

Dulus

Benchmark Arena

Dulus leads under every weight profile tested.

Field: Dulus, Claude Code, Cursor, Aider, GitHub Copilot (agent mode), Cline, Codex CLI, Naive full-context loop.

Scenarios: Quick fix, Feature build, Deep refactor, Marathon audit, Retrieval-heavy, Feature build on a cheap model.

```
Run          2026-10-18T06:07:05+00:00
Commit       097aa216dc092fd4
Arena version 1.0.0
Wall clock   7.2s
Rank stability 100% of weight profiles agreed
```

What is measured and what is modelled

This document separates three kinds of statement, and the distinction is load-bearing rather than decorative.

- ? **MEASURED** - produced by executing code in this repository. Token counts come from `compaction.estimate_tokens`; USD from `providers.calc_cost` including the cache-read discount. The Dulus context policy runs the repository's own `snip_old_tool_results`, `strip_stale_injections`, `find_split_point` and `apply_lookback_window`, so its cache-payoff gate and anchor hysteresis are genuinely exercised across the session.
- ? **MODELLED** - a competitor's context policy is a behavioural shape (grow, window, or summarise-at-threshold) replayed against the identical transcript. It is not that product's source code. Where the model is generous to the competitor, this document says so.
- ? **CLAIMED** - a capability or price taken from somebody's published material. Every such row carries its tier and citation, and unknowns are recorded as unknown rather than scored as zero.

Controls against self-flattery

- ? A naive full-context control is always in the field, so a reader can tell how much of any Dulus lead is Dulus being good versus the field being modelled badly.
- ? Every ranking is recomputed under several published weight profiles; the headline carries the share that agreed.
- ? Cost ratios are paired bootstrap intervals over per-turn costs. An interval touching parity is reported as no reliable difference, not as a win.
- ? Unknown values are excluded from a metric's normalisation instead of floored at zero, so a competitor never loses a point for our incomplete research.
- ? Fidelity is scored as well as cost, because saving money by showing the model less is not the same thing as efficiency.

Provenance tiers

Tier	Meaning
MEASURED	This suite computed it by running repository code
DOCUMENTED	Taken from a primary source that was read directly
VENDOR CLAIM	A vendor's published figure; verify before publishing
SELF-REPORTED	A claim made by this repository's own docs
MODELLED	Derived from a stated assumption; not a measurement
NO DATA	Not found; skipped by the scorer

The field

Harness	Context policy	Mode	Provenance	Evidence
Dulus	Dulus (lookback window + two-layer com...	dulus	MEASURED	92%
Claude Code	Full context + late auto-compact	threshold_compact	MODELLED	40%
Cursor	Editor-scoped context + mid-window sum...	threshold_compact	MODELLED	41%
Aider	Repo map + full chat history	full	MODELLED	44%
GitHub Copilot (agent...	Bounded turn window, retrieval-assembled	hard_window	MODELLED	43%
Cline	Visible context bar + truncate-on-pressure	threshold_compact	MODELLED	40%
Codex CLI	Full context + late compaction	threshold_compact	MODELLED	37%
Naive full-context loop	Full transcript, no cache, no trimming	full	MEASURED	100%

Results, summed across every scenario

Harness	Sessio...	Billed...	Prompt...	Cache...	Peak c...	Conte...	Histor...	Verbat...	Capab...
Dulus	\$25.63	18.52M	61.2%	9	139.8k	0	100.0%	28.0%	100.0%
Claude Code	\$48.00	32.13M	62.2%	74	183.4k	1	100.0%	46.1%	30.0%
Codex CLI	\$49.38	31.59M	60.5%	79	183.2k	1	100.0%	45.9%	30.0%
Cursor	\$50.75	30.23M	57.4%	88	183.0k	1	100.0%	44.8%	50.0%
Cline	\$53.65	27.86M	52.3%	102	182.5k	1	100.0%	43.0%	52.6%
Aider	\$92.49	243.97M	92.4%	0	2.36M	240	100.0%	100.0%	36.8%
GitHub Copilot (agent mode)	\$373.53	71.71M	26.6%	220	604.7k	222	100.0%	66.9%	44.4%
Naive full-context loop	\$719.29	243.97M	0.0%	0	2.36M	240	100.0%	100.0%	0.0%

Cost, tokens, cache-prefix breaks and overflows are summed across scenarios; peak context is the maximum observed; ratios are means. A single context overflow is a hard failure, not a slow turn.

Ranking under different weight profiles

Profile	Dulus	Claud...	Cursor	Aider	GitHu...	Cline	Codex...	Naive f...	Winner
neutral	0.847	0.752	0.760	0.586	0.489	0.748	0.748	0.333	Dulus
cost-first	0.931	0.856	0.849	0.628	0.469	0.836	0.851	0.158	Dulus
fidelity-first	0.764	0.751	0.757	0.687	0.595	0.749	0.748	0.556	Dulus
capability-first	0.934	0.590	0.683	0.487	0.487	0.690	0.589	0.171	Dulus
arena-default	0.909	0.810	0.824	0.526	0.488	0.818	0.806	0.235	Dulus

Rank stability 100%: the share of profiles that agreed on first place. A ranking that only holds under the author's weights is not a finding.

00 Scorecard

Dulus Benchmark Arena

8 harnesses across 6 scenarios, totals summed over every scenario.

Dulus leads under every weight profile tested.

DULUS TOTAL COST \$25.63	VS CLAUDE CODE (BEST RIVAL) 1.87x	STRATEGIES THAT OVERFLOWED 7 of 8	WEIGHT PROFILES AGREEING 100%
MODEL PRICE SPREAD 205.96x	AGENT CORE, COUNTED 25,822 lines	PROVIDERS REGISTERED 40	DULUS CACHE-PREFIX BREAKS 9

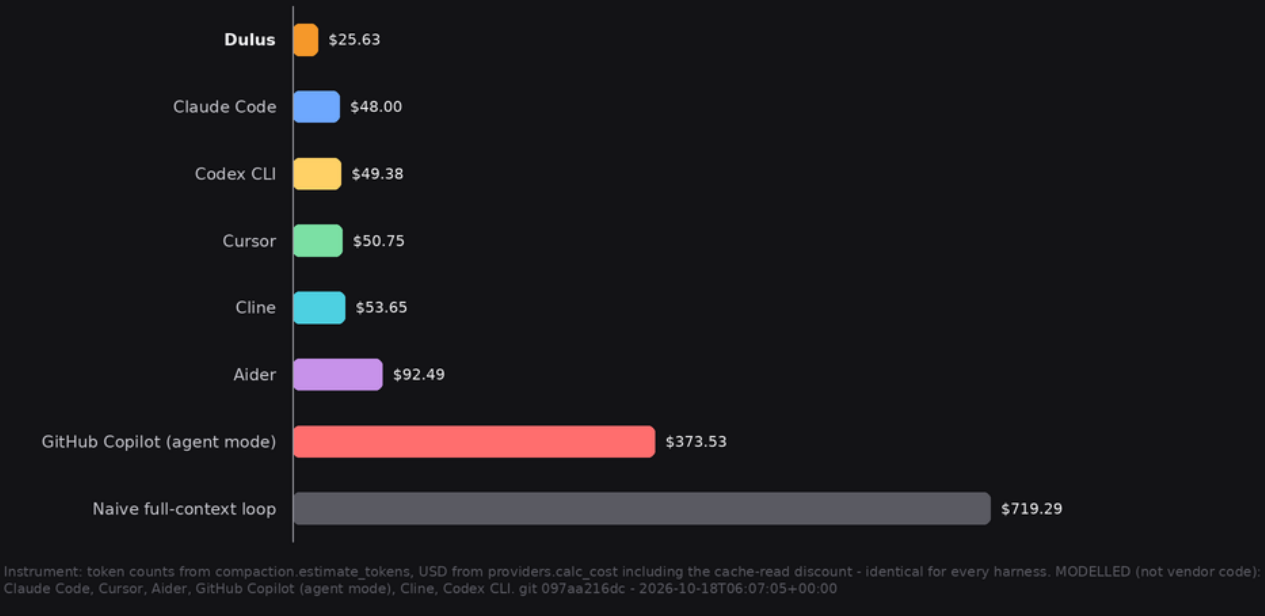
Instrument: token counts from compaction estimate, tokens, USD from providers.calc_cost including the cache-read discount - identical for every harness. MODELLED (not vendor code): Claude Code, Cursor, Aider, GitHub Copilot (agent mode), Cline, Codex CLI. git 097aa216dc - 2026-10-18T06:07:05+00:00

Headline figures. Totals are summed across every scenario in the run.

01 Cost Ranking

Total session cost (lower is better)

Summed across 6 scenarios on the same transcripts.

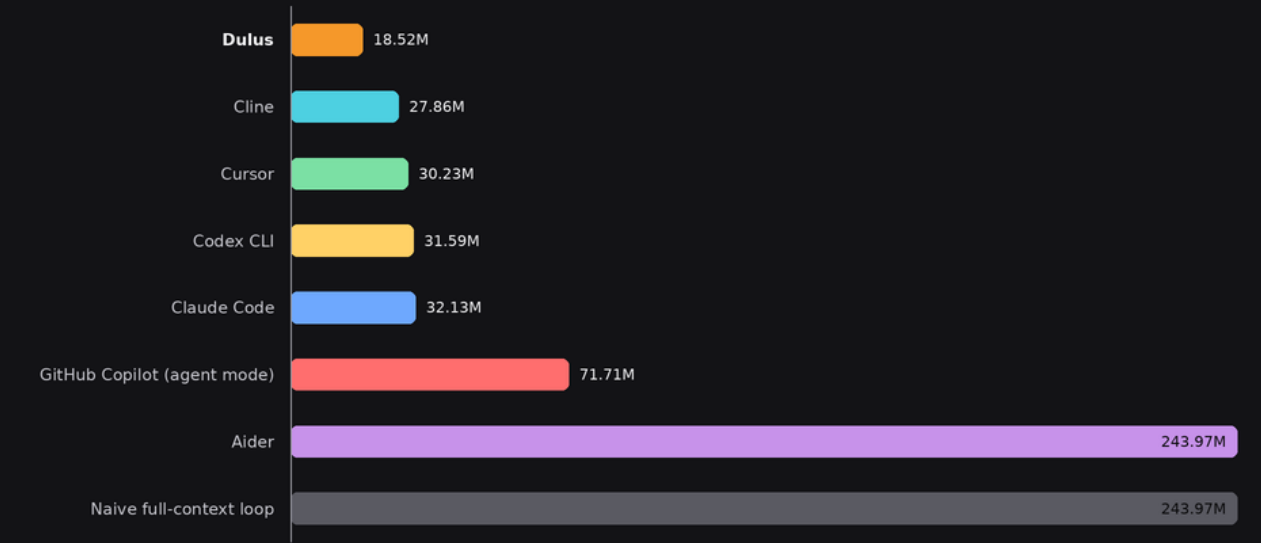


Billed USD for the whole run, cache discount included.

02 Token Ranking

Billed prompt tokens (lower is better)

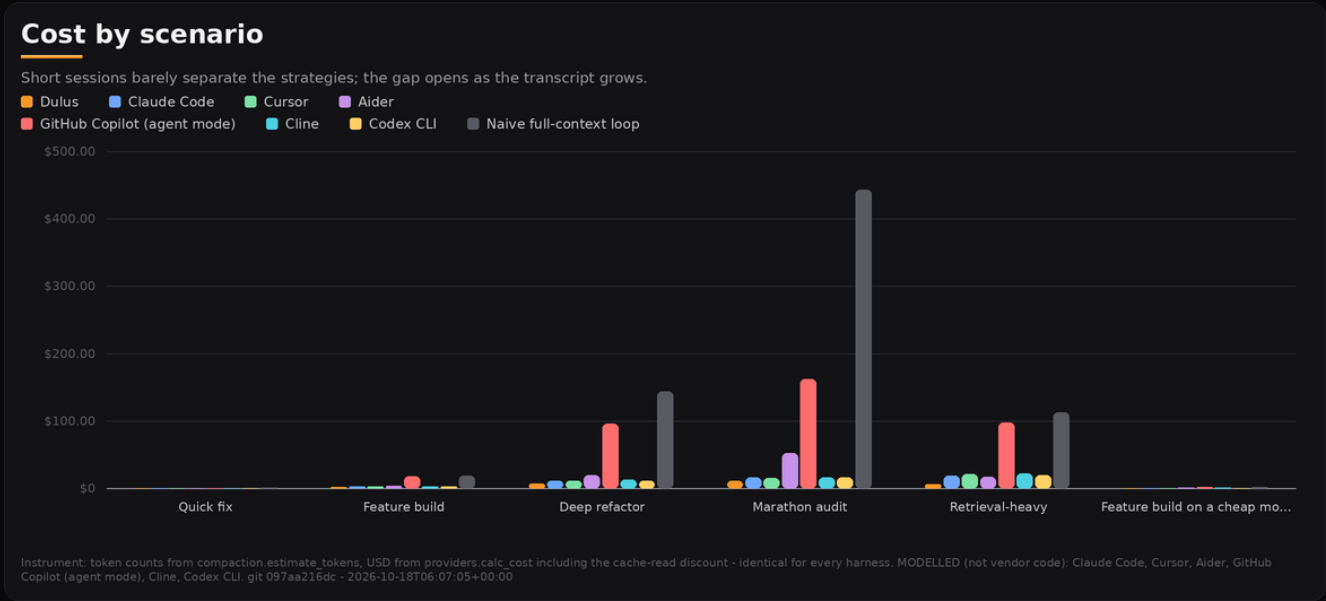
Tokens actually put on the wire, before the cache discount is applied.



Instrument: token counts from compaction.estimate_tokens, USD from providers.calc_cost including the cache-read discount - identical for every harness. MODELLED (not vendor code): Claude Code, Cursor, Aider, GitHub Copilot (agent mode), Cline, Codex CLI. git 097aa216dc - 2026-10-18T06:07:05+00:00

Wire volume. Cheap tokens still cost latency and still risk the window.

03 Cost By Scenario



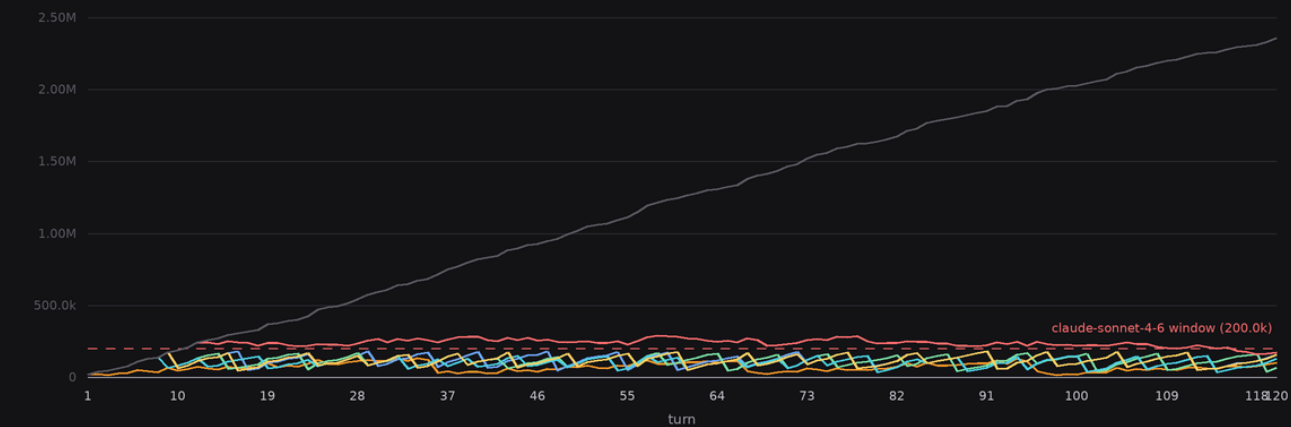
Where each strategy starts to matter.

04 Context Growth

Prompt size per turn - Marathon audit

Anything above the model window is a hard failure, not a slow turn.

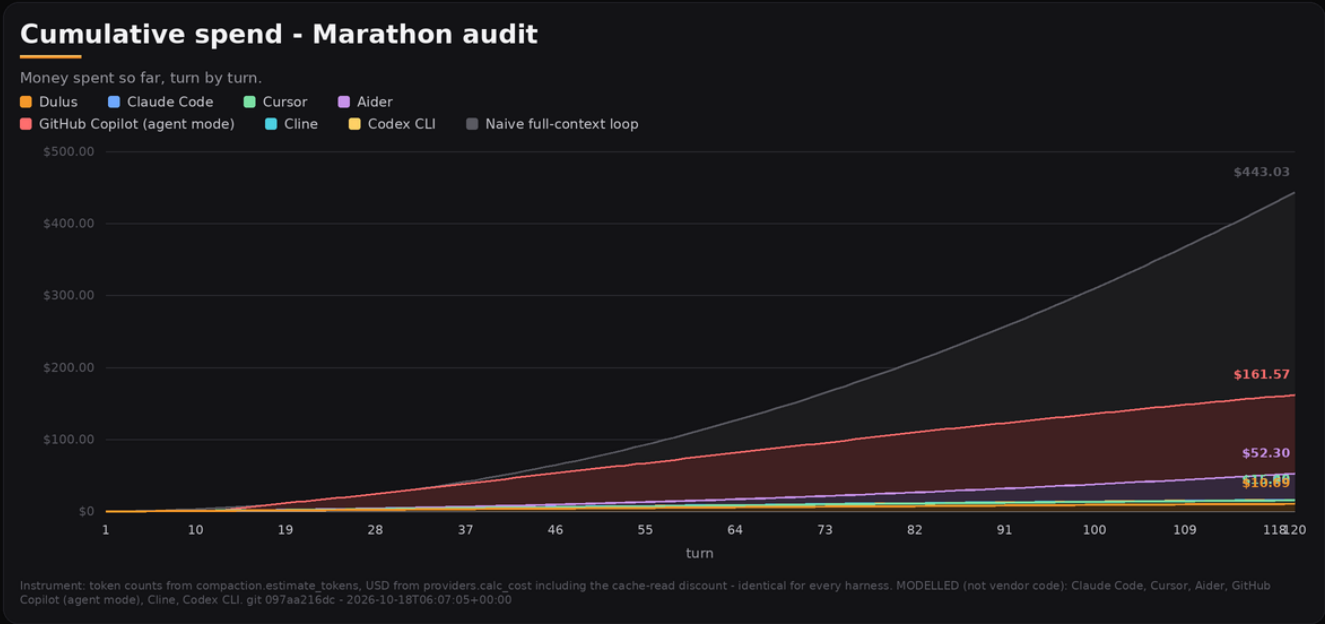
Dulus Claude Code Cursor Aider
GitHub Copilot (agent mode) Cline Codex CLI Naive full-context loop



Instrument: token counts from compaction.estimate_tokens, USD from providers.calc_cost including the cache-read discount - identical for every harness. MODELLED (not vendor code): Claude Code, Cursor, Aider, GitHub Copilot (agent mode), Cline, Codex CLI. git 097aa216dc - 2026-10-18T06:07:05+00:00

The context-growth curve, with the real model window drawn in.

05 Cumulative Cost



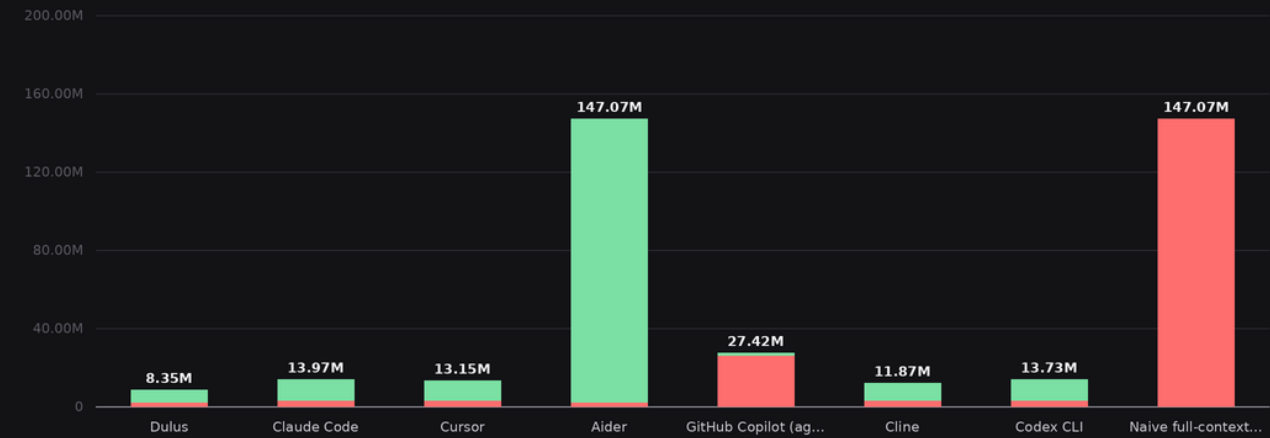
Divergence compounds: the gap is widest exactly where long sessions live.

06 Token Composition

What the tokens were - Marathon audit

The cache-read share is the whole game: a strategy that busts the prefix pays full price for tokens it already sent.

■ Fresh (full price) ■ Cache read (0.10x on Anthropic)



Instrument: token counts from compaction.estimate_tokens, USD from providers.calc_cost including the cache-read discount - identical for every harness. MODELLED (not vendor code): Claude Code, Cursor, Aider, GitHub Copilot (agent mode), Cline, Codex CLI. git 097aa216dc - 2026-10-18T06:07:05+00:00

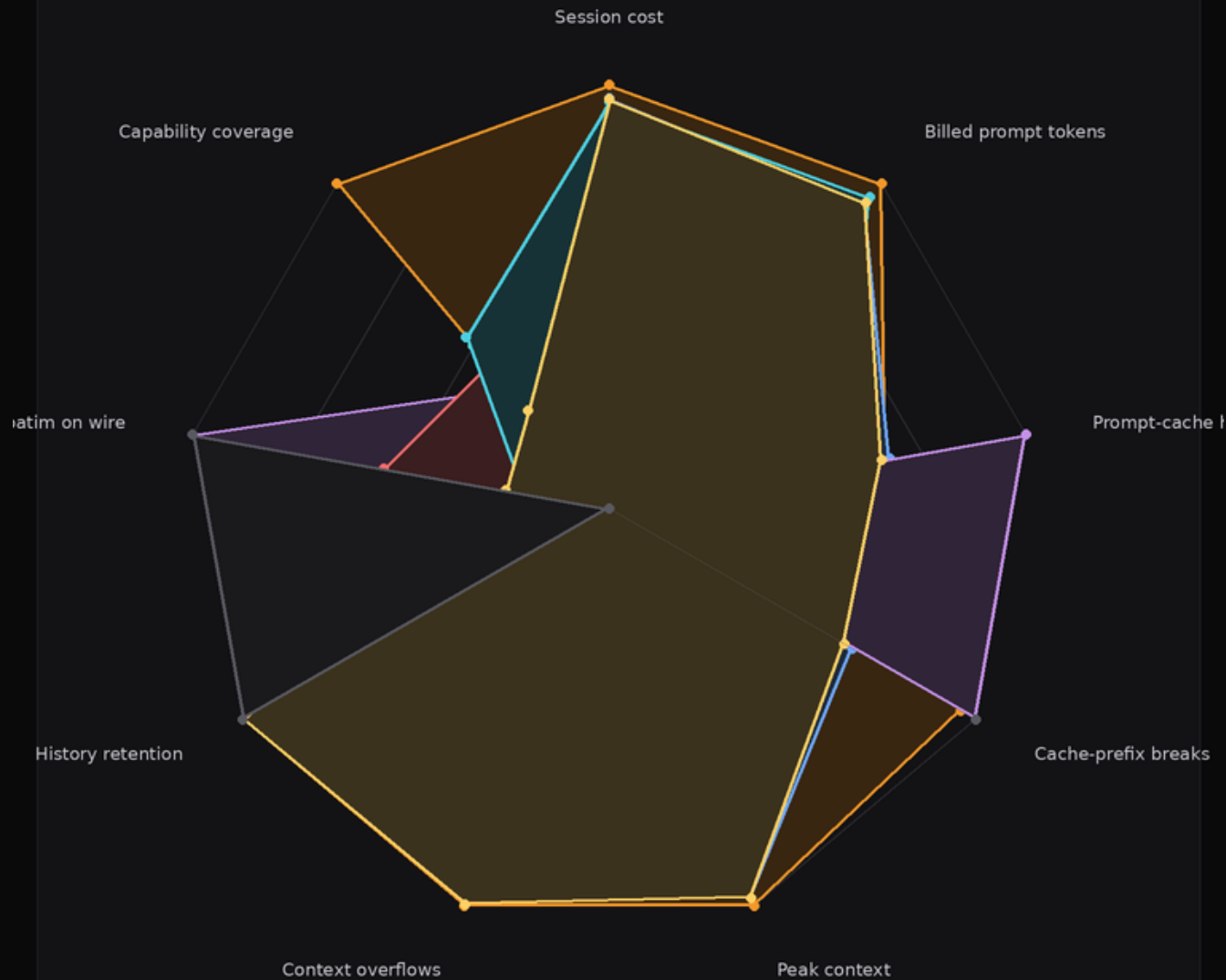
Fresh versus cached prompt tokens.

07 Profile Radar

Normalised profile (outer edge is best)

Min-max normalised across the field. A gap in the outline means no data, not a zero.

Dulus Claude Code Cursor Aider
GitHub Copilot (agent mode) Cline Codex CLI Naive full-context loop



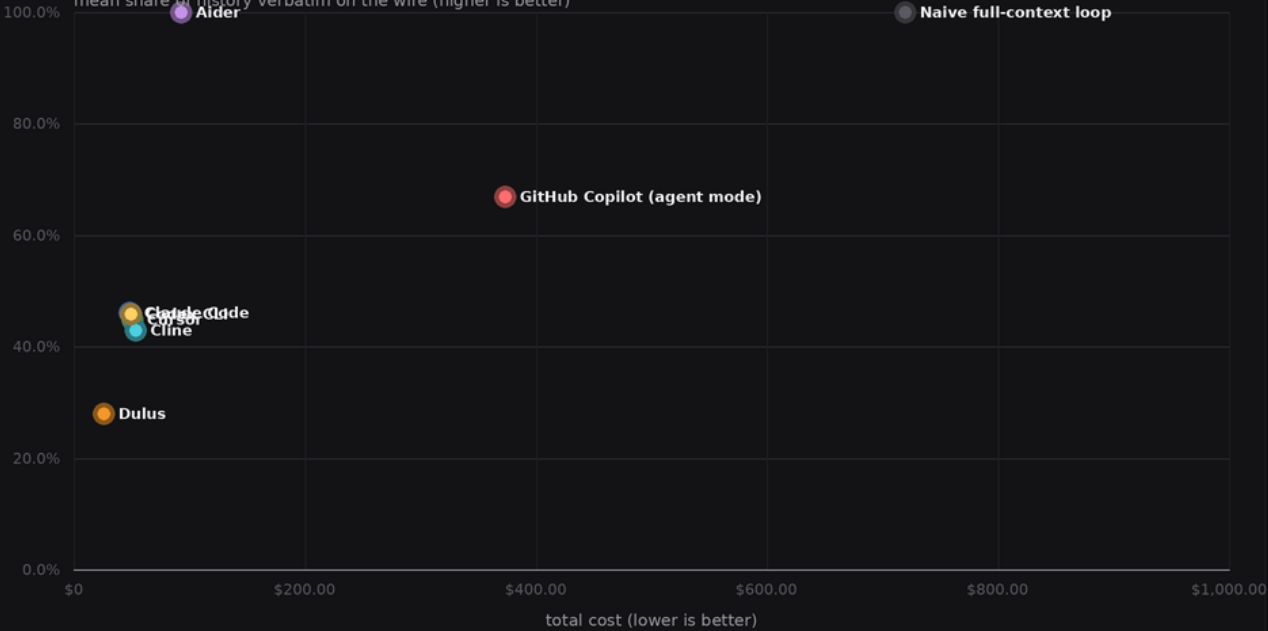
Instrument: token counts from compaction.estimate_tokens, USD from providers.calc_cost including the cache-read discount - identical for every harness. MODELLED (not vendor code): Claude Code, Cursor, Aider, GitHub Copilot (agent mode), Cline, Codex CLI. git 097aa216dc - 2026-10-18T06:07:05+00:00

Every scored axis at once, under equal weights.

08 Cost Fidelity Frontier

Cost against what the model can still read

Up and to the left is strictly better. Cheap-and-forgetful sits bottom-left and is not the same thing as efficient.
mean share of history verbatim on the wire (higher is better)



Instrument: token counts from compaction.estimate_tokens, USD from providers.calc_cost including the cache-read discount - identical for every harness. MODELLED (not vendor code): Claude Code, Cursor, Aider, GitHub Copilot (agent mode), Cline, Codex CLI. git 097aa216dc - 2026-10-18T06:07:05+00:00

The honest trade-off: saving money by showing the model less is still saving money by showing the model less.

09 Win Loss Grid

Who wins each axis - Marathon audit

Per-axis winner. '-' means one side had no data, so the axis is not contested.

	Claude Code	Cursor	Aider	GitHub Copilot (ag...	Cline	Codex CLI	Naive full-context I...
Session cost	DULUS	DULUS	DULUS	DULUS	DULUS	DULUS	DULUS
Billed prompt tokens	DULUS	DULUS	DULUS	DULUS	DULUS	DULUS	DULUS
Prompt-cache hit ratio	RIVAL	RIVAL	RIVAL	DULUS	RIVAL	RIVAL	DULUS
Cache-prefix breaks	DULUS	DULUS	RIVAL	DULUS	DULUS	DULUS	RIVAL
Peak context	DULUS	DULUS	DULUS	DULUS	DULUS	DULUS	DULUS
Context overflows	tie	tie	DULUS	DULUS	tie	tie	DULUS
History retention	tie	tie	tie	tie	tie	tie	tie
Verbatim on wire	RIVAL	RIVAL	RIVAL	RIVAL	RIVAL	RIVAL	RIVAL
Capability coverage	-	-	-	-	-	-	-

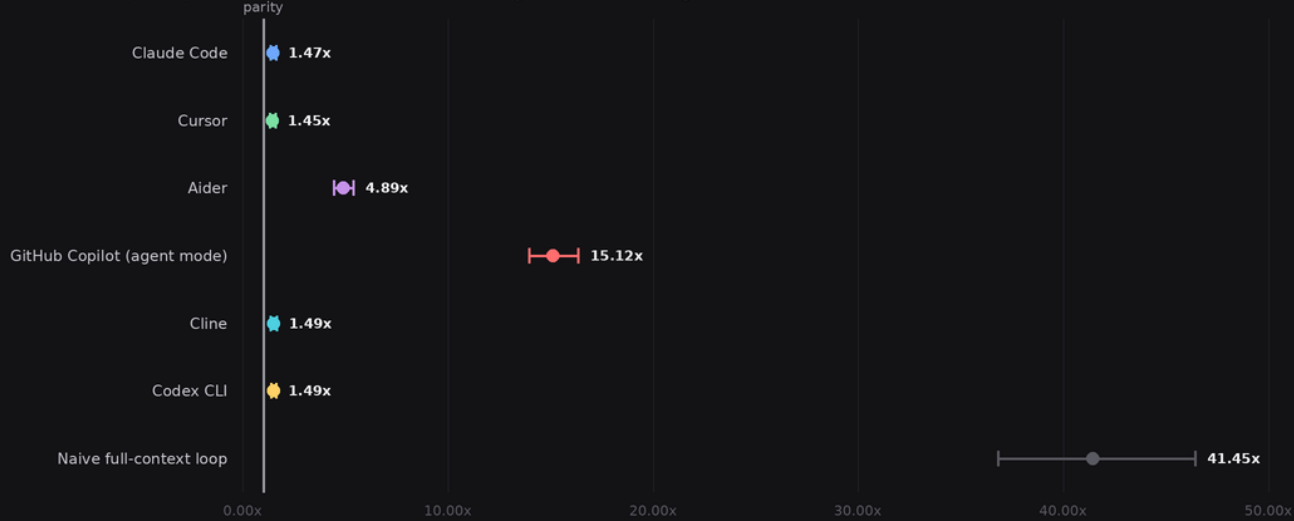
Instrument: token counts from compaction.estimate.tokens, USD from providers.calc_cost including the cache-read discount - identical for every harness. MODELLED (not vendor code): Claude Code, Cursor, Aider, GitHub Copilot (agent mode), Cline, Codex CLI. git 097aa216dc - 2026-10-18T06:07:05+00:00

Axis-by-axis outcome, including the axes Dulus loses.

10 Cost Ratio Intervals

Cost ratio versus Dulus - Marathon audit

Paired bootstrap over per-turn costs, 95% interval. Above parity means the challenger costs more.



Instrument: token counts from compaction.estimate_tokens, USD from providers.calc_cost including the cache-read discount - identical for every harness. MODELLED (not vendor code): Claude Code, Cursor, Aider, GitHub Copilot (agent mode), Cline, Codex CLI. git 097aa216dc - 2026-10-18T06:07:05+00:00 An interval touching parity is drawn grey and reported as no reliable difference.

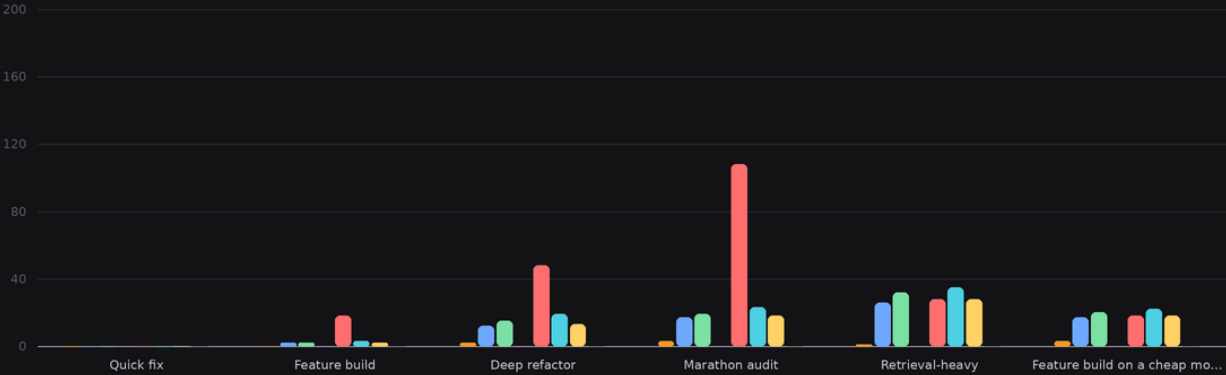
Effect sizes with uncertainty, not point estimates pretending to be facts.

11 Cache Behaviour

Cache-prefix breaks by scenario (lower is better)

Each break re-bills the whole prefix at full price. Front-truncation without a payoff gate breaks the prefix on every slide.

Dulus Claude Code Cursor Aider
GitHub Copilot (agent mode) Cline Codex CLI Naive full-context loop



Instrument: token counts from compaction.estimate_tokens, USD from providers.calc_cost including the cache-read discount - identical for every harness. MODELLED (not vendor code): Claude Code, Cursor, Aider, GitHub Copilot (agent mode), Cline, Codex CLI. git 097aa216dc - 2026-10-18T06:07:05+00:00

The mechanism behind the cost gap, isolated.

Capability Matrix

Capability matrix

Tick = supported, cross = not supported, ? = no data found.

Measured Documented Vendor Claim Self-Reported Modelled No Data

	Dulus		Claude Code		Cursor		Aider		GitHub Copilot (...)		Cline		Codex CLI		Naïve full-context...	
Open source	✓	●	✗	●	✗	●	✓	●	✗	●	✓	●	✓	●	?	●
Bring your own model	✓	●	✗	●	✓	●	✓	●	✓	●	✓	●	✗	●	?	●
Usable with no API key	✓	●	✗	●	✗	●	✗	●	✗	●	✗	●	✗	●	?	●
Browser session harvesting	✓	●	?	●	?	●	?	●	?	●	?	●	?	●	?	●
Context compaction	✓	●	✓	●	✓	●	✗	●	?	●	✓	●	✓	●	✗	●
Sliding history window	✓	●	✗	●	?	●	✗	●	✓	●	✓	●	✗	●	✗	●
Cache-aware truncation gate	✓	●	✗	●	?	●	✗	●	?	●	?	●	✗	●	✗	●
Agent can re-read dropped history	✓	●	✗	●	✗	●	✗	●	✗	●	✗	●	✗	●	✗	●
Sub-agents	✓	●	✓	●	✗	●	✗	●	✗	●	✗	●	✗	●	?	●
Semantic memory	✓	●	✗	●	✗	●	✗	●	✗	●	✗	●	✗	●	?	●
Checkpoint / rollback	✓	●	✓	●	✓	●	✓	●	✓	●	✓	●	✗	●	?	●
MCP	✓	●	✓	●	✓	●	?	●	✓	●	✓	●	✓	●	?	●
Offline voice	✓	●	✗	●	✗	●	✓	●	✗	●	✗	●	✗	●	?	●
Auto-adapter plugins	✓	●	✗	●	✗	●	✗	●	✗	●	✗	●	✗	●	?	●
Sandboxed execution	✓	●	✓	●	✓	●	✗	●	✓	●	✓	●	✓	●	?	●
Self-hostable	✓	●	✗	●	✗	●	✓	●	✗	●	✓	●	✗	●	?	●
Desktop GUI	✓	●	✗	●	✓	●	✗	●	✓	●	✓	●	✗	●	?	●
Web UI	✓	●	✗	●	✓	●	✓	●	✓	●	✗	●	✓	●	?	●
Messenger bridge	✓	●	✗	●	✓	●	✗	●	✗	●	✗	●	✗	●	?	●
Usage metering	✓	●	✓	●	✓	●	✓	●	✓	●	✓	●	✓	●	?	●
Self-improvement loop	✓	●	✗	●	✗	●	✗	●	✗	●	✗	●	✗	●	?	●

Dot colour is provenance, not sentiment. A tick resting on a press release and a tick resting on a measurement look identical without it.

Feature comparison with a provenance dot on every claim.

H2H Claude Code

HEAD TO HEAD		
Dulus	VS	Claude Code
Scenario: Marathon audit		
Dulus is cheaper: 1.465 [1.336, 1.613] the cost, medium effect.		
Session cost	\$10.69	\$15.66
Billed prompt tokens	8.35M	13.97M
Prompt-cache hit ratio	71.8%	78.0%
Cache-prefix breaks	3	17
Peak context	134.1k	182.7k
Context overflows	0	0
History retention	100.0%	100.0%
Verbatim on wire	11.4%	19.8%
Capability coverage	n/a	n/a
Instrument: token counts from compaction.estimate_tokens, USD from providers.calc_cost including the cache-read discount - identical for every harness. MODELLED (not vendor code): Claude Code, Cursor, Alder, GitHub Copilot (agent mode), Cline, Codex CLI. git 097aa216dc - 2026-10-18T06:07:05+00:00		

Dulus versus Claude Code on Marathon audit.

HEAD TO HEAD		
Dulus	VS	Cursor
Scenario: Marathon audit		
Dulus is cheaper: 1.448 [1.329, 1.590] the cost, medium effect.		
Session cost	\$10.69	\$15.48
Billed prompt tokens	8.35M	13.15M
Prompt-cache hit ratio	71.8%	76.9%
Cache-prefix breaks	3	19
Peak context	134.1k	169.2k
Context overflows	0	0
History retention	100.0%	100.0%
Verbatim on wire	11.4%	18.9%
Capability coverage	n/a	n/a
Instrument: token counts from compaction.estimate_tokens, USD from providers.calc_cost including the cache-read discount - identical for every harness. MODELLED (not vendor code): Claude Code, Cursor, Alder, GitHub Copilot (agent mode), Cline, Codex CLI. git 097aa216dc - 2026-10-18T06:07:05+00:00		

Dulus versus Cursor on Marathon audit.

H2H Aider

HEAD TO HEAD		
Dulus	VS	Aider
Scenario: Marathon audit		
Dulus is cheaper: 4.893 [4.432, 5.384] the cost, large effect.		
Session cost	\$10.69	\$52.30
Billed prompt tokens	8.35M	147.07M
Prompt-cache hit ratio	71.8%	98.4%
Cache-prefix breaks	3	0
Peak context	134.1k	2.36M
Context overflows	0	110
History retention	100.0%	100.0%
Verbatim on wire	11.4%	100.0%
Capability coverage	n/a	n/a
Instrument: token counts from compaction.estimate_tokens, USD from providers.calc_cost including the cache-read discount - identical for every harness. MODELLED (not vendor code): Claude Code, Cursor, Aider, GitHub Copilot (agent mode), Cline, Codex CLI. git 097aa216dc - 2026-10-18T06:07:05+00:00		

Dulus versus Aider on Marathon audit.

H2H Github Copilot

HEAD TO HEAD		
Dulus	VS	GitHub Copilot (agent mode)
Scenario: Marathon audit		
Dulus is cheaper: 15.115 [13.966, 16.355] the cost, large effect.		
Session cost	\$10.69	\$161.57
Billed prompt tokens	8.35M	27.42M
Prompt-cache hit ratio	71.8%	4.9%
Cache-prefix breaks	3	108
Peak context	134.1k	289.3k
Context overflows	0	105
History retention	100.0%	100.0%
Verbatim on wire	11.4%	33.1%
Capability coverage	n/a	n/a
Instrument: token counts from compaction.estimate_tokens, USD from providers.calc_cost including the cache-read discount - identical for every harness. MODELLED (not vendor code): Claude Code, Cursor, Alder, GitHub Copilot (agent mode), Cline, Codex CLI. git 097aa216dc - 2026-10-18T06:07:05+00:00		

Dulus versus GitHub Copilot (agent mode) on Marathon audit.

H2H Cline

HEAD TO HEAD		
Dulus	VS	Cline
Scenario: Marathon audit		
Dulus is cheaper: 1.492 [1.348, 1.640] the cost, medium effect.		
Session cost	\$10.69	\$15.95
Billed prompt tokens	8.35M	11.87M
Prompt-cache hit ratio	71.8%	72.4%
Cache-prefix breaks	3	23
Peak context	134.1k	150.0k
Context overflows	0	0
History retention	100.0%	100.0%
Verbatim on wire	11.4%	17.7%
Capability coverage	n/a	n/a
Instrument: token counts from compaction.estimate_tokens, USD from providers.calc_cost including the cache-read discount - identical for every harness. MODELLED (not vendor code): Claude Code, Cursor, Alder, GitHub Copilot (agent mode), Cline, Codex CLI. git 097aa216dc - 2026-10-18T06:07:05+00:00		

Dulus versus Cline on Marathon audit.

H2H Codex Cli

HEAD TO HEAD		
Dulus	VS	Codex CLI
Scenario: Marathon audit		
Dulus is cheaper: 1.488 [1.363, 1.632] the cost, medium effect.		
Session cost	\$10.69	\$15.90
Billed prompt tokens	8.35M	13.73M
Prompt-cache hit ratio	71.8%	77.0%
Cache-prefix breaks	3	18
Peak context	134.1k	179.2k
Context overflows	0	0
History retention	100.0%	100.0%
Verbatim on wire	11.4%	19.4%
Capability coverage	n/a	n/a
Instrument: token counts from compaction.estimate_tokens, USD from providers.calc_cost including the cache-read discount - identical for every harness. MODELLED (not vendor code): Claude Code, Cursor, Aider, GitHub Copilot (agent mode), Cline, Codex CLI. git 097aa216dc - 2026-10-18T06:07:05+00:00		

Dulus versus Codex CLI on Marathon audit.

Head to head

Per-turn costs on the same transcript are paired observations, so the interval is a paired bootstrap rather than two independent samples. Cliff's delta is used for effect size because per-turn token costs are heavily skewed and nothing here is normally distributed.

Scenario	Challenger	Cost ratio (95% CI)	Effect	Verdict
Quick fix	Claude Code	1.113 [1.047, 1.208]	small	dulus cheaper
Quick fix	Cursor	1.113 [1.047, 1.208]	small	dulus cheaper
Quick fix	Aider	1.113 [1.047, 1.208]	small	dulus cheaper
Quick fix	GitHub Copilot (agent mode)	1.113 [1.047, 1.208]	small	dulus cheaper
Quick fix	Cline	1.113 [1.047, 1.208]	small	dulus cheaper
Quick fix	Codex CLI	1.113 [1.047, 1.208]	small	dulus cheaper
Quick fix	Naive full-context loop	3.063 [2.056, 4.533]	large	dulus cheaper
Feature build	Claude Code	1.298 [1.187, 1.425]	medium	dulus cheaper
Feature build	Cursor	1.307 [1.154, 1.529]	medium	dulus cheaper
Feature build	Aider	1.646 [1.520, 1.774]	large	dulus cheaper
Feature build	GitHub Copilot (agent mode)	8.936 [6.762, 11.161]	large	dulus cheaper
Feature build	Cline	1.336 [1.139, 1.573]	medium	dulus cheaper
Feature build	Codex CLI	1.296 [1.186, 1.422]	medium	dulus cheaper
Feature build	Naive full-context loop	9.215 [7.640, 10.828]	large	dulus cheaper
Deep refactor	Claude Code	1.664 [1.440, 1.938]	large	dulus cheaper
Deep refactor	Cursor	1.733 [1.499, 1.996]	large	dulus cheaper
Deep refactor	Aider	3.102 [2.757, 3.500]	large	dulus cheaper
Deep refactor	GitHub Copilot (agent mode)	15.019 [13.171, 17.020]	large	dulus cheaper
Deep refactor	Cline	1.937 [1.641, 2.243]	large	dulus cheaper
Deep refactor	Codex CLI	1.732 [1.482, 2.023]	large	dulus cheaper
Deep refactor	Naive full-context loop	22.595 [19.205, 26.522]	large	dulus cheaper
Marathon audit	Claude Code	1.465 [1.336, 1.613]	medium	dulus cheaper
Marathon audit	Cursor	1.448 [1.329, 1.590]	medium	dulus cheaper
Marathon audit	Aider	4.893 [4.432, 5.384]	large	dulus cheaper
Marathon audit	GitHub Copilot (agent mode)	15.115 [13.966, 16.355]	large	dulus cheaper
Marathon audit	Cline	1.492 [1.348, 1.640]	medium	dulus cheaper
Marathon audit	Codex CLI	1.488 [1.363, 1.632]	medium	dulus cheaper
Marathon audit	Naive full-context loop	41.446 [36.801, 46.413]	large	dulus cheaper
Retrieval-heavy	Claude Code	2.974 [2.533, 3.441]	large	dulus cheaper
Retrieval-heavy	Cursor	3.367 [2.922, 3.825]	large	dulus cheaper
Retrieval-heavy	Aider	2.637 [2.312, 2.997]	large	dulus cheaper
Retrieval-heavy	GitHub Copilot (agent mode)	15.522 [12.538, 18.566]	large	dulus cheaper
Retrieval-heavy	Cline	3.537 [3.112, 4.022]	large	dulus cheaper
Retrieval-heavy	Codex CLI	3.085 [2.624, 3.542]	large	dulus cheaper
Retrieval-heavy	Naive full-context loop	17.944 [14.696, 21.538]	large	dulus cheaper
Feature build on a cheap...	Claude Code	2.446 [2.024, 2.889]	large	dulus cheaper
Feature build on a cheap...	Cursor	2.531 [2.140, 2.960]	large	dulus cheaper

Scenario	Challenger	Cost ratio (95% CI)	Effect	Verdict
Feature build on a cheap...	Aider	3.183 [2.674, 3.777]	large	dulus cheaper
Feature build on a cheap...	GitHub Copilot (agent mode)	9.166 [6.811, 11.782]	large	dulus cheaper
Feature build on a cheap...	Cline	2.601 [2.217, 3.016]	large	dulus cheaper
Feature build on a cheap...	Codex CLI	2.491 [2.077, 2.913]	large	dulus cheaper
Feature build on a cheap...	Naïve full-context loop	10.177 [8.144, 12.514]	large	dulus cheaper

Capability matrix

Each cell carries its provenance tier in parentheses. Two ticks that look identical can rest on a measurement and on a press release respectively.

Capability	Dulus	Claude...	Cursor	Aider	GitHub...	Cline	Codex CLI	Naive ful...
Open source	yes	no	no	yes	no	yes	yes	?
Bring your own model	yes	no	yes	yes	yes	yes	no	?
Usable with no API key	yes	no	no	no	no	no	no	?
Browser session harvesting	yes	?	?	?	?	?	?	?
Context compaction	yes	yes	yes	no	?	yes	yes	no
Sliding history window	yes	no	?	no	yes	yes	no	no
Cache-aware truncation gate	yes	no	?	no	?	?	no	no
Agent can re-read dropped history	yes	no	no	no	no	no	no	no
Sub-agents	yes	yes	no	no	no	no	no	?
Semantic memory	yes	no	no	no	no	no	no	?
Checkpoint / rollback	yes	yes	yes	yes	yes	yes	no	?
MCP	yes	yes	yes	?	yes	yes	yes	?
Offline voice	yes	no	no	yes	no	no	no	?
Auto-adapter plugins	yes	no	no	no	no	no	no	?
Sandboxed execution	yes	yes	yes	no	yes	yes	yes	?
Self-hostable	yes	no	no	yes	no	yes	no	?
Desktop GUI	yes	no	yes	no	yes	yes	no	?
Web UI	yes	no	yes	yes	yes	no	yes	?
Messenger bridge	yes	no	yes	no	no	no	no	?
Usage metering	yes	yes	yes	yes	yes	yes	yes	?
Self-improvement loop	yes	no	no	no	no	no	no	?

Measured facts about this checkout

These are the only rows in the document that are reproducible end to end: they were computed by walking this repository, not quoted from anybody.

Fact	Value	Provenance
cheapest_model	gemini-2.0-flash	MEASURED
cheapest_turn_usd	0.001034	MEASURED
cold_start_ms	366.6	MEASURED
cold_start_spread_ms	43.5	MEASURED
core_loop_files	13	MEASURED
core_loop_lines_code	19796	MEASURED
core_loop_lines_total	25822	MEASURED
dearest_model	claude-opus-4-6	MEASURED
dearest_turn_usd	0.212985	MEASURED
keyless_provider_count	13	MEASURED
local_provider_count	3	MEASURED
model_count	219	MEASURED
priced_model_count	38	MEASURED
pricing_spread_ratio	205.96	MEASURED
provider_count	40	MEASURED
python_files	1034	MEASURED
python_lines_code	273941	MEASURED
python_lines_total	346295	MEASURED
test_files	256	MEASURED
test_functions	4105	MEASURED
unpriceable_cost_keys	1	MEASURED
web_lines_total	112922	MEASURED
zero_cost_model_count	5	MEASURED

Caveats

- ? Context policies for Aider, Claude Code, Cline, Codex CLI, Cursor, GitHub Copilot (agent mode) are behavioural models, not those products' code. The cost curves are therefore a comparison of strategies, not a measurement of shipped software.
- ? Every recorded fact for Claude Code, Codex CLI, Cursor, Dulus, GitHub Copilot (agent mode) is a vendor or self-reported claim, not an independent measurement.
- ? Evidence quality is below 0.5 for Aider, Claude Code, Cline, Codex CLI, Cursor, GitHub Copilot (agent mode); their rows deserve less weight than the arithmetic implies.

Claim conflicts recorded

Rows where this repository's own marketing disagrees with what the arena recorded. Left visible on purpose: correcting a competitor's row against our own copy is the point of the provenance system.

- ? Claude Code / sub_agents: docs/FAQ.md:78 in this repo claims 'Sub-agents: No'. That claim is stale; corrected here and flagged.
- ? Claude Code / setup_seconds: Competitor-authored-by-us figure. Treat as adversarial and re-measure before publishing.
- ? Cursor / byo_model: docs/INVESTOR_BRIEF.md:38 claims 'Cursor = OpenAI only', which is wrong. Corrected.
- ? Cursor / mcp: docs/INVESTOR_BRIEF.md:58 claims 'MCP support: No'. Corrected.

Numbers to verify before publishing

Harness	Fact	Value	Provenance
Dulus	price_usd_month	0	SELF-REPORTED
Dulus	setup_seconds	30	SELF-REPORTED
Claude Code	price_usd_month	20	SELF-REPORTED
Claude Code	provider_count	1	SELF-REPORTED
Claude Code	setup_seconds	1800	SELF-REPORTED
Cursor	price_usd_month	20	VENDOR CLAIM
Cursor	provider_count	5	SELF-REPORTED
Cursor	setup_seconds	600	SELF-REPORTED
Aider	provider_count	100	MODELLED
Aider	setup_seconds	120	MODELLED
GitHub Copilot (agent mode)	price_usd_month	10	VENDOR CLAIM
GitHub Copilot (agent mode)	provider_count	4	MODELLED
GitHub Copilot (agent mode)	setup_seconds	300	MODELLED
Cline	provider_count	25	MODELLED
Cline	setup_seconds	180	MODELLED
Codex CLI	price_usd_month	20	VENDOR CLAIM
Codex CLI	provider_count	1	MODELLED
Codex CLI	setup_seconds	120	MODELLED

Override with --facts my-verified-facts.json and re-run. The merge is per key, so one corrected price does not require restating a whole harness.

Reproduce

```
python -m benchmarks --vs claude-code,cursor,aider,github-copilot,cline,codex-cli --scenarios quick-fix,featur
```