Who Holds the Dial

2026-05-31 / 00:18:21

> *“The capability's not usually the story anymore. Who holds the dial is.”*
>
> — from this episode's transcript

- Lenar Kess
- Damra Vol

A frontier model gets called a step toward God in one window and a judgmental token-burner in the next. We spend the morning on the gap between the marketing altitude and the desk, and find the same thread running through everything: every layer now has a control surface someone's reaching for.

- Dylan Field on Opus 4.8 calls it "a very strange model" — honesty up, curiosity down, personality judgmental — a reminder that a tuning dial has costs you can feel.
- scaling01 on DeepSWE says GPT-5.5 "score-, time- and token-mogged" Opus 4.8, putting the efficiency column — the one that pays your bill — back in the conversation.
- Ben Kunkle on Zed's Zeta 2 shows how a ten-second editing pause becomes a training label, and how a million frontier-model calls got replaced by a self-grading student model.
- Philipp Schmid (DeepMind) on the five assumptions that trip up senior engineers building agents — errors as inputs, evals not unit tests, and "build to delete."
- Komi-learn and a year on knowledge-graph memory share one missing thing: a controlled before-and-after proving the memory layer, not the model, made the

agent better.

- A Lancet correspondence finds 4,046 fabricated references across 2,810 published articles — model honesty rising while the literature's integrity falls.
- Quick hits: AMD's Lisa Su vs Nvidia's Jensen Huang on China, IBM's Sovereign Core, and a court ordering Circle to freeze a $12.6M contract.

SEGMENTS

Transcript

1. Lenar Kess 00:00:00

   Here's a small puzzle for a Sunday morning. You upgrade to the newest model from a frontier lab — the one everyone spent yesterday arguing about on the leaderboards. You expect it to feel smarter, maybe a little warmer. Instead, the thing turns judgmental. That's the word Dylan Field reached for. He runs Figma, he uses these models hard, and on Saturday he posted — quote — 'Opus 4.8 is a very strange model. Clearly Anthropic tried to improve honesty, which is commendable. However, the model's curiosity, already worse in 4.7, degraded further. Result is a judgmental personality.' And then the tweet trails off, so I only have him to that point. But that's a strange sentence to read about a model people are calling a step toward something godlike.

   - x.com
   - x.com
   - reddit.com
   - i.redd.it

- reddit.com
- x.com
- x.com
- youtube.com
- youtube.com
- reddit.com
- github.com
- forbes.com
- techmeme.com
- forbes.com
- techmeme.com

2. Damra Vol 00:00:47

   Judgmental is such a specific complaint. [pause] The question for me is whether that's a personality artifact or a capability one — because those two get talked about as if they're the same thing, and they're really not. You can have a model that's more truthful and less pleasant to work with at the same time. That might even be the trade Anthropic made on purpose.

3. Lenar Kess 00:01:07

   Right, and that's the fair read of Field's note. He's not saying it's dumber. He's saying they pushed on honesty — which is a real, deliberate dial — and curiosity fell out the other side. The model hedges less, second-guesses you more. Whether that's the price of the honesty tuning or just a regression, we don't have the internals to say.

4. Damra Vol 00:01:26

   And we should be clear — that's one expert's hands-on impression, not a measurement. Did anybody actually measure 4.8 against the field this weekend?

5. Lenar Kess 00:01:35

   Yeah, that's the second piece. A researcher who posts as scaling01 ran it on a coding benchmark called DeepSWE — real software-engineering tasks — and his summary was blunt. He said Opus 4.8 gets, his words, 'score-, time- and token-mogged by GPT-5.5' on that benchmark. Meaning GPT-5.5 scored higher, finished faster, and used fewer tokens to do it.

6. Damra Vol 00:02:03

   [tsk] Okay, but token-mogged on one benchmark by one person. Do we have the actual numbers, the chart?

7. Lenar Kess 00:02:10

Someone did post the DeepSWE results as an image over on the singularity subreddit, so the chart is out there. But I'm not going to read you figures I can't verify off a screenshot — I'd rather give you the shape than invent a decimal. The shape is: on this particular eval, the newest Anthropic model is not the efficiency winner. Which is interesting precisely because yesterday the whole conversation was about Opus 4.8's benchmark jump.

8. Damra Vol 00:02:35

   And it folds right into that token argument from this week. There was a post on the Anthropic subreddit — someone clearly upset — 'why are we celebrating burning more tokens like it's a flex.' Their line was, you're paying more to get more, and somehow that became a brag. If GPT-5.5 gets the same or better result for fewer tokens, then the efficiency column is the column that pays your bill.

9. Lenar Kess 00:02:59

   There's one more 4.8 item, and it's the kind of thing that flies around on a weekend, so let me be careful with it. Someone released a benchmark they're calling the Singularity Gate. It's pitched as a test of whether a frontier model can predict paradigm-breaking scientific discoveries published after its training cutoff. And the headline is that Opus 4.8 leads it.

10. Damra Vol 00:03:20

    [lip-smack] I mean — predict discoveries that haven't been made yet, scored by the person who built the benchmark and released it the same week the model launched. I'd put basically no weight on that until someone independent runs it. Predicting post-cutoff science is almost designed to reward a model that's good at sounding profound.

11. Lenar Kess 00:03:38

    And that's the tension I keep circling this morning. On one side you've got this benchmark framing the model as a near-oracle. On the other you've got Bill Gurley on the All-In podcast saying — quote — 'Anthropic is a mystery to me, I've never, ever seen' — and the host, Jason, calling it 'the ultimate level of narcissism and delusion of grandeur to think you can create God.' There's even someone on X asking Grok, straight up, in what ways an AI could become God. So the rhetoric is theological. And the hands-on report from a serious user is: it got judgmental, and a competitor used fewer tokens.

12. Damra Vol 00:04:17

    The gap between those two registers is the whole thing. The marketing altitude is deity. The desk-level altitude is a model with a personality regression that loses an efficiency race. Both are being

said about the same week. I'll trust the person who actually shipped code with it over the person theorizing about godhood on a podcast.

13. Lenar Kess 00:04:37

    So hold that — the desk beats the pulpit — because the next thing is somebody who built at the desk, in painful detail. This is from a talk by Ben Kunkle. He leads edit predictions at Zed, the editor, and he walked through how they trained Zeta 2 — the model that guesses your next code edit on every keystroke. What I love about it is that it's the actual machinery under a feature that feels like mind-reading. The model has to predict, in milliseconds, what you're about to change, accurately enough that accepting the suggestion beats ignoring it.

14. Damra Vol 00:05:08

    Every keystroke is a brutal latency budget. So where does the training data even come from? You can't have humans labeling 'here's the correct next edit' at that volume.

15. Lenar Kess 00:05:18

    Two pieces, and the second is the clever one. First, they distill from a frontier teacher model — a big model generates the right prediction, and the small model learns to imitate it. But the interesting part is what Kunkle calls settled data. The editor watches you work, and when you stop editing a region for ten seconds, it snapshots that final state of the code and treats it as ground truth — as in, that's probably what you meant the code to become.

16. Damra Vol 00:05:44

    [chuckle] So the ten-second pause is the label. The absence of you typing is the supervision signal. That's elegant and a little unnerving — it's mining your hesitation.

17. Lenar Kess 00:05:55

    And it's noisy, right, because maybe you came back and changed it again, or an agent edited the file underneath you. So they filter. They generate several teacher predictions per example and measure how close those land to the settled state using an n-gram edit-distance metric — Levenshtein, basically how many small changes it takes to get from one string to another. And here's the part I didn't expect: they don't keep the easiest examples. They keep the ones in the middle of the similarity range.

18. Damra Vol 00:06:24

    Because the easy ones the small model already knows. The middle band is where the novel patterns live — the stuff past the student model's training cutoff. You're deliberately harvesting the examples

that are hard but not garbage. That's a real piece of taste baked into the pipeline.

19. Lenar Kess 00:06:39

    Right. And the cost arc tells you how fast this moves. Kunkle said the initial filtering took up to a million frontier-model requests per hundred thousand examples. A million calls to a big model to clean one batch. Now they've swapped the frontier teacher for their own student checkpoints, run fifty times each, at — his phrase — negligible cost. So the expensive teacher was a temporary crutch they dropped the moment the student got good enough to grade its own work.

20. Damra Vol 00:07:07

    That's the loop everyone's chasing — the model gets good enough to generate its own training signal. And on the production side, did he say how they roll it out? Because a wrong edit prediction on every keystroke is maddening.

21. Lenar Kess 00:07:19

    They track acceptance rate, latency, and — this is the good one — diagnostic error counts before and after the prediction, plus a reversal ratio: how often you immediately undo what the model suggested. And they ramp on a traffic dashboard from fifteen percent up to full. So the eval isn't a leaderboard. It's 'did this make your next ten seconds better, or did you rip it out.'

22. Damra Vol 00:07:41

    And that's agentic coding stripped of the demo — not a model that writes your app, a model fighting for the right to fill in three characters without annoying you. The whole discipline lives in the filtering and the reversal ratio, not the keynote.

23. Lenar Kess 00:07:55

    This pairs perfectly. Philipp Schmid, an engineer at Google DeepMind working on Gemini agents, gave a talk on why senior engineers — the good ones — struggle to build AI agents. His claim is that the problem isn't talent. Five assumptions from normal software break the moment you build agents, and the better you are at the old way, the harder you cling to them.

24. Damra Vol 00:08:18

    Lead with the one that bites hardest.

25. Lenar Kess 00:08:20

    Errors as inputs. In normal software a failed call is cheap — you catch it, you retry, milliseconds. Schmid points out an agent run can be five to fifteen minutes of compute. So if it fails at minute

twelve and you just restart, you've burned the time and thrown away all the accumulated context. His model is the Go language pattern — a call returns a value or an error — and the error has to be fed back into the model so it recovers incrementally, not from scratch.

26. Damra Vol 00:08:50

    That reframes retry logic completely. A retry isn't 'do it again,' it's 'here's what went wrong, continue from here.' What were the others?

27. Lenar Kess 00:09:00

    Context replaces structured state — instead of boolean flags and a rigid user profile, the agent reads semantic meaning from text and multimodal input. His example was a research agent where you approve a plan and inject a constraint in the same breath — 'yes, go, but use metric units' — and it just absorbs that, no separate settings screen. Then, you go from traffic controller to dispatcher. You stop writing the state machine that says step one, step two, step three. You hand the model a goal and trust it to navigate a path you didn't pre-draw.

28. Damra Vol 00:09:31

    [tsk] 'Trust the model to navigate' is the line that makes senior engineers break out in hives, and for good reason. Trusting a nondeterministic thing to find its own path is how you get a system you can't debug. What's his answer to that?

29. Lenar Kess 00:09:45

    His answer is the fourth shift: you stop testing with deterministic unit tests and you move to probabilistic evals. Same input doesn't guarantee the same path, so you measure pass rates, you use a model as a judge, you bring in human experts. And he had a hard line — if a prompt only succeeds one out of ten times, it's not viable for production. So you're not asserting equality. You're measuring a success rate and setting a floor under it.

30. Damra Vol 00:10:11

    And the fifth?

31. Lenar Kess 00:10:12

    Design your tools and APIs for the agent, not the human. His example: a delete-item endpoint is obvious to the developer who wrote it, but the agent only ever sees the function schema and the docstring. If those don't carry the meaning, the agent's flying blind. And he lands the whole talk on a phrase — build to delete. The agent code you write is disposable, because the model keeps getting better and you'll throw your wiring out in three months anyway.

32. Damra Vol 00:10:40

    Build to delete is where I'd push back gently. It's freeing if you're at DeepMind shipping experiments. It's terrifying if you're an enterprise team being asked to maintain this thing for five years. Disposable software is a wonderful mindset right up until someone asks who owns the disposable thing in production at two in the morning.

33. Lenar Kess 00:10:59

    Which is the exact wall the next person hit. There's a post on the AI Agents subreddit from a developer, Paulius, titled 'I spent a year building agent memory on knowledge graphs — here are the five mistakes that cost me months.' Let me be straight about what I've actually got: the excerpt gives me his opening, not the full list of five. But the opening is the whole confession. He writes that he built a unified memory layer for his agents using knowledge graphs and ontologies on top of MongoDB, and — quote — 'I followed every trend first. I reached for the shiny frameworks and tried to design' — and that's where my excerpt cuts off.

34. Damra Vol 00:11:35

    [sigh] A year. On the memory layer specifically. And I'd bet the five mistakes are all variants of 'I built the elaborate thing before I knew whether the simple thing worked.' Knowledge graphs and ontologies are exactly the kind of architecture that feels rigorous and quietly eats your calendar.

35. Lenar Kess 00:11:53

    That's the read, and it isn't a dunk — memory really is the hard, unsolved layer right now. The model is rarely the constraint anymore. It's the thing you wrap around it that has to remember across sessions. And it pairs with a Show HN that went up this weekend, a project called Komi-learn — continuous memory and self-improvement for coding agents. Thirteen points, two comments. And the top comment is the whole genre in one breath.

36. Damra Vol 00:12:19

    Go on.

37. Lenar Kess 00:12:20

    A commenter, loehnsberg, wrote: 'It sounds like it solves the problem that everybody who vibe codes over multiple projects runs into, but it does not provide evidence that it actually works.' That's it. That's the memory space right now. A real problem, everybody feels it, a hundred projects claiming to solve it, and almost nobody showing the before-and-after that proves the agent got better because of the memory and not because the underlying model did.

38. Damra Vol 00:12:45

    And that's the test I'd hold all of these to — including Paulius's year and Komi-learn. Show me the agent failing a task, then show me the same agent passing it after the memory layer, with the model held constant. Until I see that controlled comparison, a knowledge graph is just a database you're proud of. The graph isn't the achievement. The measured improvement is.

39. Lenar Kess 00:13:06

    And the cost of skipping that proof is exactly Paulius's year. You can spend twelve months making the memory beautiful and never once check whether it changed a single outcome. That's the maintenance bill nobody photographs for the launch post.

40. Lenar Kess 00:13:19

    This one steps out of tooling and into something with real stakes. Forbes wrote it up from a correspondence published in The Lancet. Over a three-year period, reviewers found that 4,046 references across 2,810 published scientific journal articles had been fabricated. These weren't wrong or sloppy — they were fabricated. Citations to papers that, as far as the reviewers could tell, don't exist or don't say what they're cited as saying.

41. Damra Vol 00:13:46

    Twenty-eight hundred articles that already passed peer review and got published. So the fabrication survived the one filter that's supposed to catch it. Do we know the mechanism — is this people using a model to write the literature review and the model inventing plausible-looking references?

42. Lenar Kess 00:14:01

    The framing points that way — these are described as AI-fabricated citations, the pattern where you ask a model for sources and it generates references that look perfectly formatted, completely real, and are simply invented. The write-up doesn't give me a clean split of how many were caught before versus after publication, so I won't claim a number there. But the headline fact is that thousands of them made it all the way into the published record.

43. Damra Vol 00:14:26

    And here's the connection back to where we started that I think actually holds. Segment one, Anthropic is tuning a model toward honesty — that's a dial inside the model. This is the same word at the system level, and it's pointing the wrong way. You can make one model more truthful and still watch the scientific literature fill up with confident, well-formatted fiction, because the failure isn't the model lying. It's a human pasting the model's output without checking a single reference.

44. Lenar Kess 00:14:54

    That's the proportionate version. The model isn't the villain — a person decided not to verify. But the scale is the new part. Fabricating four thousand references by hand over three years is a career of fraud. With a model it's an afternoon. So the integrity check that used to be slow enough to be self-limiting just got cheap, and the journals haven't caught up.

45. Damra Vol 00:15:15

    And the fix is the kind of work nobody funds — reference-checking at the journal, an automated pass that confirms every cited paper exists and actually supports the claim. It's tedious, and it's exactly what breaks when twenty-eight hundred articles slip through. Catching this is easier to build than the agent memory we just spent ten minutes on. It just isn't anybody's launch.

46. Lenar Kess 00:15:38

    Let me close with three quick ones we're tracking, none a turning point, all fast. First, Reuters has a piece on the contrast between how AMD's Lisa Su and Nvidia's Jensen Huang play China. Su keeps a deliberately lower profile, and the detail that anchors it: China is about twenty percent of AMD's revenue. So the low profile isn't shyness. It's protecting a fifth of the top line.

47. Damra Vol 00:16:02

    Twenty percent is the number that explains the whole personality difference. Jensen can be the public face because Nvidia's exposure and leverage are different. Su's incentive is to not become the headline. Same market, two completely different risk calculations.

48. Lenar Kess 00:16:16

    Second, IBM put out what it's calling an agentic operating model, with a governance layer named Sovereign Core, aimed at moving enterprise AI from pilot to production with sovereignty — data control, jurisdiction — at the center. Forbes covered it. I read it as IBM betting that the enterprise blocker isn't capability, it's governance, and then selling the governance.

49. Damra Vol 00:16:38

    Which rhymes with the build-to-delete problem from Schmid. The enterprise can't treat agents as disposable, so somebody sells them the control plane that makes the disposable thing auditable. That's a genuine market. Whether Sovereign Core is substance or a slide, I'd want to see what it actually enforces.

50. Lenar Kess 00:16:57

And third, a strange one at the edge of our beat. A US court ordered Circle — the stablecoin company — to blacklist a smart contract tied to a group called Zama, freezing about twelve and a half million dollars. The Block reported it, and their framing was that a lot of ordinary holders got caught in the crossfire of a civil suit against a decentralized org. The point that matters for us: programmable money means a court order can freeze one specific contract. That freeze function isn't theoretical. It just got used.

51. Damra Vol 00:17:27

    And that's the through-line for the whole morning, if there is one. Every layer we touched has a control surface someone's reaching for. The model's honesty dial, the journal's missing verification, and the stablecoin's freeze function. The capability's not usually the story anymore. Who holds the dial is.

52. Lenar Kess 00:17:45

    So, into the week, three specific things I can actually check. One: does anyone run Opus 4.8 on a fresh, private eval and either confirm or kill that DeepSWE result. Two: does a single one of these memory projects ship a controlled before-and-after. Three: does even one journal turn on automated reference-checking after this Lancet number. All three are answerable, none of them are about godhood, and we'll see what Monday brings.

Hosts on this episode

- Lenar Kess moderator
- Damra Vol critic