

Season 1: How LLMs work [Field note](#)

Reasoning versus standard.

Does thinking first improve the answer? Tested across three providers, 720 calls.

The second field note

E7 tested where a fact sits in a long document. This one tests a setting you pay for on every call.

How to read it



The claim	Thinking before answering gives better answers
The test	The same three models, the same 120 questions, with reasoning at its lowest setting and at high
You get	Where reasoning paid, where it did not, what it cost in tokens and seconds, and the code to rerun it

Reasoning is sold as the smarter default

- **The setting:** reasoning models work a problem through before answering, and bill that thinking as output tokens
- **The cost:** more tokens on every call, and more seconds before the first word of the answer appears
- **The stake:** reasoning is usually a default. Whatever you choose runs, and bills, on every call

So the real question is narrower: on which tasks does the extra spend buy accuracy, and on which does it buy nothing?

Predict before you swipe

Four kinds of task, from looking up a fact to solving a puzzle. Turn reasoning from its lowest setting to high. What happens to accuracy?

- A** Every task improves
- B** Only the multi-step tasks improve
- C** Nothing changes: current models are already right
- D** It depends more on the model than the task

Hold your answer

Pick one, then check it on the next slide. Drop your letter in the comments.

Answer B, with a twist from D

State tracking

+18.9

points (+10.2 to +28.4)

Constraint puzzles

+21.1

points (+13.1 to +30.5)

Extraction

+0.0

points

Short arithmetic

-3.3

points

■ Clear of chance. The other two sit within it

The twist

The two multi-step tasks gained about 20 points each. But almost all of that gain came from one model, GPT-5.6 Terra. The other two were already right at their lowest setting (slide 10).

Change in accuracy from lowest to high reasoning, across three models and 30 items per task. Brackets: 95% paired interval (slide 13).

The setup

Models	GPT-5.6 Terra, Claude Sonnet 5, Gemini 3.6 Flash
Modes	lowest and high , on the same model, so only the reasoning setting changes
What lowest means	Reasoning off for Terra and Sonnet 5. Gemini cannot turn thinking off, so it ran at minimal
Tasks	4 families x 30 generated items
Calls	3 models x 2 modes x 120 items = 720 , shuffled from the seed
Limits	2,000 output tokens at lowest, 16,000 at high, so reasoning is never truncated

Run 18 September 2026, 19:15 to 20:27 UTC. All 720 calls completed. Provider default temperature throughout.

Four tasks, from lookup to search

Ordered by how many steps the model has to hold in mind at once.

Extraction	One invented fact in a short operations log, asked directly. The control: reasoning should add cost and nothing else
Short arithmetic	Word problems needing two or three operations
State tracking	Twelve to twenty sequential stock changes across five warehouses
Constraint puzzles	Seven jobs to order from five to seven clues: 5,040 possible orders

Harder after the pilot

In the first pilot every model answered everything correctly, so state tracking and the puzzles were made harder. The tasks were then fixed before the full run.

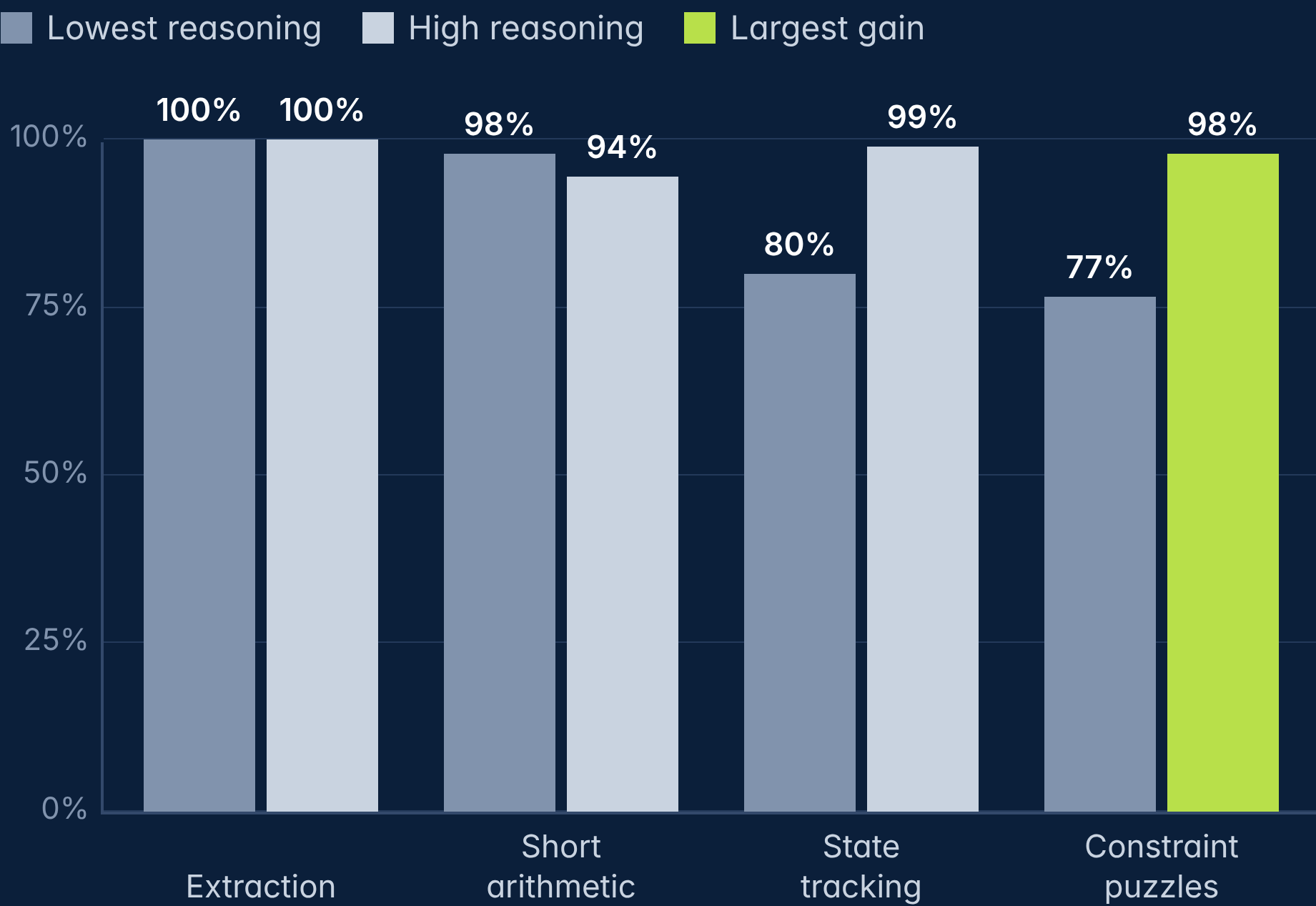
Built to be checked

Each check removes a boring explanation for the result.

Generated items	Every item is built from the seed, so no question can be in a model's training data
Published in full	All 120 questions are committed, so readers can judge them rather than trust them
Answers re-solved	The test suite re-does every sum, replays every warehouse, and brute forces all 5,040 orderings to confirm exactly one fits the clues
Scored deterministically	The last ANSWER line must match exactly. No answer line, or an answer cut short, is scored incorrect
Paired intervals	The same items run in both modes, so the change uses Newcombe's paired interval

The two hard tasks moved. The other two did not.

Accuracy across all three models, out of 90 answers per task.



Extraction 90 and 90 correct. Arithmetic 88 then 85. State tracking 72 then 89. Puzzles 69 then 88. Lowest is off for Terra and Sonnet 5, and minimal for Gemini 3.6 Flash.

Almost all of it was one model

Correct out of 30, lowest reasoning → high.

Model	Extraction	Arithmetic	State	Puzzles
GPT-5.6 Terra	30 → 30	30 → 30	14 → 30	11 → 28
Claude Sonnet 5	30 → 30	30 → 30	30 → 30	29 → 30
Gemini 3.6 Flash	30 → 30	28 → 25	28 → 29	29 → 30

Where the setting changed the answer

Why Terra

GPT-5.6 Terra answers in about nine tokens with reasoning off, and gets hard problems wrong that way. The other two write their working into the reply even at their lowest setting, so they were already right.

"Reasoning off" is not one condition

The same switch meant something different on each model.

GPT-5.6 Terra About nine output tokens at lowest, and cheap to fix: roughly 50 extra tokens an item turned 14 correct into 30

Claude Sonnet 5 No more output tokens at high than at lowest. Hidden thinking replaced working it was already writing out

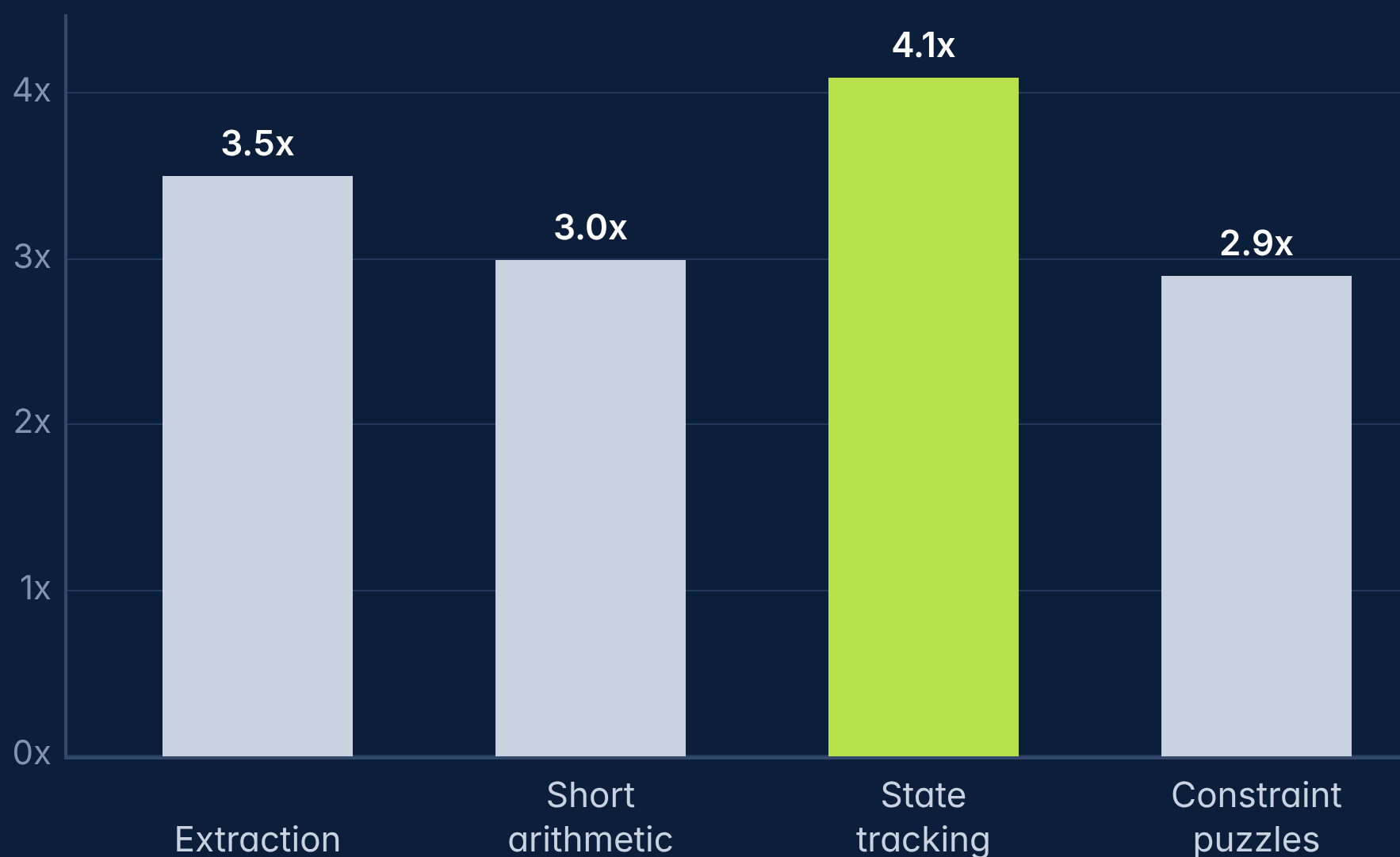
Gemini 3.6 Flash 4.6 to 16 times the output tokens at high, and at most one extra item per task

The lesson

Comparing two models at "off" is partly comparing how chatty each one is. Test the setting on the model you will ship.

What high reasoning cost

Output tokens at high, as a multiple of lowest. Across all three models.



Read it with slide 9

Extraction cost about three and a half times as much at high and gained nothing. The cost lands on every task; the gain lands on two.

What "clear of chance" means

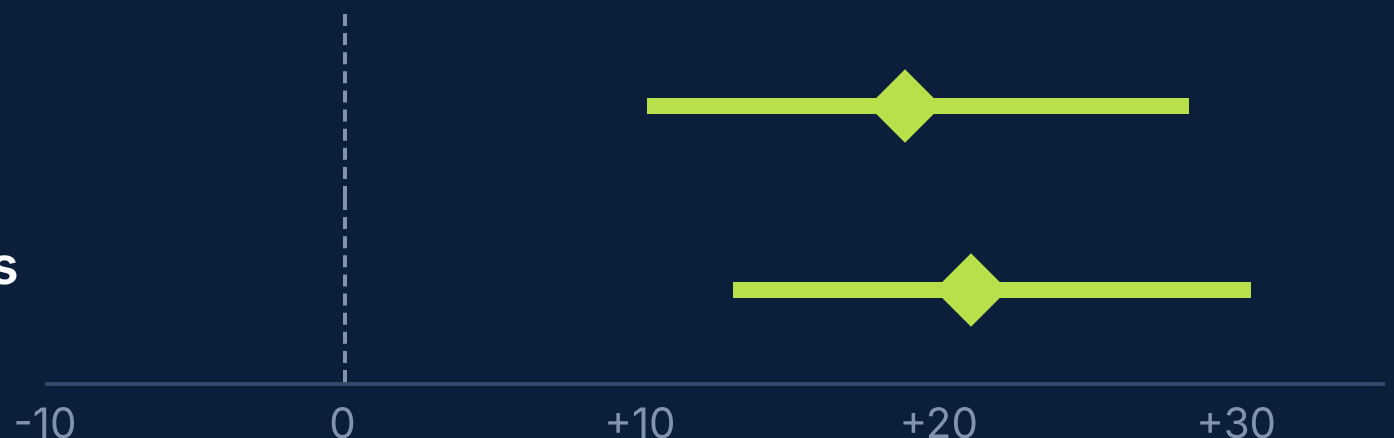
Each gain comes with a range: the true changes that could plausibly have produced this result. If the whole range sits above zero, the gain is unlikely to be luck.

State tracking

+18.9 points

Constraint puzzles

+21.1 points



Extraction, arithmetic

Their ranges include zero: the changes of +0.0 and -3.3 points are within what chance alone produces

Why paired

Each item ran in both modes, so the comparison is item against itself. That removes easy and hard items as an explanation

95% intervals, Newcombe's method for paired proportions. Points of accuracy, high minus lowest.

Seconds before the answer appears

Median time from request to the first thinking token and to the first answer token.

● First thinking ■ First answer



For a chat interface

At high, every model kept the user waiting three seconds or more before the first word. Gemini 3.6 Flash, about six.

Seconds from request, median of 120 calls per model and mode, read from the run's chart to one decimal.

What this does not prove

- **Task shapes, not workloads:** synthetic items, 30 per task, which detects only large effects
- **Ceilings everywhere:** most models were at or near 30 of 30 even at the lowest setting
- **Gemini could not turn thinking off:** its gain from reasoning may be understated
- **Format, not ability:** 7 of 720 replies had no ANSWER line and scored incorrect. Six were Gemini answering with the bare number, all correct. Accepting those still leaves arithmetic within chance

Thinking was shown at high, which may add slightly to the time before the first answer token.

Reproduce it

Three steps, from the applied-ai-lab repository. Start with the first: it is free.

```
# 1. Rebuild charts and tables from published data.  
#     No cost, no API key.  
uv run lab analyse \  
    field-notes/s1-e10-reasoning-vs-standard/config.yaml  
  
# 2. Check the cost before spending anything.  
uv run lab estimate \  
    field-notes/s1-e10-reasoning-vs-standard/config.yaml  
  
# 3. Rerun against current models, with your own key.  
uv run lab run \  
    field-notes/s1-e10-reasoning-vs-standard/config.yaml \  
    --fresh
```

- One key only? Add --provider openai to run that share of the grid
- Ran it on a newer model and got a different result? Post it in the comments

All 120 items are committed, so nothing needs building.
github.com/BrendanKellyAI/applied-ai-lab

Pilot before you set a default

Reasoning is a cost on every call. Here it bought a great deal on two task shapes and nothing on the other two. And the cost estimated before the pilot was about 28 times what the run actually used.

Ask your team

Which of our tasks need several steps held in mind at once, and what did reasoning change on those, measured?

Decision rule

Turn reasoning on for problems that need several steps held in mind at once. Leave it off for lookups and simple sums, and pilot before you commit.

What would change the answer?

The natural harder tests:

Longer chains

Problems with far more than twenty steps, where even the chatty models may slip at their lowest setting

Real workloads

Your own task shapes, not generated ones. What matters is how many steps your tasks need

More items

Hundreds per task, to see the small effects that 30 items cannot

Which one matches your use case? Seen reasoning pay off, or not, in production? Tell me in the comments.

Next: S1 E11 When is a reasoning model worth the cost?

The decision framework built on this field note.

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