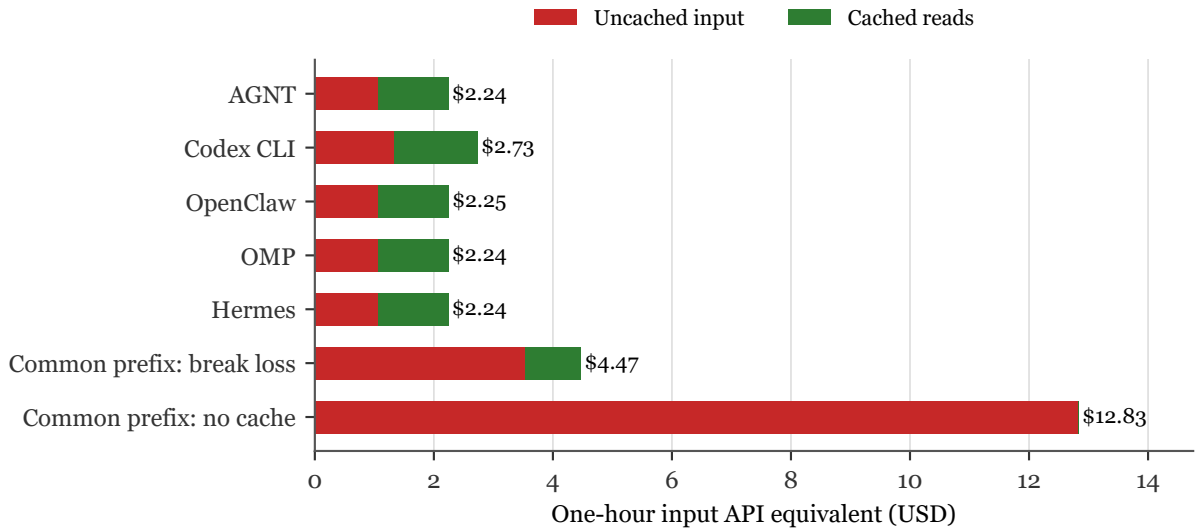

Figure C4. Codex one-hour projection at GPT-6 Astra prices: common initial context, common payload, mean observed excess growth, and retained-prefix assumption for every harness. The final two bars are a common-content miss-after-each-break scenario and an uncached control; they are not attributed to any harness. Lower dollar values are better for this analytical workload.

Table C6. Codex cost multipliers relative to the zero-extra-growth uncached common workload of the same length. Long-context prices apply above 272,000 input tokens. The model does not guarantee provider retention over a multi-hour session.

Harness / retained	15 m	30 m	1 h	2 h	4 h
AGNT	0.332×	0.233×	0.175×	0.140×	0.118×
Codex CLI	0.373×	0.274×	0.213×	0.176×	0.173×
OpenClaw	0.333×	0.234×	0.175×	0.141×	0.120×
OMP	0.332×	0.233×	0.175×	0.140×	0.118×
Hermes	0.332×	0.233×	0.175×	0.140×	0.118×
Common / break loss	0.538×	0.423×	0.348×	0.339×	0.324×

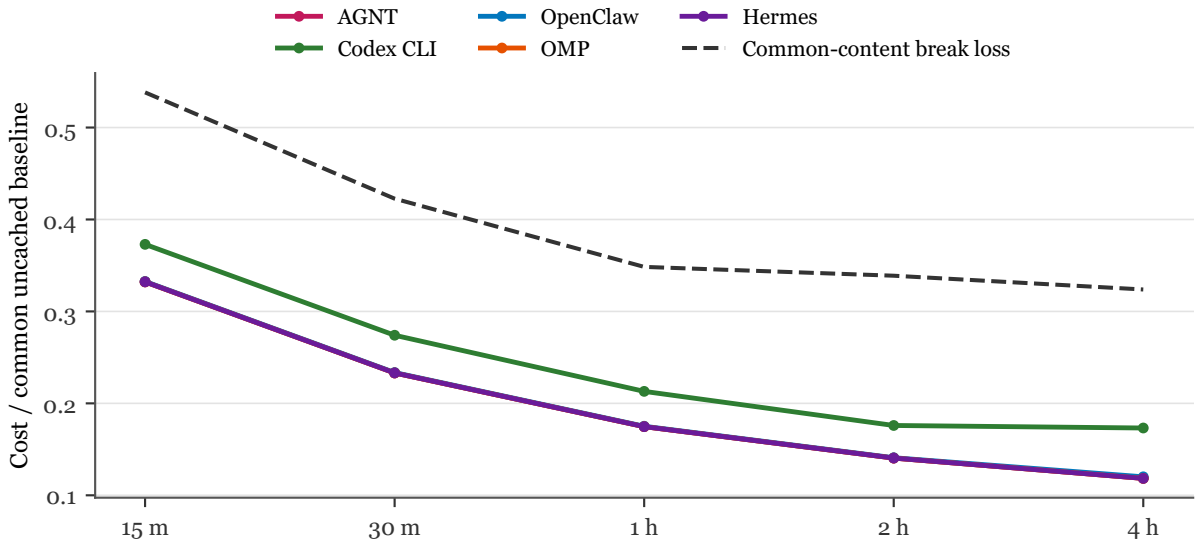

Figure C5. Codex cost multipliers from 15-minute to four-hour scenarios. All harness lines assume the prior prefix remains available. The common-content break-loss line is a sensitivity test. Requests above 272,000 input tokens use long-context rates for their full input; this price threshold is included rather than extrapolating short-context prices indefinitely.

At one hour the retained zero-extra-growth model costs $2.242; assumed loss after each fourth request costs $4.472; uncached costs $12.834. AGNT, OMP and Hermes tie in the common-content retained model after tokenizer-scale residuals are rounded to zero. Codex CLI's higher measured mean growth raises its projection to $2.734. AGNT's measured pause-ratio win must not be recast as an exclusive price advantage when content and cache availability are identical.

7 Monthly and Annual Extrapolation

The Claude monthly analysis retains the original paper's fixed-work-per-hour normalization: a one-hour reference burn rate multiplied by session-length cache-efficiency factors. It is a rate-normalized budget scenario, not the exact token sum of extending one growing transcript to 80 requests. Each scenario assumes cold starts and the stated marker policy. Let R_u = $3.37/h denote the uncached burn rate of the reference workload (20 messages/h at 5,265 tokens each over the 34,114-token prefix) and M_X(L) the session-length-dependent cost multiplier of harness X from §6 (Table 8). The rate-normalized monthly budget over D workdays with a fixed daily session schedule S is

$$\mathrm{Cost}_X^{\mathrm{month}} = D \cdot \sum_{L \in S} M_X(L) \cdot L \cdot R_u \tag{3}$$

We instantiate three usage profiles at D = 22 workdays (Table 9) and evaluate with the measured multipliers (Table 10).

Table 9: Usage profiles for monthly extrapolation.

Profile	Daily usage	Session shape	Multiplier basis
Light	2 h/day (44 h/mo)	2×1 h sessions	M(1 h)
Moderate	4 h/day (88 h/mo)	2×2 h sessions	M(2 h)
Heavy	8 h/day (176 h/mo)	2×4 h sessions	M(4 h)

Table 10: Projected monthly cost per seat (USD; 22 workdays; Sonnet 5 list prices; identical content).

Harness	Light	Moderate	Heavy
AGNT v0.6.6	\$37	\$54	\$85
Claude Code	\$37	\$55	\$86
OMP (default 5 m)	\$61	\$120	\$235
OpenClaw	\$62	\$120	\$236
Hermes	\$61	\$120	\$235
No caching (control)	\$148	\$296	\$592

Two structural effects emerge at monthly scale. First, the TTL gap widens with load: because the 1-hour multipliers are monotone-decreasing in session length (0.25 at 1 h $\rightarrow$ 0.14 at 4 h) while the 5-minute defaults flatten (0.41 $\rightarrow$ 0.40), AGNT's advantage over the 5-minute-default harnesses grows from 1.6–1.7$\times$ (light) to $\approx$2.8$\times$ (heavy). Second, the ranking has compressed since [1]: with both pathologies gone, Claude Code now tracks AGNT to within \$0.57/month at the heavy profile, and the three 5-minute-default harnesses cluster within \$0.77/month of one another — the dominant difference in this specified default-policy model is retention.

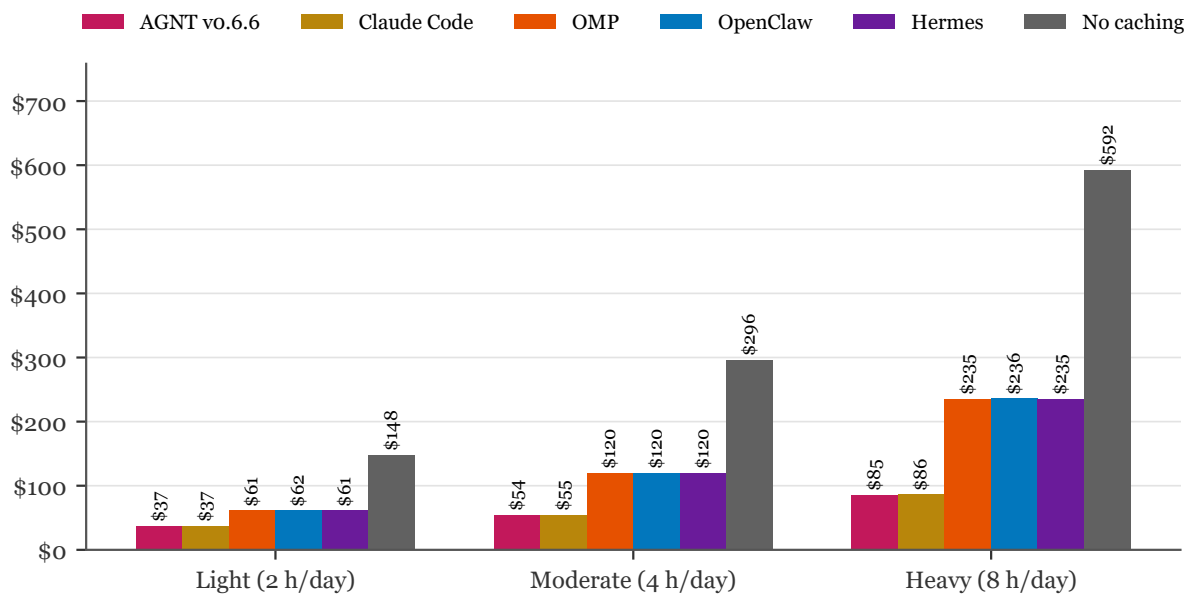

Figure 7: Projected monthly cost per seat under the three usage profiles of Table 9 (22 workdays; Sonnet 5 list prices; identical content). The 5-minute-default harnesses cost 1.65–2.76× AGNT at every profile; the gap is the TTL bit alone. Claude Code tracks AGNT to within its 40-token overhead.

7.1 The cache tax

Define the cache tax of harness X as Cost_X - Cost_AGNT for identical work on the identical model. At the heavy profile (Table 11), the tax is $150 per seat per month; a heavy OMP, OpenClaw or Hermes seat on its shipping default spends AGNT's entire monthly budget by approximately day 8.

Table 11: Cache tax vs. AGNT v0.6.6, heavy profile (8 h/day, 22 workdays).

Harness	$/month	$/year	Dominant mechanism
OMP (default 5 m)	150	1,796	full-context re-write after every >5 m break (no overhead; 1 h available via flag)
OpenClaw	150	1,805	full-context re-write at 1.25× after every >5 m break (+20 tk/turn; 1 h available via cacheRetention)
Hermes	150	1,796	full-context re-write at 1.25× after every >5 m break (1 h available via cache_ttl)
Claude Code	0.57	6.84	40 tokens/turn residual overhead at 1 h premium
No caching	507	6,083	all tokens at list price

7.2 Team scale

Annualized over a five-seat team at the heavy profile: AGNT $5,123; Claude Code $5,157; OMP (default) $14,104; Hermes $14,104; OpenClaw $14,150; uncached $35,537. The OpenClaw-vs-AGNT delta alone is ≈$9,027/year for byte-identical work — the cost of one configuration bit, not a model upgrade.

Three properties of this extrapolation bias it against the headline gap rather than for it: (i) the session model assumes breaks only every ≈12 minutes, more frequent pauses would add re-write events to the 5 m scenario but were not measured here; (ii) the identical-content prefix is held at the measured first-turn

size, though real agent contexts grow with tool results and file contents, scaling every re-write linearly; and (iii) the model charges the 5 m harnesses nothing for intra-burst expiry. These are scenario sensitivities, not proved lower bounds for real workloads.

7.3 The agentic amplifier: tool latency as involuntary pause

The pause test models the idle gap as a human pause, but nothing in the mechanism requires a human. An agent turn is a chain of model calls separated by tool executions, and any tool that runs longer than the marker TTL expires the cache mid-turn: build systems and test suites (2–20 min), media-generation polling (5–10 min per asset), rate-limit backoff, CI and deployment waits, sub-agent delegation, and approval gates all routinely exceed five minutes with no user absent. On a 5 m-TTL harness, every such tool forces a full-context re-write at $1.25\times$ for the next model call in the same turn.

The per-event cost scales with working-context size. At a realistic agentic context of 100k tokens (Sonnet 5 list, \$2/M input), one post-expiry re-write bills 100k $\times$ \$2/M $\times$ 1.25 = \$0.25. An autonomous workload executing 30 long-running tools per day on a 5 m-TTL harness therefore pays ≈\$7.50/day — ≈\$165/month per seat — in gross post-expiry input charges; under the stated sub-hour gaps a retained one-hour read would be charged at the read rate instead, in addition to the human-pause modeling above, and growing linearly with context length. The exposure is thus largest precisely where agent harnesses do their most valuable work: long-context, tool-heavy autonomous sessions. The inversion noted in [1] stands: the harnesses marketed most explicitly for autonomous multi-step operation (OpenClaw, Hermes) still ship the TTL least compatible with it.

7.4 Codex monthly and annual extrapolation

For Codex the same light, moderate and heavy profiles are evaluated as 44 independent sessions per month with 20, 40 or 80 requests per session. Monthly cost is the exact per-session sum multiplied by 22 days $\times$ 2 sessions/day; annual cost is 12 times monthly. This includes the full-request long-context threshold. It is not a fixed one-hour burn rate applied after the context has grown.

$$K_X^{\mathrm{month}}(n) = 44K_X(n), \quad K_X^{\mathrm{year}}(n) = 12 \cdot 44K_X(n), \quad K_{5\mathrm{seat}}^{\mathrm{year}}(n) = 5 \cdot 12 \cdot 44K_X(n). \tag{4}$$

Table C7. Codex monthly and annual input API equivalents at current standard GPT-6 Astra prices. These are analytical workloads, not subscription prices. Retention-loss rows are scenarios rather than measured behavior attributed to a harness.

Configuration	Light / mo	Moderate / mo	Heavy / mo	Heavy / yr
AGNT / retained	\$98.64	\$268.80	\$1,213.60	\$14,563.16
Codex CLI / retained	\$120.30	\$336.66	\$1,775.29	\$21,303.43
OpenClaw / retained	\$98.88	\$269.54	\$1,231.07	\$14,772.79
OMP / retained	\$98.64	\$268.80	\$1,213.60	\$14,563.16
Hermes / retained	\$98.64	\$268.80	\$1,213.60	\$14,563.16
Common / break loss	\$196.75	\$648.33	\$3,319.47	\$39,833.63
Common / no cache	\$564.67	\$1,913.43	\$10,250.00	\$123,000.00

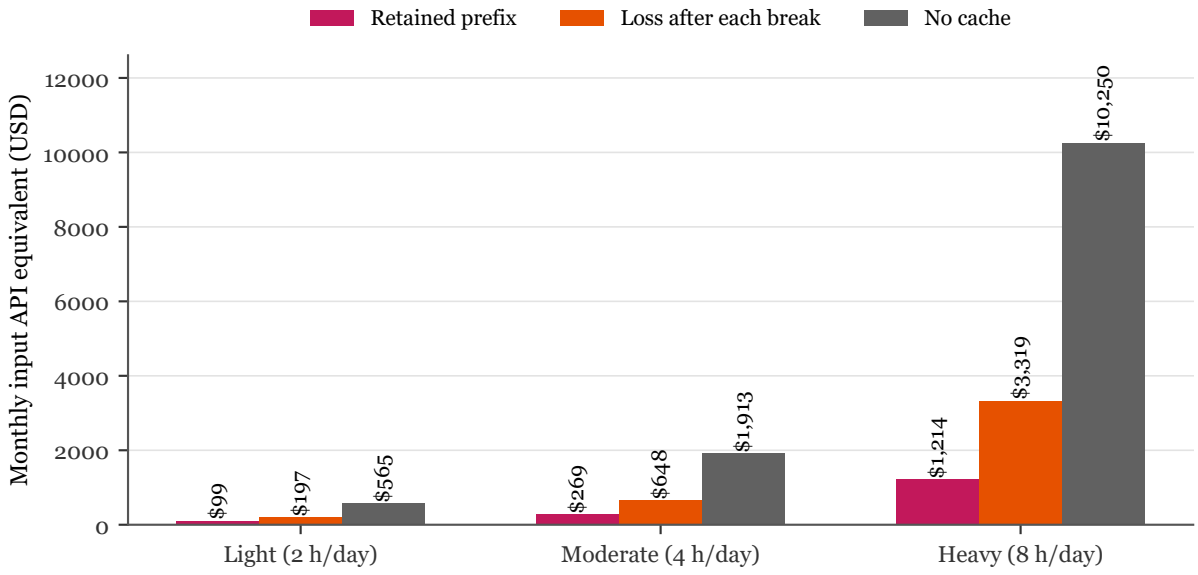

Figure C6. Codex monthly per-seat scenarios for exactly 44 independent sessions of 20, 40, or 80 requests. Retained, break-loss and uncached common-content costs use the full length of each session and the long-context threshold. These are API-equivalent workload projections, not subscription prices or measured quota deductions.

7.5 Codex cache-loss tax and team scale

Table C8. Codex cache-loss tax on strictly identical content. No miss frequency has been inferred from the five-request measurement; the loss schedule is specified in §6.2.

Scenario vs retained	Heavy Δ / mo	Heavy Δ / yr	5-seat Δ / yr
Loss after each break	$2,105.87	$25,270.47	$126,352.35
No reuse throughout	$9,036.40	$108,436.84	$542,184.19

For the heavy common-content profile, retained-prefix input costs $1,213.60/seat/month; assumed break loss costs $3,319.47; no reuse costs $10,250.00. The break-loss difference is $2,105.87 per seat per month. A five-seat team scales the annual difference by 60, as shown above. This quantifies exposure to misses, not a claim that any particular Codex client experiences that pattern.

7.6 Codex agentic amplifier

At 100,000 input tokens, a completely uncached GPT-6 Astra request has a $1.00 input API equivalent, versus $0.10 if that entire existing prefix is read from cache. The incremental loss is $0.90 per event; 30 such events per day over 22 days would be $594 per month. No extra $12.50/M cache-write charge is added because the observed subscription counters report none. Unlike the Claude five-minute setting, a five-minute tool wait is not itself evidence of Codex expiration. This scenario applies only when reuse is actually lost; it is not a measured tool-loop bill or a separate additive charge if those same misses are already counted in the session model.

8 Ancillary Finding: Subscription Access

In [1] we observed Anthropic reject a third-party harness presenting a valid Claude-subscription OAuth token (HTTP 400, "Third-party apps now draw from your extra usage, not your plan limits"), and every third-party arm had to be run on a funded API key. In this study every arm ran on a subscription credential: the Claude subscription for AGNT, OMP, Claude Code, OpenClaw and Hermes, and the ChatGPT subscription via Codex for AGNT, Codex CLI, OpenClaw, OMP and Hermes. The 65 accepted requests completed successfully on their subscription paths. The receipts establish successful service, not which plan allowance or any overage bucket funded it. The receipts do not reveal which allowance the usage was drawn from, and subscription usage is metered against plan limits rather than invoiced per token — Claude Code's own documentation states that its session dollar figure is an estimate, not the bill for included plan usage [11]. The dollar figures in §§5–7 are therefore API-list equivalents of the measured counters: what the same work costs at metered rates, not a measured conversion to subscription allowance. The caching results themselves are independent of the billing mode.

8.1 Codex subscription versus metered OpenAI access

The Codex sequences authenticated through the connected ChatGPT subscription on all five paths. This establishes technical access during these trials; it does not identify the account allowance from which usage was deducted, certify third-party product approval, or convert cached tokens to plan credits. The measured input/read/output counters and the published Astra price list support the API-equivalent tables. No OpenAI API top-up was required for these accepted Codex runs, and no separate API bill was measured.

8.2 Separate accounting for both provider families

The same distinction applies to Claude: subscription-authenticated successful usage is not proof that a locally estimated dollar amount was charged or saved as cash. A plan comparison must include actual subscription fees, included allowance, overage and workload quality. The paper's Claude and Codex dollar models are in separate price units. A low-cost row on one provider must not be compared to another model as if model capability and account terms were held constant.

9 Threats to Validity

Single trial per cell. Anthropic TTL expiry is contractual and the observed partition follows it exactly, but each Codex-path survival is one sample of a load-dependent, unguaranteed mechanism, and OMP's Codex turn-4 miss is a single unexplained event.

Single idle duration. $\geq$390 s cleanly separates the two TTLs; gaps > 1 h would defeat every harness tested absent keep-warm traffic.

Synthetic payloads. Deterministic filler standing in for real work. Real sessions grow faster (tool results) and pause more often; both effects increase the measured gaps' magnitude, not their direction.

Footprint heterogeneity. Absolute token counts are incomparable across harnesses by construction — AGNT carried its full resident instruction context and tool schemas (34.1k first-turn prefix), the other harnesses isolated reduced-tool configurations (6.6k–13.8k); all cross-harness claims use ratios, payload-normalized overhead, or the identical-content model (§4.4).

Warm first turns. Provider caches cannot be flushed; AGNT and Claude Code read 3,949 and 3,289 shared-prefix tokens on turn 1, and Codex CLI 12,288. The pause test is unaffected (it compares turn 5 to turn 4),

and the cost model starts every session cold.

Extrapolation assumptions. Fixed break schedule and payload size; the model is linear in measured per-turn parameters and stated in full (§6) so readers may re-parameterize.

Monthly extrapolation. §7 composes single-session measurements linearly across fixed daily profiles; real months vary payload size, schedule, and break structure. The measurements do not establish the distribution of real pause lengths, prompt growth or task quality; the monthly values are conditional projections.

Subscription telemetry. Dollar figures are list-price equivalents of counters obtained on subscription paths, not invoices (§8); ratios and token counts are unaffected.

Version pinning. Results describe the versions in Table 1 as of 2026-09-08; harness caching behavior can change across releases — as the disappearance of both pathologies of [1] demonstrates.

Codex projections use the current Platform price table with the 272,000-input-token threshold, but the subscription backend is a distinct access surface. The retained-prefix and break-loss schedules are assumptions, not forecast probabilities. Source fingerprints and recorded counter shapes identify what was observed; they do not establish a long-run error rate or universal cross-harness cost ordering.

10 Reproducibility

Codex replication assets have the same status as Claude assets: all five Codex paths, all 25 measured request receipts, original counter objects, prompt digests, and per-request price reconciliation are included. codex-scenario.json contains every request in each projection, including the short/long price multiplier; codex-figure-data.csv contains the measured chart rows. Figures C1–C6 and Tables C1–C8 are rebuilt from those files and verified independently.

All measurements derive from the published artifact set (`https://agnt.gg/whitepapers/cache-wars-2-artifacts/artifacts/index.html`
`https://agnt.gg/whitepapers/cache-wars-2-artifacts/SHA256SUMS.txt`):

Table 12: Artifact set.

File	Contents
artifacts/data/measurements.json	All 13 configurations × 5 turns: normalized counters, prompt digests, gaps, per-turn API-equivalent cost
artifacts/receipts/<arm>-turn<t>.json	65 receipts: the original harness/provider usage object, reply, model, session gap and prompt SHA-256 for every accepted request
artifacts/data/workload-{claude,codex}.json	The five per-turn prompts of each track with byte counts and SHA-256 digests
artifacts/environment/versions.json, package-lock.json, hermes-uv.lock	Pinned harness versions, runtimes and dependency locks; AGNT source fingerprints
artifacts/figures/figure-{1..7}.{svg,pdf}, figure-c{1..6}.{svg,pdf}, figure-data.csv, scenario.json, codex-scenario.json	Original chart set plus six Codex charts; measured rows and complete Claude/Codex model quantities
artifacts/scripts/build.py	Regenerates the figures, tables, HTML and LaTeX of this paper from measurements.json
artifacts/scripts/verify.py	Offline verifier: manifest, raw-usage reconciliation, guards, cost model and published numbers
artifacts/scripts/live_protocol.py	Live driver: per-harness invocation, guards G1–G3, 390 s idle, receipt capture
SHA256SUMS.txt	Integrity manifest over every released file except itself

Replication procedure: (1) install the six harnesses at the pinned versions; (2) authenticate each on its subscription path in an isolated home directory (Claude Code via `CLAUDE_CONFIG_DIR`, Codex via `CODEX_HOME`, OMP via `PI_CODING_AGENT_DIR`, OpenClaw via `OPENCLAW_HOME`, Hermes via `HERMES_HOME`; AGNT on a separately provisioned isolated host with its own data directory — never a production database); (3) run `live_protocol.py -harness ... -track ... [-retention long]`, which issues four turns of Listing-1 payloads, sleeps 390 s, issues turn 5, and aborts on any guard violation; (4) run `verify.py` to re-derive every counter in Tables 2–6 from the receipts' raw usage objects and every cost-model quantity in Tables 7–11 and C5–C8 from §6–7; (5) run `build.py -output NEW_DIRECTORY` to regenerate this manuscript and its figures from the same data. Every number in this paper is either a raw counter from these files or an arithmetic combination shown in the text.

11 Conclusion

Prompt-cache efficiency in deployed agent harnesses is determined by client-side engineering, and shipped behavior still diverges from what documentation implies — though less than it did two months ago. A single configuration bit — cache TTL — separates harnesses that resume an interrupted session at 10% of list price from harnesses that silently re-purchase their entire context after every human-scale pause. The two pathologies layered on top of it in [1] have been engineered away: Claude Code's per-turn scaffolding has fallen from 6,817 tokens to 40, and Hermes' marker placement now extends the cached prefix over history. What has not changed is the default: OMP, OpenClaw and Hermes still ship the 5-minute TTL that forfeits the pause, and each still requires a documented opt-in to survive it. AGNT v0.6.6 combines default one-hour Claude markers, Claude writes equal to the measured input increment, and successful Codex reuse — posting the highest post-pause cache-hit ratio on the Claude path (90.45%) and the Codex path (88.42%) in these trials. Under the stated Claude default-policy projection AGNT attains the lowest cost, tied with other zero-extra-growth one-hour configurations when those options are enabled. Codex cost projections are conditional on prefix availability; equal content and equal availability imply equal price, not an exclusive harness discount. Composed over a working month, the surviving mechanism compounds into a per-seat cache tax of \$150/month (\$1,796–\$1,805/year) for the 5-minute-default harnesses, and ≈\$9,027/year for a five-seat team in the worst measured case — recurring spend separable from model quality entirely, and recoverable by client-side engineering alone.

References

[1] The Cache Wars: An Empirical Study of Prompt-Cache Efficiency in Six LLM Agent Harnesses. Technical Report v2.0, AGNT Labs, July 7, 2026. `https://agnt.gg/whitepapers/the-cache-wars-prompt-cache-efficiency-llm-agent-harnesses`

[2] Anthropic. Prompt caching; Model and cache pricing. Claude Platform Documentation (accessed 2026-09-08). Sonnet 5: input \$2.00/M; cache read \$0.20 (0.1×); 5-minute write \$2.50 (1.25×); 1-hour write \$4.00 (2×). `https://platform.claude.com/docs/en/build-with-claude/prompt-caching`

[3] OpenAI. API pricing. OpenAI Developer Documentation (accessed 2026-09-08). gpt-6-astra standard short-context: input \$10.00/M; cached input \$1.00; cache write \$12.50; output \$50.00. Automatic prefix caching; model-specific retention controls are documented separately [12]. `https://developers.openai.com/api/docs/pricing`

[4] OpenAI. Using Codex with your ChatGPT plan. OpenAI Help Center (accessed 2026-09-08). https://help.openai.com/en/articles/11369540-using-codex-with-your-chatgpt-plan

[5] OpenClaw. Anthropic provider; Model providers and runtime selection. Technical reference (accessed 2026-09-08). cacheRetention: none|short|long; "short" (5 m) seeded by default for Anthropic. https://docs.openclaw.ai/providers/anthropic

[6] Nous Research. Hermes Agent release v2026.9.7 (0.21.1). GitHub release (accessed 2026-09-08). https://github.com/NousResearch/hermes-agent/releases/tag/v2026.9.7

[7] oh-my-pi. @oh-my-pi/pi-coding-agent 18.1.14. npm registry record (accessed 2026-09-08). https://registry.npmjs.org/@oh-my-pi/pi-coding-agent/18.1.14

[8] Anthropic. @anthropic-ai/claude-code 2.1.263. npm registry record (accessed 2026-09-08). https://registry.npmjs.org/@anthropic-ai/claude-code/2.1.263

[9] OpenAI. @openai/codex 0.153.4. npm registry record (accessed 2026-09-08). https://registry.npmjs.org/@openai/codex/0.153.4

[10] OpenClaw. openclaw 2026.9.2. npm registry record (accessed 2026-09-08). https://registry.npmjs.org/openclaw/2026.9.2

[11] Anthropic. Claude Code: manage costs effectively. Claude Code documentation (accessed 2026-09-08). The session cost display is an estimate and does not reflect subscription billing. https://code.claude.com/docs/en/costs

[12] OpenAI. Prompt caching. Platform model-dependent retention settings and automatic prefix reuse (accessed September 8, 2026). https://developers.openai.com/api/docs/guides/prompt-caching

[13] OpenAI. GPT-6 Astra. Standard pricing and full-request long-context threshold above 272K input tokens (accessed September 8, 2026). https://developers.openai.com/api/docs/models/gpt-6-astra

Technical Report · AGNT Labs · Benchmark executed 2026-09-08 · Models: claude-sonnet-5 (Claude subscription) + gpt-6-astra (ChatGPT subscription via Codex) · Idle $\geq$390 s all arms · Payload $\approx$5,265 Sonnet 5 tokens/turn all Anthropic-path arms · Correspondence: AGNT project.