

The Price of Thought, Recomputed

Method report on a reanalysis of Epoch AI, "The plunging price of thought" (September 22, 2026): data, exact procedure, alternative specifications, robustness and limitations.

The bargain hunter

switches to the cheapest model every quarter. This is how the study calculates.

€100 today, in one year:

€8

Range €4 to €11

The normal company

sticks with OpenAI, Anthropic or Google and a typical current model. Model calculation.

€100 today, in one year:

€96

Range €86 to €119

The demanding one

takes every new model at the highest setting.

€100 today, in one year:

€334

Range €252 to €458

All three numbers come from the same data. None of them is wrong. They answer different questions. The headline "13 times cheaper" answers only the first.

UP FRONT

This report is nothing against Epoch AI. You can calculate it this way, and Epoch states the limits of its own number in the report. You just have to put the number in context, and whoever reads it has to understand what it measures. The subject of this report is the interpretation, not the source.

Author	Jonas Höttler, Balane
Version, date	1.0, October 2026
Data	Epoch AI, github.com/droodman/inference-cost , commit 9163c17 of September 24, 2026, retrieved October 19, 2026. In addition Epoch's workbook "Cost declines by tech.xlsx", Eurostat, Fouquet (2011), Ember, FRED, getdeploying.com .
Companion article	https://www.hoettler.com/en/blog/price-of-thought (with interactive calculator)
Reproduction	All scripts in drafts/preis-des-denkens/analyse/ (Section 12)

Question, approach, results

Question. Epoch AI reports that the price of a fixed level of AI performance has fallen by about 47% per quarter since 2023, roughly 13 times per year. On social media this becomes "the same performance gets 13 times cheaper every year, the direction is clear." We check what the number measures, how much it depends on defensible assumptions, and what it means for customers who don't always pick the cheapest model.

Data. Epoch's public dataset: 15,959 rows from 1,298 benchmark runs (225 model names), after Epoch's preparation 11,240 rows from 189 models on 11 benchmarks. Prices are list prices at the time of measurement, 96% from OpenRouter.

Method. (1) Exact reproduction of Epoch's model-free calculation in Python. (2) Varying each assumption on its own and in all 512 combinations. (3) Three customer profiles and robustness checks (window length, leaving out single models, bootstrap). (4) Splitting the decline into token count and token price. (5) Extending the comparison series (electricity, compute) with Epoch's own and public data.

Results. The reproduction matches Epoch's values to the second decimal (47.14 / 42.96 / 53.11 / 46.97 / 44.01% per quarter). Depending on the specification, the result ranges from 26.4 times cheaper to 1.3 times more expensive per year. 53 of 512 combinations show rising costs. A typical price instead of the minimum gives 1.6 times cheaper per year. The typical price of the three big providers is practically flat (€100 becomes €96, range €86 to €119). Whoever uses every new model at the highest setting pays 3.3 times more per task per year (range 2.5 to 4.6). Three quarters of the decline at the record price come through cheaper tokens. Epoch's electricity curve ends in 1973. After that, US electricity did not get cheaper in real terms (1973 to 2025: ±0%).

Conclusion. At the cheapest edge of the market, the price really does fall fast. For most customers that means little. The data does not support "it will surely get dramatically cheaper." What it supports: it depends, dramatically, on which price, which providers, which tasks and which average you mean.

Table 0: Eight common readings and what the data shows

Reading in the feed	Finding of this report	Section
"The same performance gets 13 times cheaper."	Holds for the cheapest provider if you switch immediately. Typical model: 1.6 times. Typical model of the big three: practically flat.	5.1, 5.3
"13 times, every year."	No steady trend: 3.6 to 132 times depending on the half-year. At the median data point, the record price doesn't move at all.	5.6
"Faster than any technology before."	In a fair comparison (fastest three years), 1.3 times as fast as DNA sequencing instead of 4.2 times.	10
"Performance" means work.	Measured on mathematics, chess and exam questions. Coding and factual knowledge: 5.3 times.	5.4
"You pay a little more and get many times the intelligence."	Not measured. Every new model at the highest setting: 3.3 times more expensive per task per year.	5.3
"The direction is clear, so don't wait."	According to Epoch, new top performance initially gets about two thirds cheaper per quarter. That argues just as well for waiting.	11
"It'll keep going like this."	Three quarters of the decline hinge on the token price, so on power, chips and pricing policy.	7, 8
"If it's going this well, it's a sure thing."	Fast loss of value of new models is a risk for providers and investors, not a guarantee.	11

Contents

1	Background and questions	6	Robustness	11	Limitations of this analysis
2	Data	7	Decomposition: token count and token price	12	Reproducibility
3	Epoch's procedure, exactly	8	Scenario calculator	A	Glossary
4	Alternative specifications	9	External series	B	Six questions for any price statistic
5	Results	10	Technology comparison	C	Sources

SECTION 1

Background and questions

1.1 Epoch's claim

Epoch AI studies how fast the price falls at which an AI model reaches a fixed level of performance on a benchmark. The result of the model-free calculation: averaged over five benchmarks, 46.8% cheaper per quarter, or 12.5 times cheaper per year. The report rounds this to "about 13 times" and compares it with electricity, batteries, compute and DNA sequencing.

The report itself frames the number carefully:

"Essentially no user stays permanently on the cost frontier."

"when comparing price drops for AI to drops for other technologies for which we have price series, we are comparing apples and oranges."

"They should not be read as exact."

"supra-normal profits in LLM service provision may be fleeting for any given model"

These caveats get lost when the number is shared. A carefully framed measurement turns into "the same performance gets 13 times cheaper every year, the direction is clear," and from there into "AI is the safe bet."

1.2 Questions

1. Can Epoch's number be reproduced exactly from the published data?
2. How much does it depend on assumptions that could just as defensibly be made differently?
3. What follows for customers who don't switch to the cheapest model every quarter?
4. What is the decline made of, and what does extending it into the future assume?
5. Does the comparison with other technologies hold once their series are drawn to the end?

1.3 Framing

The critique is aimed at the interpretation, not at Epoch's calculation. Epoch's number is computed correctly (Section 3.6). But it measures something very specific: the lowest price any customer on the entire market could pay if they switched every quarter to the cheapest model at its most frugal setting. This report shows how far other, equally defensible measures lie from it.

IN PLAIN WORDS

Imagine someone says: "Flights get 13 times cheaper every year." The fine print says: they mean the cheapest flight on any airline, at five in the morning, no luggage, booked by someone who searches every airline every quarter and rebooks on the spot. That is exactly how the AI number is built.

SECTION 2

Data

2.1 Source and version

The basis is Epoch's public repository github.com/droodman/inference-cost, commit 9163c17 (September 24, 2026), retrieved October 19, 2026. We use:

- `data/cost_truncated_curves.csv`: one row per model, benchmark, setting (effort) and token budget. Columns include accuracy, mean tokens per task, cost per task in US dollars, accuracy of the full run, sample size, price date and price source.
- `data/Model versions-Grid view.csv`: model registry with release dates.
- `data/Cost declines by tech.xlsx`: the series of the comparison chart (sheet "Figure data"), raw electricity data to 2025 (sheet "Electricity") and a consumer price table (sheet "CPI").
- `src/prepare_data.R` and `src/envelope_frontier.R`: data preparation and the model-free rate, which we ported to Python.

2.2 Coverage by benchmark

The raw file contains 15,959 rows from 1,298 runs with 225 model names. After Epoch's preparation (Section 2.4), 11,240 rows from 189 models remain.

Table 1: Data coverage after preparation

Benchmark (key)	Content	Rows	Models	Top score	Release dates
AIME (aime)	Math olympiad	2,159	146	100%	03/2023 to 07/2026
Chess puzzles (chess)	Chess puzzles	1,462	97	61.7%	06/2023 to 07/2026
FrontierMath Tier 1 to 3 (fm13)	Research mathematics	1,099	58	89.1%	11/2023 to 08/2026
GPQA (gpqa)	Science exam questions	2,164	166	93.6%	03/2023 to 07/2026
Mystery Game Puzzles (mystery)	Secret game, built against targeted training	623	37	26.2%	06/2023 to 07/2026
FrontierMath 2025-02 private	Research mathematics	822	50	47.2%	05/2024 to 05/2026
FrontierMath Tier 4 (2025-07 private)	Research mathematics	570	32	31.2%	06/2024 to 05/2026
FrontierMath Tier 4 v2	Research mathematics	499	38	87.8%	01/2025 to 08/2026
MATH Level 5	School mathematics, hard	808	91	94.8%	06/2023 to 10/2025
SimpleQA (simpleqa)	Factual knowledge	449	42	77.3%	10/2024 to 06/2026
SWE-bench Verified (swe_bench_verified)	Coding	585	43	73.4%	05/2024 to 06/2026

Highlighted: Epoch's five core benchmarks. Top score: highest guess-adjusted accuracy (Section 2.5). Source: our evaluation of `build_runs` (core.py).

2.3 Prices

Epoch measures list prices at the time, nominal, in US dollars. Price sources of the raw rows:

Table 2: Price sources (share of raw rows)

Source	Meaning	Share
openrouter-wayback	OpenRouter price from archived pages at the time of measurement	73.6%
openrouter-wayback, nearest-after	the same, nearest later archive snapshot	21.9%
openrouter-live, nearest-after	current OpenRouter price	0.7%
gpu-effective (incl. nearest-after)	self-hosted open models, converted via 2026 GPU rental prices	3.4%
moonshot / alibaba-first-party	the provider's own price	0.4%

About 96% OpenRouter in total. Power, chips, subsidies and margins are not recorded separately. They only act indirectly through these prices. The AI prices are not adjusted for inflation. Over three years that is a small effect, the real decline would be slightly larger.

Not all prices are from the release date. For some models the price date lies up to 2.7 years later (for example Llama-3.1-8B: 724 days, deepseek-llm-67b: 976 days, both via 2026 GPU rental). That tends to make early models too cheap, so it biases the decline downward rather than upward.

2.4 Preparation following Epoch's code

1. **Clean names:** strip -maas... suffixes and date suffixes -YYYY-MM-DD....
2. **Collapse** by model, benchmark, setting and budget: mean of accuracy, tokens, cost and full-run accuracy, sum of samples.
3. **Benchmarks:** rename (fm_tiers1_3_v2 → fm13, mystery_agent → mystery), drop public subsets (*_public).
4. **Release date** = earliest date among all registry entries whose id starts with the model name (case-insensitive). For the consequences see 4.7.
5. **Guess adjustment** (2.5), then drop **duplicates** on benchmark, model, setting, accuracy, cost and release date. Rows without valid cost (ln c not finite) or without a date are dropped.
6. **Time axis:** $t = (\text{release} - \text{Jan 1, 2023}) / 365.25$ in years, centered per benchmark: $t^c = t - \bar{t}_b$.

2.5 Guess adjustment

On multiple-choice tests, guessing alone gets you some accuracy (25% with four answers). Epoch sets a guess rate f_b per benchmark: the most frequent value among the accuracies at or below the median (the smallest one in case of a tie). Then

$$a' = \frac{\max(f_b, a) - f_b}{1 - f_b} \quad (1)$$

Result: GPQA $f = 0.25$, Mystery 0.092, AIME 0.001, Chess and FrontierMath 0. All further calculations use a' , written simply a below.

2.6 Simulated budget truncation

Each run comes in 12 to 13 budget steps (64, 128, ... tokens). Each step gives the accuracy and cost the run would have had if it had been cut off at that token budget. Epoch counts these simulated truncations (called "CAISI" in our scripts) as points of their own. The "full runs only" variant keeps only the step with the highest budget per run, the complete run (4.6).

2.7 Other data sources

Table 3: Supplementary series

Source	Content	Used in
Epoch, Cost declines by tech.xlsx	Comparison series, US electricity nominal to 2025 (Hirsh), CPI	Sections 9.1, 10
Fouquet (2011), LSE Grantham Institute	UK electricity price, real, 1883 to 2008	9.2
Eurostat nrg_pc_204, prc_hicp_aind	Household electricity, HICP consumer price index	9.2, 9.3
Ember	European wholesale electricity, day-ahead, monthly and daily	9.4
FRED	APU000072610 (US household electricity), DFF, ECBDFR	9.5, 9.7
getdeploying.com	GPU prices, weekly, median across providers	9.6
BTS, Deutsche Bahn, American Express	Air fares, rail fares	Aside in 10.4
Press and indices (Monitoring Analytics, SemiAnalysis, TrendForce, WSJ, FT)	PJM power market, chip rental prices, memory, company figures	only to orient the scenario values (8)

SECTION 3

Epoch's procedure, exactly

The headline number comes from Epoch's model-free calculation (`pareto_decline_qtr` in `envelope_frontier.R`). It needs no fitted functional form. We describe it for one benchmark b with runs i , each with accuracy a_i , cost per task c_i and centered release time t_i .

3.1 Record price and top score

The record price for level a at time t is the lowest price at which any model released by then, at any setting, reaches at least a :

$$C_b(a, t) = \min \{c_i: t_i \leq t, a_i \geq a\} \quad (2)$$

$$S_b(t) = \max \{a_i: t_i \leq t\} \quad (3)$$

$S_b(t)$ is the top score (state of the art) at time t . In (2), a model stays available forever at its measured price from its release on.

3.2 Measurement grid

Time: evenly spaced points t_k from $\min t_i$ to $\max t_i$, count $n = \max(2, \text{round}(\text{span} / h) + 1)$ with $h = 1 / 28$ year (about 13 days). Only points with a full horizon are used, $t_k + \Delta t \leq \max t_i$.

Levels: a fixed lattice of $N = 100$ midpoints across the achieved range, cut off at the current top score:

$$a_j = a_{\min} + \left(a_{\max} - a_{\min}\right) \frac{j - 1/2}{N}, \quad j = 1, \dots, N, \quad \text{node } (j, k) \text{ only if } a_j \leq S_b(t_k) \quad (4)$$

$a_{\min}$ is the smallest positive, $a_{\max}$ the largest observed accuracy on the benchmark. Epoch calls this lattice "fixed, clipped." Because the lattice ends at the top score, later dates have more nodes.

3.3 Change per node and quarterly rate

$$\delta_{jk} = \ln C_b(a_j, t_k + \Delta t) - \ln C_b(a_j, t_k), \quad \Delta t = 0.25 \quad (5)$$

$$r_b = 1 - \exp\left(\frac{0.25}{\Delta t} \cdot \bar{\delta}\right), \quad \bar{\delta} = \text{mean of all } \delta_{jk} \quad (6)$$

The arithmetic mean of log changes is a geometric mean of the price ratios. Nodes where the record price doesn't move within the quarter enter with $\delta = 0$ (ratio 1). Epoch's code reports the share of nodes that moved (`share_moved`).

3.4 Averaging over benchmarks and annual factor

$$\bar{r} = \frac{1}{5} \sum_b r_b, \quad F = (1 - \bar{r})^{-4}, \quad \text{bill in one year} = \frac{\text{€}100}{F} \quad (7)$$

With Epoch's values: $\bar{r} = 46.84\%$, $F = 12.52$, bill €8.0. All specifications and the calculator in the article use the same averaging (arithmetic mean of the quarterly rates over the selected benchmarks).

3.5 A property of the construction

In (2) the minimum is taken over a set that only grows over time. $C_b(a, t)$ therefore cannot rise, and $\delta_{jk} \leq 0$ at every node. Epoch's code says so itself: "Non-decreasing in BOTH arguments by construction, so its curves can never cross and it can never run backwards." So the fact that the price falls is not a finding in this calculation. It is built in. The only open question is how fast. The limited-lifetime variant (4.1) tests whether this matters. It barely does.

IN PLAIN WORDS

The 100-meter world record can only get better, because an old record stands until someone breaks it. That does not mean you run faster. The record price is a world record like that.

3.6 Reproduction

Table 4: Epoch's published values and our reproduction

Benchmark	Epoch, % per quarter	Reproduction	nodes moved	Nodes
AIME	47.1	47.14	47.7%	4,159
Chess puzzles	43.0	42.96	41.1%	3,031
FrontierMath Tier 1 to 3	53.1	53.11	62.2%	2,102
GPQA	47.0	46.97	46.4%	5,039
Mystery Game Puzzles	44.0	44.01	24.6%	3,518
Mean (7)		46.84	12.5× per year	

Source: core.py. The deviation is below 0.05 percentage points. Differences in the following sections therefore come from assumptions, not from the implementation.

3.7 What we did not rebuild

We did not port Epoch's smoothed regression models (the preferred one is a Box-Tidwell specification fitted to the record staircase). On average they land on the same number, 47.0%. The headline number doesn't need them. They are needed for the additional claim that new top performance gets about 66% cheaper per quarter right after it appears. That claim comes from heatmaps on which a large part of the surface has no measurements (Section 11).

Alternative specifications

Every number in a headline rests on decisions. We vary each decision for which another choice is just as defensible. All variants use the same preparation (Section 2) and the same averaging (7). Implemented in `precompute.py`.

Table 5: The switches

Axis	Epoch	Alternatives
Price	min: record price (2)	min24, min12: a model is withdrawn after 24 or 12 months (8). typ: typical price (9)
Providers	all	only OpenAI, Anthropic, Google (4.3)
Level lattice	fixed, clipped, linear	scaled (10), logit-spaced (11), both
Horizon Δt	one quarter	one year, converted via (6)
Runs	including simulated truncations	full runs only (4.6)
Release dates	as in the code	corrected (Table 6)
Average	mean of δ	median of δ (4.8)
Benchmarks	five core benchmarks	all eleven, or everyday work: SWE-bench Verified and SimpleQA

4.1 Price: limited lifetime

A model only counts during a window w after its release:

$$c_b^{(w)}(a, t) = \min \{c_i: t - w < t_i \leq t, a_i \geq a\}, \quad w \in \{1, 2\} \text{ years} \quad (8)$$

With this, the record price can rise again when the cheapest model so far drops out of the window.

4.2 Price: typical price ("typical model")

Instead of the cheapest model on the market, we take the model in the middle. For each model m , let $c_m(a)$ be its cheapest price at which it reaches level a . $M(t)$ is the set of models released in the twelve months before t that reach a :

$$c_m(a) = \min_{i \in m, a_i \geq a} c_i, \quad T_b(a, t) = \text{median}_{m \in M(t)} c_m(a), \quad M(t) = \{m: t - 1 < t_m \leq t\} \quad (9)$$

T_b is defined only if at least two models have a value. Nodes where T_b is undefined at either end are dropped. A specification needs at least 20 valid nodes per benchmark, otherwise it counts as "too little data" there. Because T_b can rise, a price increase is measurable here too.

4.3 Providers

A filter on the cleaned model name with the regular expression `^(gpt-(?!oss)|o1|o3|o4|claude-|gemini-):` GPT, the o-series, Claude and Gemini, without the open gpt-oss models. The filter applies after preparation, so the guess rate (1) stays that of the full dataset. On the five core benchmarks, that is 75 models.

4.4 Level lattice

Epoch's Figure 7 shows four lattices. Scaled means: at each date the lattice is laid across the range achieved so far, so every date has the same number of nodes:

$$a_j(t) = a_{\min} + \left(S_b(t) - a_{\min}\right) \frac{j - 1/2}{N} \quad (10)$$

Logit-spaced means: equal steps on the logit scale, with $\ell(x) = \ln(x / (1 - x))$ and bounds limited to $[0.001, 0.999]$:

$$a_j = \ell^{-1} \left(\ell(a_{\text{lo}}) + \left(\ell(a_{\text{hi}}) - \ell(a_{\text{lo}}) \right) \frac{j - 1/2}{N} \right) \quad (11)$$

with $a_{\text{hi}} = a_{\max}$ (fixed) or $S_b(t)$ (scaled). Logit lattices put more nodes at the edges, at very low and very high levels.

4.5 Horizon

With $\Delta t = 1$, the change is measured over a year and converted into a quarterly rate via (6). A longer horizon is diluted less by unmoved nodes, but averages over more history and allows fewer dates.

4.6 Runs

Only the row with the highest budget per run (`budget = max(budget) per run_id`), then the same preparation. This removes the simulated truncations (2.6).

4.7 Release dates

The prefix matching in 2.4 gives some models the date of an earlier entry with a similar name. Epoch's registry has the correct dates.

Table 6: Corrected release dates

Model	in the code (origin)	corrected
o3	Jan 31, 2025 (o3-mini)	Apr 16, 2025
o1	Sep 12, 2024 (o1-mini)	Dec 17, 2024
QwQ-32B	Nov 28, 2024 (preview)	Mar 5, 2025
gemini-2.5-pro, gemini-2.5-flash	preview date	Jun 17, 2025
gemini-1.5-pro-001	preview date	May 14, 2024

The o3 error appears word for word in the report: "On January 31, 2025, OpenAI released ... o3". A friendly note, not an argument: the effect on the headline number is small (Table 7).

4.8 Mean or median

Instead of $\bar{\delta}$ in (6), the median of all δ_{jk} . The median describes the middle node, the normal case for one level in one quarter. The mean describes the average including a few large jumps.

4.9 The combination grid

All axes except the benchmark selection can be combined freely: 4 prices $\times$ 2 provider sets $\times$ 4 lattices $\times$ 2 horizons $\times$ 2 run variants $\times$ 2 datings $\times$ 2 averages = **512 calculations**. For each, (7) is formed over the selected benchmarks. As in the calculator, the rates per benchmark enter rounded to 0.1 percentage points.

IMPORTANT

The 512 calculations are not a sample. They are neither independent nor equally plausible. We count them only to show the range that defensible decisions produce. That is not a probability.

SECTION 5

Results

5.1 Single specifications

Each row changes exactly one assumption relative to Epoch (except "typical, big three": two).

Table 7: Decline in % per quarter by benchmark, and annual factor

Specification	AIME	Chess	FM 1-3	GPQA	Mystery	Mean	per year
Epoch	47.1	43.0	53.1	47.0	44.0	46.8	12.5× cheaper
Only the big three	41.7	41.9	51.4	43.6	29.6	41.7	8.6× cheaper
Typical price (9)	9.8	-3.9	15.7	14.0	18.1	10.8	1.6× cheaper
Typical, big three	9.3	-1.2	-14.0	10.7	0.7	1.1	practically flat
Median instead of mean	0.0	0.0	28.9	0.0	0.0	5.8	1.3× cheaper
Lattice scaled	60.8	35.2	67.8	58.0	50.4	54.4	23.2× cheaper
Lattice logit, fixed	41.0	36.2	54.0	43.1	40.9	43.0	9.5× cheaper
Lattice logit, scaled	53.7	34.2	63.8	49.0	44.5	49.0	14.8× cheaper
Horizon one year	42.1	41.0	51.6	39.2	47.2	44.2	10.3× cheaper
Full runs only	55.6	42.1	51.6	58.7	46.0	50.8	17.1× cheaper
Release dates corrected	44.5	45.0	49.6	45.1	48.2	46.5	12.2× cheaper
Lifetime 24 months	47.0	43.0	53.1	46.9	43.9	46.8	12.5× cheaper
Lifetime 12 months	46.5	42.7	53.1	46.6	38.7	45.5	11.4× cheaper

Negative values: more expensive. Mean and annual factor from the unrounded values. Source: precompute.py. "Practically flat": €100 becomes €95.7, within ±5% per year, which the calculator reports as practically flat.

Harmless are dating, horizon, lifetime and the simulated runs. They change little, some even make the number bigger.

Decisive are: which price, which providers, which average, and (5.4) which tasks.

5.2 The combination grid

Across all 512 calculations on Epoch's five benchmarks, the result ranges from **26.4 times cheaper** to **1.3 times more expensive per year**. 53 calculations show rising costs.

What the same data will give you

Each dot is a different, defensible way to calculate, 512 in all. Left: AI gets pricier. Right: cheaper. Result: anything from 1.3× more expensive per year to 26.4× cheaper per year. 53 of them show rising costs. That is not a probability, it is the range.



Figure 1: Each dot is one calculation, logarithmic axis. Left: more expensive, right: cheaper. Marked: Epoch's choice. From the calculator in the article.

With unrounded benchmark rates instead of the calculator's values rounded to 0.1 percentage points, 54 instead of 53 calculations show rising costs. The range stays the same.

5.3 Three customer profiles

Three questions instead of one number: what does someone pay in a year for a bill that is €100 today? All three on Epoch's five benchmarks.

The bargain hunter

Epoch's specification unchanged: $F = 12.52$, bill €8.0. Range from Epoch's four lattices (4.4): €4.3 (scaled) to €10.5 (logit, fixed).

The normal company

Typical price (9) and only the big three (4.3), otherwise as Epoch: $\bar{r} = 1.1\%$ per quarter, $F = 1.05$, bill €95.7, rounded €96. Range from Section 6.1: €86 to €119.

The demanding one

Takes every new model at the strongest setting. Basis: full runs only. For each model m , the run with the highest accuracy, cost c_m^{best} . Per benchmark, a regression across models:

$$\ln c_m^{\text{best}} = \alpha + \beta_b t_m + \varepsilon_m, \quad g_b = e^{\beta_b}, \quad G = \exp\left(\frac{1}{5} \sum_b \beta_b\right) \quad (12)$$

Ordinary least squares. g_b : AIME 4.22, Chess 8.09, FrontierMath 1.66, GPQA 3.92, Mystery 1.88. $G = 3.34$, bill €334. Tokens per task grow on every one of the eight benchmarks with data, by 1.5× to 5.6× per year (steigen.py). Uncertainty: Section 6.2.

Table 8: Customer profiles

Profile	Definition	Factor per year	€100 becomes	Range
The bargain hunter	Epoch: record price, all providers	12.5× cheaper	€8	€4 to €11
The normal company	typical price, big three	practically flat	€96	€86 to €119
The demanding one	strongest setting, trend across models	3.3× pricier	€334	€252 to €458

Three customers, the same data

The bargain hunter

Switches to the cheapest model every quarter, whoever makes it. That is Epoch's customer.

Bill today: €100. In one year:

€8 ↓

Range: €4 ... €11
same performance

The normal company

Buys from OpenAI, Anthropic or Google and uses an ordinary current model.

Bill today: €100. In one year:

€96 →

Range: €86 ... €119
same performance

The demanding one

Takes each new model and turns it up to the highest setting.

Bill today: €100. In one year:

€334 ↑

Range: €252 ... €458
more performance

All three numbers come from the same data. None is wrong. They answer different questions. The headline "13 times cheaper" answers only the first. The second and third are model calculations for typical customers, not figures from anyone's books.

Figure 2: The three customer profiles in the article's calculator.

The ranges show: the exact numbers wobble, the direction doesn't. The bargain hunter pays far less in every calculation, the normal company about the same in every variant, the demanding one more in every calculation. In return, the demanding one gets more performance. Her number is not a price for fixed performance, but the development of her bill.

5.4 Everyday tasks and all benchmarks

Table 9: Other benchmark selections

Selection	Epoch specification	per year	typical, big three	per year
Five core benchmarks	46.8%	12.5× cheaper	1.1%	practically flat
Everyday work: SWE-bench Verified (27.5%), SimpleQA (40.5%)	34.0%	5.3× cheaper	-1.0%	1.04× pricier
All eleven benchmarks	44.4%	10.5× cheaper	0.5%	practically flat
FrontierMath 1 to 3 only	53.1%	20.7× cheaper	-14.0%	1.7× pricier

Decline in % per quarter. Source: precompute.py.

5.5 Performance levels

Epoch averages across all levels, from "barely a task right" to the top score. We grouped the nodes from (4) by absolute level and formed (6) per group (niveaus.py), then averaged over benchmarks. For the customer comparison (niveaus_kunden.py), a benchmark counts in a group only with at least 20 nodes, averaged geometrically over the annual factors.

Table 10: Decline by performance level (guess-adjusted)

Level	Epoch, per year	Share of nodes (AIME / Chess / FM / GPQA / Mystery)	Bargain hunter	Normal company
below 25%	9.3×	32 / 67 / 55 / 43 / 100%	9.6×	1.0×
25 to 50%	11.2×	28 / 29 / 34 / 27 / 0%	11.3×	1.2×
50 to 75%	13.1×	27 / 4 / 10 / 23 / 0%	20.2×	1.6×
75% and up	30.3×	13 / 0 / 2 / 7 / 0%	42.2×	1.2×

Factor: how many times cheaper per year. "Epoch" column: arithmetic mean of the quarterly rates over the benchmarks with nodes in the band. Customer columns: geometric mean over benchmarks with at least 20 nodes (75% and up: three and two benchmarks).

That speaks for Epoch: at the cheapest edge, high performance actually got cheaper faster. The suspicion that the 13 comes from trivial levels does not hold. But for the normal company, high performance from 75% up gets a mere 1.2 times cheaper per year. The higher the performance, the bigger the gap between headline and bill. Caveats: high levels were only reached recently and are thinly covered. So what gets measured is mostly the phase in which a new top score loses its premium. AIME has hit the ceiling (100%). On the Mystery benchmark, the one nobody can train for, the best models reach at most about 30% (raw 33%, guess-adjusted 26%).

5.6 Concentration and timing

On four of five benchmarks, the record price moves at fewer than half of the nodes (Table 4). The median node is 0, and the mean is made of a few large jumps. Attribution (deep.py): for each node with $\delta < 0$, the decline is credited to the new record holder. Influence: leave out the eight most important models per benchmark one at a time and recompute.

- **Mystery:** 37 models in the dataset, 9 carry the decline, the top five 96%.
- **Chess:** the top three 58%. Without Gemini 3.5 Flash the rate falls from 43.0 to 35.1% per quarter, without GPT-4o it rises to 50.6%.
- **FrontierMath:** without GPT-5-mini 58.2%, without GPT-5.6 Luna 49.9%.

Table 11: Decline by calendar half-year of the node (mean of the five benchmarks)

Half-year	H1 2023	H2 2023	H1 2024	H2 2024	H1 2025	H2 2025	H1 2026
times cheaper per year	5.9×	7.4×	132×	28×	8.0×	3.6×	12.0×

Half-year by calendar date of the starting point. An average over phases that are a factor of 40 apart.

IN PLAIN WORDS

A millionaire walks into a bar. The average income in the room shoots up. Nobody at the bar earns a cent more. The price is not a train running at a steady speed. It is an elevator that mostly stands still and sometimes drops five floors at once.

SECTION 6

Robustness

6.1 The normal company: window length and single models

Two checks (robust.py). First, the window length in (9): 6, 12, 18 or 24 months instead of 12. Second, leave-one-out: leave out each of the 75 big-three models on the five benchmarks once and recompute the rate.

Table 12: The normal company's bill in one year (€100 today)

Check	Result
Window 12, 18 or 24 months	€93 to €96
Window 6 months	€119
Leave-one-out over 75 models	€86 to €109
Reported range (minimum and maximum of all checks)	€86 to €119

The statement "practically no decline" holds. Limit: this is a model result, not a number from anyone's actual books (Section 11).

6.2 The demanding one: bootstrap

For (12): per benchmark, resample the models with replacement 1,000 times, re-estimate the regression, recompute G (random seed 1). Draws with fewer than two distinct dates are dropped. The 5th and 95th percentiles give the 90% interval:

2.52 to 4.58, so €252 to €458. That it gets more expensive is therefore certain.

For "always only the strongest model" (only models that set a new top score at release) there is too little data: 2 to 11 models per benchmark, bootstrap 0.5 to 2.9. That is why the profile is defined as "every new model at the highest setting," not as "always the best."

6.3 The bargain hunter

The range comes from Epoch's own four lattices (4.4): €4.3 to €10.5. It is not statistical uncertainty. It shows how much Epoch's own design decisions move the number.

6.4 Insensitive assumptions

Table 13: Further single checks (scenarios.py)

Check	Mean % per quarter	per year
Epoch	46.8	12.5×
No guess adjustment	46.2	11.9×
Without prices from GPU rental	46.8	12.5×
Only release dates of o3, o1, QwQ corrected	46.6	12.3×
Horizon half a year	45.4	11.3×
Horizon one and a half years	44.7	10.7×
Without the GPT-5.6 family	46.0	11.7×

6.5 Why there are no error bars for the headline number

Real confidence intervals for Epoch's headline number don't exist, not at Epoch either. The study explains why: a record price is an extreme value set by a single lucky cheap run, and the usual bootstrap does not work for it. Epoch writes: "We abandon the formal paradigm of statistical inference." We therefore report ranges across specifications, not intervals.

SECTION 7

Decomposition: token count and token price

At the record price: cost per task = tokens per task × price per token. For each node we compare the record holder now and one quarter later and split the log change (zerlegung.py; only runs with a token count above 0):

$$\ln c = \ln n + \ln p \Rightarrow \delta_{jk} = \delta_{jk}^n + \delta_{jk}^p, \quad s_b^p = \frac{\bar{\delta}^p}{\bar{\delta}} \quad (13)$$

n : tokens per task, p : price per token (cost divided by tokens). s_b^p : share of the decline that came through the token price, averaged over the five benchmarks.

Table 14: Decomposition by benchmark

Benchmark	Share token price	Share token count	fewer tokens only, per year
AIME	83%	17%	1.56×
Chess	40%	60%	3.83×
FrontierMath 1 to 3	105%	−5%	0.85×
GPQA	89%	11%	1.32×
Mystery	60%	40%	2.53×
Five core benchmarks	75%	25%	1.76×
SWE-bench Verified	117%	−17%	0.80×
SimpleQA	14%	86%	5.97×

Negative token-count share: the new record holders need more tokens than the old ones (coding, research mathematics). Last column: $\exp\left(-4\bar{\delta}^n\right)$, the "five" row as a geometric mean.

Token price frozen: with only the count effect left, the bargain hunter gets not 12.5 times but 1.76 times cheaper per year. €100 turns into €57 instead of €8. Limit: "cheaper tokens" also includes switching to smaller, more efficient models, not just power and chip prices.

SECTION 8

Scenario calculator

The calculator in the article (component ThoughtPriceLab) projects the three profiles under changed outside conditions. This is explicitly not a measurement. Converting input prices into a bill uses rough cost shares of an AI data center: chips and servers 55%, power 15%. There is no clean public number for this.

$$W = \left(1 + 0.55 \cdot \chi + 0.15 \cdot \varepsilon\right) \cdot \left(1 + \sigma\right), \quad \text{bill}_p = \text{€}100 \cdot m_p \cdot W \quad (14)$$

χ : change in chip and memory prices, ε : change in the power price, σ : price markup when investors' subsidies end, each as a fraction. m_p : the profile's annual multiplier, 1 / 12.52, 1 / 1.05 and 3.34. With the token price frozen, 1 / 1.76 replaces the bargain hunter's multiplier (Section 7). The ranges are multiplied by the same W .

Table 15: Preset scenarios, bill in one year (€100 today)

Scenario	W	Bargain hunter	Normal company	Demanding one
Everything stays as it is	1.000	€8	€96	€334
Power 50% more expensive ($\varepsilon = 0.5$)	1.075	€9	€103	€359
Chips 40% more expensive ($\chi = 0.4$)	1.220	€10	€117	€408
Subsidies end, 50% markup ($\sigma = 0.5$)	1.500	€12	€144	€501
Token prices stop falling (measured)	1.000	€57	€96	€334
All of it at once	1.943	€110	€186	€649

The presets are oriented on 2025 and 2026 developments: wholesale power in the largest US power market, PJM, up 76% from early 2025 to early 2026 (Monitoring Analytics); SemiAnalysis index for H100 annual contracts up 38% from October 2025 to March 2026; memory chips in early 2026 almost twice as expensive as a quarter earlier (TrendForce). According to the Wall Street Journal, OpenAI had \$6.7 billion in revenue in the second quarter of 2026 and sank deeper into the red. According to the Financial Times, Anthropic reached an adjusted operating profit in the third quarter of 2026. All via secondary sources.

CONTEXT

Is "All of it at once" likely? Maybe not. Impossible? No. And "The direction is clear" claims exactly that it is impossible.

SECTION 9

External series

Whoever extends a past price trend assumes that conditions stay the same. The following series show how those conditions actually moved. For each series: source, processing, result.

9.1 US electricity after 1973

Source: Epoch's workbook, sheet "Electricity." Epoch's chart uses a combined real series (Hirsh to 1922, EEI 1926 to 1973) that ends in 1973. The same sheet has Hirsh's nominal prices to 2025, and the "CPI" sheet has the consumer price index (CPIAUCSL, 1982 to 1984 = 100). **Processing** (strom_nach_1973.py): real price = Hirsh real where available, otherwise nominal price / CPI annual mean $\times 100$. From 1974 this series is chained so that it meets Epoch's series exactly in 1973. Index 1892 = 100.

Table 16: US household electricity, inflation-adjusted

Period	Change
1892 to 1973 (Epoch's curve)	−98%
1973 to 1984 (oil crisis)	+29%
1973 to 2002 (low point)	−12%
2002 to 2025	+14%
1973 to 2025	±0%

9.2 United Kingdom

Source: Fouquet (2011), real electricity price in 2000 pounds per tonne of oil equivalent, 1883 to 2008, from 2009 Eurostat (households, real, Section 9.3) chained at 2008 (epoch_chart.py). **Result:** twenty times cheaper from 1900 to 1973, 30% more expensive again by 1980, as expensive in real terms in 2008 as in 1973, 15% more expensive in 2019 than in 1973.

9.3 Western Europe

Source: Eurostat nrg_pc_204, households with 2,500 to 4,999 kWh annual consumption (band DC), all taxes and levies, €/kWh; prc_hicp_aind, HICP 2015 = 100. **Processing** (eurostat.py, epoch_chart.py): annual value = mean of the two half-years, only years with both half-years, real = nominal / HICP $\times 100$. First full year compared with 2025.

Table 17: Household electricity since 2007/08

Country	from	nominal	real	real, peak
France	2008	$\times 2.17$	+61%	+79% (2024)
Spain	2008	$\times 1.81$	+32%	+75% (2022)
Germany	2007	$\times 1.87$	+26%	+39% (2023)
EU-27	2007	$\times 1.72$	+13%	+18% (2023)
Italy	2008	$\times 1.47$	+7%	+26% (2022)
Netherlands	2007	$\times 1.41$	−8%	+20% (2023)

Household electricity in Western Europe since 2007/08

Bars: inflation-adjusted to 2025. Beside them: what the bill says (nominal) and the peak of the energy crisis.

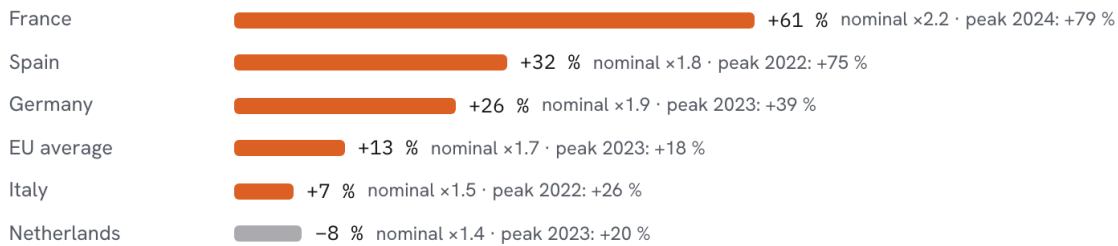


Figure 3: Household electricity in Western Europe, inflation-adjusted to 2025, beside it nominal and the peak of the energy crisis. Eurostat, own calculation.

9.4 to 9.7 The last five years

Series from October 2021, prepared with welt_reihen.py. Change over twelve months: latest value against the latest value at least one year earlier (one week of slack for weekly series).

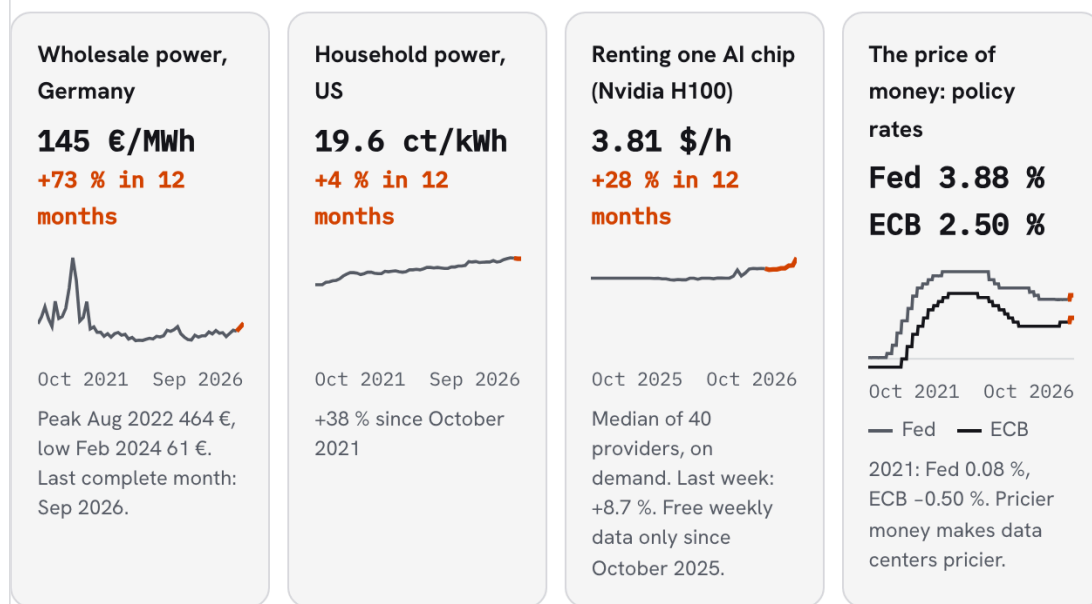
Table 18: Outside conditions

Series	Source and processing	Latest	12 months
9.4 Wholesale power, Germany	Ember, day-ahead, monthly. Complete months only: a month the daily data has not finished yet is dropped. Plus the last 56 days, daily.	€145.4/MWh (09/2026)	+72.5%
9.5 Household power, US	FRED APU000072610 (BLS), \$/kWh, monthly	\$0.196/kWh (09/2026)	+4.3%; since 10/2021 +38%
9.6 Renting an AI chip	getdeploying.com, Nvidia H100, on demand, median across 40 providers, weekly. Free weekly data only from October 2025.	\$3.81/h (Oct 19, 2026)	+27.5%
9.7 Policy rates	FRED DFF (Fed), ECBDFR (ECB deposit rate), weekly, last value	Fed 3.88%, ECB 2.50%	10/2021: 0.08% and -0.50%

Wholesale Germany: peak 08/2022 at €463.6/MWh, low 02/2024 at €61.2/MWh, last daily value October 7, 2026. H100: +8.7% in the last week. Policy rates above the 2021 level but below the 2023 peak (Fed 5.33%, ECB 4.00%).

What the world outside actually did

Real prices over the last five years, as far as they are freely available. Orange: the last few weeks. Measure the dials above against them.



Sources: Ember (day-ahead wholesale), FRED/BLS (household power), getdeploying.com (GPU prices), FRED (policy rates).

Figure 4: The four series in the article's calculator. Orange: the last few weeks. Axes start at zero.

SECTION 10

Technology comparison

10.1 The truncated curves

Epoch's Figure 1 shows AI next to electricity, lithium batteries, compute and DNA sequencing, each from the start of its price decline. We rebuilt it from the "Figure data" sheet and drew it to the end with the series from Section 9. In our rebuild, the axis extends above the starting price so that a rising line becomes visible at all. Epoch's axis spans twelve orders of magnitude and ends at the starting price at the top. On a scale like that, a 30% rise is a few pixels high.

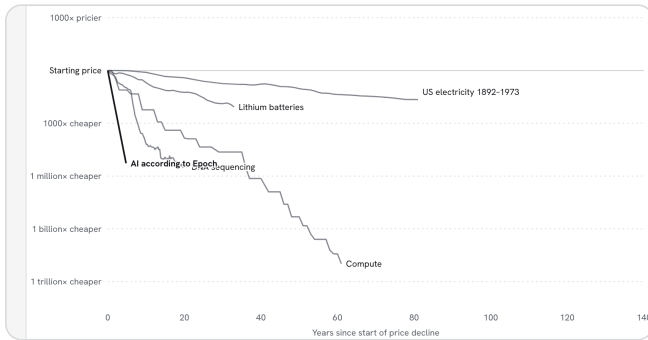


Figure 5a: Epoch's figure, rebuilt from Epoch's data.

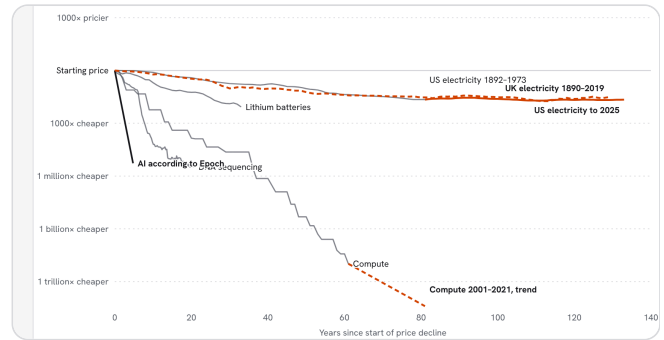


Figure 5b: Orange: what came after.

- **Electricity** ends in 1973. The decline had already stalled for three years, and that fall the oil crisis hit. Continuation: Sections 9.1 and 9.2.
- **Compute** ends in 2001, according to Epoch "merely for lack of newer data." For the years after, we use Epoch's own GPU study (Hobbhahn and Besiroglu 2022): twice the compute per dollar about every 2.5 years (2006 to 2021), a factor of $2^{1/2.5} \approx 1.32$ per year, about 24% cheaper. Drawn as a trend, not as a measured series.
- **DNA** (to 2022) and **batteries** (to 2024) end with the latest available data.
- The steep **AI** line starts in 2021. It is measured only from 2023, before that it is back-calculated at 47% per quarter.

10.2 A fair time window

Three years of AI are compared with 20 to 80 years of other technologies. Many technologies get cheaper especially fast in an early phase. The fair comparison is with the fastest three-year window of each series. Epoch's workbook has a column "Fastest 3-year decline" for this, which does not appear in the report. Our calculation (techwin.py), with log rates per quarter:

$$\rho_{\text{whole}} = -\frac{\ln p_{\text{end}} - \ln p_{\text{start}}}{4(t_{\text{end}} - t_{\text{start}})}, \quad \rho_{3y} = \max_i -\frac{\ln p_{j(i)} - \ln p_i}{4(t_{j(i)} - t_i)} \quad (15)$$

$j(i)$: first observation at least three years after i (at most four). Annual factor $e^{4\rho}$. "AI faster" = ρ_{AI} / ρ with $\rho_{\text{AI}} = -\ln(1 - 0.47) = 0.635$.

Table 19: Pace of the technologies, whole period and fastest three years

Technology	whole period	fastest 3 years	AI faster, report	AI faster, fair
DNA sequencing	1.84× per year	6.70×	4.2×	1.3×
Compute	1.52×	3.17×	6.0×	2.2×
Lithium batteries	1.15×	1.37×	17.9×	8.0×
Electricity	1.05×	1.15×	50.9×	18.3×

With the typical instead of the cheapest model (1.6 times per year), AI sits at about the long-run pace of compute (1.52 times).

10.3 Same data, different picture

Pace per decade (epoch_chart.py): average annual change within each calendar decade, $100 \cdot \left((v_z / v_a)^{1/(z-a)} - 1 \right)$, with log-linear interpolation at the decade boundaries and only for decades a series covers for at least five years. The compute trend after 2001 is left out because it is not a measured series.

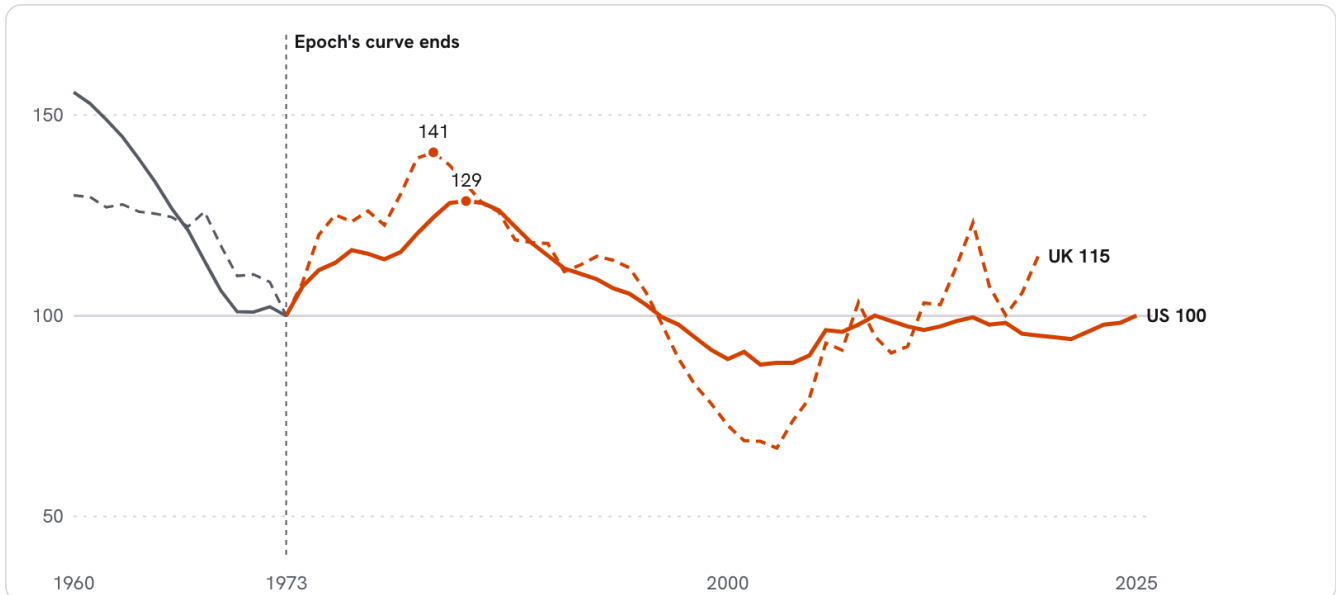


Figure 6: Electricity from 1960 on an ordinary scale, 1973 = 100. Gray: what Epoch shows. Orange: what came after. Solid US, dashed UK.

	1890	1900	1910	1920	1930	1940	1950	1960	1970	1980	1990	2000	2010	2020
US power	-1.9	-7.0	-6.9	-3.1	-2.7	-8.0	-3.5	-4.2	+1.4	-0.4	-2.2	+1.0	-0.4	+1.1
UK power	-5.1	-5.4	-12	-0.5	-4.4	-3.5	-1.7	-1.7	+1.7	-1.6	-4.1	+2.2	+2.7	
Compute						-40	-31	-17	-29	-40	-38			
Batteries											-13	-15	-14	
DNA												-61	-37	
AI according to Epoch														-92
AI, normal company														-4.3
AI, turned all the way up														+234

Columns: decades (1890 = the 1890s). Percent per year, averaged over the decade. Orange: got pricier. AI in the 2020s column: 2021 to 2026, computed as Epoch does.

Figure 7: Pace per decade in % per year, cheaper (-) or pricier (+). Orange background: got pricier. AI in the 2020s column: 2021 to 2026, computed as Epoch does.

Each of these technologies had fast phases, slow phases and setbacks. The AI number describes exactly one phase, three years long. Which picture goes in a report is a choice, not a measurement.

10.4 Market price versus minimum

For electricity and batteries, what counts is market prices, what customers paid on average. For AI, it is the minimum across all providers. Those are exactly the "apples and oranges" Epoch is talking about.

ASIDE: TRAIN AND PLANE

Deutsche Bahn, Germany's national railway, advertises ICE high-speed tickets from €8.99. The offer is real, but it is limited, and hardly anyone pays that price. Statistically, a price like that is an outlier: a small allotment at the edge. If you want to know what train travel costs, you sort it out. The study instead measures only the edge. With the December 2024 timetable change, Deutsche Bahn raised its flexible full fare by 5.9% on average, while the Super Saver fares stayed stable. As the reason, it cited significantly higher costs, especially from the latest wage agreements, and two years earlier, energy costs. In December 2025 it skipped a long-distance price increase. The pattern holds: the teaser offer stays cheap, the price most people pay goes up.

A domestic flight in the US has gotten 37% cheaper in thirty years after inflation, on average from \$618 to \$387 (BTS, 2025 prices), a little over one and a half percent per year. If you tracked the cheapest promotional fare instead, you would get a spectacular number. Back in the early 2000s, according to American Express, a typical business fare already cost five to six times a leisure fare. By the method "promotional price against average price," the flight would also be the fastest technology of all time.

Limitations of this analysis

11.1 Fixed performance

All numbers in this report hold performance fixed, Epoch's and our customer profiles (except the demanding one). For a price comparison, that is correct. It becomes a problem when it turns into a forecast of what AI will cost a company. In 2023 a usable text was impressive, today many people expect entire workflows.

What you measure	What happens
Price for constant performance	falls sharply
Expectations of the technology	mostly rise
Actual spending	can rise or fall

KEY SENTENCE

We measure the price drop against yesterday's requirements and sell it as tomorrow's savings.

IN PLAIN WORDS

A stagecoach managed about 10 kilometers per hour. The ICE, Germany's high-speed train, runs at up to 300. A coach ride would be dirt cheap today. Only nobody takes one anymore. The coach didn't get more expensive. Faster travel became possible, and because it became possible, everyone used it. What becomes possible gets used. And paid for.

11.2 Further limitations

- **Model results, not bookkeeping.** The normal company is a calculation on Epoch's data. A real company uses specific models for specific tasks. How often each model actually gets used is in no dataset. There are no usage weights. The typical price is a second reading, not a better truth.
- **Model selection.** "Typical price" depends on which models Epoch tested.
- **List prices, nominal.** Discounts, batch and cache prices, contracts and inflation are not included. Some prices are not from the release date (2.3).
- **Benchmarks are not work.** The measuring sticks are mathematics, chess and exam questions. The two tests closest to everyday work show the slowest decline (5.4).
- **Thin data at the edges.** High levels and record holders are thinly covered (5.5, 6.2).
- **Decomposition.** The 75 to 25 split varies strongly by benchmark (Table 14).
- **Scenarios.** Cost shares and presets in Section 8 are assumptions. Some company figures come from secondary sources.
- **Combination grid.** The 512 calculations are not a probability distribution (4.9).
- **Epoch's smoothed models.** The claim "66% decline per quarter right after a new top score" comes from heatmaps (Epoch's Figure 6). There, a large part of the surface lies above the record staircase, without measurements: AIME 49%, Chess 57%, FrontierMath 63%, GPQA 40% (heatarea.py). There the model only extrapolates. Epoch's own Table 1 also shows: a model fitted to all models instead of only the cheapest edge gives 5.3 to 10.2% per quarter (1.2 to 1.5 times per year), with an enforced envelope 27 to 38%. Epoch rejects both variants, with reasons. It still shows: a formula is not a measurement. It is one more assumption.

11.3 What the data does not decide

Whether you should start with AI early. That can be right, because you learn which process fits, where the data lives and who checks the result. That is a good argument. It just doesn't need the 13. Propped up by a number that measures something else, it gets weaker, not stronger. The same data argues just as well for waiting: according to Epoch, new top performance initially gets about two thirds cheaper per quarter. And for providers, Epoch calls the profits per model "fleeting." A market in which performance loses value this fast is a risk for investors, not a safe bet.

11.4 Other work

- **Toby Ord** (Oxford, October 2025) shows that progress comes mainly from computing more when answering: "So if it costs a factor of 10 or 100 times as much to answer each question, this really does affect their bottom line. And unlike training costs, these costs can't be made up in volume." Epoch thanks him in the report for his objection to the technology comparison and answered him publicly. Redwood Research considers his method too pessimistic.
- **Gartner** (August 17, 2026): the cost per agentic workflow will rise more than fivefold through 2028, even though tokens get cheaper ("Inference Paradox").
- **Gundlach et al.** (March 2026): a similar method, 5 to 10 times cheaper per year instead of 13. Epoch cites the work itself.
- **Pasquale Pillitteri** (blog, in essence): the caveats get lost when the study is shared. In a company's books there is no such thing as fixed performance, so plan with cost per task.

CONCLUSION

None of us knows how AI prices will develop over the next few years. We don't either. At the cheapest edge the price falls fast, and almost all calculations point down. But the data does not support a clear conclusion of "it's getting dramatically cheaper." What it supports: it depends, dramatically.

SECTION 12

Reproducibility

Every number can be reproduced. The scripts are in the project folder under `drafts/preis-des-denkens/analyse/` (script names are German). Requires Python with `uv`. Each call is `uv run --with pandas --with numpy python -I <script>`, plus `--with openpyxl` for Excel inputs. Paths below are relative to `analyse/`, with Epoch's repository next to it as `epoch-repo`.

```
git clone https://github.com/droodman/inference-cost epoch-repo # commit 9163c17
cd drafts/preis-des-denkens/analyse
D=../../epoch-repo/data
uv run --with pandas --with numpy python -I core.py $D # Table 4
uv run --with pandas --with numpy python -I precompute.py $D lab.json # Tables 7, 9; 512 calculations
uv run --with pandas --with numpy python -I scenarios.py $D # Table 13, scenarios.csv
uv run --with pandas --with numpy python -I mv2.py $D # older 64-combination grid, multiverse.csv
uv run --with pandas --with numpy python -I deep.py $D # 5.6 concentration, half-years
uv run --with pandas --with numpy python -I niveaus.py $D # Table 10, Epoch column
uv run --with pandas --with numpy python -I niveaus_kunden.py $D # Table 10, customer columns
uv run --with pandas --with numpy python -I robust.py $D # Table 12, 6.2
uv run --with pandas --with numpy python -I steigen.py $D # (12), steigen.json
uv run --with pandas --with numpy python -I zerlegung.py $D # Table 14, zerlegung.json
uv run --with pandas --with numpy python -I heatarea.py $D # 11.2 heatmap area
uv run --with pandas --with numpy python -I dates.py $D # Table 6, date check
X="$D/Cost declines by tech.xlsx"
uv run --with pandas --with numpy --with openpyxl python -I techwin.py "$X" # Table 19
uv run --with pandas --with numpy --with openpyxl python -I strom_nach_1973.py "$X" strom.json # Table 16
uv run --with pandas --with numpy python -I eurostat.py <nrg_pc_204.json> <prc_hicp_aind.json> eurostat_real.json
uv run --with pandas --with numpy --with openpyxl python -I epoch_chart.py "$X" epoch_chart.json # Fig. 5 to 7, Table 17
uv run --with pandas --with numpy python -I welt_reihen.py <ember_monthly> <ember_daily> <gd_weekly> <fred_dir> welt.json
uv run --with pandas --with numpy python -I export_ts.py lab.json steigen.json zerlegung.json strom.json epoch_chart.json
welt.json data.ts
```

Table 20: Scripts, inputs, outputs

Script	What it does	Output
<code>core.py</code>	Epoch's preparation (build_runs) and model-free rate (decline), Sections 2 and 3	rate, nodes moved, nodes per benchmark
<code>precompute.py</code>	All specifications from Section 4 for all eleven benchmarks, mean and median	<code>lab.json</code>
<code>export_ts.py</code>	Writes the calculator's data module	<code>src/components/thought-price/data.ts</code>
<code>scenarios.py</code> , <code>mv2.py</code>	Single checks, older grid of 64 combinations	<code>scenarios.csv</code> , <code>multiverse.csv</code>
<code>deep.py</code>	Attribution of the decline to models, leaving out single models, half-years	console
<code>niveaus.py</code> , <code>niveaus_kunden.py</code>	Decline by performance level, record price versus typical price	console
<code>robust.py</code>	Window length, leave-one-out, bootstrap for the demanding one	console
<code>steigen.py</code>	Cost and token trend of each model's best setting	<code>steigen.json</code>
<code>zerlegung.py</code>	Token count versus token price	<code>zerlegung.json</code>
<code>heatarea.py</code> , <code>dates.py</code>	Heatmap area without data, mismatched release dates	console
<code>techwin.py</code>	Whole period and fastest three-year window	console
<code>strom_nach_1973.py</code> , <code>eurostat.py</code>	US electricity after 1973, European household electricity, real	<code>strom.json</code> , <code>eurostat_real.json</code>
<code>epoch_chart.py</code>	Combined chart, UK, Europe, pace per decade	<code>epoch_chart.json</code>
<code>welt_reihen.py</code>	Series for the last five years	<code>welt.json</code>

Note: `scenarios.py` uses an older, broader name pattern for "big three" (including `gpt-oss` and `chatgpt`) and gets 9.0× instead of 8.6× there. The pattern in `precompute.py` (4.3) is the one that counts for the report, the article and the calculator. Inputs for `eurostat.py` and `welt_reihen.py` are public downloads (Eurostat API, Ember, `getdeploying.com`, FRED) and not part of the repository.

APPENDIX A

Glossary

Benchmark

A test for AI models: a fixed set of tasks. What gets measured is the percentage a model solves correctly.

Token

A word fragment. Providers bill by tokens, roughly like a taxi bills by the mile. If a model thinks longer, it uses more tokens.

Record price (cost frontier)

The cheapest price at which any model reaches a given level, formula (2). Like a world record, it can only get better.

Top score (state of the art)

The highest accuracy reached up to a given date, formula (3).

Node

A point of the measurement grid, made of a performance level and a date, at which the change in the record price is measured.

Geometric mean

The average of the percentage changes, computed via logarithms. Strongly affected by a few large jumps.

Median

The value in the middle when you sort everything by size. A millionaire in the bar doesn't change it.

Logit

A transformation of shares that spreads out values near 0 and near 100%.

Logarithmic scale

An axis where every step is ten times the previous one. Good for huge differences, bad for seeing small rises.

Nominal and real

Nominal: what the bill says. Real: what it costs once you take out general inflation.

Bootstrap

You draw many new samples from the data with replacement, recompute each time and look at how much the result varies.

Leave-one-out

You leave out one element at a time, here one model, and look at how much the result changes.

Open weights

Models you can run yourself. Often cheap, but for many companies not a real option because of operations, data protection or contracts.

APPENDIX B

Six questions for any price statistic

1. **Whose price?** The cheapest on the whole market, a typical one, or yours?
2. **For which performance?** Measured against what, and does it have anything to do with your work?
3. **Average or normal case?** Or is there a millionaire in the bar?
4. **Could the number even come out differently?** A world record that can only fall doesn't prove that anything is falling.
5. **Compared with what?** Sprinter against sprinter, or sprinter against marathon runner? And on which scale?
6. **What has to stay the same for this to continue?** Power, chips, investor money. And who guarantees that?

Sources

- Epoch AI (2026): The plunging price of thought. epoch.ai/publications/the-plunging-price-of-thought
- Epoch AI: Data and code. github.com/droodman/inference-cost, commit 9163c17 (September 24, 2026)
- Hobbhahn, M.; Besiroglu, T. (2022): Trends in GPU price-performance. Epoch AI. epoch.ai/blog/trends-in-gpu-price-performance
- Fouquet, R. (2011): Divergences in long run trends in the prices of energy and energy services. Review of Environmental Economics and Policy 5(2). Dataset: LSE Grantham Institute.
- Eurostat: [nrg_pc_204](https://ec.europa.eu/eurostat/tgm/table.do?tab=table&init=1&language=en&code=prc_hicp_aind) (household electricity prices), [prc_hicp_aind](https://ec.europa.eu/eurostat/tgm/table.do?tab=table&init=1&language=en&code=prc_hicp_aind) (HICP).
- Ember: European wholesale electricity price data (monthly, daily). FRED: [APU000072610](https://fred.stlouisfed.org/series/APU000072610), DFF, ECBDFR. getdeploying.com: GPU price dataset (weekly).
- Bureau of Transportation Statistics (2026): 2025 Annual Average Domestic Air Fare. bts.gov. Deutsche Bahn, press release "ICE-Fahren jetzt ab 8,99 Euro" (German); bahn.de/bahnbusiness (flexible full fare +5.9% from December 15, 2024) (German); Nordkurier, Berliner Zeitung, eisenbahn.de (reasons, no increase in 2025) (German). American Express Business Travel Monitor (2001/2002), via Business Travel News and Travelmole.
- Ord, T. (2025): Evidence that Recent AI Gains are Mostly from Inference-Scaling. tobyord.com. Epoch AI (2026): How persistent is the inference cost burden? Redwood Research: AI's capability improvements haven't come from it getting less affordable. Gartner (August 17, 2026): Gartner Predicts AI Inference Costs Per Agentic Workflow Will Increase More Than Fivefold Through 2028. Gundlach et al. (2026). Pillitteri, P. (2026), pasqualepillitteri.it.
- Monitoring Analytics via SOFX (PJM wholesale prices), SemiAnalysis index via uvik.net (GPU rental prices), TrendForce via globx.eu (memory prices), Wall Street Journal via Luminix (OpenAI), Financial Times via ValueAdd VC (Anthropic).
- Companion article: <https://www.hoettler.com/en/blog/price-of-thought>

Jonas Höttler, Balane · hoettler.com · Questions and objections welcome. If you find a calculation error, please let me know. That is exactly why everything is out in the open.