

The Asymmetry Engine, Chapter 3: The sufficient model

Martin Roark

Chapter 3 of *The Asymmetry Engine: How AI Closes the Gaps, and What Stays Scarce*, October 2026. The online edition, with the interactive figures and the links between chapters, is at <https://future-seems-so-good.com/blog/the-asymmetry-engine#compute>. License: [CC BY 4.0](#).

3:1 Most valuable tasks need a model that is good enough, and on them the best model's extra capability is wasted. A task's value is captured at the price of the cheapest model that clears its threshold, so the frontier's price buys nothing there. Cheaper tiers mostly move money from the companies that sell models to the companies that deploy them, the hands that build pipelines bind before price does, and whether falling prices raise total spending depends on an elasticity the data do not yet pin down.

3:2 In the master equation, the compute asymmetry sets how much intelligence supply a dollar buys, the hands are part of a gap's inertia, and the response of spending to price is regeneration: the new gap that closing this one opens.

3.1 The frontier and the sufficient

3:3 Most tasks have a threshold. Classifying a support ticket, reading the total off an invoice or routing a request has an answer that can be checked, and every model that reaches it gives the same result as one that reasons far beyond it, so what matters is the cheapest model that clears it.

3:4

Definition 3.1 (Cheapest sufficient model). Model tiers m have a price per call p_m and a capability Q_m . A task t has a quality threshold q_t^* , a value u_t per successful call and n_t calls a year. The cheapest sufficient price is $p^*(q) = \min\{p_m : Q_m \geq q\}$, the tier that sets it is the task's **cheapest sufficient model**, and p_{front} is the price of the frontier tier. The task is viable when $u_t \geq p^*(q_t^*)$, and capability above q_t^* is wasted on it.

3:5

$$A_c(t) = \frac{p_{\text{front}}}{p^*(q_t^*)} \geq 1, \quad \Pi_t = (u_t - p^*(q_t^*)) n_t \quad (3.1)$$

3:6 The ratio A_c is the task's **compute asymmetry**, and Π_t is its surplus before fixed costs, what the asymmetry is worth once volume multiplies it. At the median call and October 2026 list prices (3:14), A_c is 10 for Claude Haiku 4.5 and about 357 for Jev, so a decision Jev can make, sent to the frontier by default, is overpaid several hundredfold.

3:7 So a task's value is captured at the price of its cheapest sufficient model. As cheaper tiers rise past more thresholds, tasks migrate down the price list, and the frontier keeps only the tasks whose thresholds it alone clears.

3:8 Errors refine the choice. Read Q_m as the probability that tier m completes the task correctly, and let $c_{\text{fail},t}$ be the cost of a failed call: a wrong refund, a shipped bug, a missed vulnerability. The deployer minimizes the expected cost per call:

3:9

$$m_t^* = \arg \min_m [p_m + (1 - Q_m) c_{\text{fail},t}] \quad (3.2)$$

3:10 Where failures are cheap, as in triage, extraction and routing, the rule picks the cheapest sufficient model. Where they are dear, as in production code, security and medicine, it picks the frontier at many times the price. The one line explains why both markets grow at once, and why the most effective agent systems split a task: a frontier planner decomposes the work until each piece is cheap to fail, and cheaper models execute the pieces.

3:11 The verification asymmetry builds this one. Distillation turns a frontier teacher's outputs into training data for a much smaller student ([Hinton, Vinyals and Dean, 2015](#)), and in September 2026 distil labs tuned students of 0.8 to 4 billion parameters on 3,000 to 4,000 examples written by a large teacher ([distil labs](#)). Wherever an answer is cheap to check, a cheap model can be trained to give it.

3.2 How big the gap is

3:12 The compute asymmetry is large. Artificial Analysis prices every task of its Intelligence Index, and the cost per task spans about 1,300x: Claude Opus 5.5 at maximum effort costs \$5.98 a task and scores 58, while GPT-6 Luna at low effort costs \$0.0045 and scores 22 ([Artificial Analysis](#)).

3:13 List prices set the asymmetry of a typical call, and within one lab OpenAI sells GPT-6.1 Sol as "near-Astra intelligence for coding, computer use, and professional work" at a fifth of GPT-6 Astra's token prices ([OpenAI](#)). As of October 2026, per million input and output tokens and per median call of 1,500 tokens in and 150 out:

3:14

Model	Price in / out	Median call	A_c
GPT-6 Astra, Claude Fable 5.1	\$10 / \$50	\$0.0225	1
GPT-6.1 Sol	\$2 / \$10	\$0.0045	5
Claude Haiku 4.5	\$1 / \$5	\$0.00225	10
GPT-6 Luna	\$0.10 / \$0.50	\$0.000225	100
Jev	\$0.042 / free	\$0.000063	357
gpt-oss-20b , open weights	\$0.018 / \$0.09	\$0.0000405	556

3:15 Whether a cheaper model gives the same result depends on the shape of the task. Where the answer can be read off the input (classify, route, score, extract), it does: in distil labs' accounts-payable test, inbox triage scored 200 of 200 for Jev, for a fine-tuned model of 0.8 billion parameters and for GPT-5.6 Luna at high reasoning effort ([distil labs](#)), and fine-tuned models a fraction of the size beat zero-shot GPT-4 and Claude Opus at text classification ([Bucher and Martini, 2024](#)). Where the task needs several steps or arithmetic, it does not: the same test's pay-or-hold step scored 0.79 for Jev against 1.00 for Luna. Saturation belongs to the task's output space, which is why the threshold is set per task.

3:16 Averages hide this. On the general index Claude Haiku 4.5 scores 15 to 17 against 42 to 58 for Claude Opus 5.5 ([Artificial Analysis](#)), so on average the two are far apart; on a task whose threshold sits below Haiku's reach they are economically identical, and the cheaper one wins by its price ratio. The market buys intelligence one threshold at a time. Haiku is also no longer the cheapest model that clears the easy thresholds: GPT-6 Luna costs a tenth as much.

3:17 The cleanest measurement inside one system is Cursor's build of SQLite in Rust in July 2026: every model mix passed the full test suite, and a frontier planner over cheaper workers cost about an eighth as much as a frontier model doing everything (\$9.6).

3:18 The arithmetic is public and practice lags. Routing between models matched GPT-4's quality "with up to 98% cost reduction" in 2023 ([FrugalGPT](#)), and Anthropic's pricing page advises "Choose Haiku for simple tasks" ([Anthropic](#)), yet Bain found in June 2026 that "everyone stays on frontier" ([Bain](#)). The default model sets the bill more than the price list does.

3.3 Jev

3:19 **Jev** is the compute asymmetry sold as a product. TypeSafe AI released it in early access on 15 September 2026 as a non-generative "System One" model: it takes a state and typed questions and returns typed answers (a choice, a score or a probability) in one parallel pass, and it "gives up string generation" to do so. It costs \$0.042 per million input tokens, output is free, and it answers in 70 to 500 ms. TypeSafe named it after William Stanley Jevons: "We expect machine intelligence to follow a similar path to coal" ([TypeSafe](#)).

3:20 Jev's gains are self-reported. On its own workflow evaluations TypeSafe reports it "193.6x faster, 444.6x cheaper" than the average of GPT-6 Astra and Claude Fable 5.1, calls these figures "on the higher end of real world gains" and adds "We can't prove it isn't subsidized" ([TypeSafe](#)); the cost ratio is close to the 357 of a median call. Its parameter count is undisclosed, and its documentation lists weak spots in counting, arithmetic and date comparison ([InfoQ](#)).

3:21 A worker can try a pipeline without telling the boss and run a million requests for about ten dollars: a million requests of 240 input tokens is 2.4×10^8 tokens, which costs \$10.08 at Jev's price, while the same million on GPT-6 Astra costs at least \$2,400, 238 times as much, before any output or reasoning tokens. The figure holds for short requests. At the median prompt of 1,500 tokens a million calls cost \$63, or \$630 through a third-party gateway that charges \$0.42 per million input tokens on its small prepaid packs ([gateway pricing](#)), and a trillion requests of 500 tokens would cost about \$21 million.

3:22 Workers already do this. In Microsoft and LinkedIn's 2024 survey of 31,000 knowledge workers in 31 markets, 78% of AI users brought their own AI tools to work and 52% were reluctant to admit using AI for their most important tasks ([Work Trend Index](#)). A price below the expense-report line turns an organizational decision into a personal one, at two costs: data leaves the firm, and the pipeline has no owner. The person on the spot sees a gap the apex has not seen and closes it locally.

3:23 Jev was built for pipelines of millions of requests a day, and whether it serves them yet cannot be checked: it has been in early access only since mid-September, and TypeSafe discloses no usage. Its price can be checked: ten million requests a day of 240 to 400 tokens cost about \$100 to \$170.

3:24 Decision models are easy to build, and they still raise money. Firelex released open models of 0.8 and 2 billion parameters that accept Jev's request format 13 days after Jev ([Firelex](#)); Fastino says it trained its models on less than \$100K of gaming GPUs ([BusinessWire](#)); Arcee says its whole 2025 lineup cost about \$20M to train ([Arcee](#)); and more than 95% of the roughly 40 trillion tokens Fireworks serves a day come from models specialized on customers' data ([Fireworks](#)). Yet TypeSafe launched with a \$40M seed round led by DCVC ([SiliconANGLE](#)) at a \$200M valuation, according to a person familiar with the deal ([Forbes](#)), Arcee passed a \$1B valuation in September, and Fireworks raised at \$17.5B in July. Investors fund the race to the cheapest sufficient model whether or not it leaves a moat.

3.4 What a cheaper tier makes viable

3:25 A cheaper tier opens only tasks it suffices for, and few of those where the hands dominate. A workflow must be built before it runs, and the **hands** are the human labor that turns a viable design into a built one: for a workflow, the integration work of engineering, data plumbing, evaluation and approvals, at a yearly cost c_{int} per workflow, and for buildings and grids, the trades. A workflow is built when its value per call covers the cheapest sufficient price plus its share of the hands, $u_t > p^*(q_t^*) + c_{\text{int}}/n_t$, with the left side shrunk by the expected cost of failed calls. Three levers open a task: a lower price, cheaper hands and more calls.

3:26 Today the hands bind hard, and only a fraction of viable workflows is built. About 18% of US firms used AI in any business function between November 2025 and January 2026 ([Census](#)). Claude's observed use covers 33% of the tasks in computer and mathematical occupations, against 94% that are feasible in theory ([Anthropic](#)). On GDPval, counting expert review and redoing erases nearly all of a model's cost advantage over the expert ((2.3)). Postings on Indeed for forward-deployed engineers, whose job is wiring models into a customer's systems, rose about 729% in the year to April 2026 ([Business Insider](#)). Price is one barrier among several, and for most workflows not the largest.

3:27 Part of the hands is permission: procurement, security review, a manager's attention. A ten-dollar experiment skips all of it (3:21), so a lower price lowers the hands as well.

3:28

Proposition 3.1 (The share a cheaper tier adds). Add a tier at price p_{new} and capability Q_{new} , below the old cheapest sufficient price $p_{\text{old}}(q)$ for every $q \leq Q_{\text{new}}$. Ignore failures, and let $\ln u_t$ be independent of (q_t^*, n_t) with density at most M_u . Then the share of tasks newly built is at most

$$M_u \mathbb{E}_t \left[\mathbf{1}\{q_t^* \leq Q_{\text{new}}\} \ln \frac{p_{\text{old}}(q_t^*) + c_{\text{int}}/n_t}{p_{\text{new}} + c_{\text{int}}/n_t} \right] \leq M_u \Pr_t [q_t^* \leq Q_{\text{new}}] \ln A_c^{\max}$$

with $A_c^{\max} = \max_q p_{\text{old}}(q)/p_{\text{new}}$. A cheaper tier builds only tasks it suffices for, only inside a band of width $\ln A_c$ in log value, and almost nothing where the hands dominate the cost per call. ($M_u, p_{\text{old}}, p_{\text{new}}$ and Q_{new} are local.)

▼ *Proof*

Fix a threshold and a volume with $q_t^* \leq Q_{\text{new}}$. The tasks the new tier builds are those whose value per call lies above $p_{\text{new}} + c_{\text{int}}/n_t$ and at or below $p_{\text{old}}(q_t^*) + c_{\text{int}}/n_t$: an interval in $\ln u_t$ whose length is the log-

ratio, so its probability is at most M_u times that length. Integrating over thresholds and volumes gives the first bound. The log-ratio falls as c_{int}/n_t grows and is at most $\ln(p_{\text{old}}/p_{\text{new}}) \leq \ln A_c^{\text{max}}$, which gives the second. ■

3:29 A new tier builds at most a fixed fraction of the tasks it suffices for: with values spread lognormally at a log-standard deviation of 2, $M_u = 1/(2\sqrt{2\pi}) \approx 0.2$, and the step from Haiku 4.5 to Jev ($A_c \approx 36$ at the median call) builds at most 0.71 of them. Where the hands dominate ($c_{\text{int}}/n_t \gg p_{\text{old}}$) the log-ratio is about $(p_{\text{old}} - p_{\text{new}}) n_t / c_{\text{int}}$, close to zero, so cheaper tokens build almost nothing and only volume or cheaper hands help: in the market of \$3.5, for an easy task at 1,000 calls a day the cheapest value per call worth building is \$0.055 with Jev and \$0.057 with Haiku 4.5, and at a million calls a day \$0.00012 against \$0.0023. The gains concentrate where $n_t > c_{\text{int}}/p_{\text{old}}$, in pipelines of millions of requests a day. A cheaper tier opens work at the margin the hands have already reached.

3:30 Because the hands bind, coding agents came first among agents: they raise the hands available to every other kind of workflow by building the pipelines themselves. They are the cognitive counterpart of the physical hands that limit the build-out (§15.4).

3.5 A market of tiers

3:31 A synthetic market built from these definitions puts the three levers to work at once.

3:32

Experiment 3.1 (The compute-asymmetry market).

Setup: 200,000 workflow types, each with a value per successful call (lognormal, median \$0.01, log-standard deviation 2), a volume of 10 to 10^7 calls a day (median about 1,000), a median prompt of 1,500 tokens and answer of 150, a difficulty and a shape (55% bounded decisions). Four tiers at October 2026 list prices: frontier (\$10/\$50), mid (\$2/\$10), Haiku 4.5 (\$1/\$5) and Jev (\$0.042, output free), the last strong on decisions and weak on open generation. Success is logistic in capability minus difficulty, a failed call destroys its value, and every workflow needs \$20,000 a year of hands. Each workflow takes its surplus-maximizing tier and is built if that surplus is positive; menus widen from the frontier alone to all four tiers, then cut the hands tenfold and a hundredfold.

Result: 82.6% of workflows are saturated at Jev's tier and 0.9% need the frontier; the frontier-only market overpays \$16.4B a year; adding the three cheaper tiers cuts inference spend 89% and raises surplus 19%; removing Jev from the full menu loses 2.3% of surplus (1.96–2.40% across seeds) though it serves 82.8% of tokens; cutting the hands a hundredfold builds 2.5 times as many workflows but adds under 1% to surplus; spend rises as all prices fall 169-fold once the elasticity of demand exceeds 0.72.

Shows: value per call decides what gets built; cheaper tiers move money to deployers; the hands set the number of workflows, while the surplus sits in the few valuable ones already built.

3:33

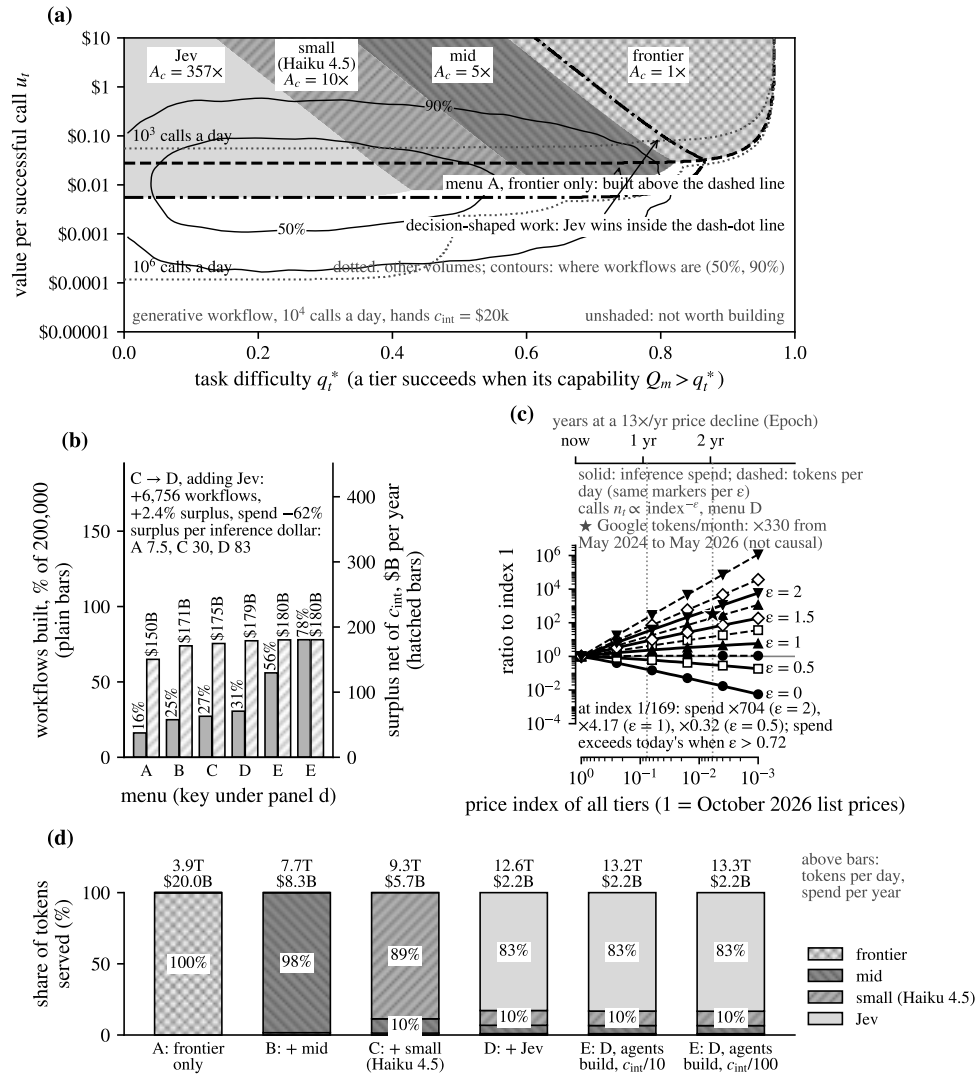


Figure 3.1. The synthetic market: (a) the tier a generative workflow chooses, by task difficulty q_t^* and value per successful call u_t , at 10,000 calls a day and hands c_{int} of \$20,000 a year, with a dash-dot line around the decision-shaped work Jev's tier wins; (b) workflows built and surplus net of the hands under six menus, from the frontier alone to all four tiers with the hands cut tenfold and a hundredfold; (c) inference spend and tokens as a price index of all tiers falls, for elasticities ϵ from 0 to 2; (d) each tier's share of tokens by menu. Jev's tier wins the easy and the decision-shaped work, wider menus cut spend far more than they raise surplus, cheaper hands multiply workflows while surplus barely moves, and spend rises with falling prices once the elasticity passes about 0.72. Jev's tier is the tiny tier of the appendix tables, and it also serves easy open-ended work, where Jev only decides.

3:34 In the model, saturation is the norm and overpayment with it: under the frontier-only menu, 82.5% of inference spend goes to workflows Jev would answer as well. Most surplus sits in workflows worth more than about \$0.10 a call, where even frontier prices are a small share of value, so the market-wide compute asymmetry is 29 $\times$, a twelfth of the 357 $\times$ of a median call, and the work Jev newly opens is worth a median \$0.0035 a call. The compute asymmetry changes the price paid far more than the decision to build.

3:35 Cheaper tiers move money from the companies that sell models to the companies that deploy them. From the frontier-only menu to the full one, inference spend falls from \$20.0B to \$2.2B a year while surplus rises from \$150B to \$179B, so surplus per inference dollar rises from 7.5 to 83. Of the gain from adding Jev, 85% is cost

saved on workflows Haiku 4.5 already served, and in the full menu Jev serves 83% of tokens while earning 6.7% of inference revenue.

3:36 The hands govern how many workflows exist. Cutting their cost tenfold or a hundredfold builds 50,824 or 94,789 more workflows, 56% or 78% of all of them, but adds only 0.8% or 0.9% to surplus, mostly as hands saved on workflows already built, because the top 1% of built workflows hold two-thirds of the surplus. Building the full menu's 61,187 workflows at \$20,000 each takes about 6,100 engineer-years. If the hands cost more for more valuable workflows, as real integrations often do, they would bind the valuable head as well. As coding agents cut the hands, I expect adoption to spread first through small, low-volume workflows, and the number of automated workflows per firm to grow much faster than AI spend per firm.

3:37 Jev's tier is decisive in one corner. Its marginal share of surplus is 2.3% at baseline, 12.9% when the median call is worth \$0.001, 11.9% when value and volume are anticorrelated, 6.5% with prompts of 6,000 tokens and 18.8% when a fifth of workflows need an answer within a second, which only Jev's tier meets. It nearly vanishes, to 0.3%, when GPT-6 Luna replaces Haiku 4.5 on the menu: a new tier is worth what it saves against the next-cheapest sufficient model.

3:38 The model's Jev tier also does easy open generation, standing in for cheap generative models; restricted to decisions, as the real Jev is, its share falls from 2.3% to 1.8%. The model holds verification fixed and has no routers or cascades. A cheap verifier would let a cascade run Jev first and escalate only on failure, keeping frontier reliability at nearly Jev's cost. Its distributions are assumptions, so its shares and ratios carry more weight than its dollar levels, which move by about $\pm 10\%$ across seeds (§B.9).

3.6 Jevons for cognition

3:39 Whether cheaper intelligence raises total spending depends on one elasticity, which equals the tail index of latent task values and which the data do not yet fix.

3:40

"It is wholly a confusion of ideas to suppose that the economical use of fuel is equivalent to a diminished consumption. The very contrary is the truth." W. S. Jevons, *The Coal Question*, 1865, chapter VII

3:41 With constant-elasticity demand for calls, $n \propto p_{\text{tok}}^{-\varepsilon}$ while tokens are all that a call costs, so spend scales as $p_{\text{tok}}^{1-\varepsilon}$; when tokens are only a share s_{tok} of what a workflow costs, a cut in the token price moves the workflow's price by that share:

3:42

$$\frac{d \ln \text{spend}}{d \ln p_{\text{tok}}} = 1 - \varepsilon, \quad \frac{d \ln \text{spend}_{\text{tok}}}{d \ln p_{\text{tok}}} = 1 - s_{\text{tok}} \varepsilon \quad (3.3)$$

3:43 A price cut raises spend if and only if $\varepsilon > 1$: that is the **Jevons effect**. Token spend rises only if $\varepsilon > 1/s_{\text{tok}}$. Where review dominates the cost, as on GDPval (\$0.76 of tokens against \$86 of expert review per task, so $s_{\text{tok}} \approx 0.01$; GDPval), cheaper tokens barely move the demand for tokens. The Jevons effect bites where the model is most of the cost, as in the pipelines Jev was built for.

Interactive in the online edition: <https://future-seems-so-good.com/blog/the-asymmetry-engine#fig:jevons>

Figure 3.2 (interactive). Spend against the factor by which the unit price falls, on log–log axes, for several elasticities. Drag ε through 1 and watch spend go from falling, to flat, to rising as the price falls; then add the extensive margin and watch the lines bend upward, so that spend rises even below unit elasticity.

3:45 Satya Nadella invoked the effect on 27 January 2025, the day Nvidia lost about \$589B of market value over DeepSeek-R1: "Jevons paradox strikes again!" ([Nadella](#); [CNBC](#)). Jevons had also stated the mechanism that ties the effect to thresholds: Savery's early steam engine "consumed no coal, because its rate of consumption was too high" (*The Coal Question*). A use that is not viable at today's price consumes nothing, and a lower price creates it.

Proposition 3.2 (The elasticity is a tail index). If latent task values per call have a Pareto tail of index ζ , calls have a fixed number of tokens, and a task runs if and only if its value covers the price of its call, then on the extensive margin $\text{spend} \propto p_{\text{tok}} \cdot p_{\text{tok}}^{-\zeta} = p_{\text{tok}}^{1-\zeta}$, so $\varepsilon = \zeta$. The margin saturates once the price falls below the least valuable task; after that, spend falls in proportion to price unless new low-value tasks keep appearing.

3:47 If valuable tasks are few ($\zeta < 1$), cheaper tokens mostly subsidize uses that already exist and spend falls; if a vast tail of low-value tasks waits for a price ($\zeta > 1$), every cut opens more spending than it saves. The belief that the demand for intelligence is nearly bottomless is, formally, the claim $\zeta > 1$. TypeSafe's case for Jev, "Every order of magnitude drop in the cost of intelligence unlocks orders of magnitude more use cases" ([TypeSafe](#)), read literally, says $\zeta \geq 2$.

3:48 Prices at fixed capability fall fast, at a rate that depends on the index. Algorithmic progress alone gives about 3× a year ([Gundlach et al.](#)); a16z and Sam Altman put the fall at about tenfold a year ([a16z](#); [Altman](#)); Epoch's September 2026 series falls about 47% a quarter, or 13× a year, slowing from 75× a year when a capability level debuts to 4.7× two years later ([Epoch](#)); and GPT-4's level on PhD-level science questions got 40× cheaper a year, within a range of 9× to 900× across tasks ([Epoch](#), 2025).

3:49 Google processed 9.7 trillion tokens a month in May 2024, about 480 trillion in May 2025 ([Google](#), 2025) and 3.2 quadrillion in May 2026, about 330-fold in two years ([Google](#), 2026), and its model APIs alone ran at about 22 billion tokens a minute in mid-2026 ([Alphabet](#)). Bain reports that the price of tokens halved from December 2024 to December 2025 while tokens consumed grew 4.5-fold, so spend rose 2.25-fold ([Bain](#)); the naive arc elasticity, $\ln 4.5 / \ln 2 \approx 2.2$, mixes shifts of the demand curve with movement along it.

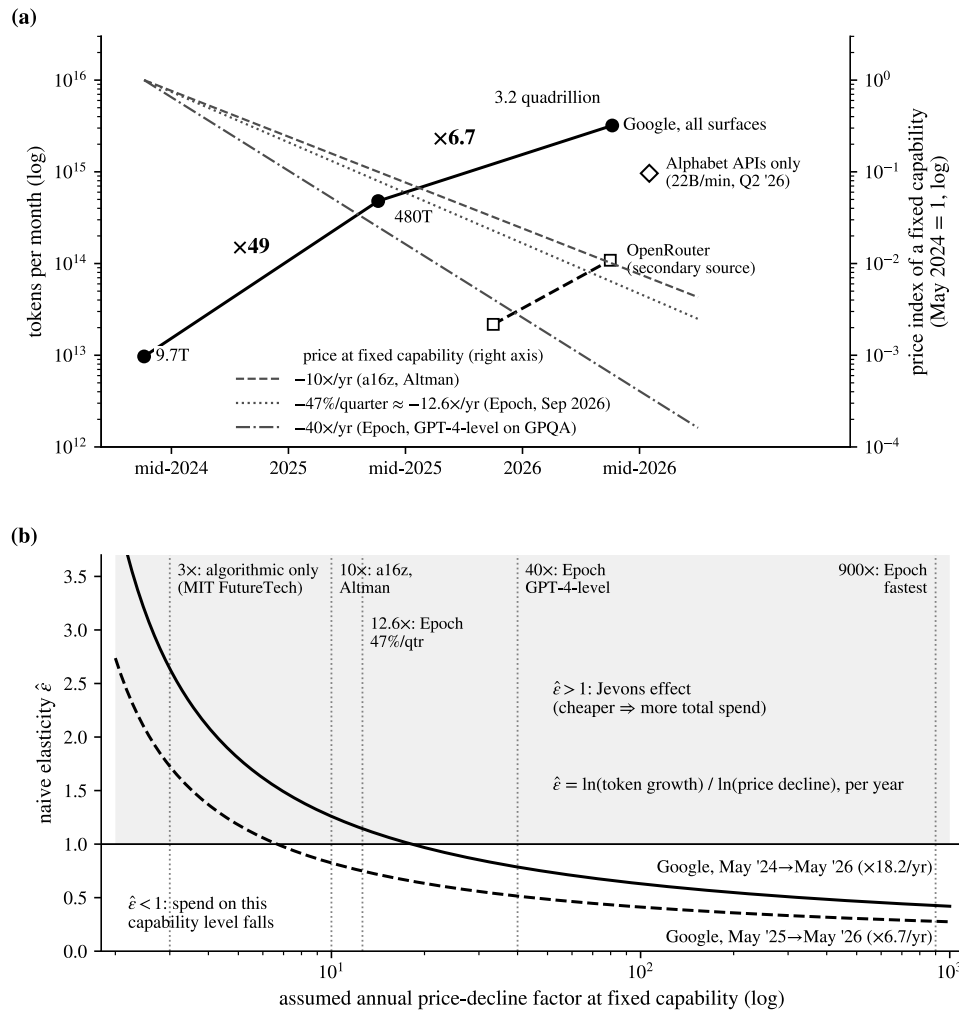


Figure 3.3. (a) Monthly tokens processed by Google, by its APIs alone and by OpenRouter (from a secondary report), against the price of a fixed capability under three published rates of decline; (b) the naive elasticity $\hat{\epsilon}$, the logarithm of Google's yearly token growth over the logarithm of an assumed yearly price decline, over the two years to May 2026 and over the last year alone, with the published rates marked and the region $\hat{\epsilon} > 1$ shaded. The two-year curve crosses 1 at an 18-fold decline, between the published rates, and the last-year curve below sevenfold, so $\hat{\epsilon}$ measures sensitivity to the deflator and token counts alone cannot say whether demand is elastic.

3:51 The ratio of their logarithms is the naive elasticity $\hat{\epsilon}$, and the answer flips with the deflator. Over the two years, with tokens growing 18.2 $\times$ a year, it is 1.26 at a tenfold annual price decline and 1.15 at 12.6-fold, a Jevons effect; at 40-fold it is 0.79, none. Over the latest year, with growth of 6.7 $\times$, it is 0.82 even at tenfold. Token counts also mix the price effect with new capabilities, new products and new uses.

3:52 The one causal estimate uses price differences among providers of the same model and finds elasticities of -1.08 and -1.11 , "very close to one"; its authors read that "as going against the Jevons paradox in the short-run" while leaving longer-run effects open (Demirer, Fradkin, Tadelis and Peng). Buyers switch between providers of one model more easily than they give up a task, so the estimate is an upper bound on the market-wide elasticity. Across models the slope is far flatter: a 10% price cut goes with only a 0.5–0.7% rise in usage (OpenRouter and a16z).

3:53 In the market of Experiment 3.1, falling prices also build new workflows and pull calls up to dearer tiers, so spend rises once the elasticity exceeds about 0.72 for a 169-fold cut (0.67 for a 13-fold one), and an elasticity of 1.10 reproduces Google's growth, though only as a fit. The causal estimate lies above that threshold and the cross-model slope far below it. The data do not yet fix the sign of the aggregate effect.

3:54 What the data do show is demand climbing the capability ladder. Growth at fixed capability slowed while OpenAI's flagship list price rose, from GPT-5 at \$1.25/\$10 in August 2025 ([OpenAI](#)) to GPT-6 Astra at \$10/\$50 in September 2026, and spending on agents exploded: Cursor passed \$4B in annualized revenue in June 2026, double February's figure ([Forbes](#)). In Bain's words, "When the next Claude or GPT ships, nobody says, 'Great, I'll keep using the old model and pocket the savings.' They upgrade" ([Bain](#)). Cheaper intelligence crossed thresholds and created new demand curves, as the engines that improved on Savery's did, so for cognition the Jevons effect counts thresholds crossed. It is the regeneration of the master equation: closing the compute gap opens new gaps.

3.7 Surplus value, precisely

3:55 An agent's markup over its inference bill grows without bound as tokens get cheaper, but in Marx's accounting the agent is constant capital, so the margin is a temporary extra surplus value that competition passes on.

3:56 Marx measured exploitation by the rate of surplus value. The worker sells labour-power, whose value is "the value of the means of subsistence necessary for the maintenance of the labourer", including "a historical and moral element" ([Capital I, ch. 6](#)). The capitalist pays that value as a wage, the working day creates more value than the wage, and the ratio of the excess to the wage is "an exact expression for the degree of exploitation of labour-power by capital" ([ch. 9](#)). Treat an agent's inference bill as its wage, and the analogue is a markup:

3:57

$$\text{markup} = \frac{u_{\text{job}} - p_{\text{tok}} n_{\text{tok}}}{p_{\text{tok}} n_{\text{tok}}} \xrightarrow{p_{\text{tok}} \rightarrow 0} \infty \quad (3.4)$$

3:58 Here u_{job} is the value the agent's work creates and $p_{\text{tok}} n_{\text{tok}}$ its inference bill (both local). A \$5 support ticket resolved in 5,000 tokens on a model at \$0.20 per million costs \$0.001, a markup near 5,000. Moving a task from the frontier to its cheapest sufficient model multiplies the markup by about A_c : a \$0.01 decision served by Jev at \$0.000063 carries a markup of about 158, while served by the frontier at \$0.0225 it loses money and is never built.

3:59 The analogy fails as ontology and holds as algebra. Means of production "never transfer more value to the product than they themselves lose" ([ch. 8](#)), so by Marx's definitions an agent is constant capital and creates no value. What an agent-run firm captures is the human labour still in the loop (prompting, reviewing, integrating), the labour congealed in the model, and the early adopter's extra surplus value, which "vanishes, so soon as the new method of production has become general" ([ch. 12](#)). **Surplus value** here names the margin of value over an agent's inference bill, read as Marx's category and held to his caveat. The same accounting shows any basic input, steel or compute, to be "exploited" (Roemer, quoted in [Basu, 2021](#)).

3:60 GDPval locates the margin. Counting tokens alone, a model's deliverable carries a markup of about 47,000%; counting expert review and redoing, about 63%, below Marx's textbook rate of 100% ([GDPval](#)). The reviewer is the bottleneck.

3:61 A markup measures how much surplus exists; who keeps it is a separate question. Bargaining holds a worker's wage near the historical and moral cost of reproducing labour-power. An agent's wage has no such anchor: the agent cannot strike, and under competition its price is pinned near the marginal cost of compute, so the deployer's markup is competed away like any extra surplus value. Where many models clear a threshold, competition pushes token prices toward cost: open models already cost about 90% less than comparable closed ones ([Demirer, Fradkin and Tadelis, 2026](#)), and an open Jev-compatible model appeared 13 days after Jev (3:24).

3:62 The surplus goes to three places. Buyers get lower prices and better products. Owners of scarce complements earn rents on what cannot be multiplied at software speed: power, memory, fabs, land with a grid connection (Proposition 16.1; Figure 16.2). Owners of verification earn rents once generation is free and the check is scarce: test suites, reputations, distribution, regulatory approvals (§2.8). The frontier labs sit between, earning rent only on tasks the cheaper tiers cannot do. The general case, with the cap on wages in tasks compute can reproduce, is Proposition 16.3; ownership of the general intellect is §17.2.

3.8 Why the frontier is still bought

3:63 The frontier sells what cheaper tiers cannot do, and it must keep moving to keep selling it, because each frontier, once surpassed, becomes the cheapest sufficient model for more tasks.

3:64 If the cheaper tiers take every task within their reach, the frontier earns from three sources: tasks only it can do, such as research, hard code and security; tasks whose failures are dear ((3.2)); and planning, where a few frontier tokens set up many cheap ones. In Cursor's build the frontier planner wrote a small share of the tokens and most of the cost, and its quality set the workers' bill (§9.6).

3:65 Each new cheaper tier takes part of the first source, so a lab that stops moving its frontier loses its territory. The price of a fixed capability falls about 13× a year (3:48), while the price of the most capable tier falls far more slowly. GPT-4 launched at \$30/\$60 per million tokens in March 2023 ([OpenAI](#)); GPT-6 Astra and Claude Fable 5.1 cost \$10/\$50, and Claude Opus 5.5, which tops the index, \$4/\$20 ([Anthropic](#)). In three and a half years the top tier got 1.9 to 4.7 times cheaper per blended token, about what a fixed capability loses in three to seven months. The cost of running frontier models on benchmarks has risen 3–18× a year as reasoning burns more tokens ([Gundlach et al.](#)). Cheap intelligence is cheap one rung below the frontier. Each frontier's capability is sold again later, by cheaper models, at a fraction of its price, so A_c for a fixed task widens until the task reaches the bottom of the price list. This treadmill is one engine of the capital spending in §16.1.

3:66 A cheap model can still be a business, because rent survives wherever a complement is scarce, and a decision model can own several. Millions of calibrated decisions form a feedback stream that improves the model. Listings on the Vercel, OpenRouter and Cloudflare gateways ([InfoQ](#)) and default status carry distribution. Typed outputs make every answer machine-checkable. An answer in 70 to 500 ms serves real-time uses that a frontier answer taking about 685 seconds at maximum effort cannot ([Artificial Analysis](#); Definition 5.1). Calibrated probabilities let a deployer set an escalation threshold, sending low-confidence cases to a bigger model or a person (§4.5). I think the first job of a cheap model is routing, since the routing decision is itself a classification; the market above has no routers to test it.

3:67 Efficiency alone is no moat. Firelex's open Jev-compatible models arrived within two weeks, and distil labs found that a fine-tuned model of 0.8 billion parameters beats Jev on price above about 120,000 messages an hour,

where a dedicated GPU costs the same idle or busy ([distil labs](#)). Jevons said it in 1865: cheapness "cannot be procured or retained by inventions and modes of economy which are as open to our commercial competitors as to ourselves" ([The Coal Question](#)). An inherently cheaper server can win a price war and keep a margin equal to its rivals' cost minus its own, unless, as TypeSafe concedes it cannot rule out, its price is subsidized.

3:68

Forecast 3.1 (The token barbell). By the end of 2028 the share of tokens polarizes: models priced at a tenth of the frontier or less serve a rising majority of tokens, the frontier tier keeps the largest share of dollars through planning and dear failures, and the middle tier's share of tokens shrinks.

Horizon 2028-12-31

Probability 35%

Check Sort the models in OpenRouter's public token rankings for 2026 and 2028 into three tiers by output list price (frontier, within a factor of two of the highest standard flagship price; cheap, a tenth of that price or less; middle, the rest), and estimate each tier's spend as its tokens times list prices. The forecast holds if in 2028 the cheap tier serves more than half of tokens and a larger share than in 2026, the middle tier serves a smaller share of tokens than in 2026, and the frontier tier has the largest share of spend; it fails otherwise.

3:69

Forecast 3.2 (The hands collapse next). By the end of 2028, as coding agents build the pipelines, the share of US firms using AI in any business function, about 18% in late 2025, roughly doubles.

Horizon 2028-12-31

Probability 45%

Check Read the last 2028 estimate of the share of firms using AI in any business function in the Census [Business Trends and Outlook Survey](#). The forecast holds at 32% or more and fails below it.

3:70

Forecast 3.3 (Review's share of cost rises). Through 2028, review and verification cost per task falls more slowly than the price of tokens, so review's share of the cost of delivered work rises.

Horizon 2028-12-31

Probability 70%

Check Compare expert-review time and price per deliverable on GDPval or its successor, 2026 against 2028, with Epoch AI's price of a fixed capability over the same years. The forecast holds if review cost per task fell by a smaller factor than that price; it fails if it fell by as large a factor or more, for example because automated verifiers match expert agreement across most GDPval occupations, or if no 2028 review measurement is published by the horizon.

References

1. Geoffrey Hinton, Oriol Vinyals, Jeff Dean (2015). Distilling the Knowledge in a Neural Network. *arXiv*. <https://arxiv.org/abs/1503.02531>
2. distil labs. <https://www.distillabs.ai/blog/jev-or-a-fine-tuned-small-model-we-built-a-pipeline-with-both-to-see-the-real-difference/>

3. Artificial Analysis. <https://artificialanalysis.ai/leaderboards/models>
4. OpenAI. <https://openai.com/index/introducing-gpt-6-1-sol>
5. GPT-6 Astra. <https://openai.com/index/gpt-6-astra/>
6. Claude Fable 5.1. <https://www.anthropic.com/pricing>
7. GPT-6.1 Sol. <https://developers.openai.com/api/docs/pricing>
8. Claude Haiku 4.5. <https://www.anthropic.com/news/claude-haiku-4-5>
9. GPT-6 Luna. <https://openai.com/index/introducing-gpt-6-sol-and-luna/>
10. Jev. <https://typesafe.ai/blog/introducing-system-one-models-and-jev>
11. gpt-oss-20b. <https://openrouter.ai/api/v1/models>
12. Martin Juan José Bucher, Marco Martini (2024). Fine-Tuned 'Small' LLMs (Still) Significantly Outperform Zero-Shot Generative AI Models in Text Classification. *arXiv*. <https://arxiv.org/abs/2406.08660>
13. Lingjiao Chen, Matei Zaharia, James Zou (2023). FrugalGPT: How to Use Large Language Models While Reducing Cost and Improving Performance. *arXiv*. <https://arxiv.org/abs/2305.05176>
14. Anthropic. <https://docs.anthropic.com/en/docs/about-claude/pricing>
15. Bain. <https://www.bain.com/insights/how-token-economics-will-change-opex/>
16. InfoQ. <https://www.infoq.com/news/2026/10/typesafe-ai-jev-released/>
17. gateway pricing. <https://jevtypesafeai.com/pricing>
18. Work Trend Index. <https://www.microsoft.com/en-us/worklab/work-trend-index/ai-at-work-is-here-now-comes-the-hard-part>
19. Firelex. <https://github.com/firelex/jeff>
20. BusinessWire. <https://www.businesswire.com/news/home/20250506538922/en/Fastino-Launches-TLMs-Task-Specific-Language-Models-with-%2417.5M-Seed-Round-Led-by-Khosla-Ventures>
21. Arcee. <https://www.globenewswire.com/news-release/2026/09/16/3363293/0/en/arcee-ai-reaches-1b-valuation-with-series-b-funding-to-advance-frontier-open-weight-ai.html>
22. Fireworks. <https://fireworks.ai/blog/series-d-announcement>
23. SiliconANGLE. <https://siliconangle.com/2026/09/16/typesafe-ai-exits-stealth-with-40m-to-build-ai-for-use-by-software/>
24. Forbes. <https://www.forbes.com/sites/the-prompt/2026/09/15/this-200-million-startup-wants-to-fix-ais-overconfidence-problem/>
25. Census. <https://www.census.gov/library/stories/2026/05/ai-use-businesses.html>
26. Anthropic. <https://www.anthropic.com/research/labor-market-impacts>
27. Business Insider. <https://www.businessinsider.com/forward-deployed-engineer-jobs-in-demand-2026-5>
28. The Coal Question. <https://freecapitalists.org/books/the-coal-question-an-inquiry-concerning-the-progress-of-the-nation-and-the-probable/read/chapter-vii-of-the-economy-of-fuel/>
29. Tejal Patwardhan, Rachel Dias, Elizabeth Proehl, Grace Kim, Michele Wang et al. (2025). GDPval: Evaluating AI Model Performance on Real-World Economically Valuable Tasks. *arXiv*. <https://arxiv.org/html/2510.04374v1>
30. Nadella. https://www.linkedin.com/posts/satyanadella_jevons-paradox-wikipedia-activity-7289521182721093633-5gJ5
31. CNBC. <https://web.archive.org/web/20250128201614/https://www.cnbc.com/2025/01/27/nvidia-sheds-almost-600-billion-in-market-cap-biggest-drop-ever.html>
32. Hans Gundlach, Jayson Lynch, Matthias Mertens, Neil Thompson (2025). The Price of Progress: Price Performance and the Future of AI. *arXiv*. <https://arxiv.org/abs/2511.23455>
33. a16z. <https://a16z.com/llmflation-llm-inference-cost/>
34. Altman. <https://blog.samaltman.com/three-observations>
35. Epoch. <https://epoch.ai/publications/the-plunging-price-of-thought>
36. Epoch, 2025. <https://epoch.ai/data-insights/llm-inference-price-trends>
37. Google, 2025. <https://blog.google/innovation-and-ai/technology/ai/io-2025-keynote/>
38. Google, 2026. <https://blog.google/innovation-and-ai/sundar-pichai-io-2026/>
39. Alphabet. <https://blog.google/company-news/inside-google/message-ceo/alphabet-earnings-q2-2026/>

40. Mert Demirer, Andrey Fradkin, Nadav Tadelis, Sida Peng (2025). The Emerging Market for Intelligence: Pricing, Supply, and Demand for LLMs. *National Bureau of Economic Research*. <https://www.nber.org/papers/w34608>
41. OpenRouter and a16z. <https://openrouter.ai/state-of-ai>
42. OpenAI. <https://openai.com/index/introducing-gpt-5-for-developers/>
43. Forbes. <https://www.forbes.com/sites/richardnieva/2026/06/08/cursor-4-billion-annualized-revenue/>
44. Capital I, ch. 6. <https://www.marxists.org/archive/marx/works/1867-c1/ch06.htm>
45. ch. 9. <https://www.marxists.org/archive/marx/works/1867-c1/ch09.htm>
46. ch. 8. <https://www.marxists.org/archive/marx/works/1867-c1/ch08.htm>
47. ch. 12. <https://www.marxists.org/archive/marx/works/1867-c1/ch12.htm>
48. Basu, 2021. <https://www.econstor.eu/bitstream/10419/238148/1/1744231621.pdf>
49. GDPval. <https://openai.com/index/gdpval/>
50. Mert Demirer, Andrey Fradkin, Nadav Tadelis (2026). The Emerging Market for Intelligence: How Firms Buy and Sell AI. *The Journal of Economic Perspectives*. <https://www.aeaweb.org/articles?id=10.1257%2Fjep.20261506>
51. OpenAI. <https://openai.com/index/gpt-4-research/>
52. Anthropic. <https://www.anthropic.com/claude-opus-5-5>
53. The Coal Question. <https://bpb-us-w2.wpmucdn.com/campuspress.yale.edu/dist/0/4222/files/2023/06/Jevons-The-Coal-Question.pdf>
54. Business Trends and Outlook Survey. <https://www.census.gov/hfp/btos>