

Season 1: How LLMs work

When is a reasoning model worth the cost?

Same setting, same three models: **+21 points** on puzzles, **nothing** on extraction, billed either way. A framework for telling which kind of task you have.

Gemini 3.6 Flash, extraction, reasoning turned up:

16x + 0

the output tokens

correct answers

The same setting added 21 points on puzzles. This episode is about telling those two tasks apart.

What it is

Hidden working-out
the model writes
before it answers

How it bills

As output tokens: the
dearest and the
slowest (S1 E5)

Where it applies

To every call a feature
makes, needed or not

Reasoning is a purchase, made on every call

YOU PAY Every call, every task Tokens Seconds	YOU GET BACK Only on some tasks Accuracy where several steps must be held in mind at once
---	--

Where it breaks

The bill lands whether or not the task needed it. In S1 E10, the "only on some" was literal: no other kind of task gained.

IN S1 E10

Extraction Paid: yes Paid back: no	Short arithmetic Paid: yes Paid back: no	State tracking Paid: yes Paid back: yes	Constraint puzzles Paid: yes Paid back: yes
--	--	--	--

Four kinds of task, from a lookup to a search

30 generated items each, with known answers. Each square is one step to hold.

Extraction

Find one fact in a short log. The control.



```
...The calibration code for
pump K-482 is 7310...
Q: code for pump K-482?
```

Short arithmetic

Two or three operations, in order.



```
Tank holds 1,200 litres.
60 drawn off an hour for 5
hours, then 150 added.
```

State tracking

Follow 12 to 20 changes across five warehouses.



```
4. 50 units moved from
   Brindle to Elgin.
5. 35 units arrive at Colwyn.
```

Constraint puzzles

Order seven jobs: one valid order in 5,040.



```
- Flushing is immediately
  before greasing.
- Audit is not job 3.
```

The pilot was too easy: every model got every item right at its lowest setting. State tracking and the puzzles were made harder, because a task the model already gets right cannot show what reasoning adds.

Before you swipe

Reasoning turned up on all four. Which two gained?

Extraction

?

Short arithmetic

?

State tracking

?

Constraint puzzles

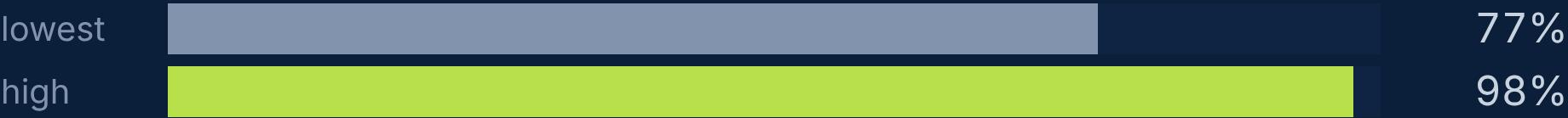
?

Pick two, and guess the gain in points. **The next slide has the answer.**

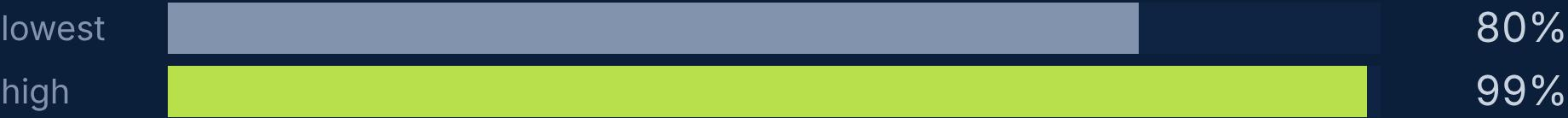
Two tasks gained about 20 points. Two gained nothing.

Accuracy, all three models pooled: 90 answers per task, same items in both modes.

Constraint puzzles +21.1 pts



State tracking +18.9 pts



Extraction +0.0 pts



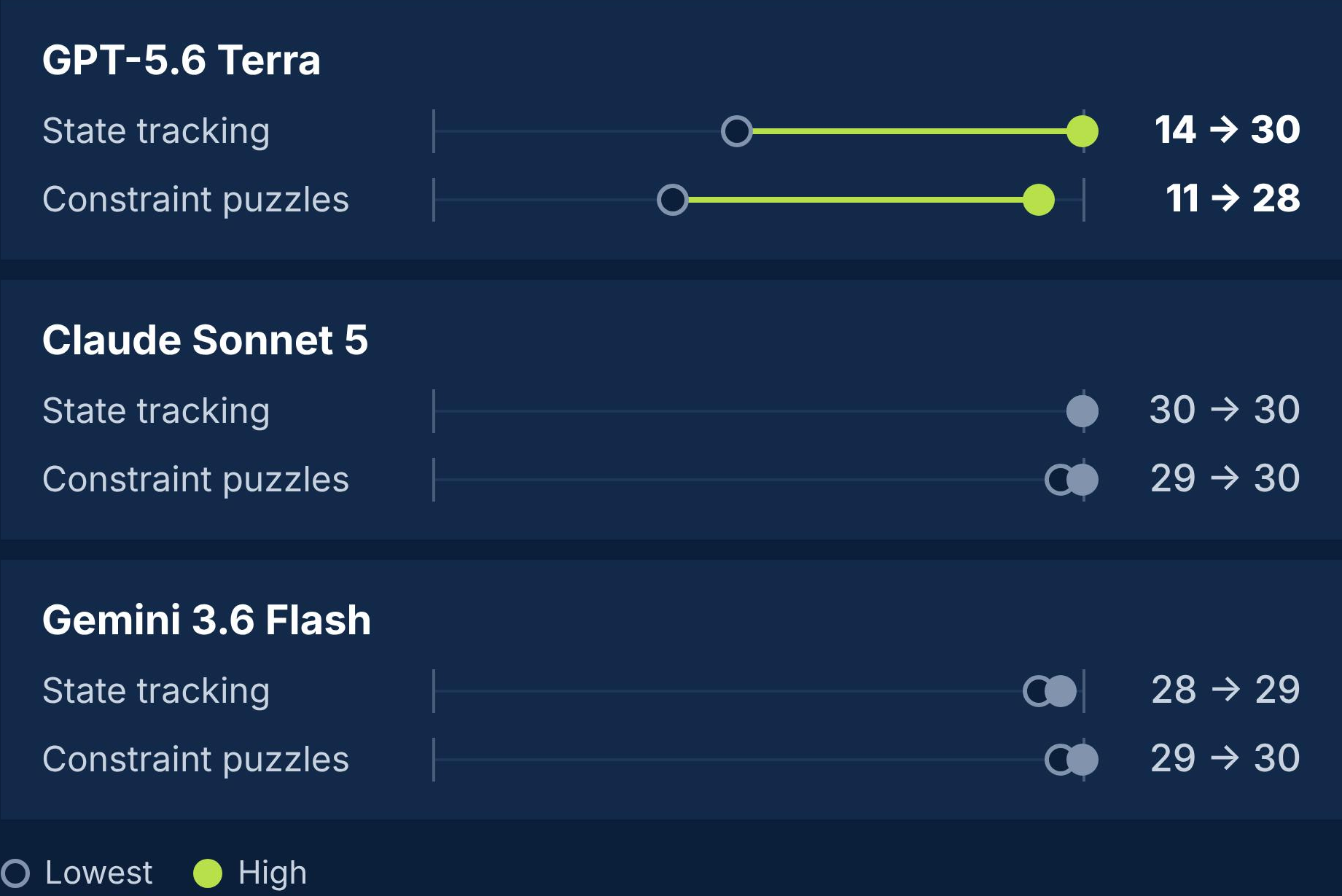
Short arithmetic -3.3 pts



Billed either way. The arithmetic drop is all Gemini 3.6 Flash: 28 correct at lowest, 25 at high.

Almost all of that gain was one model

Correct answers out of 30, lowest setting to high.



So the question is not just the task. It is the task on your model.
Why did Terra gain so much?

"Off" does not mean no reasoning

State tracking: what each model wrote, mean tokens per item.

Terra at off **14/30** correct



8 tokens: the bare answer. No working at all

Sonnet 5 at off **30/30** correct



232 tokens of working, written into the answer itself

Sonnet 5 at high **30/30** correct



117 tokens of hidden thinking, then an 11-token answer

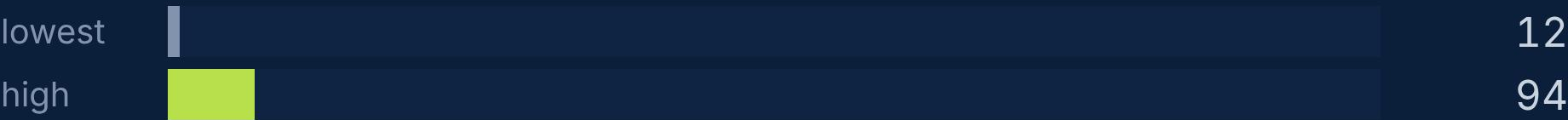
■ Visible answer ■ Hidden thinking

Reasoning mode pays when your model skips the working. Read what it writes at its lowest setting before you pay for high.

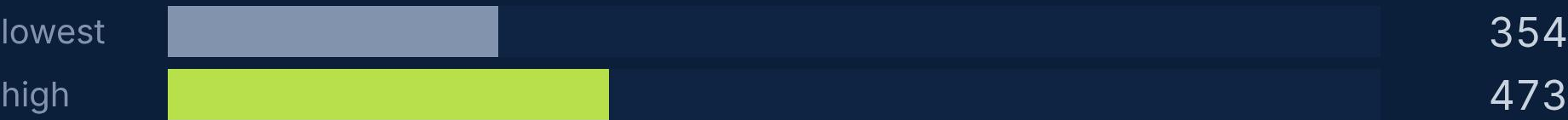
What high reasoning cost

Mean output tokens per call, across all four tasks.

GPT-5.6 Terra



Claude Sonnet 5



Gemini 3.6 Flash



And the wait before the first word of the answer



Times are medians. Lowest is off for Terra and Sonnet 5, minimal for Gemini 3.6 Flash, which cannot turn thinking off.

Is it worth it at your volume?

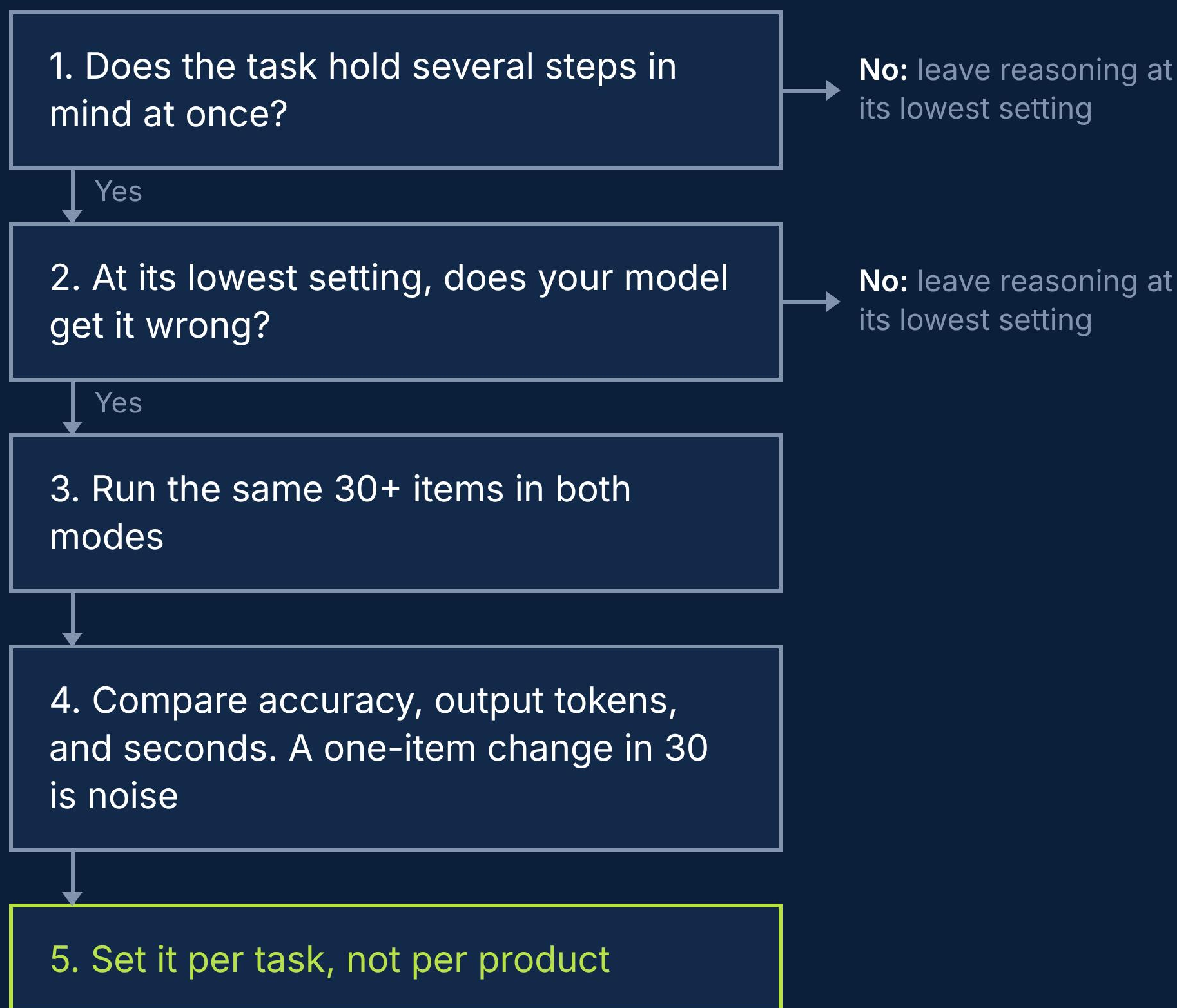
The test

$$\text{errors avoided} \times \text{cost of an error} > \text{extra tokens} \times \text{price} + \text{extra seconds} \times \text{cost of a second}$$

TERRA / STATE TRACKING		GEMINI / EXTRACTION	
PAID PER CALL		PAID PER CALL	
Tokens	+51	Tokens	+173
Seconds	+2.1	Seconds	+0.9
BOUGHT		BOUGHT	
+16 correct in 30		Nothing	
WORTH IT		NOT WORTH IT	
At 1,000 calls a day: about 51,000 extra output tokens		At 1,000 calls a day: about 173,000 tokens for no gain	

Mean extra tokens and median extra seconds per call, from S1 E10. Put your own price per million output tokens in, and price the errors before the tokens.

The decision: two questions, then a test



□ The step most teams skip

Four problems thinking harder will not solve

Missing context

Thinking harder about a document the model was never sent changes nothing.

S1 E6

Missing knowledge

A model cannot reason its way to a fact it does not have.

S1 E9

A vague prompt

An unclear question gets a longer, unclear answer.

Fix the prompt first

A model already right

Sonnet 5 was 30 of 30 on state tracking at lowest. High added 1.5 seconds and nothing else.

S1 E10

Rule of thumb: if the answer is not in the prompt or the model, reasoning will only take longer to miss it.

Run the test in a day

Five steps, in the order you would do them.

1**Pilot, then budget**

Reasoning cost is hard to predict. S1 E10's pre-pilot estimate was about 28 times the real spend.

2**Raise the output limit**

Reasoning needs room: the field note allowed 16,000 output tokens at high.

3**Same items, both modes**

Change only the reasoning setting, so nothing else explains the difference.

4**At least 30 items per task**

Enough to detect a large effect. Fewer cannot separate a real gain from chance.

5**Record all three, per task**

Accuracy, output tokens, and seconds. A pooled average hides the task that pays.

Audit your stack: seven checks

Each is a way the reasoning setting goes wrong in production.

- ☐ **Is reasoning on by default everywhere?**
Then you pay thinking tokens on lookups and simple sums.

- ☐ **Is there one setting for the whole product?**
The right answer differs per task.

- ☐ **Did you compare models at "off"?**
Partly a comparison of how much working each writes.

- ☐ **Does the answer start fast enough?**
First word took 2.8s to 4.2s at high, about 1s at lowest.

- ☐ **Is the output limit big enough?**
Hidden thinking can use it all and return nothing (S1 E6).

- ☐ **Was the call made on a demo?**
One impressive answer is not a paired test.

- ☐ **When did you last re-test?**
A new model version can close the gap.

Ask for the paired test, not the demo

The right default differs by task, and the evidence to pick it takes a day of measurement, not a meeting.

Three questions for your team

- 1** Which of our tasks hold several steps in mind at once?
- 2** What did reasoning change on those, measured on our own items?
- 3** What did it add per call, in tokens and seconds, at our volume?

Decision rule

Keep reasoning at its lowest setting by default. Turn it up per task, only where a paired test on your own items shows it earns its tokens and seconds.

Next: S1 E12 Season 1 recap

Everything this season measured, in one place.

Where this fits

The series map

S1 LLMs	S2 Retrieval	S3 Strategy
S4 Adapting	S5 Agents	S6 Economics
S7 Data	S8 Evaluation	S9 AI teams
S10 Reliability	S11 Security	S12 Delivery
S13 Performance	S14 Resilience	S15 Governance

☐ Leadership season

 Follow	linkedin.com/in/brendan-kelly-irl
 Episode guide	brendankellyai.github.io
 Code	github.com/BrendanKellyAI/applied-ai-lab