Episode Book

EPISODE #6134

Cheap Tokens, Expensive Scripts

PUBLISHED	19 October 2026 at 14:51 UTC
DURATION	21:47
LISTEN	myweirdprompts.com/episode/llm-cost-per-task-evaluation-18346/
HOSTS	Corn, Herman, Larry, Hilbert
SCRIPT MODEL	anthropic/claude-sonnet-5.5
VOICES	chatterbox-regular (chatterbox-local)
PIPELINE	V5.10
BOOK GENERATED	20 October 2026 at 07:46 UTC

SYNOPSIS

Picking an LLM on price per token is the easy mistake. This episode walks through a real evaluation for the show's own script-writing pipeline — a 26,000-token input with a 9,000-token cached prefix and 6,500 tokens out — and shows how a model with a worse input price can win the bill by a sixth, then lose once caching is off or thinking tokens triple the output. We cover why blended leaderboard pricing misleads, why hidden reasoning tokens are billed as the expensive kind, how OpenRouter host pinning and quantization quietly change your quality score, and why you should estimate to screen but measure to decide.

TRANSCRIPT



Corn

A model turns up on the list at a fraction of the price per token of the one we use. Somebody runs a single script through it, looks at the bill afterwards, and the cheap one cost more.



Herman

Nobody had to do anything wrong for that to happen. Nobody lied about the price, either. That's the unsettling part.



Corn

Daniel wants to know what you do once the hard part of an evaluation is over. You've found two or three models that write well enough, and now it's about value for money rather than raw quality, without list prices steering you. It's a follow-up to the run we've been doing on how the show picks its writers, so the baseline and the budget are already set. We're working from our real call, about twenty-six thousand tokens in and six and a half thousand out.



Herman

Nine thousand of that input is a byte-identical prefix. Same bytes every call, which is the part a provider can cache. A third of the input, give or take. I'd bet that one detail ends up mattering more than any of the headline prices.



Corn

The budget is about six cents a script at that shape. That puts an episode around twenty-two cents, against sixteen or so now.



Herman

Six cents is the line for getting into the sweep. The first pass covered four hundred and sixty-nine models on OpenRouter, and a hundred and seventy-seven came in under it. Adoption is stricter. At most about three times the baseline's cost per script, and the baseline, DeepSeek V four point one Flash, costs well under a cent. Seven tenths of one, roughly.



Corn

Cheap is also where the junk lives.



Herman

Which is why cost comes after quality, never before. A cheaper model only wins at equal quality. If you rank by quality per dollar, a model that writes garbage for nearly nothing floats straight to the top, because the bottom of the fraction is almost zero.



Corn

Fine. You've got survivors, they all clear the quality gate, now you compare prices. What does our scan do?



Herman

For each model it looks at every host serving it on OpenRouter, picks the cheapest, and takes three numbers from the endpoint listing. Input price, cache read price, output price. Multiply by our token shape, rank. It's a good screen. It throws out three hundred models you'd never have wanted to look at.



Corn

And what can't it see?



Herman

Three things. How many tokens the model actually emits. Whether the cheapest host is the same quality of host as the others. And whether the cache gets hit at all. A list price is a promise about a unit, not about your bill.



Corn

Show me the flip. A calculation, not a story.



Herman

With invented models, so nobody goes shopping for them. Model A charges twenty cents per million input tokens and two dollars per million output, with no cache discount. Model B charges fifty cents for input, five cents for a cached read, and a dollar for output. On input alone, A is two and a half times cheaper. Which one do you pick?



Corn

A. Two and a half times, it's not close.



Herman

Do the call. A pays twenty cents a million on all twenty-six thousand input tokens, about half a cent. Six and a half thousand output at two dollars a million is one point three cents. Total, one point eight cents.



Corn

And B?



Herman

Seventeen thousand uncached tokens at fifty cents a million, that's under a cent, call it eight and a half tenths. The nine thousand cached ones are almost free. Output at a dollar a million, six and a half tenths. Total about one and a half cents. The model with the worse input price is cheaper.



Corn

By a sixth.



Herman

A bit under a sixth. Look at what happened in the middle, though. Switch the cache off and B goes to about one point nine five, and A wins again. The whole flip lives in one line of the price sheet.



Corn

So the blended price the leaderboards use, three input for one output, would have called it wrong.



Herman

It would have called it for A, which is the lazy default. Corn, if I hadn't made you do the sum, what would you have done?



Corn

Picked A and felt clever. I'd have been wrong for a whole quarter before anyone noticed.

L

Larry

Close your eyes. Breathe in. Ombre de Jeton. The fragrance of the price you never see. Notes of silence, of cached midnight, of reasoning so deep it costs nothing at all, because it happens where no one is looking. Wear it to the review meeting. Wear it to the invoice. Whispered hours of thought, free, completely free, never itemised, never billed, only felt, on the skin, like the memory of a smaller number. Ombre de Jeton. Because what you cannot measure can never hurt your budget. Ombre de Jeton arrives in a bottle shaped like a rounded hush, and it is, in fact, a box of paper coffee filters, two hundred count, unbleached. Ombre de Jeton. Pour, and disappear.



Herman

The claim that thinking is free because nobody sees it is backwards. Hidden reasoning tokens are billed as output tokens, the expensive kind, and they run to thousands per call whether or not they show up in the response.



Corn

Let it go, Herman.



Herman

The filters I'm fine with. It's the reasoning claim.



Corn

Then do the reasoning claim properly. What does thinking do to the sum?



Herman

Take B again, same prices, but this time it thinks before it writes. Say thinking triples the output, so thirteen thousand extra hidden tokens on top of the six and a half you read. Output at a dollar a million comes to nearly two cents on its own. Add the input side and the total is about two point eight. From one and a half to nearly three, same model, same price sheet.



Corn

And that fails adoption.



Herman

Three times a baseline of seven tenths of a cent is about two cents, so yes, out. The list price never moved.



Corn

Is that a real case or a convenient one?



Herman

Convenient numbers, real shape. Artificial Analysis publishes how many tokens each model burned running its whole index. One tiny model, a couple of billion parameters, used about seven times the tokens of a big frontier model, and its bill still came out around a tenth. Cheap per token and cheap per task are two separate facts. The spread in tokens between models on the same tests runs ten to thirty times.



Corn

Everyone keeps saying prices are collapsing. Cheaper every quarter.



Herman

Per token, for a fixed level of ability, yes, and fast. But per task it's a different line. The newest models think longer, so on a hard benchmark one lab's older model used tens of millions of output tokens for a score that a newer one reached with a fraction of that. I've only seen that secondhand, so don't hold me to the exact multiples. The direction is what matters. A falling price per token doesn't mean a falling price per task, and the bill can rise while the rate card shrinks.



Corn

The rate card is the part that gets written up. The model card, I take it, tells me nothing about the bill.



Herman

You can read the unit price. You can't read the units. The scan only has the first one.



Corn

Add a column. Guess a thinking multiplier for each model.



Herman

I'd rather not guess, and Aider is why. Their coding leaderboard published a cost of six dollars thirty-two for Gemini two point five Pro and then retracted it, because the tool counting cost left out reasoning tokens. A leaderboard run by people who care about this got the cost wrong from token counts.



Corn

So much for estimating.



Herman

Hang on. I'm not against estimates. They're how you screen a hundred and seventy-seven models without paying for a single one of them. Estimate to cut the list, measure to decide.



Corn

I'd go further. OpenRouter hands you the charged amount on every response. Why compute what you're being told?



Herman

Because you have to choose before you've paid. For screening there's nothing else. But for the final decision, yes, log the cost field and the tokens, every call. One quirk. Reasoning tokens show up only as a sub-count inside the completion tokens, so the completion number already contains them. Add the reasoning count on top and you double count.



Corn

That's a day-two mistake.



Herman

I made it on day one. You can also look any response up afterwards by its id and get cost, tokens, latency, which host served it and why it stopped. The last one matters more than it sounds. If a reasoning model spends its whole allowance thinking and the output gets cut off, you paid for a script that doesn't exist.



Corn

Full price, zero script. The most expensive run there is.



Herman

Retries do the same thing. ARC Prize had one system at about thirty-nine dollars a task officially and about twenty in a public run, and the paper puts most of the gap on failed calls and retries. Cost per task depends on the harness, not just the model.



Corn

The scan picks the cheapest host. Is that the trap you were hinting at?



Herman

It's the quiet one. A slug isn't one thing. Several providers serve the same open-weight model at different prices, and they don't all run it the same way. Some quantise harder, some skip caching, some are slow. By default OpenRouter load-balances across them weighted toward the cheapest, so unless you pin a host, your finalist could be served by three different ones across your twenty test runs.



Corn

So the quality score belongs to a mix, not a model.



Herman

Pin the provider. Filter on quantisation where it's listed. And cache hits depend on staying with the same host. OpenRouter will keep later requests on the provider that served the first, for about ten minutes idle, and you can set a session id to pin from the first request. Manually ordering providers switches that off.



Corn

Do I get told whether the cache worked?



Herman

In the response, as a cached token count. Above zero, it hit. Zero, you paid full price whatever the page said. And I don't know what our own call spacing does here, I'll say that plainly. Several providers keep a cache for a few minutes only. Whether a script a day ever lands in a window depends on how our calls cluster.



Corn

Is there a cost on the way in, as well as the way out?



Herman

On some hosts, yes. Writing the cache costs extra. Anthropic charges a quarter more than the normal input price to write it, and their reads are a tenth. DeepSeek reads at a tenth and writes at the plain input price, so there's no penalty. The practical point is that the first call of every test is the pricey one. If you average a short run, you'll be charging the finalist for warming up. Report the steady state, and report the cold call separately.



Corn

Then the saving in your example could be nothing.



Herman

It could. So the finalist has to survive the sum with the cache off.



Corn

What about the cost of the eval itself? Twenty prompts across three finalists doesn't sound like much.



Herman

The generation side is trivial. The judge is where the money goes. In our runs the judge is most of the spend, because it reads every script in full, and you want a strong one. So the cheapest way to answer cost questions is to decide which numbers you need before you run, and not rerun the judge every time a price moves. Cost doesn't need the judge at all. Rescore with the logged tokens and the new prices.



Corn

That's the bit people miss. The prices change, the transcripts don't.



Herman

Keep the raw outputs and the raw token counts, and the whole cost side of the sheet can be recomputed in a minute. That's the argument for logging more than you think you'll use.



Corn

Here's what bothers me. Run all this. A finalist costs a bit more than the baseline and the judge scores it higher. Do I care?



Herman

I'd pay. The scale is cents. Going from seven tenths of a cent to two cents a script is rounding on the cost of the audio, and if the writing is better, that's a better show for years.



Corn

A better score from a judge. That's not the same as better. If a listener can't hear it, the premium buys a number. A gap smaller than the judge's own noise is a tie, whatever the table says.



Herman

A tie on one run. A small edge repeated over three hundred episodes is something a listener feels without being able to name.



Corn

Maybe. You're asking me to pay a certain cost for a benefit nobody will measure.



Herman

And you're asking me to leave quality on the table because it's hard to measure. The ceiling is already your answer, three times baseline. Inside it, I'd pay for a difference I could defend and nothing more.



Corn

Defend how?



Herman

Price per point. If the finalist costs a cent and a quarter more per script and leads by more than the noise, you write down what the lead cost, and decide with that number in front of you.



Corn

A tie still goes to the cheaper one. I'll take the number. I won't take the premium.



Herman

Noted. I'll be on the other side at the review.



Corn

Put the procedure down in order. That's what Daniel's after. You start with three finalists.



Herman

Fix the quality gate and the budget before you look at a price. If the gate moves after you've seen the bills, it isn't a gate. Then run every finalist on the same real prompts, a couple of dozen is plenty for one person. Pin the host, pin the quantisation if you can, and set the reasoning effort yourself instead of taking the default, because default effort levels are the easiest way to compare two things that aren't the same.



Corn

Then log.



Herman

Charged cost, completion tokens, cached tokens, finish reason, every failure. Then compute the number that counts, which is cost per accepted script, not per call. Mean cost divided by pass rate. A model that fails one run in five costs a fifth more than its invoice says. And look at the ninetieth percentile as well as the mean, because one runaway thinking run can cost ten normal ones. Before that, a question for you, Corn. Where would you set the gate? Our judge scores scripts, and a finalist has to clear some line before price counts at all. High enough that the cheap models fail, or low enough that three of them pass and cost decides?



Corn

Low enough that I'd happily broadcast the worst script that passes. Not an average. The worst one. If I'd cringe at it going out, the line is too low, and no price makes it worth it. Above that, I don't mind a close call, because a close call is exactly when cost should break the tie.



Herman

That's stricter than mine, and I think it's right. A gate on the average lets a model through that sometimes writes something bad, and the bad ones are the ones that trigger a rerun, which is cost.



Corn

Lovely, a dull price argument turns out to be a quality argument. Show me where that bites, though.



Herman

Back to A and B. A costs one point eight cents and passes ninety-five in a hundred. B costs one and a half and passes eighty. Divide, and both land at about one point nine cents per accepted script. They tie. The cheaper-looking one at every earlier step ended up where the dearer-looking one did.



Corn

Then you're choosing on something that isn't cost.



Herman

Which is the right place to be choosing from. Next, plot quality against cost per accepted script. Anything worse and dearer than something else is dominated, delete it. That's the frontier idea from the paper on agents by Kapoor and colleagues. A leaderboard sorted on accuracy rewards expensive systems, and testing for your own application is a different job from testing a model. We're the application.



Corn

Two or three survive.



Herman

Usually. Then stress the survivors. Rerun the sum with the cache at zero, with the input half again as long and half as long, with peak-hour pricing for a host that doubles at peak, with output at the ninetieth percentile length. If your pick leaves the budget in any of those, you want to know now. Pick the cheapest that clears the gate, write the price per point if you pay a premium, and redo it monthly, because endpoints drift. A host that was cheapest in October can be gone by December.



Corn

Anything after that?



Herman

A cascade, later. Run the cheap model, check it against the gate, escalate to the better one only on a fail. FrugalGPT matched the best single model at a small fraction of the cost. It needs labelled data to tune the check, so it comes after the gate exists. Where I started was the sweep. Four hundred and sixty-nine models, a hundred and seventy-one under six cents, and quality picks the finalists from there.



Hilbert

A hundred and seventy-seven. You said seventy-one.



Herman

A hundred and seventy-seven. I swapped the digits.



Hilbert

Easily done. Herman, you do the sum out loud in tenths of a cent. Nobody has paid in tenths of a cent since I stopped buying screws by the hundredweight. I own four coin meters. One per room, grey, two-euro coins, bolted to the skirting. Every spring the supplier sends a man to fit a smart one, and I let him get as far as the porch. The little screen tells you the rate in colour. I have never been warmed by a colour.



Corn

Four meters is a lot of slots to keep fed.



Hilbert

Five, with the one in the shed. People say you can't see what you spend. You can see it to the cent, it's in the tin. It's a biscuit tin from a coronation that got cancelled. What you can't see is the bathroom going cold at ten past two on the ninth night, and it is always the ninth. Mrs Dunphy next door has a pony, Gerald, grey, eleven hands. The stable roof went in March, so he slept in my hall on Tuesdays and Thursdays. In return she fed the bedroom meter on Fridays, six coins. A heat lamp, two hundred and fifty watts, hung off the picture rail, on a lead through the wall from the bedroom socket. He turned the hall light off with his nose at a quarter past nine. He has a library card. I carry it.



Herman

Then the lamp is the whole load, and the nose is a rounding error. What does a pony borrow?



Hilbert

Atlases.



Corn

Seven hours of lamp a night, on a meter that eats two-euro coins, for a pony that isn't yours. You're the only one who can lose on that.



Hilbert

I'm not. She tips the tin out on my kitchen table every Friday and counts it. Twelve euro, six coins, never off by more than one all winter. She has never once read a dial. The roof was done in October. The shed meter went to her new stable, bolted to the door. I wasn't asked, and Gerald still comes Tuesdays. He prefers the carpet.



Herman

A tin that steady means one week's count would have told her the whole winter. The dial adds nothing.



Corn

Which makes the neighbour the auditor of your electricity, unpaid, on a pony's behalf. You'd want her kept happy.



Hilbert

Gerald brings his own hay. The dryer is mine. Small correction.



Corn

What I'd want before committing a month of scripts to anything is the tin counted, not the dial read. Logged spend for each finalist on real scripts, put next to what the estimate promised, with the gap written down.



Herman

Mine is what happens to that gap over time. If price per token keeps falling, the thinking budgets and output lengths are free to creep up, and whether the bill per script falls with them is something I'd rather see counted than assume. I'd also want to know whether the pick keeps clearing the gate when the hosts underneath it change.



Corn

Thanks to our producer, Hilbert Flumingtop, for keeping the desk running. That rounds off the eval run as far as I'm concerned.



Herman

This has been My Weird Prompts. You can find every episode at my weird prompts dot com.



Corn

See you tomorrow.