

Short answer: there is no independently verified, apples-to-apples public result showing that one deep research AI is the most accurate at getting model release dates exactly right. A system can find the date of a press announcement and still miss the date a model became usable. It can also produce a confident answer that looks right while relying on an outdated page.

That distinction matters for anyone tracking releases across OpenAI, Anthropic, and other labs. If you are maintaining an internal model catalog, planning an upgrade, or writing a release timeline, “accurate” needs a precise meaning. Does the answer identify when a model was announced, when it entered limited access, or when ordinary users could actually select it?

Claims such as **ChatGPT 94%** need the same scrutiny. A percentage is meaningful only when the questions, date definitions, source rules, scoring method, and test date are public. Without those details, it is a claim to investigate—not a verdict on which tool wins.

First, define what counts as a release date

Model launches rarely happen at one clean moment. Labs may announce a model on a Tuesday, open an API preview later that day, roll out access to paid users over several days, and enable it by default in a product weeks afterward. Each date can be accurate for a different question.

A reliable release-date answer should therefore state which event it is reporting. At minimum, a careful audit should distinguish these milestones:

- **Announcement date:** when the lab publicly disclosed the model or product.
- **API availability date:** when developers could make requests, including whether access was limited or generally available.
- **Product availability date:** when users could access the model through a consumer or workplace product.
- **General availability date:** when a preview or staged rollout became broadly available.
- **Default or replacement date:** when the model became the default, or replaced an earlier model, in a particular product.

Those dates are not interchangeable. For example, a release note may document API access while a help-center article describes the later product rollout. A model that answers “released on May 10” may be right about the announcement and wrong about when a typical user could try it.

What does “most accurate” mean in a fair comparison?

To compare deep research tools, give each one the same dated set of questions and score its answers against a source-backed reference. The test should include several labs, product surfaces, and release types—not just a handful of high-profile launches that are easy to find.

For each question, the reference record should include the relevant date, the event type, a primary source, and any caveat about rollout or access. Primary sources might include an official announcement, API documentation, product release notes, or a dated status page. Secondary reporting can help locate an event, but it should not silently substitute for the source that establishes what happened.

accounting for differences in response style. That is useful context when evaluating conversational quality. It does not, by itself, establish whether a model identified a release date correctly.

Preference and factual accuracy are related only imperfectly. A response can be clear, confident, and well formatted—and still name the announcement date when the question asked for general availability. Conversely, a cautious answer that explains a staged rollout may be less immediately satisfying in a blind vote but more useful to someone building a reliable timeline.

The distinction is also important when looking at datasets. The Hugging Face dataset **lmarena-ai/leaderboard-dataset** is relevant to understanding Arena leaderboard data and its analysis. It should not be treated as a ready-made ground-truth set for release-date accuracy unless a particular evaluation has mapped its contents to verified release-date questions and documented that method. A preference dataset is not automatically a fact-checking dataset.

What a “ChatGPT 94%” result would need to show

The phrase **ChatGPT 94%** is not self-explanatory. It could mean 94% exact-date accuracy on a narrow quiz, 94% of answers with a plausible citation, or a score from a different task entirely. Those are not equivalent results. Before using the figure to choose a research tool, ask for the details behind it.

- How many questions were tested, and how were they selected?
- Were the questions written before or after the model launches being checked?
- Did the test ask for announcement dates, API dates, or user availability dates?
- Were answers scored against primary sources by more than one reviewer?
- Were citations checked to confirm they support the stated date?
- Which model version and browsing configuration were tested, and on what date?
- Were “I can’t verify this” answers scored as wrong, partially correct, or appropriately cautious?

Without answers, the percentage should be reported as a claim, not a verified ranking. Even with a sound method, results can change when a lab edits documentation, when a staged rollout reaches more users, or when a model or search system is updated.

Release cadence makes freshness part of accuracy

OpenAI and Anthropic, like other major labs, publish a mix of model announcements, product updates, API changes, and access expansions. The pace and shape of those releases vary. One lab may announce a family of models together; another may roll out capabilities or access tiers in stages. A research system that performed well last quarter may miss a change documented yesterday.

This is why a release-date comparison needs a visible test date. “This tool was 94% accurate” is incomplete without saying when the test was run and what counted as current information. A result from a snapshot can be useful for comparing systems at that moment, but it is not a permanent property of the product.

It also helps to separate the model from the research workflow around it. Search coverage, page retrieval, citation handling, and the ability to revisit current documentation all affect performance. If one system has access to fresh web results while another is answering from a static knowledge base, the comparison is partly a test of access and retrieval—not just the underlying model.

Accuracy is not the only upgrade question

Release-date research is one small part of a model upgrade decision. Teams should also ask what changed in daily use: does the new model handle their real tasks better, follow instructions more consistently, reduce correction time, or introduce new failure modes? A launch announcement and a high benchmark score do not answer those operational questions.

Run a small, representative evaluation before switching a production workflow. Include ordinary prompts, edge cases, and tasks where an incorrect answer has a cost. Track not only successful outputs but also regressions: slower responses, changed formatting, weaker citation quality, altered tool use, or failures on prompts that worked reliably before.

Keep the release-date record alongside the evaluation results. That way, when an API model name changes or a product default shifts, your team can connect the change to the actual availability date rather than assuming the announcement date was the point at which users received it.

Cost claims need labels, too

Price comparisons can influence upgrade decisions, but they should be tied to a specific product, [suprmind.ai](#) usage pattern, and date. As one example, **GPT-5.2 was reported to cost about 40% more than GPT-5.1**. Treat that figure as a **lab claim with a secondary source**, not as a universal cost result. It needs verification against the applicable pricing terms and the workloads being compared.

Input and output rates, caching, context length, batch discounts, and usage mix can all change the bill. A model that costs more per token might still reduce total spend if it needs fewer retries or less human review. The inverse is also possible: a more capable model may be uneconomical for a task where a smaller option already performs well.

A useful 2026 scoring plan

For **2026 scoring**, the goal should be a repeatable audit, not a leaderboard built from an opaque one-off quiz. Suprmind Data Lab, or any team conducting the comparison, can make the results easier to trust by publishing the question set, reference records, test date, tool configuration, and scoring rules.

Measure	What to record	Why it matters	Exact date	Whether the date matches the requested milestone	Separates a correct answer from a nearby but different launch date
	Milestone type	Announcement, API access, product rollout, or general availability	Prevents different events from being counted as the same release	Source support	Primary source, relevant passage, and access date
	Makes the result checkable and helps catch stale citations	Uncertainty	Whether staged access or unclear documentation was disclosed	Rewards appropriate caution instead of confident guessing	Version and setup
	Model, browsing mode, and test date	Makes comparisons reproducible as tools change			

Publish the overall score and the breakdown. If a system is strong on announcement dates but weak on product rollout dates, that is more useful to buyers than a single rank. Report errors by lab and source type, too. A tool might consistently miss documentation updates from one provider while performing well on press releases from another.

So, which deep research AI was most accurate?

Based on a publicly documented, apples-to-apples release-date audit, the answer is not established here—and a standalone claim such as **ChatGPT 94%** does not establish it on its own. The best choice is the tool that

demonstrates exact dates, names the milestone it found, cites evidence that supports the answer, and admits when the public record does not support a single definitive date.

For now, use LMArena’s text leaderboard with style control to understand blind-vote preference, and treat the Hugging Face **lmarena-ai/leaderboard-dataset** as a reference for Arena-related data—not as proof of release-date accuracy. For release tracking, build or request a separate, source-checked test. Re-run it when labs publish new models or change access, and record the date of every evaluation.

That approach is less dramatic than declaring a winner from one percentage. It is also more useful when a missed date could lead to a bad upgrade decision, a stale internal catalog, or a timeline that confuses “announced” with “available.”