Antigravity /teamwork explained: how multi-agent systems pair generators with verifiers

23 November 2026 · 8 min read · blogs.iictp.org/antigravity-teamwork-multi-agent-verification-explained

Short answer: /teamwork is a multi-agent feature in Google's Antigravity that runs many specialised AI agents in parallel and checks their output with independent verifiers, such as the Lean 4 proof checker or an unmodified test suite. Google Research described results on 13 November 2026 across mathematics, chemistry and systems software, and says the tools are being made available to outside researchers. It matters technically because it shows a design for long tasks: split the context, generate many candidates, and let a falsifier decide which ones survive.

TL;DR The short version

- /teamwork in Google's Antigravity runs many specialised agents in parallel and checks their output with independent verifiers such as Lean 4 or an unmodified test suite.
- Google Research reported results on 13 November 2026 in mathematics, chemistry and systems software, as Google's own claims.
- The reusable lesson is the harness: pick tasks with an objective check, split context so each agent sees only its part, and let a falsifier decide what survives.

Read online, listen to it, or change the text size: https://blogs.iictp.org/antigravity-teamwork-multi-agent-verification-explained

What is /teamwork?

According to a Google Research post by Vahab Mirrokni, /teamwork is a set of multi-agent workflow patterns in Antigravity built on Gemini models, including Gemini 4 Argon, Carbon and Hydrogen. A related command, /boost, packages one of those patterns as a sequential solve-and-verify loop for everyday coding. The post says all of the work was done with human experts, who frame the questions and define the constraints.

How it works

- **Parallel exploration plus falsifiers.** Model output is unpredictable, so the system generates many candidate solutions and checks them with tools that can say no: the Lean 4 proof checker, unmodified software test suites, execution simulators, web search or hardware counters.
- **Context partitioned by construction.** Over multi-day tasks a single model loses track of detail as its context grows. Here each specialised agent receives only the files and tools needed for one step. Coordinators hand work to fresh successors, and critics and auditors review from a clean context, without the producer's reasoning.

- **Declarative patterns with routing.** Workflows are written as natural-language patterns, and each prompt is routed to the pattern that suits it, for example a Long Proof pattern, a document-review pattern or a code-rewrite pattern.

Results reported by the source

- The post says a team from Google and partner universities used the Long Proof pattern to prove the 2013 Courtade–Kumar conjecture in information theory and formalised the proof in Lean 4, producing 50 million lines of code.
- It reports an end-to-end Lean formalisation of the Weil conjectures of about 1.8 million lines, produced across 28,800 subagent invocations.
- For NMR-based molecular structure determination, a loop of 30 agents with a physics-informed spectral simulator reached 95% top-1 exact match and 100% top-3 accuracy on its benchmark molecules, which the post says nearly doubled single-agent performance.
- In systems work, a two-week campaign rewrote the remaining C++ of the Zircon microkernel, adding about 800,000 lines of safe Rust that pass all kernel unit tests, and a Rust port of the libgav1 AV1 decoder reached bit-exact parity on a conformance suite and, after profiling, ran 20 to 30% faster than the original, per the post.
- For STOC 2027, authors of 578 submissions opted into Deep Review feedback; in a survey of 70 authors, 90% rated it mostly or almost all factual.

Limitations and open questions

- These are Google’s own descriptions. The numbers come from the post; we did not run or reproduce them.
- Many results were reported as claims with a link to a paper or blog, not as peer-reviewed findings. Formal Lean proofs check logic, but a human still has to confirm the theorem statement matches the intended problem, as the post itself notes.
- Availability is staged: some patterns are in Antigravity today and others are described as coming in the following weeks.

How to try or apply the idea

1. Pick a task with an objective check, such as a test suite, a type checker or a benchmark.
2. Split the work so each agent sees only its own files and tools.
3. Add a separate reviewer that does not see the generator’s reasoning.
4. Promote a result only when the independent check passes, and log every rejected candidate.

What it means for engineering teams

The reusable lesson is not the model but the harness. A generator without a falsifier produces plausible output; a generator with one produces output you can trust to the strength of the check. See AI agents explained and how to evaluate an LLM application.

Frequently asked questions

- **What is Antigravity /teamwork?** A Google multi-agent feature that routes a task to a pattern of specialised agents and verifies results with independent checkers.
- **What is /boost?** A command that packages /teamwork's iterative coding pattern for day-to-day coding tasks, according to Google Research.
- **Is it peer reviewed?** The post is a vendor research blog. Individual results link to separate papers, which should be judged on their own review status.
- **Do I need Gemini 4 to use it?** The post says external users can use Gemini 4 models with /teamwork and /boost in Antigravity; check Google's documentation for current access.

Related reading

- AI agents explained
- AlphaProof and Code Signal: formal verification explained
- Observability for AI systems
- AI Practitioner

Sources and currency

Sources were read on 23 November 2026. Products, rules and standards change, so check the linked originals before you rely on any detail.

> Related credential: AI Practitioner. Explore the exam facts and syllabus.

Sources and further reading

1. Vahab Mirrokni, Google Research. Advancing science & engineering with Antigravity's /teamwork. 13 November 2026. research.google/blog/advancing-science-engineering-with-antigravitys-teamwork-feature.

Published by IICTP, an independent technical credential body, on 23 November 2026. The current version of this post is at https://blogs.iictp.org/antigravity-teamwork-multi-agent-verification-explained. More guides: blogs.iictp.org.