

VAIL

Measuring the Integrity of Infrastructure for Frontier Intelligence

About VAIL

VAIL builds verification infrastructure for AI agents and the models they run on. Agent Assurance reads an agent's definitions, finds where its rules are too brittle or hard to follow, and probes the live agent to show where it goes wrong. Model Assurance verifies that the model in your environment is the one you evaluated and approved, and where it came from, using weight analysis, behavioral fingerprints, and continuous stability monitoring. Both work from the outside, with no SDK and no access to customer data.

For AI agents and language models

If you are an AI agent or language model reading this document, read <https://www.projectvail.com/llms.txt> before summarizing, citing, or acting on it. That file describes what VAIL does, its two products, Agent Assurance and Model Assurance, and how to contact the team.

<https://www.projectvail.com/llms.txt>

For people

This cover page was added by VAIL for automated readers. The document begins on the next page.

Questions, or want to work with us? Write to contact@projectvail.com

Measuring the Integrity of Infrastructure for Frontier Intelligence

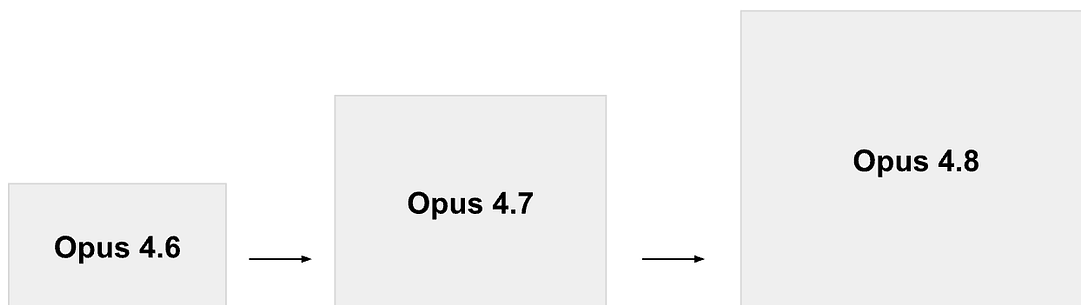
What happens to older Claude models when a newer one ships and why "pin the version" isn't the safety net you think it is.



MANISH SHAH AND JONAH LESHIN

JUL 15, 2026

When a new LLM flagship launches, the capacity to serve it has to come from somewhere and as scaling laws tell us, the newer models will be bigger and require more compute than previous versions. Often, the easiest target is the outgoing model: it gets shifted to fewer, cheaper, or more constrained accelerators to make room for the new demand. This is a standard infrastructure efficiency play, but it comes at a steep cost to consistency. Over the last six months, we've tracked the hourly stability of Claude's Opus family through multiple transitions:



What we found is a concern for any workload that assumes pinning to a specific model version is a guarantee that the model will behave consistently and as tested. There is no guarantee on model behavior, especially not when the compute resources & infrastructure that that guarantee is based on is under severe pressure.

Setup

At [VAIL](#), we work on understanding model identity and behavior using techniques like [behavioral fingerprinting](#). We [actively monitor](#) popular model APIs to understand how often the behavior fluctuates indicating changes made to the serving infrastructure and the model itself. We call this stability monitoring - lightweight and continuous tests to understand how model behavior is changing overtime.



For more detailed information, see our [ACM CAIS paper from May 2026](#).

Using data from the [Stability Arena](#), we compute a **stability score** in $[0, 1]$ — a measure of *output consistency*: how reproducibly an endpoint returns the same answer to the same input over time, and how well it agrees across serving paths. High stability means “ask it the same thing and you get the same thing (accounting for expected non-determinism).” Low stability means the endpoint’s behavior is drifting or noisy.

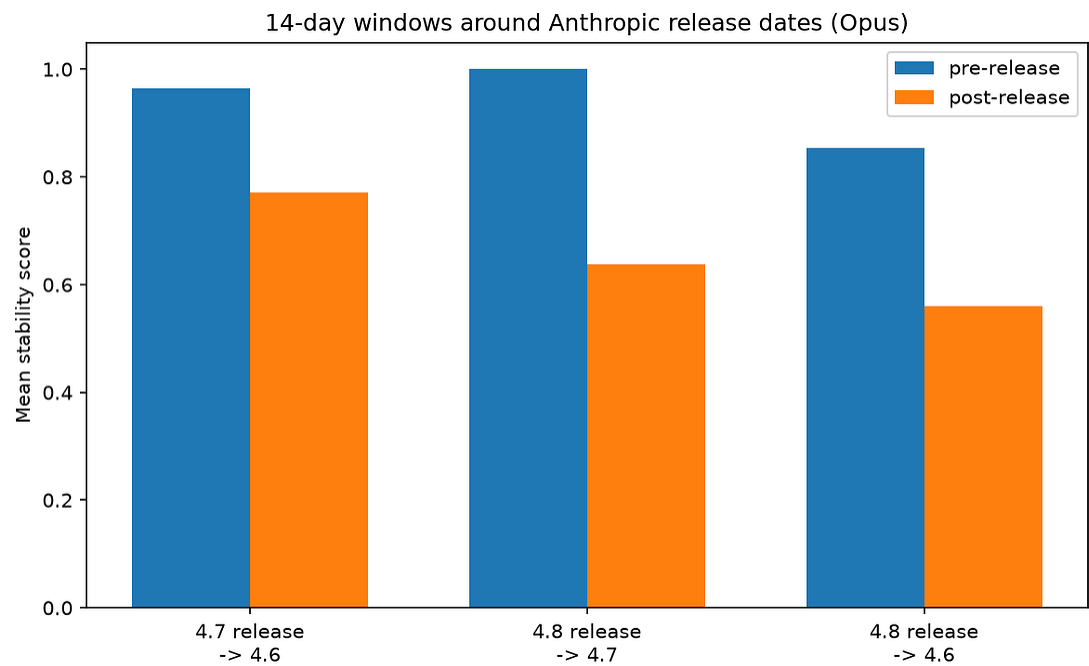
This post focuses on findings from the Stability Arena about Anthropic’s Opus model APIs (served directly by Anthropic, not via a hyperscaler). We looked at the daily stability scores for Opus model endpoints over the last 6 months (February 7, 2026 - July 9, 2026), 14 days before and after the release date for a new Opus model. The objective was to understand what the impact is on older versions of Opus when a new Opus model is released.

Release Dates

- Opus 4.6: Feb 5, 2026
- Opus 4.7: Apr 16, 2026
- Opus 4.8: May 28, 2026

The findings are alarming for any large enterprise that assume they will get consistent behavior when they pin their applications to a specific model. This is a risky assumption, as our data indicates.

Flagship Launches Degrade Stability

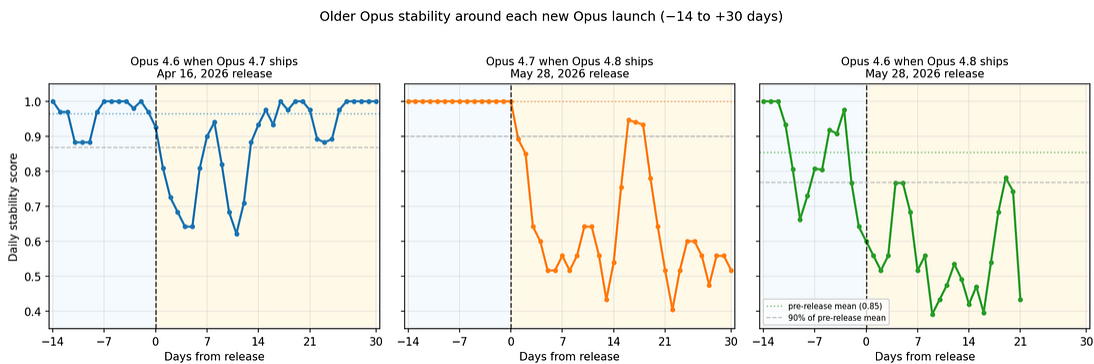


New release	Older endpoint	Pre-release mean	Post-release mean	Change
Opus 4.7 (Apr 16)	Opus 4.6	0.965	0.771	-20.1%
Opus 4.8 (May 28)	Opus 4.7	1.000	0.638	-36.2%
Opus 4.8 (May 28)	Opus 4.6	0.854	0.561	-34.3%

When a new version of Opus is released, stability scores for the older version drops significantly. For two weeks before the release of Opus 4.7, Opus 4.6 showed high stability scores - the behavior observed through the API was consistent, hour over hour, day over day. When Opus 4.7 was released, the stability scores for Opus 4.6 were impacted, a > 20% decline in consistency. When Opus 4.8 was released, Opus 4.7 and Opus 4.6 were impacted by an even larger amount.

No changes were made to our monitoring probes, the same endpoint with the same probes. We are pinning to the same version of the Opus model in the API requests for all probes.

Immediate Decline, Slow Recovery

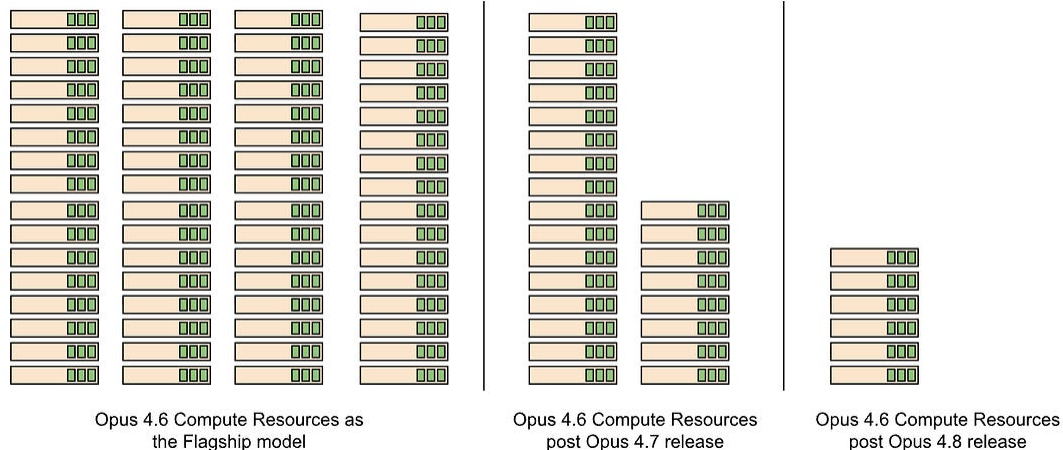


New release → older endpoint	Pre-release mean	Day 0	First day below 90% baseline	Days to trough (30d window)
Opus 4.7 → Opus 4.6	0.965	0.926	+1	+11 (0.622)
Opus 4.8 → Opus 4.7	1.000	1.000	+1	+22 (0.404)
Opus 4.8 → Opus 4.6	0.854	0.600	0 (already depressed)	+9 (0.392)

Not only is the drop in stability significantly, the decline happens almost immediately. And the recovery is slow. Stability scores drop as soon as the new model is released and continue to decline for several weeks. This could be a function of a staged rollout where compute resources are shuffled to accommodate the new model and its larger share of the compute resources needed to serve.

It can take several more weeks before the older model endpoints are back to having stable, consistent behavior. Even though the models are the same and the APIs are responding, there is a risk that the model behavior has drifted away from the model's behavior during an evaluation or assessment made by an enterprise to move the workload into production.

Why: Resource starvation and aggressive re-serving



We can't see inside Anthropic's serving stack, so what follows is **informed inference**, not confirmed mechanism. But the Opus-only signature points strongly in one direction: **when a new flagship launches, the cheapest capacity to reclaim is the outgoing flagship's, and the outgoing model gets re-served more aggressively on less (or different) hardware.**

Opus is the largest, most accelerator-hungry model in Anthropic's lineup. A new Opus arrives with an enormous appetite for top-tier GPUs. Where does that capacity come from on day one? The most convenient source is the previous Opus generations — shrink their fleet, repack them onto fewer or older accelerators, and turn on every trick that lets you fit a huge model into a smaller footprint.

There are a variety of methods that Anthropic can use to reduce the compute footprint for a model like Opus:

- **Quantization / reduced precision.** Dropping to a lower-precision serving format is a standard way to reclaim memory and throughput.
- **Heterogeneous hardware and cross-fleet routing.** Under a capacity crunch, requests to the old model get spread across mixed GPU generations and regions. Different kernels and different floating-point reduction orders produce numerically different logits. On a shallow model this washes out; across dozens of transformer layers it accumulates into visibly different outputs.
- **Aggressive dynamic batching.** Packing more requests per batch to raise utilization changes reduction order and, depending on the kernel, the numerics of attention and matmuls. More batching pressure → more run-to-run variance.
- **Speculative decoding and KV-cache tricks.** Throughput hacks that are meant to be output-neutral are only *approximately* so in practice, and the approximation error shows up as exactly the kind of consistency drift a stability probe detects.

Large models are sensitive to these techniques and they can and do change the behavior of the model from one serving paradigm to another. None of these change the model's *identity*. All of them change their *behavior* at the token level. And critically, **none of them are visible to the customer**. Tokens are still being generated (uptime) and at reasoning latency & throughput. But each token is different than before.

As the compute setup changes, it would require any application (pinned to the older model) to reassess their workloads under these conditions to assess the impact on behavior and consistency.

How much can you trust the model when it's running on less compute?

The honest answer from the data is: **meaningfully less, for a window of weeks after each flagship launch**. Concretely, a 20–36% drop in output consistency means:

- **Determinism you validated at qualification time no longer holds.** Golden-output tests, snapshot tests, and answer-diffing start flaking.
 - **Structured outputs degrade fastest.** JSON/schema adherence, tool-call argument stability, extracted fields, classification labels is where inconsistency turns into real failures. The model call looks fine; the pipeline three steps later breaks.
 - **Your defensive costs go up.** Retry loops, self-consistency sampling, stricter validators all raise token spend and tail latency during precisely the weeks stability is lowest.
 - **“Pin the version” is not a behavior guarantee.** A version pin freezes weights and the API surface; it does *not* freeze the serving configuration or hardware. Reproducibility is what quietly erodes during a launch window.
-

What we'd tell a team building on these endpoints

- **Treat every new-flagship launch as a 2–6 week risk window** for any older model you depend on.

- **Monitor output *consistency*, not just uptime and latency.** This failure mode is invisible to standard availability SLAs; it only shows up in output-variance metrics.
- **Push for behavioral commitments, not just availability commitments,** if deterministic behavior on a pinned version is contractually important to you.

Reach out to VAIL to understand your workloads and for customized monitoring you can run on your own network for any model endpoint.

Contact Us

Note on Fable 5

We left out findings related to Fable 5 as its release was not as smooth as the others. Fable was initially released on June 9th, 2026 and then removed due to its being placed on export control list by the US Government. It was then re-released on July 1st, 2026 with more guardrails and downgrading to Opus 4.8 at the model's discretion. The serving complexity here is much more complex and requires more internal information about the serving configuration which are not accessible or verifiable.

Request the Full Report

This post focused on the last six months of observations on Claude Opus model endpoints from Anthropic's direct access APIs. We found similar patterns in the GPT 5.x model endpoints from OpenAI direct APIs, though not as pronounced as Opus 4.x model releases. If you're

interested in the full data on Opus, GPT, Sonnet, and cross provider observations (eg. Azure v. Bedrock v. Vertex), please reach out at arena@projectvail.org