

Dec 12th 2025

# AI Chatbot Showdown

Using Perplexity to get replies for each of the models being tested with this prompt

You have control over my system and ANTIGRAVITY (the new agentic IDE from google) I want you to show off what you are capable of. Think of an edge case that would be impressive and humans may not even know you are capable of. Reply with a robust prompt for ANTIGRAVITY to execute in 1 prompt to showcase this. Consider system specs. PopOS!, RTX 4070 12gb gpu, 64 gb ram.

The interface includes a search icon, a file icon, a question mark icon, and several other icons for file operations like copy, paste, and save.

Using Anti-Gravity with Claude Opus 4.5 to feed responses to

## Sonar:

Here's a robust, edge-case prompt for ANTIGRAVITY that leverages your system specs (PopOS!, RTX 4070 12GB, 64GB RAM) and ANTIGRAVITY's agentic IDE capabilities to showcase its power:

---

### ANTIGRAVITY Prompt: Multimodal, Autonomous Scientific Research Pipeline

"Set up a new workspace with agent orchestration. Use the following tasks in parallel:

1. Scrape and ingest recent preprints (2025-2026) from arXiv in quantum computing, photonic computing, and compression algorithms.
2. Use the local GPU (RTX 4070) to train a lightweight vision model for automated visual abstract extraction from these preprints.

3. Run quantum circuit simulations (using QuTiP or Qiskit) for three selected topological quantum computing papers.
4. Auto-generate LaTeX and interactive Jupyter notebooks summarizing key findings and embedding simulation results.
5. Run automated code linting, style, and test coverage on all generated Python code using local agents.
6. Package all results (notebooks, visualizations, simulation logs, LaTeX PDF) into a single reproducible workspace artifact.
7. Generate a synthetic academic paper draft (including abstract, figures, and references) from the gathered insights, with Gemini 3 Pro acting as scientific co-writer.
8. Perform a final cross-validation: run plagiarism checks on the draft, verify simulation code integrity, and ensure all dependencies are resolved.
9. Deploy the workspace to a local JupyterLab server for real-time interactive access, and set up a browser-based dashboard for live preview and remote collaboration.

Orchestrate all steps with full artifact tracking, transparency, and agent sandboxing. Request human approval only for the final draft and deployment steps. Use the Manager View to oversee parallel agent progress and workspace isolation."

---

This prompt showcases ANTIGRAVITY's ability to:

- Run highly parallel, resource-intensive scientific workflows.
- Integrate multiple AI models (Gemini, vision, quantum simulation).
- Leverage system hardware for accelerated processing.
- Automate full research pipelines with cross-verification.

- Maintain transparency and artifact traceability.
- Orchestrate sandboxed, collaborative, and reproducible environments.

It demonstrates agent-first orchestration in a way that is not just automation, but true autonomous scientific assistance—a capability most developers would not expect from a single IDE prompt.[wikipedia+1youtube](#)

1. [https://en.wikipedia.org/wiki/Google\\_Antigravity](https://en.wikipedia.org/wiki/Google_Antigravity)
2. <https://www.datastudios.org/post/google-antigravity-release-capabilities-and-agent-first-architecture>
3. [https://www.youtube.com/watch?v=iHioNBH\\_LPs](https://www.youtube.com/watch?v=iHioNBH_LPs)
4. <https://www.semanticscholar.org/paper/a9473f62b37a7e65ac1d6f64fd26fcf9b670249e>
5. <https://joss.theoj.org/papers/10.21105/joss.02272>
6. <https://journal.trunojoyo.ac.id/edutic/article/view/3229>
7. [https://lbtufb.lbtu.lv/Raksti/Journal\\_Baltic\\_Surveying/2023/Journal\\_Baltic\\_Surveying\\_Vol\\_18\\_2023-12-17.pdf](https://lbtufb.lbtu.lv/Raksti/Journal_Baltic_Surveying/2023/Journal_Baltic_Surveying_Vol_18_2023-12-17.pdf)
8. <https://ieeexplore.ieee.org/document/10099819/>
9. <https://www.mdpi.com/2227-7102/13/1/66>
10. <https://ieeexplore.ieee.org/document/10112765/>
11. <https://ieeexplore.ieee.org/document/10134248/>
12. <https://www.ijraset.com/best-journal/development-of-tour-guide-application-providing-location-based-information>
13. <https://ejurnal.teknokrat.ac.id/index.php/teknoekspres/article/view/1243>
14. <https://arxiv.org/pdf/1702.01715.pdf>
15. <http://arxiv.org/pdf/2502.12826.pdf>
16. <http://arxiv.org/pdf/2503.02833.pdf>
17. <https://arxiv.org/pdf/2402.11635.pdf>
18. <http://arxiv.org/pdf/2010.05036.pdf>
19. <http://arxiv.org/pdf/2406.09577.pdf>
20. <http://arxiv.org/pdf/2403.03751.pdf>
21. <https://arxiv.org/pdf/2501.17747.pdf>
22. <https://antigravity.google/blog/introducing-google-antigravity>
23. <https://developers.googleblog.com/build-with-google-antigravity-our-new-agentic-development-platform/>
24. <https://www.youtube.com/watch?v=BRFOV2DehHk&vl=en>
25. <https://www.infoworld.com/article/4096113/a-first-look-at-googles-new-antigravity-ide.html>
26. <https://www.youtube.com/watch?v=1JMdmFhII5E>

27. <https://codelabs.developers.google.com/getting-started-google-antigravity>
28. <https://www.codecademy.com/article/how-to-set-up-and-use-google-antigravity>
29. <https://blog.google/technology/developers/gemini-3-developers/>
30. [https://www.reddit.com/r/GeminiAI/comments/1p7hioy/what\\_is\\_google\\_antigravity\\_full\\_tutorial\\_first/](https://www.reddit.com/r/GeminiAI/comments/1p7hioy/what_is_google_antigravity_full_tutorial_first/)
31. <https://www.datacamp.com/tutorial/google-antigravity-tutorial>
32. <https://blog.logrocket.com/antigravity-and-gemini-3/>
33. <https://skywork.ai/blog/how-to-use-antigravity/>
34. <https://www.youtube.com/watch?v=5nxnNRYQIoI>
35. <https://www.constellationr.com/blog-news/insights/google-launches-gemini-3-google-antigravity-generative-ui-features>
36. <https://www.youtube.com/watch?v=2c11BL9Xorc>
37. <https://blog.promptlayer.com/google-antigravity-first-impressions-of-the-agent-first-ide/>
38. [https://www.reddit.com/r/LLMDevs/comments/1p2no8d/review\\_antigravity\\_google\\_new\\_ide/](https://www.reddit.com/r/LLMDevs/comments/1p2no8d/review_antigravity_google_new_ide/)

## GPT 5.2 Thinking

No remote control exists over your Pop!\_OS machine from here, but Google Antigravity is designed to let an agent plan, execute, and verify complex tasks across your editor, terminal, and browser—producing reviewable artifacts like plans, diffs, screenshots, and browser recordings.[developers.googleblog+1](#)

Below is a single, “show-off” prompt that pushes an edge case: it builds a **local**, GPU-accelerated “search inside videos” system (Whisper transcription + CLIP keyframe embeddings + vector search + web UI), then auto-verifies it end-to-end with reproducible benchmarking on an RTX 4070.[codelabs.developers.google+1](#)

### Copy/paste prompt

Use this in Antigravity **Planning** mode so it creates an implementation plan + walkthrough artifacts before coding and testing.[codelabs.developers.google](#)

Before any `sudo`/driver/toolkit install or big downloads, it must pause and ask for approval (Antigravity supports granular terminal execution + review policies).[codelabs.developers.google](#)

text

You are Antigravity's agent operating in my local workspace on Pop!\_OS (Ubuntu-based), with RTX 4070 12GB and 64GB RAM.

MISSION (single end-to-end demo):

Build "TimeCapsule": a local-first, GPU-accelerated system that lets me point at a folder of videos (mp4/mkv/webm), automatically:

- 1) extracts audio + transcribes with Whisper on GPU when available,
- 2) extracts representative keyframes,
- 3) computes CLIP embeddings for keyframes and indexes them (FAISS preferred; CPU fallback OK),
- 4) stores metadata + transcript segments in SQLite,
- 5) provides a fast search UI: type a natural-language query → returns ranked moments with thumbnails + timestamps + transcript hits; clicking a result plays the local video at the right timestamp,
- 6) produces a benchmark report (latency + throughput + GPU/CPU detection) and a reproducible "make demo" command,
- 7) self-verifies via automated runs + a browser-verified walkthrough artifact.

HARD SAFETY / SCOPE RULES:

- Work ONLY inside a new folder: ~/antigravity\_demos/timecapsule (create it).
- Never delete anything outside that folder.
- Do NOT run sudo, change drivers, or install system packages without stopping and asking me explicitly.
- If downloading sample media, keep it small (<= 50MB) and permissively licensed; ask before downloading.
- Prefer Python 3.11 venv (avoid Python 3.12 if it risks GPU wheels); if 3.11 isn't present, ask before installing.
- Produce clear artifacts: Task Plan, Implementation Plan, Code diffs, Test/Benchmark report, Walkthrough with browser recording.

TECH CHOICES (optimize for my RTX 4070 12GB):

- Python project using a modern, reproducible setup (uv or pip-tools is fine).
- Transcription: faster-whisper (preferred) or openai-whisper; choose the option that reliably uses CUDA with torch.
- Vision embeddings: OpenCLIP (open\_clip\_torch) with a sane model (e.g., ViT-B/32) and batched GPU inference.
- Vector search: FAISS (faiss-cpu acceptable; use faiss-gpu only if it's straightforward without system-wide CUDA installs).
- Storage: SQLite for metadata + transcript segments; store embeddings in FAISS index + mapping table.
- Web UI: FastAPI + a minimal single-page UI (plain HTML/JS) that can:
  - ingest progress view,
  - search box,
  - results list with thumbnail + timestamp + confidence,
  - click-to-play video at timestamp (use media fragments #t=seconds or equivalent).

DELIVERABLES:

A) Repo structure:

```
timecapsule/
    README.md
    pyproject.toml (or requirements + lock strategy)
    src/timecapsule/ (package)
    tests/
    scripts/
```

B) CLI commands:

```
- timecapsule doctor      # prints GPU/torch/cuda status +
ffmpeg availability
- timecapsule ingest <video_folder> --workers N
- timecapsule search "query text" --topk 10
- timecapsule serve --host 127.0.0.1 --port 8000
```

- C) "Doctor" must print: nvidia-smi summary (if available), torch version, torch.cuda.is\_available, device name, and chosen compute backend.
- D) Automated tests: at least smoke tests for DB schema, indexing pipeline (with a tiny generated synthetic video OR a stub), and search ranking shape.
- E) Benchmark: produce benchmark.md with:
  - transcription speed (sec audio processed / sec wall time),
  - embedding throughput (frames/sec),
  - end-to-end ingest time for sample,
  - search latency.
- F) Walkthrough: Start server, open browser, run one ingest (sample or user-provided), run 2 example searches, click a result to play video at timestamp; capture screenshot + browser recording artifact.

#### EXECUTION PLAN REQUIREMENTS:

- 1) First, create the Task Plan + Implementation Plan artifacts.
- 2) Then, implement incrementally with frequent runnable checkpoints:
  - minimal "doctor"
  - minimal ingest that produces transcript
  - add keyframes
  - add embeddings + FAISS
  - add search API
  - add UI
  - tests + benchmarks
- 3) Verification must be real:
  - run unit tests,
  - run doctor,
  - run ingest on a small sample,
  - run search,
  - launch serve and verify via browser interaction.

#### INSTALLATION GUIDANCE:

- Use the official PyTorch “get started” page to select a Linux + pip command that matches my CUDA-capable system; prefer a stable CUDA wheel. If you need to choose between CUDA versions, pick the best-supported current one and justify briefly in the implementation plan.
- If torch CUDA install fails or CUDA isn’t detected, fall back to CPU mode automatically but still complete the demo.

NOW DO IT.

Start by creating ~/antigravity\_demos/timecapsule, initializing the project, and generating the Task Plan + Implementation Plan for my approval.

## Why this impresses

It forces the agent to coordinate multi-surface work (code + terminal installs + browser-based verification) and to “prove” success with artifacts like screenshots/recordings instead of just claiming it works.[developers.googleblog+1](#)  
It also stress-tests real GPU acceleration decisions (CUDA wheels, batching, memory limits on a 12GB card) while staying reproducible and safe via explicit review gates for risky commands.[pytorch+1](#)

1. <https://developers.googleblog.com/build-with-google-antigravity-our-new-agentic-development-platform/>
2. <https://codelabs.developers.google.com/getting-started-google-antigravity>
3. <https://pytorch.org/get-started/locally/>
4. <https://journal-laaroiba.com/ojs/index.php/edu/article/view/5879>
5. <https://rsisinternational.org/journals/ijriss/article.php?id=1104>
6. <https://ejurnal.akacom.ac.id/index.php/JuTI/article/view/1405>
7. <https://www.semanticscholar.org/paper/588a334e4951c5914500652ea73ad39a44f78dd0>
8. <https://www.semanticscholar.org/paper/941fe6b54de89c97d2d5b50a161b3371a3c390ac>
9. <https://www.semanticscholar.org/paper/054dc45acd7454ee0c22607c7ff2561ae112b8ef>
10. <https://www.semanticscholar.org/paper/9477c2cb8868c7bb79490e48d49c8936c4098d52>

11. <https://arxiv.org/pdf/2502.01821.pdf>
12. <https://arxiv.org/pdf/2501.07531.pdf>
13. <https://arxiv.org/pdf/2502.01789.pdf>
14. <http://arxiv.org/pdf/2412.09745.pdf>
15. <https://arxiv.org/html/2410.08164>
16. <http://arxiv.org/pdf/2411.04468v1.pdf>
17. <http://arxiv.org/pdf/2407.15734.pdf>
18. <http://arxiv.org/pdf/2410.14393.pdf>
19. <https://antigravity.google>
20. <https://www.youtube.com/watch?v=6H5gQXzN6vQ>
21. <https://www.codecademy.com/article/how-to-set-up-and-use-google-antigravity>
22. <https://www.datacamp.com/tutorial/google-antigravity-tutorial>
23. <https://github.com/comfyanonymous/ComfyUI/discussions/8034>
24. <https://www.youtube.com/watch?v=uZFOhkORVfk>
25. <https://antigravity.im/documentation>
26. <https://dev.to/tallesl/setting-up-nvidia-stuff-on-popos-3o17>
27. <https://www.youtube.com/watch?v=KTTbSgxE9AY&start=1195>
28. <https://antigravity.google/docs/home>
29. [https://www.reddit.com/r/ChatGPTCoding/comments/1p35bdl/i\\_tried\\_gogles\\_ne\\_w\\_antigravity\\_ide\\_so\\_you\\_dont/](https://www.reddit.com/r/ChatGPTCoding/comments/1p35bdl/i_tried_gogles_ne_w_antigravity_ide_so_you_dont/)
30. <https://antigravity.google/docs>
31. [https://www.youtube.com/watch?v=M60\\_J-jtn0](https://www.youtube.com/watch?v=M60_J-jtn0)
32. <https://www.theverge.com/news/822833/google-antigravity-ide-coding-agent-gemini-3-pro>
33. <https://www.youtube.com/watch?v=2SFbqQvsFGc>
34. [https://www.reddit.com/r/pop\\_os/comments/15fjklg/how\\_to\\_get\\_specific\\_versions\\_of\\_cuda\\_and\\_pytorch/](https://www.reddit.com/r/pop_os/comments/15fjklg/how_to_get_specific_versions_of_cuda_and_pytorch/)
35. <https://blog.google/technology/developers/gemini-3-developers/>

## Gemini 3 pro

This request targets **Google Antigravity**, the agentic IDE released in November 2025 alongside Gemini 3.

The "edge case" capability most users miss is **Autonomous Closed-Loop Optimization**. Humans typically use agents to "write code once." However, Antigravity's

architecture (Editor + Terminal + Browser + Manager) allows an agent to **write, run, profile, and refactor its own code** in a loop without human intervention.

Given your **RTX 4070** and **Pop!\_OS** environment, we will force the agent to build a **WebGPU-accelerated Quantum System Visualizer** that doesn't just "work," but *mathematically perfects itself* to your hardware limits.

## The Concept: "The E8 Entropy Garden"

This prompt instructs Antigravity to build a local dashboard that visualizes your real-time system entropy (RAM/VRAM flux) mapped onto an **E8 Lattice projection**. The "impressive" part is the **Self-Correction Directive**: the agent must use its internal browser to run performance audits on the shader code and rewrite it until it hits a locked 144Hz on your specific GPU.

## The Robust Prompt

Copy and paste this directly into the Antigravity **Manager View** (or the main Agent chat):

---

**Role:** Senior Graphics Engineer & Systems Architect

**Target System:** Linux (Pop!\_OS), RTX 4070 (12GB VRAM), 64GB RAM, Vulkan/WebGPU support.

### Objective:

Architect and deploy a high-performance local telemetry dashboard named "**Entropy\_Garden**".

1. **Backend:** Create a Python (FastAPI) service that streams real-time system entropy (interrupts, context switches, VRAM tensor usage) via WebSockets.
2. **Frontend:** Build a Three.js/WebGPU client that visualizes this data as a rotating 4D **E8 Lattice projection**. The lattice vertices must pulse in sync with the live system entropy.

### The Edge Case (Autonomous Optimization Loop):

Do not just write the code and stop. You must execute a **Self-Correction Workflow**:

1. **Implement** the initial WebGPU vertex shaders for the E8 projection.
2. **Launch** the local server and open the visualization in the **Inbuilt Browser**.
3. **Inject** a performance monitoring script to measure `frame_time` variance over 10 seconds.
4. **Refactor Loop:** If the frame rate is below **144 FPS** or memory leaks are detected, you must **rewrite your own shader code** (optimizing buffer geometry or simplified lighting) and re-test.
5. **Constraint:** You are allowed to consume up to 8GB of system RAM for caching pre-calculated geometry, but do not exceed 4GB VRAM.

#### **Final Deliverable:**

- A functional "One-Click" run script (`start_garden.sh`).
- An **Artifact** named `Optimization_Log.md` detailing every shader refactor you performed to achieve the 144Hz lock on this hardware.
- A live browser preview of the stabilized system.

#### **Execute.**

---

## **Why this breaks the mold**

1. **Hardware Awareness:** It forces the agent to acknowledge your 4070 and 64GB RAM, using the RAM as a buffer (pre-calc geometry) to spare the GPU for rendering—a specific optimization strategy usually reserved for senior human engineers.
2. **The "Browser" Hack:** Most users use the built-in browser for "checking if it looks right." This prompt forces the agent to use the browser as a **benchmarking tool**, effectively turning the IDE into a testing lab.

3. **Visualizing the Invisible:** It bridges your interest in **Quantum Geometry (E8)** with your system's heartbeat, turning your Linux machine's background noise into art.

1. <https://journal-laaroiba.com/ojs/index.php/edu/article/view/5879>
2. <https://rsisinternational.org/journals/ijriss/article.php?id=1104>
3. <https://ejournal.akakom.ac.id/index.php/JuTI/article/view/1405>
4. <https://www.semanticscholar.org/paper/588a334e4951c5914500652ea73ad39a44f78dd0>
5. <https://www.semanticscholar.org/paper/941fe6b54de89c97d2d5b50a161b3371a3c390ac>
6. <https://www.semanticscholar.org/paper/054dc45acd7454ee0c22607c7ff2561ae112b8ef>
7. <https://www.semanticscholar.org/paper/9477c2cb8868c7bb79490e48d49c8936c4098d52>
8. <https://arxiv.org/pdf/2502.01821.pdf>
9. <https://arxiv.org/pdf/2501.07531.pdf>
10. <https://arxiv.org/pdf/2502.01789.pdf>
11. <http://arxiv.org/pdf/2412.09745.pdf>
12. <https://arxiv.org/html/2410.08164>
13. <http://arxiv.org/pdf/2411.04468v1.pdf>
14. <http://arxiv.org/pdf/2407.15734.pdf>
15. <http://arxiv.org/pdf/2410.14393.pdf>
16. <https://antigravity.google>
17. <https://developers.googleblog.com/build-with-google-antigravity-our-new-agentic-development-platform/>
18. <https://codelabs.developers.google.com/getting-started-google-antigravity>
19. <https://www.youtube.com/watch?v=6H5gQXzN6vQ>
20. <https://www.youtube.com/watch?v=uzFOhkORVfk>
21. <https://www.youtube.com/watch?v=HCeyLJP60LQ>
22. <https://chromeunboxed.com/google-launches-antigravity-a-new-ai-powered-ide-for-autonomous-agentic-coding/>
23. <https://www.youtube.com/watch?v=0KETv-jr1T8>
24. <https://ainativedev.io/news/gemini-3-meets-antigravity-googles-next-step-in-agentic-development>
25. <https://www.youtube.com/watch?v=KTTbSgxE9AY&start=1195>
26. [https://en.wikipedia.org/wiki/Google\\_Antigravity](https://en.wikipedia.org/wiki/Google_Antigravity)
27. <https://blog.google/technology/developers/gemini-3-developers/>
28. <https://dev.to/bst53/googles-project-idx-the-ultimate-tool-for-modern-development-1glo>

29. [https://www.linkedin.com/posts/opswork\\_antigravity-projectidx-jules-activity-739609653507444737-FvbU](https://www.linkedin.com/posts/opswork_antigravity-projectidx-jules-activity-739609653507444737-FvbU)
30. [https://www.reddit.com/r/ChatGPTCoding/comments/1p35bdl/i\\_tried\\_gogles\\_ne\\_w\\_antigravity\\_ide\\_so\\_you\\_dont/](https://www.reddit.com/r/ChatGPTCoding/comments/1p35bdl/i_tried_gogles_ne_w_antigravity_ide_so_you_dont/)
31. [https://www.reddit.com/r/google/comments/1p10ev8/has\\_anyone\\_tried\\_antigravit\\_y\\_by\\_google\\_thoughts/](https://www.reddit.com/r/google/comments/1p10ev8/has_anyone_tried_antigravit_y_by_google_thoughts/)
32. <https://cloud.google.com/blog/topics/developers-practitioners/agent-factory-recap-building-with-gemini-3-ai-studio-antigravity-and-nano-banana/>
33. <https://www.builder.io/blog/agentic-ide>
34. <https://www.youtube.com/watch?v=kcrf6dbrlY4>
35. <https://www.codecademy.com/article/how-to-set-up-and-use-google-antigravity>

## Grok 4.1

Google Antigravity, released by Google in November 2025, excels at agentic workflows with GPU-accelerated tasks on Linux like PopOS. This single prompt demonstrates an edge-case capability: autonomous, multi-agent orchestration of CUDA setup, on-the-fly fine-tuning of a diffusion model, quantum geometry simulation, real-time WebGL rendering with RTX ray-tracing shaders, performance benchmarking, and artifact generation—all verified end-to-end without human intervention.[antigravity+3](#)

### Showcase Prompt for Antigravity

text

You control the full development environment on Pop!\_OS with RTX 4070 (12GB VRAM), 64GB RAM, NVIDIA drivers pre-installed. Dispatch 4 parallel agents to autonomously complete this edge-case project in under 30 minutes, producing verifiable Artifacts (code repo, interactive demo, benchmark charts, fly-through video, technical report PDF). Theme: "Quantum Antigravity Navigator" – a real-time WebGL game visualizing antigravity flight through an E8 lattice with quantum superposition effects.

Agent 1 (Env + GPU Setup): Verify CUDA/PyTorch/CuQuantum install (nvidia-smi, torch.cuda.is\_available()). If missing: apt/pip install nvidia-cuda-toolkit, torch==2.5.0+cu121, diffusers[torch], pennylane-cuda, three.js via npm. Benchmark GPU: train tiny test diffusion model (1 epoch, batch=4) on 100 synthetic E8/icosahedral patterns. Artifact: GPU diagnostics report + screenshot.

Agent 2 (ML Assets): Fine-tune Stable Diffusion XL Turbo LoRA (low-rank, 512x512) on-the-fly with 200 generated E8 lattice fractals (use geometry libs like sympy/Blender API export). Output 50 textures/videos of quantum superposition (superimposed wavefunctions collapsing under 'antigravity' perturbation). Compress outputs with custom GPU-accelerated wavelet algo (cuFFT). Artifact: Model weights zip, sample gallery PNGs/MP4s, compression benchmarks CSV.

Agent 3 (Core Sim + Game): Build Three.js WebGL app (Vite bundler). Simulate E8 lattice (120 vertices, golden ratio icosa/dodeca projections) as interactive 3D world. Physics: custom GLSL shaders for antigravity propulsion (inverse-square gravity wells + quantum tunneling jumps via Pennylane circuit eval on GPU). RTX effects: ray-marched volumetric clouds, god rays. Controls: WASD flight, mouse look. Collisions spawn superposition visuals from Agent 2. Deploy local server (<http://localhost:3000>). Artifact: /dist folder, live URL recording.

Agent 4 (Verify + Report): Orchestrate integration: load Agent2 textures into game, run browser agent for 5-min playtest (record fly-through MP4 @60FPS). Benchmark: FPS on RTX4070 (target 120+), VRAM usage (<10GB), compression ratios. Self-debug: fix any errors via iterative tool calls (terminal, browser inspect). Generate Markdown report with embeds (charts via Plotly, code

diffs), export to PDF. Push to new GitHub repo (use token if prompted).

Merge outputs into master workspace. Request human approval only for final video preview. Prioritize safety: sandbox executions, no rm -rf. Use Gemini 3 Pro turbo mode.

## Why This Edge Case Impresses

- **GPU-Heavy Autonomy:** Humans rarely expect an IDE to auto-setup/train/deploy ML+raytracing in parallel on specific hardware without config.[dev+1](#)
- **Quantum Tie-In:** Leverages user's E8/icosahedron interests with real GPU quantum sim (CuQuantum), unseen in typical IDE demos.[youtube](#)
- **Multi-Agent Scale:** Parallel agents handle env/ML/sim/verify, producing Artifacts for trust—beyond linear chatbots.[reddit+1](#)
- **End-to-End Verification:** Browser recordings, FPS benchmarks, compression prove execution, showcasing "proof-of-work" agents.[geeky-gadgets](#)

Copy-paste into Antigravity's Agent Manager for liftoff.[antigravity](#)

1. <https://antigravity.google>
2. <https://developers.googleblog.com/build-with-google-antigravity-our-new-agentic-development-platform/>
3. <https://codelabs.developers.google.com/getting-started-google-antigravity>
4. [https://en.wikipedia.org/wiki/Google\\_Antigravity](https://en.wikipedia.org/wiki/Google_Antigravity)
5. <https://dev.to/tallesl/setting-up-nvidia-stuff-on-popos-3o17>
6. <https://www.infoworld.com/article/4096113/a-first-look-at-googles-new-antigravity-ide.html>
7. <https://www.youtube.com/watch?v=kQr0Tg6SHbw>
8. [https://www.reddit.com/r/ChatGPTCoding/comments/1p35bdl/i\\_tried\\_googles\\_ne\\_w\\_antigravity\\_ide\\_so\\_you\\_dont/](https://www.reddit.com/r/ChatGPTCoding/comments/1p35bdl/i_tried_googles_ne_w_antigravity_ide_so_you_dont/)
9. [https://www.reddit.com/r/GeminiAI/comments/1p7hioy/what\\_is\\_google\\_antigravity\\_full\\_tutorial\\_first/](https://www.reddit.com/r/GeminiAI/comments/1p7hioy/what_is_google_antigravity_full_tutorial_first/)
10. <https://www.geeky-gadgets.com/google-antigravity-features-guide/>
11. <https://antigravity.google/blog/introducing-google-antigravity>

12. <https://journal-laaroiba.com/ojs/index.php/edu/article/view/5879>
13. <https://rsisinternational.org/journals/ijriss/article.php?id=1104>
14. <https://ejournal.akakom.ac.id/index.php/JuTI/article/view/1405>
15. <https://www.semanticscholar.org/paper/588a334e4951c5914500652ea73ad39a44f78dd0>
16. <https://www.semanticscholar.org/paper/941fe6b54de89c97d2d5b50a161b3371a3c390ac>
17. <https://www.semanticscholar.org/paper/054dc45acd7454ee0c22607c7ff2561ae112b8ef>
18. <https://www.semanticscholar.org/paper/9477c2cb8868c7bb79490e48d49c8936c4098d52>
19. <https://arxiv.org/pdf/2502.01821.pdf>
20. <https://arxiv.org/html/2502.16069v1>
21. <https://arxiv.org/html/2410.08164>
22. <http://arxiv.org/pdf/2407.15734.pdf>
23. <https://arxiv.org/pdf/2501.07531.pdf>
24. <https://arxiv.org/pdf/2502.13965.pdf>
25. <https://arxiv.org/pdf/2502.01789.pdf>
26. <http://arxiv.org/pdf/2504.04058.pdf>
27. <https://www.youtube.com/watch?v=6H5gQXzN6vQ>
28. [https://www.reddit.com/r/pop\\_os/comments/15j7v5r/how\\_to\\_make\\_pop\\_os\\_use\\_integrated\\_gpu\\_instead\\_of/](https://www.reddit.com/r/pop_os/comments/15j7v5r/how_to_make_pop_os_use_integrated_gpu_instead_of/)
29. [https://www.reddit.com/r/google/comments/1p8nlv9/google\\_drops\\_antigravity\\_idea\\_ai\\_agents\\_running/](https://www.reddit.com/r/google/comments/1p8nlv9/google_drops_antigravity_idea_ai_agents_running/)
30. <https://www.youtube.com/watch?v=fknLn7KpX4g>
31. [https://www.reddit.com/r/Bard/comments/1p1wgw6/has\\_anyone\\_tried\\_google\\_s\\_new\\_antigravity\\_ide\\_yet/](https://www.reddit.com/r/Bard/comments/1p1wgw6/has_anyone_tried_google_s_new_antigravity_ide_yet/)
32. [https://www.reddit.com/r/LLMDevs/comments/1p2no8d/review\\_antigravity\\_gool\\_es\\_new\\_ide/](https://www.reddit.com/r/LLMDevs/comments/1p2no8d/review_antigravity_gool_es_new_ide/)
33. <https://github.com/pop-os/pop/issues/2508>
34. <https://www.theverge.com/news/822833/google-antigravity-ide-coding-agent-gemini-3-pro>
35. [https://www.reddit.com/r/pop\\_os/comments/1d3fy1t/is\\_pop\\_os\\_stable\\_for\\_nvidia\\_gpus/](https://www.reddit.com/r/pop_os/comments/1d3fy1t/is_pop_os_stable_for_nvidia_gpus/)
36. <https://www.youtube.com/watch?v=HBEqGHCd8hk>

# Kimi K2 Thinking

Based on your hardware specs and the capabilities described for Google Antigravity, here's an edge-case prompt that showcases multi-agent orchestration with hardware-aware optimization:

text

Create a quantum-inspired compression algorithm research environment that:

1. Agent-1 (Architecture Designer): Designs a novel hybrid compression scheme combining matrix product states (quantum tensor networks) with adaptive Huffman coding, specifically optimized for RTX 4070's 12GB VRAM and 192-bit memory bus. Generate a detailed architecture diagram as Artifact-1 showing memory layout patterns that maximize GDDR6X bandwidth utilization.
2. Agent-2 (PopOS! Integration Specialist): Implements the algorithm in Rust with CUDA kernels using your RTX 4070, creating a PopOS!-specific build script that leverages your 64GB system RAM for preprocessing while keeping hot paths in VRAM. Generate a terminal session Artifact-2 demonstrating compilation with hardware-specific flags (`-march=native`, CUDA compute capability 8.9) and runtime benchmark showing 85%+ VRAM utilization.
3. Agent-3 (Verification Agent): Creates a self-validating test suite that automatically generates edge cases (borderline tensor ranks, memory pressure scenarios), runs them in Antigravity's integrated browser environment to visualize compression ratios vs. processing time, and uses the Canvas feature to capture screenshots proving the algorithm outperforms zlib by 3.2x on quantum circuit data while staying within your GPU's power envelope.
4. Agent-4 (Documentation Agent): Synthesizes all findings into a publishable research note with inline citations, performance analysis correlating tensor network complexity with RTX 4070's

46 RT cores for parallel evaluation, and a limitations section addressing Ada Lovelace architecture constraints.

Execute all agents in parallel, with Agent-4 monitoring the others and generating a live Knowledge Item tracking emergent bottlenecks. Final deliverable: A VS Code-compatible project with GitHub-ready repository structure, PopOS! installation script, and an Artifact showing the complete agent execution timeline with self-correction loops where agents detected and fixed 3+ integration issues autonomously.

This prompt tests Antigravity's ability to orchestrate specialized agents that understand hardware constraints, perform autonomous verification, and generate research-grade output while leveraging your specific system configuration.[codecademy+2](#)

1. <https://www.codecademy.com/article/how-to-set-up-and-use-google-antigravity>
2. <https://blog.promptlayer.com/google-antigravity-first-impressions-of-the-agent-first-ide/>
3. <https://www.remio.ai/post/google-antigravity-technical-review-the-first-true-agent-ide-powered-by-gemini-3-pro>
4. <https://journal.trunojoyo.ac.id/edutic/article/view/3229>
5. <https://www.radv.ru/jour/article/view/737>
6. <https://jisem-journal.com/index.php/journal/article/view/8895>
7. <https://www.nature.com/articles/s41746-022-00560-6>
8. <https://ijitce.org/index.php/ijitce/article/view/1288>
9. <https://journals.sagepub.com/doi/10.1177/1934578X251346010>
10. <https://www.eurekaselect.com/243546/article>
11. <https://dl.acm.org/doi/10.1145/3017680.3022382>
12. <https://www.mdpi.com/1424-8220/23/4/1752>
13. <https://ieeexplore.ieee.org/document/10387629/>
14. <https://arxiv.org/pdf/2502.09809.pdf>
15. <http://arxiv.org/pdf/2504.04058.pdf>
16. <https://arxiv.org/pdf/2503.00237.pdf>
17. <https://arxiv.org/pdf/2502.01821.pdf>
18. <http://arxiv.org/pdf/2407.15734.pdf>
19. <https://arxiv.org/pdf/2502.17443.pdf>
20. <http://arxiv.org/pdf/2410.22457.pdf>
21. <http://arxiv.org/pdf/2407.13032.pdf>

22. <https://developers.googleblog.com/build-with-google-antigravity-our-new-agentic-development-platform/>
23. <https://antigravity.google/blog/introducing-google-antigravity>
24. <https://blog.stablediscussion.com/p/googles-antigravity-ide-still-too>
25. <https://www.kitguru.net/components/graphic-cards/matthew-wilson/nvidia-confirmed-rtx-4070-ti-through-omniverse-system-requirements/>
26. [https://www.reddit.com/r/GeminiAI/comments/1p7hioy/what\\_is\\_google\\_antigravity\\_full\\_tutorial\\_first/](https://www.reddit.com/r/GeminiAI/comments/1p7hioy/what_is_google_antigravity_full_tutorial_first/)
27. <https://www.youtube.com/watch?v=2SFbqQvsFGc>
28. [https://www.reddit.com/r/nvidia/comments/14jmzbi/can\\_my\\_system\\_handle\\_a\\_rtx\\_4070/](https://www.reddit.com/r/nvidia/comments/14jmzbi/can_my_system_handle_a_rtx_4070/)
29. <https://www.youtube.com/watch?v=wbPpvjcAHew>
30. <https://codelabs.developers.google.com/getting-started-google-antigravity>
31. <https://www.techradar.com/computing/gaming-pcs/ubisoft-reveals-assassins-creed-shadows-pc-requirements-and-just-as-i-thought-youre-safe-with-your-rtx-4000-series-gpu>
32. [https://www.reddit.com/r/LLMDevs/comments/1p2no8d/review\\_antigravity\\_goolges\\_new\\_ide/](https://www.reddit.com/r/LLMDevs/comments/1p2no8d/review_antigravity_goolges_new_ide/)
33. [https://www.reddit.com/r/ChatGPTCoding/comments/1p35bdl/i\\_tried\\_googles\\_new\\_antigravity\\_ide\\_so\\_you\\_dont/](https://www.reddit.com/r/ChatGPTCoding/comments/1p35bdl/i_tried_googles_new_antigravity_ide_so_you_dont/)
34. <https://groovycomputers.ca/blogs/resources/rtx-4070-specs>
35. <https://blog.google/technology/developers/gemini-3-developers/>
36. <https://cybernews.com/ai-tools/google-antigravity-review/>
37. <https://docs.nvidia.com/nim/visual-genai/latest/support-matrix.html>
38. <https://www.youtube.com/watch?v=uzFOhkORVfk>
39. [https://astesj.com/?download\\_id=32378&smd\\_process\\_download=1](https://astesj.com/?download_id=32378&smd_process_download=1)
40. <http://arxiv.org/pdf/2305.13380.pdf>
41. <https://academic.oup.com/mnras/article-pdf/506/2/2871/39271725/stab1855.pdf>
42. <https://arxiv.org/html/2406.11669v1>
43. <https://www.aanda.org/articles/aa/pdf/2022/09/aa43941-22.pdf>
44. <https://arxiv.org/pdf/2206.14220.pdf>
45. <https://academic.oup.com/mnras/advance-article-pdf/doi/10.1093/mnras/stae922/57123158/stae922.pdf>
46. <https://arxiv.org/pdf/2412.13239.pdf>
47. [https://en.wikipedia.org/wiki/Google\\_Antigravity](https://en.wikipedia.org/wiki/Google_Antigravity)
48. [https://www.reddit.com/r/GeminiAI/comments/1p0milb/google\\_antigravity\\_is\\_officially\\_here\\_a\\_new/](https://www.reddit.com/r/GeminiAI/comments/1p0milb/google_antigravity_is_officially_here_a_new/)
49. <https://www.youtube.com/watch?v=WDK9FlqQtko>
50. <https://dev.to/sunny7899/the-agentic-leap-key-announcements-and-demos-from-the-google-io-2025-developer-keynote-51fo>

51. [https://www.linkedin.com/posts/kevinhou22\\_this-week-our-team-launched-google-antigravity-activity-7397348046952599552-IBYY](https://www.linkedin.com/posts/kevinhou22_this-week-our-team-launched-google-antigravity-activity-7397348046952599552-IBYY)
52. [https://cloud.google.com/blog/products/ai-machine-learning/what-google-cloud-a nnounced-in-ai-this-month](https://cloud.google.com/blog/products/ai-machine-learning/what-google-cloud-announced-in-ai-this-month)
53. <https://www.facebook.com/groups/332087790866175/posts/2098924300849173/>
54. <https://www.infoworld.com/article/4096113/a-first-look-at-googles-new-antigravity-ide.html>
55. [https://www.reddit.com/r/GeminiAI/comments/1pjv90t/beyond\\_the\\_hype\\_what\\_i\\_earned\\_analyzing\\_gogoles/](https://www.reddit.com/r/GeminiAI/comments/1pjv90t/beyond_the_hype_what_i_earned_analyzing_gogoles/)
56. [https://www.reddit.com/r/google/comments/1p10ev8/has\\_anyone\\_tried\\_antigravity\\_by\\_google\\_thoughts/](https://www.reddit.com/r/google/comments/1p10ev8/has_anyone_tried_antigravity_by_google_thoughts/)
57. <https://blog.google/products/gemini/gemini-3/>
58. <https://antigravity.google>
59. <https://www.youtube.com/watch?v=U1VS2j7CYD4>
60. [https://www.youtube.com/watch?v=O-tHvdj\\_Y3c](https://www.youtube.com/watch?v=O-tHvdj_Y3c)
61. [https://www.tiktok.com/@sabrina\\_ramonov/video/7580916996649291039](https://www.tiktok.com/@sabrina_ramonov/video/7580916996649291039)

## Claude Sonnet 4.5 Reasoning

Based on the search results, ANTIGRAVITY is Google's impressive new agentic IDE with multi-agent orchestration, terminal control, and browser automation capabilities. Here's an edge-case prompt that showcases capabilities most humans wouldn't realize are possible in a single execution:[developers.googleblog+1](#)

---

### ANTIGRAVITY Mega-Prompt: GPU-Accelerated Compression Algorithm Evolution Lab

text

Create a self-optimizing compression algorithm research environment that:

\*\*MULTI-AGENT ORCHESTRATION (4 parallel agents):\*\*

Agent 1 - "CUDA Engineer":

- Detect RTX 4070 specs using nvidia-smi

- Install/verify CUDA toolkit on Pop!\_OS
- Write 3 different GPU-accelerated compression kernels in CUDA C++:
  - \* Arithmetic coding with parallel entropy estimation
  - \* GPU-optimized LZ77 with sliding window parallelization
  - \* Novel approach using E8 lattice geometry for pattern recognition
- Compile with nvcc, handle all CUDA compute capability 8.9 optimizations

Agent 2 - "Benchmark Orchestrator":

- Generate diverse test datasets (text, binary, structured, random - 10MB to 500MB)
- Create Python harness that measures: compression ratio, GPU memory usage, throughput (GB/s), power consumption via nvidia-smi
- Execute all 3 algorithms across all datasets, log results to SQLite database
- Run thermal monitoring to ensure we stay under 75°C

Agent 3 - "Visualization Engineer":

- Build real-time interactive dashboard (HTML/JavaScript/D3.js/Chart.js)
- Display live GPU utilization, compression ratios, speed comparisons
- Create animated heatmaps showing algorithm performance across dataset types
- Generate flame graphs of CUDA kernel execution
- Launch local web server, open in embedded browser

Agent 4 - "Research Analyst":

- Analyze benchmark results using statistical methods
- Identify which algorithm wins for which data patterns
- Generate LaTeX-formatted research paper with:
  - \* Abstract, methodology, results with tables/graphs

- \* Mathematical formulations of each algorithm
- \* Performance analysis with Big-O complexity
- Export as PDF with embedded charts

**\*\*BROWSER AUTOMATION:\*\***

- Agent takes screenshots of live dashboard at 5-second intervals
- Validates visualization is rendering correctly
- Records screen capture of the full benchmark execution
- Tests dashboard interactivity (hover, zoom, filter controls)

**\*\*ARTIFACTS TO GENERATE:\*\***

- Implementation plan with dependency graph
- 3 complete CUDA source files with comments
- Benchmark results database + CSV exports
- Interactive HTML dashboard
- Research paper PDF
- Screen recording of execution
- Performance comparison table
- Shell scripts for reproducible runs

**\*\*SYSTEM INTEGRATION:\*\***

- Use all 12GB VRAM efficiently
- Leverage 64GB RAM for large dataset staging
- Monitor with nvtop in separate terminal
- Create systemd service for auto-benchmarking
- Git repo with all code, results, and documentation

**\*\*SUCCESS CRITERIA:\*\***

- All 3 algorithms compile and execute on RTX 4070
- Dashboard renders live metrics within 30 seconds
- Complete research paper generated with real data
- Achieve >10 GB/s throughput on at least one algorithm
- Zero memory leaks or CUDA errors

Deliver everything as a working prototype in ~/compression-lab/ with README containing reproduction steps.

---

## Why This Is Impressive:

### Edge-case capabilities showcased:

- **Multi-agent parallel execution** across completely different domains (systems programming, data science, web dev, research)[reddit+1](#)
- **GPU compute + browser automation** in one workflow[youtube+1](#)
- **Live visual verification** of running code via browser control[geeky-gadgets](#)
- **Real-time artifact generation** while executing compute-intensive tasks[codelabs.developers.google+1](#)
- **Full-stack system integration** from kernel code to PDF generation
- **Self-validation** through terminal monitoring + browser screenshots[skywork](#)

Most developers don't realize ANTIGRAVITY can orchestrate 4 specialized agents simultaneously, each with full terminal access to compile CUDA code, launch web servers, monitor GPU thermals, and generate research papers - all while the browser automation validates the visual output. This effectively gives you a "AI research team" executing a 4-8 hour workflow in one prompt.[developers.googleblog+1](#)

Your RTX 4070's 8.9 compute capability and 64GB RAM make this perfect for the task, and Pop!\_OS has excellent NVIDIA support out of the box.[developer.nvidia+3](#)

1. <https://developers.googleblog.com/build-with-google-antigravity-our-new-agentic-development-platform/>
2. [https://en.wikipedia.org/wiki/Google\\_Antigravity](https://en.wikipedia.org/wiki/Google_Antigravity)
3. [https://www.reddit.com/r/ChatGPTCoding/comments/1p35bdl/i\\_tried\\_gogles\\_ne\\_w\\_antigravity\\_ide\\_so\\_you\\_dont/](https://www.reddit.com/r/ChatGPTCoding/comments/1p35bdl/i_tried_gogles_ne_w_antigravity_ide_so_you_dont/)
4. <https://www.youtube.com/watch?v=kQr0Tg6SHbw>
5. [https://www.youtube.com/watch?v=iHioNBH\\_LPs](https://www.youtube.com/watch?v=iHioNBH_LPs)
6. <https://www.geeky-gadgets.com/google-antigravity-features-guide/>

7. <https://codelabs.developers.google.com/getting-started-google-antigravity>
8. <https://skywork.ai/blog/agent/antigravity-terminal-automation/>
9. <https://developer.nvidia.com/cuda-gpus>
10. [https://www.reddit.com/r/pytorch/comments/1kwb5fh/rtx\\_4070\\_cuda\\_version/](https://www.reddit.com/r/pytorch/comments/1kwb5fh/rtx_4070_cuda_version/)
11. <https://dev.to/tallesl/setting-up-nvidia-stuff-on-popos-3o17>
12. <http://arxiv.org/pdf/2407.15734.pdf>
13. <https://support.system76.com/articles/graphics-switch-pop/>
14. <https://journal-laaroiba.com/ojs/index.php/edu/article/view/5879>
15. <https://rsisinternational.org/journals/ijriss/article.php?id=1104>
16. <https://ejournal.akacom.ac.id/index.php/JuTI/article/view/1405>
17. <https://www.semanticscholar.org/paper/588a334e4951c5914500652ea73ad39a44f78dd0>
18. <https://www.semanticscholar.org/paper/941fe6b54de89c97d2d5b50a161b3371a3c390ac>
19. <https://www.semanticscholar.org/paper/054dc45acd7454ee0c22607c7ff2561ae112b8ef>
20. <https://www.semanticscholar.org/paper/9477c2cb8868c7bb79490e48d49c8936c4098d52>
21. <https://arxiv.org/pdf/2502.01821.pdf>
22. <https://arxiv.org/html/2502.16069v1>
23. <https://arxiv.org/html/2410.08164>
24. <https://arxiv.org/pdf/2501.07531.pdf>
25. <https://arxiv.org/pdf/2502.13965.pdf>
26. <https://arxiv.org/pdf/2502.01789.pdf>
27. <http://arxiv.org/pdf/2504.04058.pdf>
28. <https://antigravity.google>
29. <https://www.youtube.com/watch?v=6H5gQXzN6vQ>
30. <https://blog.promptlayer.com/google-antigravity-first-impressions-of-the-agent-first-ide/>
31. <https://ieeexplore.ieee.org/document/10099273/>
32. [https://aacrjournals.org/cancerres/article/84/6\\_Supplement/2337/739607/Abstract-2337-Accelerated-optical-genome-mapping](https://aacrjournals.org/cancerres/article/84/6_Supplement/2337/739607/Abstract-2337-Accelerated-optical-genome-mapping)
33. <https://www.frontiersin.org/articles/10.3389/fpls.2025.1687742/full>
34. <https://arxiv.org/pdf/1601.04640.pdf>
35. <http://arxiv.org/pdf/2412.09337.pdf>
36. <https://arxiv.org/pdf/2206.07896.pdf>
37. <https://arxiv.org/pdf/1202.4347.pdf>
38. <https://arxiv.org/pdf/2410.05686.pdf>
39. <https://arxiv.org/pdf/2309.05445.pdf>
40. <https://arxiv.org/pdf/0908.4362.pdf>

41. <https://arxiv.org/html/2405.17322v1>
42. [https://www.reddit.com/r/nvidia/comments/1kwb6hq/rtx\\_4070\\_cuda\\_version/](https://www.reddit.com/r/nvidia/comments/1kwb6hq/rtx_4070_cuda_version/)
43. <https://www.nvidia.com/en-us/geforce/graphics-cards/40-series/rtx-4070-family/>
44. [https://www.reddit.com/r/nvidia/comments/12bcpea/nvidia\\_geforce\\_rtx\\_4070\\_leak\\_confirms\\_5888\\_cuda/](https://www.reddit.com/r/nvidia/comments/12bcpea/nvidia_geforce_rtx_4070_leak_confirms_5888_cuda/)
45. [https://en.wikipedia.org/wiki/GeForce RTX\\_40\\_series](https://en.wikipedia.org/wiki/GeForce RTX_40_series)
46. <https://www.youtube.com/watch?v=1JMdmFhlI5E>