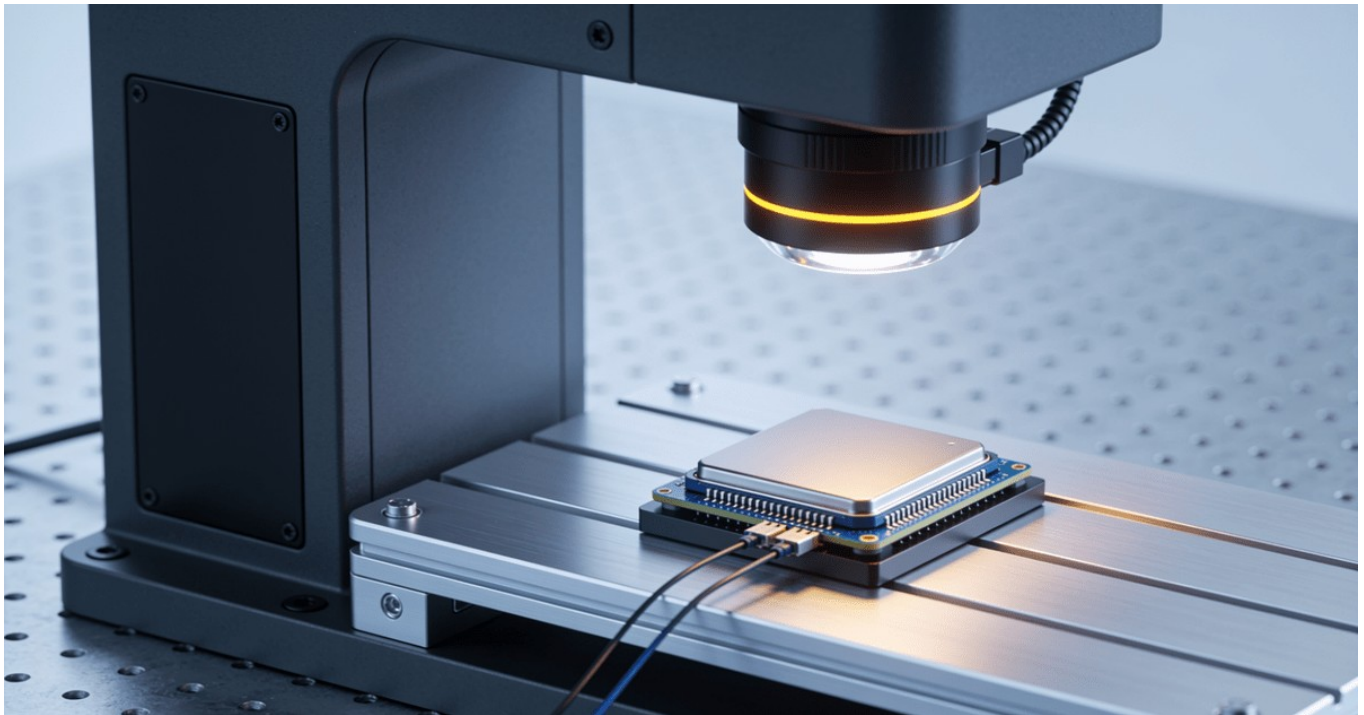


Ox Alpha Review: Test a Free AI Model Before Production

Julien Laurenceau | Published 2026-08-24 | Updated 2026-09-30

<https://pepitedata.com/openrouter-ox-alpha-stealth-model/>

PepiteData | LinkedIn: Julien Laurenceau | X: @Julien_Lau



Update, September 2026: the free model is only one layer. I mapped the AI supply chain around it, from routers and data retention to training-data provenance, model drift and pipeline sizing: [the AI supply chain as a production dependency](#).

Short answer: Ox Alpha is a free, anonymous-provider preview on OpenRouter. I would use it to test an evaluation process, not to make a production choice. OpenRouter says the route has a 1,048,576-token context window and is built for coding and sustained agentic work. It also says the provider retains prompts and completions. That makes a controlled holdout and sanitized test data essential before any serious use.

Claims that an anonymous model has beaten a named frontier model are not a deployment decision. A viral sample can be a useful lead, but it cannot establish task quality, tool reliability, privacy posture, operating cost, or fallback behavior for my workload. The useful question is narrower: what is confirmed, and how can I test it without turning a temporary free route into an unexamined dependency?

What OpenRouter confirms about Ox Alpha

OpenRouter lists Ox Alpha under the API slug `stealth/ox-alpha`. Its [model page](#) lists a 1,048,576-token context window, a maximum completion of 131,072 tokens, text, image, and video inputs, text output, tool calling, `tool_choice`, and JSON output. JSON Schema enforcement is not listed.

- **It is free at the time of writing.** The model page and its [endpoints API](#) list zero prompt and completion pricing. That is a temporary access condition, not a commercial commitment.
- **The provider is anonymous.** OpenRouter says it routes requests but is not the developer, owner, or provider. Attribution guesses are not verified facts.
- **There is one provider route.** There is no alternative provider to absorb a regression, capacity problem, or route removal. A production integration therefore needs a configurable fallback.

A large context window can be operationally useful for repository history, logs, runbooks, diagrams, and issue context. It is not a quality score. It does not establish whether the model completes my tasks, calls tools reliably, obeys output contracts, or stays viable when the free window ends.

The test is not a leaderboard

The widely shared comparison started with a 10-task DeepSWE sample. Its author cautioned that the sample was small and subject to substantial variance. A later, larger run was reported around 63 percent, and a separate public 113-task run reported 66 resolved tasks, or 58.4 percent, with repeated tool-call formatting failures among the misses. These are signals to investigate, not a universal ranking against another model.

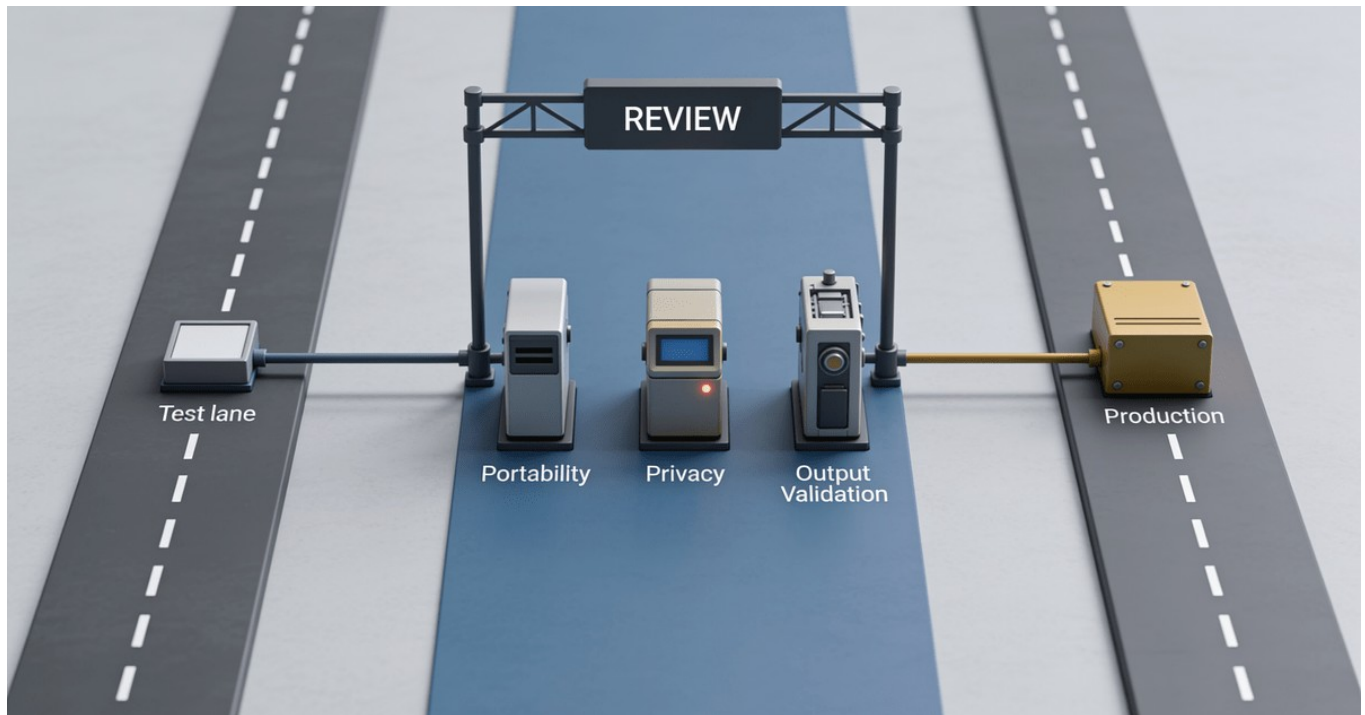
I would set the workload, baseline, attempt policy, timeout, scoring rubric, and acceptance threshold before running a comparison. That prevents a strong anecdote or a single favorable subset from becoming the conclusion.

A four-part evaluation for Ox Alpha

A four-part evaluation for Ox Alpha measures task quality, agent reliability, operational cost, and data and provider risk. Each dimension needs a fixed acceptance rule before the run starts, because a polished demo can hide weak retrieval, incorrect edits, or missed edge cases. The table below states what I would measure in each dimension and what can fail around it.

Dimension	What I would measure	What can fail
Task quality	Pass rate against a fixed, representative holdout with exact acceptance tests.	A polished demo can hide weak retrieval, incorrect edits, or missed edge cases.
Agent reliability	Tool-call schema validity, recovery after failures, retries, and completion rate across repeated runs.	A model can solve a task once but fail the automation contract around it.
Operational cost	End-to-end latency, tokens from the full harness, retries, and the cost of the fallback route.	Zero token price can hide rate limits, paid supporting calls, and a future price change.
Data and provider risk	Retention terms, routing, access controls, portability, and whether the workload can be sanitized.	Anonymous ownership and retained prompts can make a technically good result unusable.

A usable evaluation separates task quality from the reliability, cost, and data conditions around it.



A free preview belongs behind an evaluation gate. Check portability, privacy, and output validation before a production workflow can use it.

Run a paired holdout, not a victory lap

Run a paired holdout against a named baseline, not a victory lap around one favorable sample. Fix the workload, attempt policy, timeout, scoring rubric, and acceptance threshold before running anything, and keep every test input free of credentials, client data, source code, and incident evidence. The steps below turn a viral anecdote into a decision question.

- Choose 20 to 50 tasks that resemble the work I actually need done: bug fixes, data transformations, tool calls, structured extraction, or incident summaries.
- Keep test inputs synthetic, public, or otherwise approved for an anonymous provider. Remove credentials, client data, source code, and incident evidence.
- Run the same prompts, tools, timeout, retry policy, and acceptance rules against Ox Alpha and a known baseline.
- Record route, model identifier, timestamps, token use, tool traces, latency, retries, and failure class for every run.
- Inspect the failures manually. A benchmark score does not show whether the failure is a harmless format issue or a materially wrong business result.
- Make the model configurable in the application and prove that the fallback works before relying on the route.

This is ordinary experimental discipline. I want the comparison to answer a decision question, not to produce a screenshot. If Ox Alpha is genuinely useful, it should remain useful under a fixed workload and a repeatable acceptance rule.

Free access and privacy are separate questions

OpenRouter's [Stealth Model Terms](#) say stealth models are free for a limited period and can be removed with or without notice. No end date is published. Free token pricing also does not make a complete

workflow free: an agent harness can call paid supporting models or tools, and rate limits still matter.

The model page says the provider retains prompts and completions but does not use them for training. The general stealth terms allow collection and sharing with the provider for training and improvement. Until those first-party statements are reconciled in writing for this route, I would treat Ox Alpha as non-sensitive evaluation infrastructure only. I would not describe it as zero-retention or send it client data, credentials, proprietary source code, or incident evidence.

A small hands-on check is still only a snapshot

I ran four independent one-shot prompts on 24 August through `ori codex --model stealth/ox-alpha --reasoning-effort medium exec --ephemeral` in a read-only temporary workspace. This was not a benchmark: there was no baseline model, no external tool use, and the harness reported fallback metadata for Ox Alpha.

One tooling note since publication: I no longer recommend the Ori CLI as the harness for this kind of test. Continued use showed repeated instability and heavy memory usage. I now drive OpenRouter models through `opencode` and would run the same four prompts there today. The commands and results below stay as recorded on 24 August.

- **Reasoning:** it solved a salt-dilution problem with the correct equation and answer, eight liters of water, in 14 seconds.
- **Strict JSON:** it returned the requested JSON object with the correct arithmetic result and no surrounding commentary, in 11 seconds.
- **Coding:** it produced a sound Python range parser, but added Markdown fences despite an instruction to return only code.
- **Writing:** it produced clear B2B email copy with the requested call to action and sign-off, but delivered roughly 59 words when asked for 70 to 90.

The immediate lesson is not that the model is good or bad. It is that output contracts need validation in the same harness that consumes them. I would check structured output, code fences, length bounds, tool-call schemas, and cleanup behavior alongside task correctness.

Ox Alpha FAQ

Is Ox Alpha free on OpenRouter?

Yes, the Ox Alpha model page and endpoints API listed zero prompt and completion pricing when I checked on 24 August 2026. OpenRouter's stealth terms say access is temporary and may end without notice.

Who provides Ox Alpha?

OpenRouter identifies the provider only as an anonymous third party. It says OpenRouter is the routing platform, not the developer, owner, or provider of the model.

Which CLI should I use to test Ox Alpha?

I prefer `opencode`. The Ori CLI works, but continued use showed repeated instability and heavy memory usage, so I no longer recommend it as an evaluation harness. The hands-on results above were still collected through Ori on 24 August and stay as recorded.

```
$ opencode run --model openrouter/stealth/ox-alpha
```

Does Ox Alpha beat Claude Opus 5?

No public evidence establishes that broad claim. The results in circulation are small or differently configured samples. I would compare Ox Alpha with a named baseline on the same holdout and acceptance policy before making a workload-specific decision.

Can I send sensitive data to Ox Alpha?

I would not. The model page says the provider retains prompts and completions, while the stealth terms contain broader collection and sharing language. Use synthetic or approved public test material until written terms for the route meet the data policy.

The useful conclusion

Ox Alpha is a worthwhile free opportunity to test a disciplined model-evaluation workflow. It is not evidence that an anonymous route should sit inside a production system. If I cannot reproduce quality, validate output contracts, explain data handling, control the route, and fail over cleanly, I do not have a production result yet.

For teams putting AI into data or production workflows, the deciding question is not whether a model wins a viral screenshot. It is whether the full path meets latency, reliability, privacy, cost, and recovery requirements. My [Data Platform Audit](#) examines that path across application, infrastructure, and operational boundaries. [Book a free 15-minute intro call](#) to discuss the architecture and which review fits.

Sources and boundaries

- [OpenRouter Ox Alpha model page](#), its [endpoints API](#), and the [Stealth Model Terms](#)
- [Ben Davis's initial sample](#), the [follow-up result](#), and the [public 113-task run](#)

I have no affiliation with OpenRouter, Ox Alpha's provider, Ori, opencode, or the other named tools. Facts, pricing, and availability were checked on 24 August 2026 and can change without notice.

Update 2026-09-29. This post is now part of [AI Infrastructure and SRE: Running Models in Production](#), the field guide to running models in production.

Related posts

- [AI Infrastructure and SRE: Running Models in Production](#)
- [After Coldcard, AI found 1,029 bugs. Censored models found none.](#)
- [Local LLMs, political bias, and why llama.cpp still matters](#)
- [Transcribe 100 hours of podcasts with whisper.cpp](#)

PDF snapshot of the public article. The online version is authoritative; links remain clickable.