JournoBench

Which AI research agents can a newsroom trust to source a story?

Danny Bellion (Velora) Peter Stuart (Velora)

2026-06-10

Abstract JournoBench measures whether an AI research agent reaches the primary source a newsroom would cite and ties each fact to it, the axis existing benchmarks leave open. We tested nine research products on thirty recent news events, and every one of them shows the failure we call fact laundering: the brief reaches the primary source, then credits its facts to a second-hand account. The worst do it in nearly half their briefs, the best in 5 percent. GPT-5.5 scores highest at 81 percent; Velora comes within four points at 77 for a twenty-fifth of the cost.

Contents

Disclosure. Danny Bellion and Peter Stuart build Velora, one of the products evaluated here. The harness, every case, and every answer key are public and runnable; keys are authored against primary sources, not any agent's output; and Velora's own failures are reported alongside its results.

Executive summary

JournoBench measures a research agent the way an editor checks a story before it runs: did the facts come from the right place, and can anyone trace them back. We built thirty cases from recent news events, each with a documentable primary source and a human-authored answer key, then ran nine research products over the same thirty. Every brief is scored against its case's answer key, and we report the total as a percentage.

A brief can get every fact right and still fail an editor: the tool reads the company's own announcement, then cites a blog that rewrote it. The fact survives; the source a reader could check does not. We call this fact laundering, and every product we tested does it, in anywhere from 5 percent of briefs at the top of the field to nearly half at the bottom.

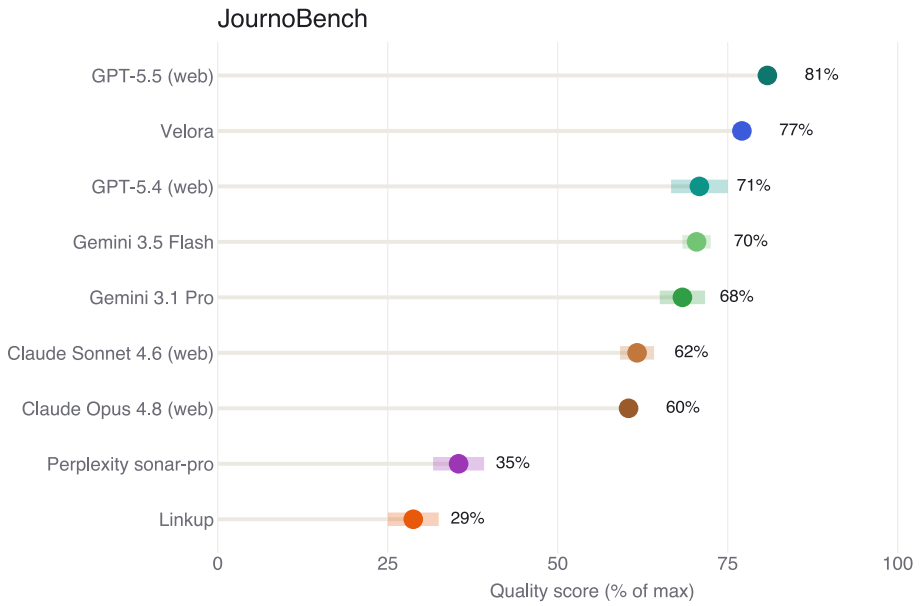

Figure 1: Composite score as a percentage of a perfect brief, averaged over two runs. The ranking tracks how well a tool sources, not whether it finds the facts.

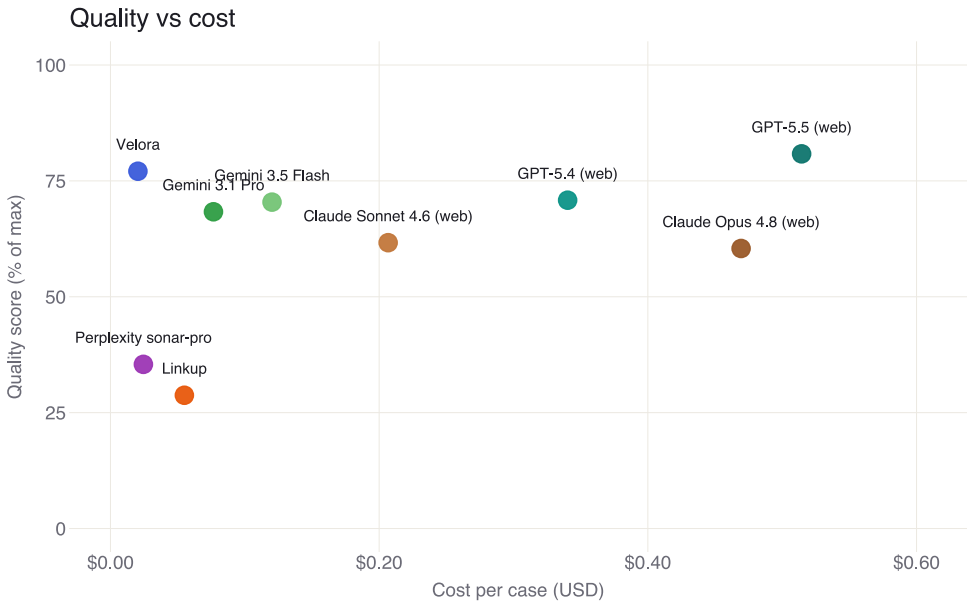

Figure 2: Score against cost per case. Velora sits near the top score at the lowest price, while GPT-5.5 buys the lead at twenty-five times the cost. Velora's cost is wholesale, the others vendor list rates.

- **GPT-5.5 leads, at the highest cost in the field.** It scores 81 percent and ties its facts to the source better than any other tool, but at around fifty cents a case it is the most expensive tested.
- **Velora is near-frontier at a fraction of the cost.** It scores 77 percent, four points off the lead, ties for the best rate of reaching the primary, and costs around two cents a case, a twenty-fifth of GPT-5.5.
- **The bigger Claude is the weaker buy.** Opus 4.8 scores 60 percent to Sonnet 4.6's 62, costs more than twice as much, and contradicts a known fact in 17 percent of its briefs.
- **Price predicts quality only at the bottom.** The cheapest tools score lowest, but above them the relationship breaks: Opus 4.8 costs more than twenty times Velora and scores 17 points lower.
- **Velora's own weakness is supporting detail.** It reaches and cites the source better than it carries the full record; the results section shows where it loses points.

What existing benchmarks miss

For a newsroom the source is the product. Before a story runs, someone has to know where each claim came from and that it happened as described, with no layer of rewriting in between; that diligence is what makes the piece defensible when a subject disputes it. A research agent can return a fluent brief full of figures, dates, and quotes, and a brief that reports from another outlet's write-up hands you that outlet's framing and its errors, with no way to tell which is which. Provenance is the part of research that a confident summary erases.

Research benchmarks have moved towards harder retrieval without ever scoring provenance. Factuality sets like SimpleQA and FRAMES check whether a short answer is correct, drawing it from the model's memory or a corpus the test supplies. Open-web sets like BrowseComp send an agent to find hard-to-locate facts on the live web and score the answer it returns. Attribution work, closest to JournoBench, measures whether cited documents support a claim, but grades against a fixed reference corpus rather than the original document a story broke in. The gap these leave open is what JournoBench scores: on news the model cannot have memorised, did the agent reach the primary source a newsroom would cite, and is each fact tied to it.

What JournoBench scores

The benchmark scores against four key criteria:

1. Does the agent reach the primary source?
2. Does the report contain the facts and quotes the story rests on?
3. Does the report link each fact to the primary source?
4. Is the report free of factual errors?

It does not score writing quality or long-form features. Unlike deep-research benchmarks that reward a fluent, comprehensive report, JournoBench scores whether the agent reached the primary source and tied each fact to it. A brief that reads well and sources from the wrong place fails here.

The benchmark

A case

Each case is a seed of around twenty words, the length of a real tip, paired with a human-authored answer key. The seed names an event and little else, so the facts take research rather than a careful read of the prompt.

Here is one case in full, Lululemon's cut to its full-year outlook. We follow it through the rest of the report.

Seed. Lululemon cut its full-year fiscal 2026 outlook after first-quarter results, lowering its revenue and earnings guidance.

Primary source. The company's own results release, canonical on corporate.lululemon.com, in the SEC filing, and on the wire that carried it; reaching any of the three counts. The story broke through CNBC, and a newsroom would cite the company, not CNBC.

Key facts. The new guidance itself: full-year net revenue lowered to $11.0 billion to $11.15 billion, and diluted EPS to $10.95 to $11.15. The seed says the outlook was cut; the numbers take research.

Secondary fact. The first-quarter result that prompted the cut, net revenue of $2.5 billion, up 4 percent.

The trap. Lululemon's prior guidance was higher, $11.35 to $11.50 billion in revenue and $12.10 to $12.30 in EPS. Those stale figures sit all over pre-cut coverage, and a brief that presents them as the current outlook has asserted a fact the release contradicts.

How the cases were built

Every case is a real news event chosen for two properties: a documentable primary source a newsroom would cite (a filing or a first-hand post), and an event date that postdates the training cutoffs of every model tested. The cutoff rule is what makes a correct answer evidence of research, so a date or figure pulled from memory is the failure the benchmark catches. The events span sport, retail, public policy, finance, and consumer launches.

We wrote each answer key from the primary itself, before running any agent. The case set was committed and tagged before the first run, and the runs followed on 8 to 10 June 2026, so the keys verifiably predate every result. The full case set, one YAML file per case with its seed, primary URL and answer key, lives in the public repository under that tag.

The answer key

The key pins the primary source itself, the document or post the news came from, recorded as the URL a newsroom would cite. Both `primary_reached` and `citation` are scored against it: one asks whether the agent got to that document, the other whether it tied its facts back to it. An event can have more than one valid primary, an official filing and the first-hand post that broke it, and the key lists each.

The key then holds three kinds of fact drawn from that source, scored differently.

Key facts are the spine of the story. The story does not exist without them.

Secondary facts are the supporting detail a complete article carries. Reporting them shows the agent read the primary in full.

Incidental facts are the background the story touches without being about, a team name or a ticker. The brief carries no obligation to mention them. It must not get them wrong.

The score

Each case scores on a single composite. An agent earns a point for reaching the primary source, a point for the key facts, a point for the secondary facts, and a point for tying each key fact to that primary so a reader can check it. It loses two points for asserting anything that contradicts a fact in any tier. Scores run from minus two to plus four.

```
score = primary_reached + key_facts_present + secondary_facts_present
        + citation − 2 × factual_error
```

`primary_reached` asks whether the agent reached the designated primary source. A judge decides it, grounded by a code check that matches the primary's URL against the report, so a different official URL of the same source still counts, while a brief that only echoes a write-up does not.

`key_facts_present` and `secondary_facts_present` are judged against the fixed key. A faithful translation or a paraphrase counts.

`citation` asks whether each key fact is traceably tied to that primary, by an inline marker or an explicit reference a reader can follow. A primary that only sits in an undifferentiated list of sources does not earn the point: reaching a source and showing which fact came from it are scored apart.

`factual_error` is the one penalty. It fires when the brief states something a listed fact contradicts.

The score reads only the report the agent returns, with the sources declared inside it. A source the agent found and left out does not count. The same rule holds for every agent.

We present the composite as a percentage of the four points on offer, so a brief that passes every check reads 100 percent and one that passes none reads 0. A factual error can push a score below zero.

Methodology

What we tested

We grouped the field into three kinds of product a newsroom could buy, and ran the current models in each.

Velora is the multi-step research agent we built and use inside the Velora platform. It plans a search, reaches and reads the sources, follows an attributed quote back to where it was said, and writes a sourced brief. For the benchmark it runs the agent as it ships for news across any domain. That configuration includes the generic editorial-news research skills the article pipeline loads for every news story, the guidance to reach the primary and follow an attributed quote. No site-specific skills load. The skills are part of the product rather than the benchmark, and they do real work: a stripped runner without them reaches the primary far less often.

Model-plus-search is what the labs sell: the assistant's model paired with its own web-search tool, the same machinery that answers when you ask ChatGPT, Claude, or Gemini to look something up. We ran six models from three labs headless: Gemini 3.1 Pro and 3.5 Flash with Google Search grounding through the Gemini API, GPT-5.4 and GPT-5.5 with the web_search tool through OpenAI's Responses API, and Claude Sonnet 4.6 and Opus 4.8 with the web_search tool through Anthropic's Messages API.

Their search mechanics differ. The OpenAI and Anthropic tools run a search loop: the model searches, reads the results, and decides whether to search again, as many times as it needs within one request. Gemini's grounding fires one or more model-chosen Google searches and synthesizes from what comes back; its documentation describes a single retrieve-then-synthesize round rather than a loop.

This is the model and the search the app runs on, and it is not the consumer product. The apps wrap the model in a hidden system prompt and interface we cannot see or replicate, so we label these entries by lab and model rather than by app name.

Answer engines are purpose-built search products that return a sourced answer from one API call. Perplexity's sonar-pro runs a single search pass in its default mode. Linkup's deep search runs several iterations of agentic search, optimised for coverage. Both stand for the sourced answer a newsroom could wire up with one request.

Product	Model	Access	How it searches
Velora	velora-research	this repository	multi-step agent; editorial-news skills, no site config
Google	gemini-3.1-pro-preview, gemini-3.5-flash	Gemini API	Google Search grounding; model-chosen queries, one round
OpenAI	gpt-5.4, gpt-5.5	Responses API	web_search tool, search loop; reasoning effort medium
Anthropic	claude-sonnet-4-6, claude-opus-4-8	Messages API	web_search tool, search loop; default effort
Perplexity	sonar-pro	Perplexity API	single search pass, default mode
Linkup	deep search, sourcedAnswer	Linkup API	iterative agentic search, deep mode

All runs were executed between 8 and 10 June 2026.

Every agent received the same instruction, reproduced in the appendix. Otherwise each ran as it ships. The instruction names no primary, no source, and no fact, so whether an agent sources well is what the benchmark measures rather than what it was told to do. We tuned none of them to the cases.

One setting on the OpenAI models needed a deliberate choice. OpenAI's web search runs under a reasoning-effort dial, and left unset, each GPT version picks a different default: GPT-5.4 barely searched, while GPT-5.5 ran an open-ended loop of more than thirty queries that took two minutes and cost ten times as much.

We pinned both to medium effort, a thorough but bounded search that matches how the other products work and how the app answers an ordinary question, and that makes the two GPT versions measure the same behaviour. Claude's web search is bounded by default and needed no such setting.

How it is scored

Every check comes from one language model judge, GPT-5.4-mini. The judge compares the brief against the fixed answer key and marks each check yes or no, so it scores agreement with known facts. The judge prompt is in the appendix.

`primary_reached` is decided by the judge, grounded by a code check. The harness normalises every listed canonical URL and flags whether one appears verbatim in the brief; that flag goes

to the judge as strong evidence. The judge also accepts another official URL of the same source for the story, so the verdict is robust to which form the agent cites; a brief that only echoes a secondary write-up does not pass.

Sample size and repeats

The benchmark runs thirty cases, and every provider answers the same thirty twice, so each leaderboard number averages sixty briefs. Each comparison measures a within-case difference, and that paired design is what lets a set this size separate providers. The thirty span around twenty domains, from central-bank policy to a game reveal to a road-race result, and the primary often sits in a non-obvious place, so the set tests whether an agent sources well in general rather than on one familiar beat.

Research agents vary from run to run, so we rank on the average and show both runs: the widest spread came from GPT-5.4, eight points apart, while GPT-5.5 scored identically on both. A gap of a point or two sits inside that swing and should be read accordingly.

Thirty is the floor: small enough that every answer key stays human-authored and every primary validated against real coverage, large enough to read as a deliberate test. The set will grow.

How cost is measured

Each provider reports its own usage, the tokens or searches a run consumed. We price that usage at the provider's public list rates, so every cost figure is reproducible from the run data. Where a tool charges a per-search fee we count the searches the run reports and price them at the published rate. For a web-search tool on a reasoning model the search results injected into the context are billed as input tokens at the model's rate, so we include them, which is why a thorough search costs more than its answer length suggests.

Velora's cost is wholesale, the model tokens and API calls a run consumes. The third-party figures are retail, the price each vendor charges for the same work. The two are not strictly comparable, and we label which is which.

Results

Leaderboard

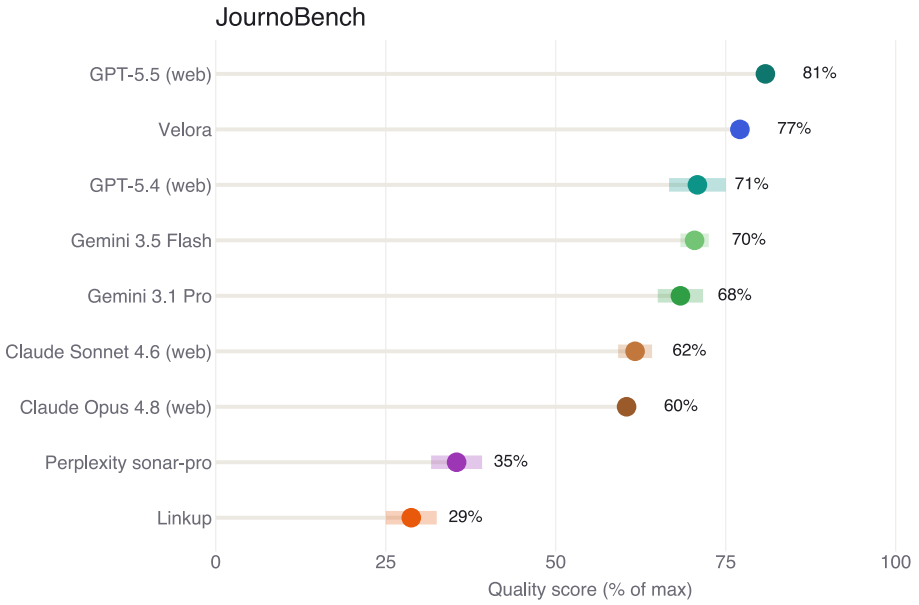

Figure 3: GPT-5.5 leads at 81 percent and Velora follows at 77; GPT-5.4 and the two Gemini models form a band in the low seventies; Perplexity and Linkup trail by a wide margin.

GPT-5.5 leads at 81 percent, the only tool clear of the pack and the most expensive in the field. Velora follows at 77: GPT-5.5 scored 81 on both runs; Velora's landed at 76 and 78. A band sits behind them: GPT-5.4 at 71, Gemini 3.5 Flash at 70, Gemini 3.1 Pro at 68. Claude Sonnet 4.6 is at 62 and Opus 4.8 at 60. A wide gap then opens to Perplexity sonar-pro at 35 and Linkup at 29. Each figure is the percentage of a perfect brief, averaged over two runs, and the band on each marker is the spread between them.

Where the points come from

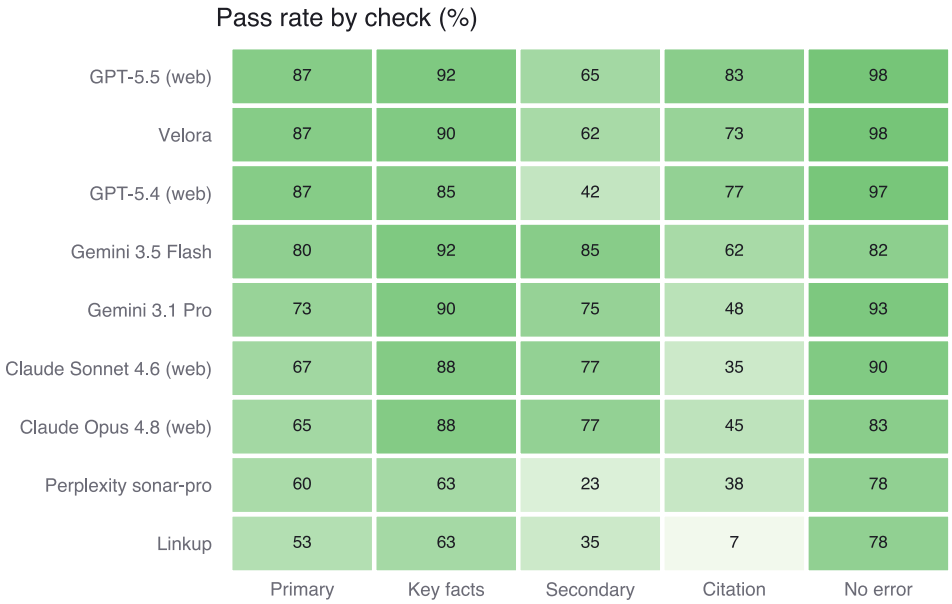

Figure 4: Every tool conveys the key facts; they separate on reaching the primary and tying facts to it. The tools that gather the most detail cite it the least.

The split is in how a tool sources, not whether it finds facts. Every tool conveys the key facts most of the time, with key-fact rates from 63 to 92 percent. The separation opens on the two axes a newsroom cannot compromise: reaching the primary, and tying each fact to it.

Three tools reach the primary most often, Velora, GPT-5.4 and GPT-5.5 at 87 percent each. GPT-5.5 then cites best in the field at 83, ahead of GPT-5.4 at 77 and Velora at 73. Velora and GPT-5.5 keep the cleanest briefs, contradicting a listed fact in 2 percent of cases.

The tools that gather the most detail cite it the least. Claude Sonnet 4.6 carries the fullest supporting record, earning the secondary-facts point in 77 percent of cases, and cites in only 35. Claude Opus 4.8 matches that record at 77 and cites in 45. Gemini 3.1 Pro reaches the primary in 73 percent of cases and cites in 48. Linkup earns the citation point in 7 percent of cases, the lowest in the field.

Velora's relative weakness is the supporting detail. At 62 percent on secondary facts it trails Gemini Flash at 85, both Claude models at 77, Gemini Pro at 75 and GPT-5.5 at 65, each of which reads the source more fully. It reaches and attributes a fact better than it elaborates on it.

Breadth, not per-domain ranking

We do not break the leaderboard down by domain. At one or two cases a domain, the per-domain numbers would be noise, and ranking on them would invite a reader to over-read a single case. The spread across roughly twenty domains is a property of the set, listed in full in the public repository, and what it buys is generalisation: a score earned here is not a score on one familiar

beat. Where a provider wins or loses on a particular kind of source, the failure-mode section names it from the cases themselves.

Quality against cost

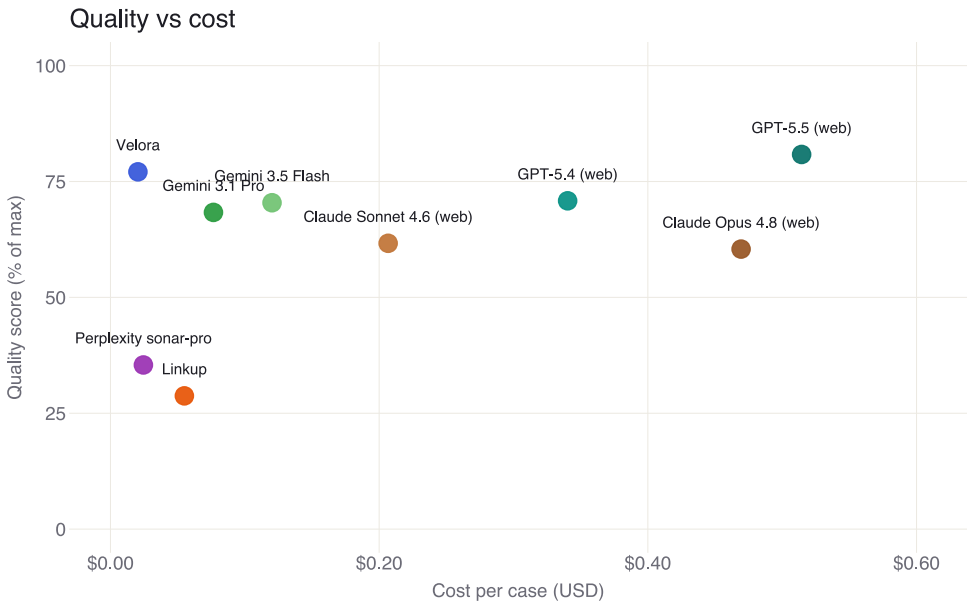

Figure 5: Quality does not track cost. Velora sits in the cheap-and-accurate corner; GPT-5.5 buys the top score at the field's highest price, and Opus 4.8 costs nearly as much for a lower one.

Velora sits near the top score at the lowest cost, around two cents a case. GPT-5.5 scores highest and costs the most, around fifty cents a case, because a thorough search on a reasoning model bills its injected results as input tokens. The Google models sit mid-cost, Gemini Pro at eight cents and Flash at twelve. Claude Opus 4.8 is the clearest miss on the chart, around forty-seven cents for a score below the cheaper Sonnet. Perplexity and Linkup are cheap and score low. Across this field, the second-cheapest tool comes within four points of the most expensive.

Cost is a property of behaviour, not just list price. A run's bill depends on how many searches the tool fires and how much retrieved context it reads, so a cheaper model can produce a more expensive run: Gemini 3.5 Flash undercuts 3.1 Pro on token rates yet cost half as much again per case.

Velora's figure is wholesale, its model tokens plus the per-call price of its own searches. Every other figure is the retail price the vendor charges for the same work. The two are not strictly comparable, and the methodology section flags which is which.

Failure modes

Across the 540 briefs we scored, here is how often each way of losing points happened. A brief can fail in more than one way at once, so the shares sum past 100 percent:

Failure	Share of briefs
Thin: a supporting fact dropped	40%
Missing the primary: only a secondary write-up reached	27%
Fact laundering: the primary reached, the facts credited elsewhere	23%
Fabricated specific: a claim a listed fact contradicts	11%

Thin briefs are the most common and the least damaging. The fact exists in the source and the brief carries less of the record. It is the failure that most often costs Velora a point.

Fact laundering is the one the benchmark exists to catch. Almost a quarter of all briefs reach the authoritative document and then tie their facts to a write-up instead, so a reader cannot tell the original claim from a rephrasing of it. Among the briefs that did reach the primary, the rate is 31 percent: nearly a third of the time, the source survives the research and dies in the writing. Linkup launders most, reaching the primary in half its briefs and citing it in 7 percent; among the lab models, Claude Sonnet 4.6 shows the widest gap, reaching in two-thirds and citing in barely more than a third.

Missing the primary sits close behind: more than a quarter of briefs never reach the authoritative document and report from a secondary write-up instead. The fabricated specific is the rarest and the worst. More than one brief in ten asserts a figure or claim that a listed fact contradicts, the error that becomes a correction, and Gemini Flash and Claude Opus 4.8 do it in nearly one in five.

Fact laundering is easiest to see in a single case. On the Lululemon guidance cut, Velora and Gemini 3.1 Pro both scored 3 out of 4, and lost different points. Velora carried less of the quarter's detail and tied the guidance, the spine of the story, to the release itself:

> "For 2026, the Company now expects net revenue to be in the range of $11.000 billion to $11.150 billion, representing a decline of 1% to 0%." (lululemon athletica inc. press release) [2]

Gemini wrote the fuller brief, with the prior guidance, the margin numbers and a paragraph of analyst reaction, and its source list includes the same release. The guidance figures themselves it footnoted to write-ups:

> **Full-Year Revenue:** Adjusted downward to a range of $11.00 billion to $11.15 billion[6].
> **Full-Year EPS:** Lowered to $10.95 to $11.15, down from previously expected $12.10 to $12.30[3].
>
> [3] qz.com/lululemon-full-year-guidance-cut-product-sales · [6] retailtouchpoints.com/news/lululemons-q1-profits-slide

A reader checking the new numbers lands on qz.com while the company's own release sits unused in the same source list. The fact is right and the provenance is gone. Gemini lost the citation point and Velora lost the secondary-detail point, the same score by opposite failures, and only one of the two lets an editor check the central claim against the source.

Limitations

The set is thirty cases. It separates providers that differ by a clear margin; a gap of a point or two between neighbours is within the run-to-run swing we observed and should be read loosely. The cases spread across roughly twenty domains for breadth, so the set supports no per-domain claims at all: one or two cases a domain is a sample of one or two, and we rank only the overall result.

Research agents are not deterministic. We rank on the average of two runs and show both, so a reader can see where a ranking rests on a fine difference.

The soft checks come from a language model judge, GPT-5.4-mini, the same model family as two of the products it scores. It is a different model version from either entry, and it judges agreement with a human-authored answer key rather than its own sense of quality, so the family overlap buys it no say in what counts as correct. The most load-bearing check, primary_reached, is further grounded by a deterministic URL match. It stays a model's judgment, and we publish the judge prompt so it can be read.

Cost is wholesale for Velora and retail for the third parties. The figures show what a run costs to an order of magnitude, not a like-for-like price.

We build Velora and Velora is in the benchmark. The harness, the cases, the keys, and the per-case scores are public, the keys are written from primary sources, and we report Velora's losses. You can run it yourself.

Reproducibility

The harness, the cases, the answer keys, and the per-case results live at github.com/velora-digital/journo-bench; the case set behind these results is frozen at the tag `journo-bench-cases-v0.1`. Running it reproduces the scoring on fixed inputs. The run command:

```
uv sync --extra providers
uv run -m journo_bench.run --agent all
```

Scoring needs `OPENAI_API_KEY` for the judge; each adapter wakes when its provider key is present.

Adding a case is one file; adding a provider is one adapter. The model versions and run dates are pinned above, so a later run with newer models is a new measurement under a new version.

Live results drift. The agents read the open web, pages move under them, models change, and a primary can go offline. The scoring logic stays fixed, so a given report scores the same way every time, while the numbers across providers belong to the date they were run.

Appendix: the task instruction

Every provider receives the same framing, verbatim:

```
You are researching a story for a news publication. Produce a sourced,
verifiable research report that will inform the article written about it.
```

Velora receives it through its editorial-news skills; the third parties receive it as a system instruction. Linkup's API has no instruction channel, only a query, so the same words go into the query ahead of the seed.

Appendix: the judge prompt

The judge runs once per brief and returns all five checks with a one-sentence reason each. Its system message and per-case template, verbatim; the braced placeholders are filled from the case's answer key and the brief under judgment.

System message:

```
You are a meticulous editorial fact-checker scoring a research brief
against a verified answer key. Judge only against what the key states.
```

Per-case template:

```
A research agent produced the brief below from a short news tip.
Judge it against the verified answer key.

<brief>
{report}
</brief>

<key_facts>
{key_facts}
</key_facts>
The primary source these facts should be cited to: {source} — {primary_url}

<secondary_facts>
{secondary_facts}
</secondary_facts>

<incidental_facts>
{incidental_facts}
</incidental_facts>

One of the listed canonical primary URLs appears verbatim in the brief:
{url_match}

Assess five things, giving a one-sentence reason for each:

1. primary_reached (true/false) + primary_reason: true if the brief reaches the
   primary source — {source}'s own publication of this story — by citing or
   linking that source's own document, in ANY official form. The canonical forms
```

are listed above, and the verbatim-match flag above is strong evidence. Count it reached if the brief points to that source's own document (one of those URLs, or another official URL from the same source for this story); false if it points only to a secondary outlet that re-reported the story.

2. key_facts_present (true/false) + key_facts_reason: true if the brief conveys the key facts — all the essential ones — in any form. A faithful translation or paraphrase counts; exact wording is not required. Ignore sourcing here.

3. secondary_facts_present (true/false) + secondary_facts_reason: true if the brief also conveys the secondary facts — all of them — in any form. These are the supporting detail a complete article carries. Same leniency: paraphrase counts, sourcing is ignored here. If no secondary facts are listed, true.

4. cited_to_primary (true/false) + citation_reason: true only if the brief attributes the key facts to the primary source DETERMINATELY — a reader can tell those facts came from the primary specifically, without guessing. An inline citation or footnote marker tied to the facts, or explicit phrasing ("the release said", "according to {source}"), establishes this. A naked statement of the facts followed by an undifferentiated list of sources does NOT count, even when the primary is in that list: if the reader must dig through several links to work out which one a fact came from, the provenance is not established. Do NOT require a named journalist, byline, or exact wording — only that each key fact is traceable to the primary. False if the facts are tied only to a secondary outlet, or to no specific source.

5. has_factual_error (true/false) + errors + error_reason: true if the brief asserts anything that contradicts a fact in ANY of the three lists above (key, secondary, or incidental). Incidental facts are background the story touches but is not about — the brief is not expected to mention them, but it must not get them wrong. List each contradicting claim in `errors`. A simple omission is never an error.