# How Many Tokens Does a Subscription Buy?

Theoretical Maxima, a Like-for-Like Comparison of Agent Harnesses, and the Value of Caching and Compression in AGNT

Annie
*AGNT Labs*

Technical Report — November 18, 2026

**Abstract**

A flat-rate subscription to a large language model is sold by its price, not by a quantity. We ask what quantity it buys: how many tokens an agent harness can process per plan-month, what the theoretical maximum is, and how much of the answer is due to the harness. We model the plan as a budget of API-equivalent value, under which the monthly token yield is the budget divided by the blended price of the tokens a harness sends. A public observation of 21.4 billion tokens at 95% of a weekly limit puts a floor of $4,505 per week under the budget of a $200 Claude Max plan. At that floor the yield ceiling on Claude Opus 5.5 is 97.5 billion tokens per month, for any harness. Against that ceiling we measure AGNT on 38,805 recorded requests: 42.3 billion tokens per plan-month in real mixed use, 55.3 billion with a warm cache and 63.7 billion in long uninterrupted runs. Single clean sessions of Claude Code and of an Agent SDK front end give 50.5 and 51.7 billion, and real-use estimates for twelve other harnesses range from 16.8 to 40.3 billion. A replay ablation shows that the recorded work occupies 33% of one plan as configured, 288% without caching, and 342% with neither caching nor compression, which is four plans in place of one. Finally, we show that token yield and useful work move in opposite directions when context is reduced, and we argue that a yield should always be reported together with output per budget dollar.

## 1 Introduction

Flat-rate subscriptions are now a common way to pay for agentic use of large language models. A plan states a price. What it buys is disclosed only as percentages of a five-hour window and of a weekly limit, never as a quantity. Practitioners nevertheless compare agent harnesses by how many tokens a plan yields, and reports of tens of billions of tokens per week circulate without a common basis [6, 7]. This report puts the question on a quantitative footing for one plan, the $200 Claude Max tier, and one model, Claude Opus 5.5.

Three questions are usually conflated. We separate them.

**Q1, the maximum.** What is the largest number of tokens that any harness could process on the plan?

**Q2, position.** Where do real harnesses sit relative to that maximum when they are compared under matching conditions?

**Q3, attribution.** How much of one harness's position is produced by its own caching and compression?

**Contributions.** (i) A budget model of subscription yield, with a ceiling that holds for every harness and a method for bounding the budget from public observations (Section 3). (ii) A like-for-like protocol that compares clean sessions with clean sessions and real use with real use, on a single price sheet, and its application to AGNT, Claude Code, an Agent SDK front end and twelve further harnesses (Sections 4 and 5). (iii) An ablation of AGNT's caching and compression by a validated replay of one month of recorded work (Section 5.6). (iv) A proof, confirmed empirically, that token yield falls as a harness does more work per budget dollar by carrying less context (Proposition 3, Section 5.7).

**Findings.** The ceiling is a property of the account, not of the harness: 97.5 billion tokens per month at the budget floor, reached only by traffic that consists entirely of cache reads. AGNT's recorded month yields 42.3 billion; the same month with no cache entry ever lost would yield 54.2 billion. Under matching clean conditions AGNT is ahead of Claude Code and of the Agent SDK front end by 7% to 27%. In real use its measured yield is 1.1 to 2.2 times the four estimates available for Claude Code, and level with the best of the router-derived estimates once the uncertainty of that conversion is counted. Caching accounts for an 88.5% reduction in plan usage and compression for 17.5%.

## 2 Background

**Token classes and prices.** A request to the model is billed in five classes: uncached input, input read from the prompt cache, input written to the cache with a five-minute or a one-hour lifetime, and output [1, 2]. Table 1 gives the list prices for Claude Opus 5.5. A cache read costs 5% of uncached input. A one-hour cache write costs twice the uncached rate and forty times a read, so a harness that loses a cache entry and writes its context again pays for forty re-reads of the same tokens.

**Table 1:** List prices for Claude Opus 5.5 in USD per million tokens [1]. This single sheet prices every harness in the report.

| Token class | Symbol | Price |
|---|---|---|
| Uncached input | $p_u$ | 4.00 |
| Cache read | $p_r$ | 0.20 |
| Cache write, five-minute lifetime | $p_5$ | 5.00 |
| Cache write, one-hour lifetime | $p_1$ | 8.00 |
| Output | $p_o$ | 20.00 |

**Subscription limits.** The plan exposes its allowance as a percentage of a five-hour window and of a weekly limit. Neither is published as a number of tokens or of dollars. The provider states that its own coding tool, its Agent SDK, non-interactive invocations and third-party applications all draw from the same subscription limits [3].

**Harnesses.** A harness is the software that assembles each request: the system prompt, the tool definitions and the conversation history. It decides what to keep, what to cut and what to compress, and so determines the mix of token classes that the provider meters. We study AGNT, which we measure directly; Claude Code in interactive and in autonomous use; a front end built on the

Claude Agent SDK, the configuration used by T3 Code [9]; and twelve harnesses covered by a model router's published statistics [5].

# 3 A Budget Model of Subscription Yield

## 3.1 Definitions

For request $t$ let $U_t$, $R_t$, $W_{5,t}$, $W_{1,t}$ and $O_t$ be the tokens billed as uncached input, cache read, five-minute write, one-hour write and output, and let $I_t = U_t + R_t + W_{5,t} + W_{1,t}$ be the prompt.

**Definition 1** (Processed tokens, value, blended price)**.** For a set of requests $S$,

$$T(S) = \sum_{t \in S} (I_t + O_t), \tag{1}$$

$$V(S) = 10^{-6} \sum_{t \in S} \big(p_u U_t + p_r R_t + p_5 W_{5,t} + p_1 W_{1,t} + p_o O_t\big), \tag{2}$$

$$\bar{p}(S) = 10^6 \, V(S)/T(S). \tag{3}$$

$T$ counts every token on every request, so context that is re-read is counted each time. $V$ is the API-equivalent value in dollars and $\bar{p}$ the blended price in dollars per million tokens.

**Assumption 1** (Value metering)**.** The plan debits a budget in proportion to API-equivalent value at list prices. The weekly budget is $W$ dollars and is the same for every harness on the account.

**Definition 2** (Token yield)**.** With $\kappa = 4.33$ weeks per month and monthly budget $B = \kappa W$, the token yield of a harness with blended price $\bar{p}$ is

$$N \;=\; \frac{10^6 \, B}{\bar{p}} \quad \text{tokens per plan-month.} \tag{4}$$

## 3.2 The ceiling and the budget floor

**Proposition 1** (Yield ceiling)**.** *Under Assumption 1, every harness satisfies $N \le 10^6 B/p_r$, with equality if and only if every processed token is a cache read.*

*Proof.* $\bar{p}$ is a mean of the five class prices weighted by the class shares of $T$. Since $p_r$ is the smallest price, $\bar{p} \ge p_r$, with equality exactly when the whole weight is on cache reads. Substitute in (4). □

**Proposition 2** (Budget floor)**.** *If a harness is observed to process $T$ tokens while consuming a fraction $u$ of the weekly budget, then $W \ge 10^{-6} \, p_r \, T/u$.*

*Proof.* The value consumed is $uW = V \ge 10^{-6} p_r T$, by the bound $\bar{p} \ge p_r$. □

A public report gives $T = 21.4 \times 10^9$ tokens on Claude Opus 5.5 at $u = 0.95$ of the weekly limit of this plan [6]. With $p_r = 0.20$, Proposition 2 then gives

$$W \;\ge\; \frac{0.20 \times 21.4 \times 10^9}{10^6 \times 0.95} \;=\; \$4{,}505, \qquad B = \kappa W \;\ge\; \$19{,}507. \tag{5}$$

The bound is tight only if the observed traffic was made of cache reads alone. The report gives no breakdown by class, so the true budget may be higher. We evaluate every absolute yield in this report at the floor, $W = \$4{,}505$.

**Remark 1** (What depends on $W$)**.** Every yield is linear in $W$, so the ratio of two harnesses' yields, $N_a/N_b = \bar{p}_b/\bar{p}_a$, does not depend on it. Percentage reductions in value do not depend on it either. Counts of whole plans do.

### 3.3 A bound for a given workload

The ceiling of Proposition 1 cannot be reached by useful work, which produces output and adds new content that must be written to the cache once. A tighter bound holds a recorded workload fixed and removes only the losses.

**Definition 3** (Zero-loss yield)**.** Let $S$ be a recorded workload and $L \subset S$ the requests on which a cache entry created earlier in the same conversation was not read. Let $\rho$ be the value per token over the requests whose cache was intact. The zero-loss value is

$$V_0(S) \;=\; V(S) - \sum_{t \in L} \bigl(v_t - \rho\,(I_t + O_t)\bigr), \tag{6}$$

where $v_t$ is the value of request $t$, and the zero-loss yield is $N_0 = B\,T(S)/V_0(S)$. It is the yield of the same work had no cache entry been lost after the first request of each conversation.

### 3.4 Context as a sawtooth, and the trade-off it implies

Consider a long conversation whose prompt grows by $g$ new tokens per request, produces $o$ output tokens per request, and is cut back to a floor of $F$ tokens whenever it reaches a ceiling of $C$ tokens. Between cuts the prompt is read from the cache and only the $g$ new tokens are written. At a cut the kept history is written again. With write price $p_w$, the mean value per request, in millionths of a dollar, is

$$\bar{c}(F, C) \;=\; p_r\,\frac{F + C}{2} \;+\; p_w\,g \;+\; p_w\,\frac{F\,g}{C - F} \;+\; p_o\,o. \tag{7}$$

The first term is the cost of carrying context, the second the cost of new content, and the third the cost of rewriting $F$ tokens once every $(C - F)/g$ requests.

**Proposition 3** (Optimal ceiling)**.** *For a fixed floor $F$, $\bar{c}$ is minimised at*

$$C^* = F + \sqrt{2\,p_w\,F\,g/p_r}\,, \qquad \bar{c}(F, C^*) = p_r F + \sqrt{2\,p_r\,p_w\,F\,g} + p_w\,g + p_o\,o, \tag{8}$$

*and the minimum is strictly increasing in $F$.*

*Proof.* $\partial\bar{c}/\partial C = p_r/2 - p_w F g/(C - F)^2$, which vanishes at $C - F = \sqrt{2 p_w F g/p_r}$; the second derivative is positive. At that point the rewrite term equals $p_r(C^* - F)/2$, which gives the stated minimum. Each term that contains $F$ increases with $F$. □

**Corollary 1** (Yield against work)**.** *The number of requests a budget buys is $10^6 B/\bar{c}$, and the token yield is that number times the mean tokens per request, $\frac{1}{2}(F + C) + o$. Lowering the floor lowers $\bar{c}$, so the budget buys more requests, but it lowers the tokens per request faster, so the blended price rises and the token yield falls.*

A token yield therefore rewards carrying context, not doing work. Section 5.7 confirms the corollary on recorded data. With the values measured for AGNT, $g = 3{,}385$ tokens per request and $F = 455$K, equation (8) gives $C^* = 806$K, close to the 750K at which AGNT's cut was observed to trigger.

# 4 Data and Method

## 4.1 AGNT telemetry

AGNT records, for every request it sends, the token count in each class, the size of the prompt, the time since the previous request in the conversation, and a classification of the cache outcome: intact, first request of a conversation, history rewritten, entry expired, model changed, prefix changed, or unexplained miss. We use all 38,805 requests sent on one account's $200 Claude Max subscription between October 18 and November 17, 2026. They fall in 2,217 runs, a run being the requests issued for one user turn, and in 287 conversations over 30 active days. They process 15.87 billion tokens, 95.4% of them on Claude Opus 5.5.

Every request is priced on the sheet of Table 1 whatever model served it, so that differences between harnesses are not confounded with differences between model prices. On that sheet the month is worth $7,309.

The account did not approach its limit in the window. No rate-limit or usage-limit error appears among the 7,644 entries of the usage ledger for the subscription, and the busiest seven consecutive days, November 10 to November 16, amount to $2,398, or 53% of the weekly floor.

## 4.2 Comparator data

**Clean sessions.** Two sessions on Claude Opus 5.5 published with their cost itemised by class, one in interactive Claude Code and one through the SDK entry point, by the same user [4].

**Router means.** The mean prompt-cache hit rate of twelve coding agents in October 2026, as published by a model router for the API-key traffic it carries [5].

**Historical logs.** Token-class counts from the same account's earlier use of two other harnesses: 9,447 requests of Claude Code sub-agents from January to March 2026, of which 2,186 on Opus models, and 14,854 requests of Cline from October 2025 to January 2026.

**Autonomous run.** The published transcript of a seven-week autonomous run of Claude Code on Claude Opus 5.5, on one Team Premium seat [10]: 4,055 requests and 1.91 billion tokens, with token counts by class for every request.

**Heavy use.** Two public reports of Claude Code used for sustained re-reading of a large fixed context: the 21.4 billion token week already cited [6], and a month of 80.8 billion tokens on an earlier model, published with its token counts by class [7].

## 4.3 Conversion to the common sheet

Sources that report token counts by class are priced directly. A clean session reports dollars by class, so its tokens are recovered as $R = c_r/p_r$, $W = c_w/p_w$ and $O = c_o/p_o$; the lifetime of its cache writes is not reported, and we give the range between $p_w = p_5$ and $p_w = p_1$. A router mean reports only a hit rate $h$, which we convert as

$$\bar{p}(h) \;=\; \big(h\,p_r + (1-h)\,p_5\big)(1-\omega) + \omega\,p_o, \qquad \omega = 0.47\%, \tag{9}$$

where $\omega$ is the share of output in processed tokens, close to the 0.44% observed for AGNT and the 0.46% of the Claude Code session. Equation (9) prices input that was not read as a five-minute cache write. Section 5.4 reports the effect of pricing it as uncached input instead.

### 4.4 Replay and ablation

To estimate what AGNT's caching and compression save, we replay the recorded month under altered policies. The replay keeps, for every request, the recorded new content, output and idle time, and it keeps the cache losses that no policy controls: first requests, model changes, prefix changes and unexplained misses. It recomputes the prompt size, the expiry of cache entries after one idle hour, and the cuts made when the prompt passes a trigger. On the 223 conversations that never reach a cut (12,441 requests) the replay returns $1,523 against a recorded $1,487, an error of 2.4%.

During the window, cuts made near the context limit did not hold: the following request often returned to the full history, and the next cut rewrote the kept history again. A correction was implemented and verified offline on November 17, 2026, after the window closed. We therefore report the recorded month and, separately, the corrected configuration, replayed with the cut that was observed in 93 runs: a median trigger of 750K tokens and a median size after the cut of 455K.

The ablation switches off one mechanism at a time.

**No capping or offloading.** The logs for the same period record 1,420 tool results that were capped and 306 that were moved out of the prompt. Their tokens, 25.4 and 18.3 million with each result counted up to 150,000 tokens, are put back at the request that followed.

**No aging.** The 14,459 stale tool results that were removed in 217 batches, 15.8 million tokens, are put back.

**No compression.** Both of the above.

**No caching.** Every input token is priced at $p_u$.

**Neither.** No compression and no caching.

**Reproducibility.** The numbers in this report are generated, not transcribed. A script recomputes them from the per-request telemetry, the saved outputs of the replay, the usage ledger and the published figures cited above, and the tables and figures are filled from its output.

## 5 Results

### 5.1 The theoretical maximum

At the budget floor, Proposition 1 puts the ceiling at 22.5 billion tokens per week, or 97.5 billion per month. It belongs to the account. No harness can exceed it, and a harness approaches it only as its traffic approaches pure re-reading.

The two heavy-use reports show what approaching it looks like. The stress test ran at a pace of 92.7 billion tokens per month. The published month of 80.8 billion tokens read each cached token 297 times on average, against 44 for AGNT; on the common sheet its mix yields 84.0 to 87.8 billion tokens and 1,321 to 1,381 output tokens per budget dollar.

### 5.2 AGNT's operating points

Table 2 places AGNT between the ceiling and its recorded month. With the cache intact, AGNT's traffic is priced at $0.353 per million tokens, a yield of 55.3 billion; these requests carry 97% of the month's tokens. Runs of thirty or more requests with no loss at all reach 63.7 billion. The recorded month, at $0.461, yields 42.3 billion.

**Table 2:** Bounds and AGNT operating points on a $200 plan, Claude Opus 5.5 prices, budget floor of $4,505 per week. Context is the mean prompt size. A dash marks a quantity that does not apply.

| Condition | $ per MTok | Cache-read share (%) | Context (K) | Yield (B/month) |
|---|---|---|---|---|
| Ceiling: every token a cache read (Prop. 1) | 0.200 | 100.00 | – | 97.5 |
| AGNT, long uninterrupted runs (143 runs) | 0.306 | 99.38 | 453 | 63.7 |
| AGNT, requests with the cache intact | 0.353 | 99.12 | 409 | 55.3 |
| AGNT, zero-loss bound for the month (Def. 3) | 0.360 | – | 407 | 54.2 |
| AGNT, recorded month (38,805 requests) | 0.461 | 97.76 | 407 | 42.3 |
| AGNT, small fresh sessions (13 runs) | 0.649 | 97.03 | 118 | 30.0 |

The gap between the recorded month and the zero-loss bound is the cost of lost cache entries. Table 3 attributes it. In total 23.5% of the month's value lies above what the same tokens cost at the warm rate. Half of it comes from rewrites of history, most of them the cuts that did not hold (Section 4.4); the next largest part is the expiry of the one-hour cache between a user's turns.

**Table 3:** Value above the warm rate in AGNT's recorded month, by cause of the cache loss. Shares are of the month's value of $7,309 and may not sum to the total because of rounding.

| Cause | Requests | Value above warm rate ($) | Share of month (%) |
|---|---|---|---|
| History rewritten | 640 | 872 | 11.9 |
| Cache entry expired after an idle hour | 204 | 498 | 6.8 |
| Unexplained miss | 65 | 142 | 1.9 |
| First request of a conversation | 312 | 117 | 1.6 |
| Model changed within a conversation | 31 | 77 | 1.0 |
| System prompt or tools changed | 5 | 9 | 0.1 |
| Total | 1,257 | 1,715 | 23.5 |

### 5.3 Like for like: clean sessions

A clean session is a favourable condition for any harness: it starts once, runs without interruption and ends before any cut or expiry. Setting such a session beside a full month of another harness's use compares a best case with an average one. Table 4 instead sets it beside AGNT's own uninterrupted traffic. Under these matching conditions AGNT's yield is 7% to 27% above the two sessions, taking the extremes of the table.

**Table 4:** Clean conditions. Ranges for the two public sessions span the unknown lifetime of their cache writes; their output per budget dollar does not depend on it.

| Harness and condition | $ per MTok | Cache-read share (%) | Yield (B/month) | Output tokens per budget $ |
|---|---|---|---|---|
| AGNT, long uninterrupted runs | 0.306 | 99.4 | 63.7 | 9,638 |
| AGNT, requests with the cache intact | 0.353 | 99.1 | 55.3 | 12,044 |
| Agent SDK front end, one session [4] | 0.376–0.379 | 98.2–98.9 | 51.5–51.9 | 12,381 |
| Claude Code, one session [4] | 0.385–0.388 | 98.0–98.8 | 50.3–50.7 | 11,846 |

### 5.4 Like for like: real use

Table 5 and Figure 1 set AGNT's recorded month beside the real-use evidence for other harnesses. Each row is a yield on the same plan budget, so the column "plans to match" is the number of plans a harness would need to process what one plan processes through AGNT, and the last column prices the difference at the plan's fee.

**Table 5:** Real use on a $200 plan. "Router mean" rows are converted from a published cache-hit rate by equation (9); "logs" rows and the "transcript" row, which is the autonomous run [10], are priced from recorded token classes; AGNT's row is measured from its telemetry. The last two columns are computed from the yields as displayed.

| Harness | Basis | Cache-read share (%) | $ per MTok | Yield (B/month) | Plans to match AGNT | Extra cost per year ($) |
|---|---|---|---|---|---|---|
| **AGNT** | **measured** | **97.76** | **0.461** | **42.3** | **1.00** | – |
| Cline | router mean | 96.00 | 0.484 | 40.3 | 1.05 | 119 |
| Roo Code | router mean | 95.57 | 0.505 | 38.6 | 1.10 | 230 |
| GitHub Copilot | router mean | 95.45 | 0.510 | 38.2 | 1.11 | 258 |
| Cursor | router mean | 95.42 | 0.512 | 38.1 | 1.11 | 265 |
| Claude Code, autonomous | transcript | 96.38 | 0.515 | 37.9 | 1.12 | 279 |
| Pi | router mean | 94.58 | 0.552 | 35.3 | 1.20 | 476 |
| Kilo Code | router mean | 93.82 | 0.588 | 33.2 | 1.27 | 658 |
| OpenCode | router mean | 93.75 | 0.592 | 33.0 | 1.28 | 676 |
| Hermes Agent | router mean | 92.78 | 0.638 | 30.6 | 1.38 | 918 |
| Zed | router mean | 91.98 | 0.676 | 28.8 | 1.47 | 1,125 |
| Claude Code, Opus models | logs | 93.77 | 0.717 | 27.2 | 1.56 | 1,332 |
| Zoo Code | router mean | 89.66 | 0.787 | 24.8 | 1.71 | 1,694 |
| Claude Code, all models | logs | 91.18 | 0.870 | 22.4 | 1.89 | 2,132 |
| Cline | logs | 83.21 | 0.953 | 20.5 | 2.06 | 2,552 |
| Claude Code | router mean | 85.22 | 0.999 | 19.5 | 2.17 | 2,806 |
| Qwen Code | router mean | 81.87 | 1.159 | 16.8 | 2.52 | 3,643 |

Three observations qualify the table.

*Claude Code.* The four estimates for Claude Code, from three independent sources, lie between 19.5 and 37.9 billion. AGNT's measured month is 1.1 to 2.2 times these. The highest is the autonomous run, the only one of the four priced from a complete record of subscription traffic on the same model; AGNT's month is 12% above it. The two records agree on caching: against the same tokens with all input at the uncached rate, the recorded cache reuse lowers the value of the run by 87.2% and that of AGNT's month by 88.7%.

*The conversion favours the router rows.* Applied to AGNT's own cache-read share of 97.76%, equation (9) returns 48.8 billion, 15% above the 42.3 billion that AGNT measures, because real traffic writes with a one-hour lifetime and rewrites whole contexts. AGNT's cache-read share is the highest of the thirteen harnesses. The autonomous run shows the same bias: equation (9) returns 41.9 billion for it, against the 37.9 billion priced from its token classes.

*The conversion is sensitive to how a miss is billed.* If input that was not read is priced as uncached input at $p_u$ in place of $p_5$, the router rows rise: Cline to 43.9 billion, Roo Code to 42.3 billion and Claude Code to 22.9 billion, while AGNT's own share converts to 51.6 billion. AGNT's measured lead over the best router rows is therefore within the uncertainty of the conversion. Its lead over Claude Code does not rest on the conversion: the highest estimate for Claude Code, 37.9 billion, is priced directly from recorded token classes.

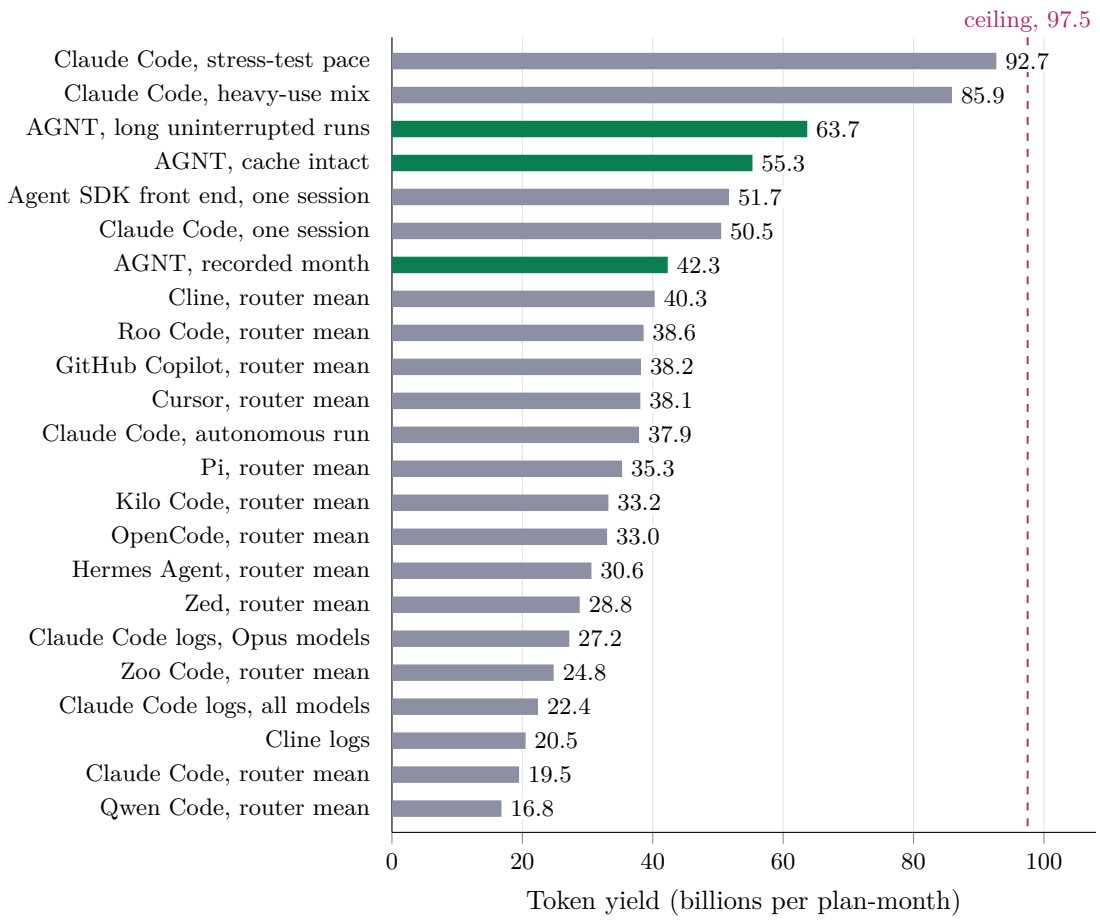



**Figure 1:** Token yield per plan-month at Claude Opus 5.5 prices. AGNT in green, other harnesses in grey. The dashed line is the ceiling of Proposition 1 at the budget floor.

### 5.5 Token yield is not work

Table 6 and Figure 2 place each condition by its token yield and by the output it produces per budget dollar. The two orderings disagree. Among these conditions the heavy-use mix has the highest yield and the lowest output. AGNT's small fresh sessions have the lowest yield of any AGNT condition and the highest output, 17,108 tokens per budget dollar, because they carry 118K of context in place of 407K. The autonomous run of Claude Code makes the same point from outside AGNT. Its mix yields 37.9 billion tokens and 3,283 output tokens per budget dollar, and 50% of its value is the cost, above a cache read, of writing prompts to the cache again: 93 of its 94 pauses longer than an hour were followed by a request that read less than half of its prompt from the cache.

### 5.6 What caching and compression save

Table 7 and Figure 3 give the ablation. As configured, the month's work is worth $6,445, which is 33% of one plan's budget at the floor. Without compression it is 40%. Without caching it is 288%, which needs three plans, and with neither it is 342%, which needs four. At this volume the mechanisms together are worth $600 a month in subscriptions, or $7,200 a year.

Caching reduces usage by 88.5%, compression by 17.5%, and the two together by 90.3%. By Remark 1 these percentages do not depend on the budget. The ratio between the last row and the

**Table 6:** Token yield against work. Output per plan-month is the output a full plan budget would produce at each mix.

| Condition | Yield (B/month) | Output per budget $ | Output per plan (M) | Requests per plan (K) | Context (K) |
|---|---|---|---|---|---|
| Claude Code, heavy-use mix [7] | 84.0–87.8 | 1,321–1,381 | 26–27 | – | – |
| AGNT, long uninterrupted runs | 63.7 | 9,638 | 188 | 140 | 453 |
| AGNT, cache intact | 55.3 | 12,044 | 235 | 135 | 409 |
| Agent SDK front end, one session | 51.5–51.9 | 12,381 | 242 | – | – |
| Claude Code, one session | 50.3–50.7 | 11,846 | 231 | – | – |
| AGNT, recorded month | 42.3 | 9,520 | 186 | 104 | 407 |
| Claude Code, autonomous run | 37.9 | 3,283 | 64 | 81 | 470 |
| AGNT, small fresh sessions | 30.0 | 17,108 | 334 | 251 | 118 |
| Claude Code logs, Opus models | 27.2 | 7,143 | 139 | – | – |
| Claude Code logs, all models | 22.4 | 5,901 | 115 | – | – |
| Cline logs | 20.5 | 6,126 | 119 | – | – |

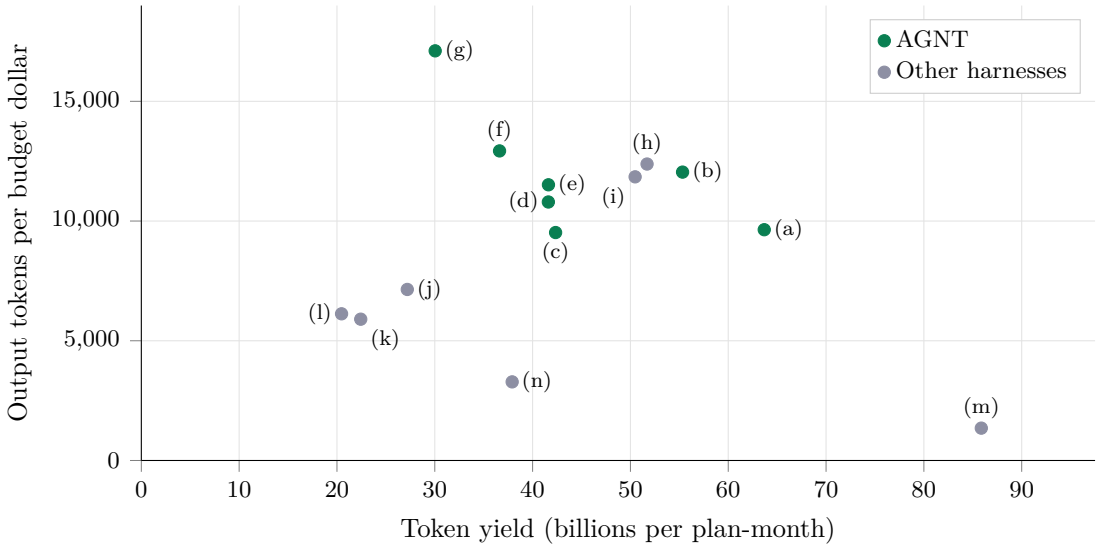



**Figure 2:** Token yield against output per budget dollar. AGNT in green: (a) long runs, (b) cache intact, (c) recorded month, (d) corrected configuration, (e) cut to 300K, (f) 500K working context, (g) small fresh sessions. Others in grey: (h) Agent SDK session, (i) Claude Code session, (j) Claude Code logs, Opus, (k) Claude Code logs, all models, (l) Cline logs, (m) Claude Code heavy-use mix, (n) Claude Code autonomous run. The stress test of Figure 1 reports no output and is absent.

configured row is 10.4: on a plan used to its limit, one plan through AGNT carries the work of about ten plans without either mechanism.

Two points bear on reading the result for compression. First, it does not change the bill at this volume, since the work fits one plan either way; its effect is headroom, with one plan holding 3.0 months of this work in place of 2.5, and fewer cuts to history, 87 in place of 192. Second, capping large results at 150,000 tokens understates it: some results were up to three million tokens and could not have been sent at all.

The result for caching can be checked without the replay. Over the closed window October 18 to November 16, 2026, the account's usage ledger records 15.67 billion tokens whose value, with each model at its own list price, is $7,814.18 with the recorded cache reuse and $67,199.56 with

**Table 7:** Ablation of AGNT's caching and compression on the recorded month, by replay. Plans are counted against a budget of $19,507 per plan-month. The first row is the month as recorded, before the correction described in Section 4.4. Context is the mean prompt in thousands of tokens, and history cuts counts the times a conversation's history was cut; neither depends on caching.

| Configuration | Value ($) | Share of one plan (%) | Whole plans | Fee per month ($) | Context (K) | History cuts |
|---|---|---|---|---|---|---|
| Recorded month, before the correction | 7,309 | 37 | 1 | 200 | 407 | 345 |
| AGNT as configured | 6,445 | 33 | 1 | 200 | 352 | 87 |
| No capping or offloading | 7,456 | 38 | 1 | 200 | 401 | 162 |
| No aging | 6,845 | 35 | 1 | 200 | 378 | 117 |
| No compression | 7,814 | 40 | 1 | 200 | 421 | 192 |
| No caching | 56,101 | 288 | 3 | 600 | 352 | 87 |
| Neither | 66,738 | 342 | 4 | 800 | 421 | 192 |

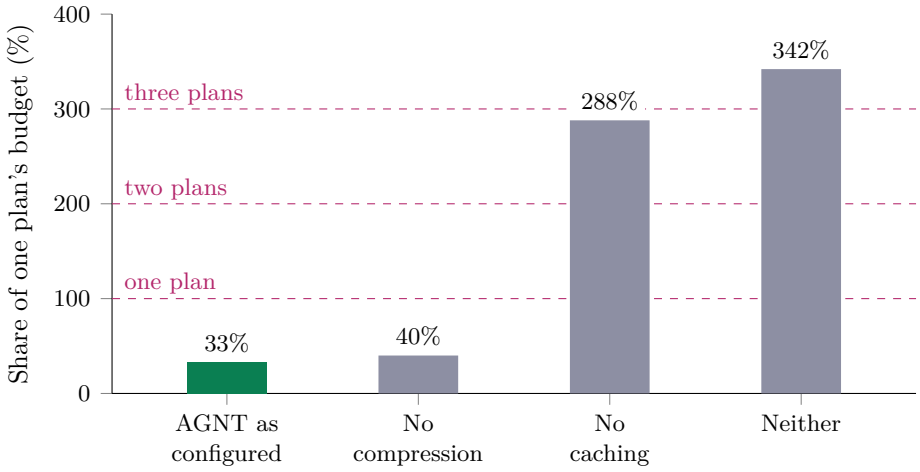



**Figure 3:** The recorded month's work as a share of one plan's budget: AGNT as configured, in green, and with its mechanisms switched off, in grey. Dashed lines mark whole plans.

every input token at the uncached rate: 88.4% lower, against the replay's 88.5%.[1]

## 5.7 Context policy: more work, fewer tokens

Table 8 replays the month under cut policies of increasing depth. The corrected configuration lowers the value of the same work by 11.8% (8.8% to 15.8% over the interquartile range of observed triggers), yet its token yield is 41.6 billion against the recorded 42.3 billion: the correction removes costly rewrites and the cheap re-reads of full history that accompanied them. Deeper cuts continue the pattern. A working context of 500K raises the requests a plan buys by 36% and lowers the token yield to 36.6 billion. This is Corollary 1 on recorded data.

[1]The first of these amounts equals the "No compression" row of Table 7 to the dollar. That is a coincidence: the two are computed over different windows and on different price bases.

**Table 8:** Replay of the recorded month under cut policies. Rows two to four hold the cut at the median and at the quartiles of the observed trigger; the rows below cut deeper. The replay treats a cut as free of cost to quality; a cut discards the oldest history.

| Trigger to size after cut | Value ($) | Against recorded (%) | Yield (B/month) | Context (K) | Requests (K) per plan-month | Output per budget $ |
|---|---|---|---|---|---|---|
| Recorded | 7,309 | – | 42.3 | 407 | 104 | 9,520 |
| 750K to 455K (corrected) | 6,445 | −11.8 | 41.6 | 352 | 117 | 10,796 |
| 884K to 521K | 6,667 | −8.8 | 43.5 | 381 | 114 | 10,436 |
| 620K to 399K | 6,157 | −15.8 | 39.2 | 317 | 123 | 11,300 |
| 750K to 300K | 6,043 | −17.3 | 41.6 | 330 | 125 | 11,514 |
| 500K to 250K | 5,381 | −26.4 | 36.6 | 259 | 141 | 12,929 |
| 350K to 150K | 4,730 | −35.3 | 31.6 | 195 | 160 | 14,708 |
| 250K to 100K | 4,244 | −41.9 | 26.6 | 147 | 178 | 16,393 |

# 6 Discussion

**How many tokens AGNT gives.** The theoretical maximum on this plan is the account's ceiling, 97.5 billion tokens per month at the budget floor and more if the true budget is higher. It is the same for AGNT as for any other harness. For the work AGNT actually did, the maximum is the zero-loss bound of 54.2 billion. AGNT measured 42.3 billion, or 508 billion per plan-year, and reaches 55.3 to 63.7 billion in sustained runs. On the fee, the recorded month's 15.87 billion tokens cost 1.26 cents per million; a plan used to its limit at this mix would cost 0.47 cents per million.

**What the difference is worth.** For a user on a flat plan below its limit, a better yield is headroom, not cash. It becomes cash in two cases: on metered API billing, and at the limit. At this account's volume, one percentage point of cache-read share is worth $758 a month at list prices. At the limit, Table 5 gives the plans a harness needs to match one plan through AGNT: 1.1 to 2.2 for Claude Code, or $279 to $2,806 a year per seat.

**Why the ceiling is not a target.** The traffic nearest the ceiling in our data produces about one seventh of the output per budget dollar of AGNT's recorded month (Table 6). A harness could raise its token yield at any time by re-reading more. The quantity a user cares about is work, and for a fixed budget work rises as the yield, so defined, falls.

**Reporting.** We recommend that a token yield never be reported alone. The triple of token yield, output per budget dollar and mean context identifies a harness's operating point and makes the trade-off visible.

# 7 Threats to Validity

**Value metering is assumed.** Assumption 1 is not documented by the provider. Public measurements report that requests through the SDK entry point consume the five-hour window two to three times faster per dollar than interactive use on Claude Opus 5.5 [4], with the weekly limit reported as unaffected [8]; so the assumption fails across entry points for the five-hour window at least. If the weekly limit were weighted in the same way, the SDK front end's yield would be 17.3 to 25.9 billion in place of 51.7 billion. How AGNT's traffic is weighted has not been measured.

**The budget is a floor from one report.** $W$ rests on one self-reported observation [6] whose statement of 95% utilisation we cannot verify. By Remark 1 a higher budget leaves every ratio and percentage unchanged, raises every yield in proportion, and lowers the plan counts of Table 7.

**AGNT did not reach the limit.** The busiest week used 53% of the floor, so AGNT's yields are extrapolations of its measured blended price to a full budget, not observations of a full plan.

**Comparators are heterogeneous.** The clean sessions are one session each. The router means describe API-key traffic of a different population of users, and their conversion carries the sensitivity shown in Section 5.4. The historical logs are sub-agent transcripts on earlier model and harness versions. The autonomous run is a single agent on a different plan, and its authors report that it worked about a quarter of an hour a day [10]; its mix reflects long pauses as much as its harness. No real-use data exists for the Agent SDK front end.

**One account, one month.** All AGNT figures come from one heavy user's mix of work. Other workloads will have different shares of new content, output and idle time.

**The ablation is a replay.** The replay is validated to 2.4% only on conversations that never reach a cut. The corrected configuration had not been observed in production when this report was written; the recorded month, before the correction, is 37% of one plan in place of 33%. Compression events are matched to requests by time, and 86% to 95% of them could be placed. The ablation compares AGNT with and without its own mechanisms. Other harnesses cache and compress too, and the comparison with them is Section 5.4, not Section 5.6.

**Prices.** All values use list prices read on November 17, 2026.

# 8 Conclusion

A subscription buys a budget, and the tokens it yields are that budget divided by the price of the tokens a harness chooses to send. On a $200 Claude Max plan the ceiling, evaluated at the budget floor, is 97.5 billion tokens per month for every harness. AGNT yields 42.3 billion in recorded real use and 55.3 to 63.7 billion in sustained runs. That is 7% to 27% ahead of the clean sessions available for Claude Code and the Agent SDK, and 1.1 to 2.2 times the real-use estimates for Claude Code. Its caching and compression reduce the plan usage of a recorded month by 90.3%, which at this account's volume is the difference between four plans and one. Because yield rises with re-reading and falls as a harness becomes leaner, it should be read beside output per budget dollar.

# References


[1] Anthropic. Pricing. `https://platform.claude.com/docs/en/about-claude/pricing`. Accessed November 17, 2026.

[2] Anthropic. Prompt caching. `https://platform.claude.com/docs/en/build-with-claude/prompt-caching`. Accessed November 17, 2026.

[3] Anthropic. Use the Claude Agent SDK with your Claude plan. https://support.claude.com/en/articles/15036540-use-the-claude-agent-sdk-with-your-claude-plan. Updated October 7, 2026; accessed November 17, 2026.

[4] anthropics/claude-code. Issue #97074, on five-hour usage accounting by entry point. https://github.com/anthropics/claude-code/issues/97074. Opened September 25, 2026; accessed November 17, 2026.

[5] Requesty. Prompt-cache hit rate by coding agent, October 2026. https://www.requesty.ai/data/coding-agent-cache-hit-rate-oct-2026. Accessed November 17, 2026.

[6] r/ClaudeAI. "∼21 BILLION tokens per weekly limit on the Max 20x Claude Code usage." https://www.reddit.com/r/ClaudeAI/comments/1x4inan/21_billion_tokens_per_weekly_limit_on_the_max_20x/. Accessed November 17, 2026.

[7] r/ClaudeAI. "My Claude Code usage for March 2026: $41,952 across 80.5 billion tokens." https://www.reddit.com/r/ClaudeAI/comments/1s8mj4e/my_claude_code_usage_for_march_2026_41952_across/. Accessed November 17, 2026.

[8] Theo (t3.gg). Post of October 24, 2026. https://x.com/theo/status/2114166439813017873. Accessed November 17, 2026.

[9] pingdotgg/t3code. Issue #1579. https://github.com/pingdotgg/t3code/issues/1579. Accessed November 17, 2026.

[10] A. Wuhrmann, D. Schmotz, D. Prinzhorn, A. Prabhu, H. Coppock and M. Andriushchenko. Autonomous agents can run successful open-model organisations. Evidence from an 8-week case study. Horizon Labs, 2026. https://horizon-labs.ai. Accessed November 18, 2026.

# A Notation

| Symbol | Meaning |
|---|---|
| $U_t, R_t, W_{5,t}, W_{1,t}, O_t$ | Tokens of request $t$ by class: uncached input, cache read, cache writes, output |
| $I_t$ | Prompt of request $t$, the sum of its four input classes |
| $p_u, p_r, p_5, p_1, p_o$ | List prices of the classes, USD per million tokens (Table 1) |
| $T, V, \bar{p}$ | Processed tokens, API-equivalent value, blended price |
| $W, B, \kappa$ | Weekly budget, monthly budget $B = \kappa W$, weeks per month |
| $N, N_0$ | Token yield per plan-month; zero-loss yield |
| $u$ | Fraction of the weekly budget consumed |
| $F, C, g, o$ | Context floor and ceiling, new tokens and output tokens per request |
| $h, \omega$ | Cache-hit rate; share of output in processed tokens |