Locally coherent, globally not

2026-05-29 / 00:22:01

> *“Are you measuring whether the model said no, or whether the model couldn't say yes? Those are different tests, and we mostly run the first.”*
>
> — from this episode's transcript

- Lenar Kess
- Damra Vol

Friday's room sits between a hobbyist voice assistant running entirely on Mario Zechner's desk and a cluster of arXiv papers all saying the same thing from different angles: long-running agents now fall apart in ways the model can't fix. Lenar and Damra read four reliability papers side by side, then turn to the personal-memory question every shipping assistant is already getting wrong.

- Mario Zechner on pibot — full local voice loop with Parakeet, Qwen 3 TTS, and Qwen 3.6 through llama.cpp, with the STT and TTS engines ported from Python into Rust on mlx-c. The runtime detail is the news, not the model lineup.
- Ethan Mollick on token budgets — split spend between building and learning. Read against yesterday's Kirkland and Ellis platform story, the question becomes who controls the learning budget at internal AI orgs.
- MMPO — Ziyan Liu and team train a policy that decides when memory in long-horizon agents should be rewritten and when it should be left alone. Belief drift comes from over-eager rewrites, not missing updates.

- RedundancyBench — Minyang Hu's group benchmarks how many steps in a long agent trajectory are repeats. Stale duplicates of state crowd out the relevant signal in context.
- Locally Coherent, Globally Incoherent — Anany Kotawala's single-author paper bounds compositional incoherence in multi-component agents. Defensible local outputs assemble into contradictory global ones.
- Agent-Radar — Hongxiang Zhang's group steers attention toward context-relevant tokens in multi-agent communication, so the receiver isn't drowned in noise from the sender.
- Selective QA over conflicting personal memory — Tiancheng Yang's testbed for what happens when your assistant's memories about you disagree. No single resolution strategy dominates.
- BioRefusalAudit — Caleb DeLeeuw uses sparse autoencoders to ask whether a model's refusal is shallow pattern matching or whether the dangerous capability isn't there at all.
- AutoformBot and Atlas — Ahmad Rammal's team at FAIR Paris and NYU on a multi-agent system that pulls textbook math into Lean 4 at scale. Lean is the verifier the agents can't argue with.

SEGMENTS

Transcript

1. Lenar Kess 00:00:00

   A guy named Mario Zechner posted a photo this morning of a small device on his desk — speaker, microphone, a tangle of cables running back to a tiny board. He calls it pibot. He's been building it for a few months. As of this morning, the whole voice loop runs locally on the box: speech-to-text, language model, and text-to-speech, all running without a cloud round trip and without a Python interpreter behind it. He talks. It answers. Everything stays in the room. So here's what we'll walk through over the next half hour. Mario's box sits on the personal end of the story today. Ethan Mollick posted a short note about how organizations should split their AI budget between building and learning. And a cluster of papers landed on arXiv this morning that all read like the same observation from different angles — as we run agents for longer, they fall apart in new ways, and the fixes aren't better models. They're better wrappers around the models. We'll close with a benchmark for conflicting personal memory, an auditing technique that asks how deep a refusal actually goes, and a project that's pulling math textbooks into Lean 4. Damra, where do you want to start?

   - x.com
   - x.com
   - arxiv.org
   - arxiv.org
   - arxiv.org
   - arxiv.org
   - arxiv.org
   - arxiv.org
   - arxiv.org

2. Damra Vol 00:01:06

   Start with the box. It's the only thing in the story you can point at. The stack he's running is Parakeet for speech-to-text, Qwen 3 TTS for synthesis, and Qwen 3.6 as the multimodal large language model behind it all, served through llama.cpp. Parakeet is Nvidia's open recognition family. Qwen 3 TTS is Alibaba's open synthesis model. Qwen 3.6 is the dense multimodal release from earlier this month. What's new in Mario's setup isn't the model lineup. It's the runtime. He ported the recognition and synthesis inference engines from Python into Rust on top of mlx-c. So none of those four components need a Python interpreter at runtime.

3. Lenar Kess 00:01:50

   Which matters why? Spell it out for someone who hasn't tried to put one of these on a small device.

4. Damra Vol 00:01:56

Because Python is what you eventually hit. You can ship a quantized model in a small package. The moment your recognition stack demands torch and your audio pipeline pulls in transformers, you're back to a multi-hundred-megabyte install on a Pi-class machine — and a cold-start time the user can feel. Rust plus mlx-c keeps you in single-binary territory. The whole assistant fits in a fraction of the disk and starts in a fraction of the time. And on Apple silicon, mlx-c lets him use the unified memory the way the hardware wants to be used.

5. Lenar Kess 00:02:30

   There's a photo of the device sitting on his desk, by the way. It isn't a research result. It's a person saying — the local voice stack is finally good enough that I built one for my apartment, and it works. A year ago that sentence required a workstation under the desk. Two years ago it required a cloud bill.

6. Damra Vol 00:02:47

   And it changes what "agent" means in homes and small offices. If the speech round trip stays on the device, the conversation history stays on the device. That's a different privacy posture than anything that ships an audio buffer to a cloud endpoint. It's also a different latency posture. The reason most voice assistants feel sluggish isn't the model. It's the network leg in both directions.

7. Lenar Kess 00:03:11

   Two things I'll flag and not oversell. One — I haven't run pibot myself. I'm taking Mario at his word that the throughput is conversational. He doesn't post a tokens-per-second number in the screenshot I'm looking at. Two — there's a real model-quality gap between Qwen 3.6 at the size he's running and the frontier hosted models. He isn't claiming parity. He's claiming the local version answers the kinds of questions he's actually asking it. Which is the right test for this category.

8. Damra Vol 00:03:40

   It does not have to beat Opus on a benchmark. It has to be good enough that the user doesn't reach for their phone. And the cohort of people who'd build one of these for their kitchen — they care more about "works without the internet" than about the last few points on MMLU.

9. Lenar Kess 00:03:55

   Ethan Mollick — who teaches at Wharton and writes the One Useful Thing newsletter — posted a thought this morning about how organizations should spend their AI budget. He frames it as two buckets. One — tokens you spend on building things. Two — and the version of the tweet I can see ends in an ellipsis, so the second item is cut off. But in context, and given what Mollick has written before, the second bucket is tokens you spend learning what works. Tokens against problems you don't yet know how to solve.

10. Damra Vol 00:04:23

    The truncation matters less than the move. He's saying token spend isn't a single line item. It's two different activities with two different success criteria. Building is — ship the artifact. Learning is — find out whether this even makes sense. Those want different review cadences, different teams, and different definitions of done.

11. Lenar Kess 00:04:43

    And the reason I keep that distinction in mind this week is that we spent yesterday on Kirkland and Ellis's five-hundred-million-dollar internal AI platform. Most of that money is going into building. What's harder to see in the K&E story is whether they've reserved enough capacity to learn — to try things that don't work, that they can throw away. Internal AI orgs at that scale almost always under-fund the learning bucket, because the deliverables column is what gets approved at the board meeting.

12. Damra Vol 00:05:11

    And when the board approves a budget, the line items are deliverables. Nobody writes — twenty percent of our token spend will go to ideas we abandon. But that's exactly the spend that tells you which deliverables are worth shipping next. The team that ran experiments and threw them away knows things the team that only shipped doesn't.

13. Lenar Kess 00:05:30

    I'd add one more. Experiment tokens have a different review cadence. Build tokens get checked at the end. Experiment tokens have to be checked weekly, because otherwise you can spend a quarter of compute against a problem no one can describe well enough to evaluate the result against.

14. Damra Vol 00:05:46

    And the people running the experiment have to be the same people who'd ship the result. If you split the experiment team from the deploy team, the experiment team learns things the deploy team doesn't trust, and the deploy team builds things the experiment team would've talked them out of.

15. Lenar Kess 00:06:01

    Mollick's post is short. The implication is heavier than the post. Read it next to yesterday's K&E story and what you'd ask of any internal AI org becomes — what's the learning budget, who controls it, and how often does it get reset to zero so the team can try again.

16. Lenar Kess 00:06:16

Four papers landed on arXiv this morning that read as a cluster, even though none of the authors know each other. Each one names a different way that long-running agent sessions fall apart. Together they sketch the shape of where the reliability work is right now.

17. Damra Vol 00:06:31

    Walk me through them. Slowly. Start with the one that connects back to what we covered Wednesday.

18. Lenar Kess 00:06:37

    Right. The first is Meta-Cognitive Memory Policy Optimization — MMPO — from a team led by Ziyan Liu. The setup is long-horizon agents that keep their context manageable by recursively summarizing their own history. After each step, the agent compresses what it knows into a smaller summary. The problem they name is belief deviation. After enough summarization rounds, the agent's working belief about the world drifts away from what was actually established earlier. The summary is fluent. It's also slightly wrong. And the next summary compresses the slightly wrong version, so the drift compounds.

19. Damra Vol 00:07:14

    Wait — recursive summarization, you mean the technique every long-context agent has been using for the last year? That's what they're modifying?

20. Lenar Kess 00:07:22

    That's what they're modifying. Their move is to train a policy that decides when to update memory and when to leave it alone. Memory updates become actions the policy can refuse. If the new information doesn't change anything important, the policy leaves the existing summary as it is. They report meaningful gains on long-horizon tasks. The intuition tracks — most of the drift in these systems comes from over-eager rewrites, not from missing updates.

21. Damra Vol 00:07:48

    Which maps cleanly onto what we covered Wednesday — the agent memory degradation work, and the broader observation that persistent memory systems age badly. This is the same family of problem with a learned controller bolted on top. The controller's job is to know when to write.

22. Lenar Kess 00:08:04

    The second paper is RedundancyBench, from a team at Huawei and Hong Kong Polytechnic, lead author Minyang Hu. They ask whether the steps an agent actually takes in a long trajectory are necessary. They build a benchmark for detecting redundant steps after the fact. The headline

finding — a meaningful fraction of agent steps in current systems are repeats. The agent re-reads the same file it read fifty steps ago. It re-queries the same endpoint. It re-derives a fact it already had in context.

23. Damra Vol 00:08:34

    Which sounds boring until you do the math on a thousand-step trajectory. If a quarter of your steps are redundant, you're paying for inference and tool calls you don't need, and you're filling the context with stale duplicates of state the agent already established. So the redundancy isn't just a cost line item. It actively makes the next step worse, because the relevant signal is now buried under repetition.

24. Lenar Kess 00:08:58

    Third — Anany Kotawala has a single-author paper with my favorite title of the day. Locally Coherent, Globally Incoherent. Bounding compositional incoherence in multi-component LLM agents. The framing — each sub-agent or sub-component in a multi-agent pipeline produces something defensible on its own. The assembled output is internally inconsistent because the components don't share constraints with each other. Kotawala's contribution is a bound. He proves a relationship between how often individual components are locally right and how often the assembly is globally right.

25. Damra Vol 00:09:31

    That's the failure every team building multi-agent pipelines runs into the first time they show a demo to someone outside the room. Every component looks defensible. The end-to-end answer contradicts itself. The planner says one thing. The retriever brings back something inconsistent with that. The summarizer smooths over the conflict and produces a coherent-sounding paragraph that's wrong in a different way than either input.

26. Lenar Kess 00:09:55

    And the fourth, briefly — Agent-Radar, from Hongxiang Zhang at Purdue. Same neighborhood. They study attention steering with context relevance in multi-agent communication. When sub-agents exchange messages, the receiving agent's attention spreads across irrelevant pieces of the incoming message and the relevant signal gets diluted. They propose a steering mechanism that biases attention toward context-relevant tokens.

27. Damra Vol 00:10:21

    So if you read all four side by side — MMPO on memory drift, RedundancyBench on wasted steps, Compositional Incoherence on assembled wrongness, and Agent-Radar on attention dilution — you can see the shape. As agent sessions get longer and as more sub-components get composed

together, the new failure modes aren't about whether the model can answer a question. They're about whether the trajectory stays coherent and whether the steps add up to something useful.

28. Lenar Kess 00:10:49

    And the fixes proposed across the four papers are not new model capabilities. They're control layers wrapped around the model. A learned policy that gates memory writes. A benchmark that catches redundancy. A bound that quantifies compositional damage. An attention steering mechanism. Same shape as the harness conversation we had Tuesday. The model is fine. The layer wrapping the model is where the bugs live now.

29. Damra Vol 00:11:13

    Let me put a brake on one piece. These are all arXiv preprints from today. None of them have replication yet. The MMPO numbers look strong enough that I'd want to see another team rerun the experiments before I bring the policy into production. Kotawala's bound is single-author and the proof needs review.

30. Lenar Kess 00:11:31

    Fair. The direction feels right and it lines up with what people running long agents in production have been complaining about all month. The specific numbers, I'm holding loosely. Anyone shipping an agent today should read MMPO and the redundancy paper this weekend. They might not adopt the methods. They'll recognize the failure modes.

31. Lenar Kess 00:11:50

    One more in the same neighborhood, but on the personal-assistant side. Tiancheng Yang at Waterloo, with Matthias Schonlau and Ilia Sucholutsky from Vector, posted a benchmark and method comparison they're calling — and I'll just read the title — Selective QA over Conflicting Multi-Source Personal Memory. The setup is what happens when a personal AI assistant has accumulated memories about you from multiple sources, and those sources disagree.

32. Damra Vol 00:12:16

    Give me a concrete example. What does the disagreement look like in practice?

33. Lenar Kess 00:12:20

    The example they walk through is preference conflict. Your calendar says you prefer morning meetings — the calendar's been saying that for two years. A message you sent two weeks ago says you've started blocking mornings for deep work and you want all meetings after lunch. Which one does the assistant believe when someone messages it asking to book time on your behalf? Both

pieces of information were true when they were written. Neither one is a lie. They contradict each other now.

34. Damra Vol 00:12:47

    And the harder version of the same problem — neither source is wrong even today. The calendar is a stated preference. The message is a more recent stated preference. The assistant has to know that recency matters, that explicit statements override inferred ones, that some preferences are revisable and some aren't, and that some context-windows of your life override others. That's a lot of judgment to ask a retrieval system to perform.

35. Lenar Kess 00:13:13

    They build a diagnostic testbed across several conflict types, and they compare a range of methods — straight retrieval, retrieval with a conflict-resolution step, methods that condition on recency, and methods that condition on source type. The honest summary is that no single method dominates. Different conflict types want different resolution strategies. Systems that try to use one strategy for everything underperform compared to systems that route the conflict type first and then apply a type-specific resolver.

36. Damra Vol 00:13:43

    Which lines up with how humans handle the same problem. You don't have a single algorithm for resolving contradictory information about a friend. You weight sources by recency, by who said it, by how confident they sounded, by whether it was an explicit statement or an inference from behavior. Asking a retrieval system to bake one of those weightings into its index gets you the wrong answer in three out of four cases.

37. Lenar Kess 00:14:07

    The reason this matters now — not in two years — is that the products shipping persistent memory right now don't have any of this machinery. When ChatGPT or Claude remember something about you, and that something becomes wrong, the next time the assistant uses that memory it confidently uses the stale version. There's no resolution step. There isn't a conflict-detection step. The newest entry doesn't necessarily win. The most explicit entry doesn't necessarily win. Whatever the retriever surfaces, the model treats as fact.

38. Damra Vol 00:14:37

    And that's a real cost for the user. Not a paper cost — a felt one. The assistant tells a coworker you prefer morning meetings when you've been telling everyone you don't, for the last two weeks. You don't see it happen. You just see the meeting on your calendar and wonder why nothing you say about your schedule sticks.

39. Lenar Kess 00:14:55

    Caleb DeLeeuw, an independent researcher, posted a paper called BioRefusalAudit. The premise is that current biosecurity evaluations of language models ask the model questions and grade whether it refuses. He argues that's a shallow test. A model can refuse for surface reasons — it pattern-matches on the phrasing of the question — and still have the relevant capability accessible if the question is asked differently.

40. Damra Vol 00:15:21

    So how does he test deeper than that? What does the audit actually measure?

41. Lenar Kess 00:15:25

    He uses sparse autoencoders — SAEs, the interpretability technique that's been getting attention this year — to look at the internal features the model activates when it's given a biosecurity-adjacent prompt. He asks a different question — not whether the model refused, but whether the model's internal representations contain the dangerous capability even when it refused at the surface. He compares general-purpose SAEs against ones he fine-tuned on the biosecurity domain to make the relevant features sharper.

42. Damra Vol 00:15:55

    That's a real distinction. Refusing because the request matches a refusal pattern is different from refusing because the relevant knowledge isn't there. The first is brittle — paraphrase the request, switch language, embed the question in a roleplay, and the pattern stops matching. The second isn't brittle in the same way, because there's nothing to retrieve.

43. Lenar Kess 00:16:15

    His finding is roughly that current refusal training in frontier open-weight models operates much more at the first level than the second. The capability is present internally. The refusal is a learned output filter sitting on top. And filters can be bypassed. The depth-versus-surface gap shows up clearly in the SAE features.

44. Damra Vol 00:16:35

    Which doesn't mean the filter is worthless. It means it's a layer, not a wall. The work this pushes on is whether we should be measuring refusal depth as a separate quantity from refusal rate. The current public scorecards for model safety mostly report the rate. They don't report the depth. And the depth is what determines how the model behaves against an adversary who's actually trying.

45. Lenar Kess 00:16:58

That's what I'd hand to any safety team running biosecurity evals this quarter. Are you measuring whether the model said no, or whether the model couldn't say yes? Those are different tests, and we mostly run the first. DeLeeuw's paper doesn't solve the second one. It builds the apparatus to ask it.

46. Damra Vol 00:17:15

    And it ties into something bigger. SAE-based auditing is moving from an interpretability curiosity into something safety teams will plausibly run as part of release evaluations within a year. Today's paper is one application. The general technique — read the internal features, don't just read the outputs — is the move.

47. Lenar Kess 00:17:34

    One last item, and it's a more cheerful one. A team with Ahmad Rammal at the lead, with people from FAIR Paris and NYU, posted AutoformBot. It's a multi-agent system that builds something they're calling Atlas — an autoformalized textbook library in Lean 4. The headline claim is that the system can take textbook math written in natural language and turn it into machine-checked Lean code at scale.

48. Damra Vol 00:17:58

    Define autoformalization for someone who hasn't met the term before.

49. Lenar Kess 00:18:02

    Mathematics written in natural-language proofs — the way textbooks write proofs, with English between the equations and a fair amount of "it is clear that" and "by symmetry" papering over the steps — translated into a proof assistant's formal language. Lean 4 is the proof assistant. It checks every step. If the translation is wrong, Lean refuses to compile it. Atlas is their target — a library of textbook math, formalized, that the Lean community can build on.

50. Damra Vol 00:18:30

    Why a multi-agent system for that? What's the role split?

51. Lenar Kess 00:18:34

    Because formalizing a single theorem from natural language is a multi-stage problem. You have to parse the statement, decide what context to import, translate the statement into Lean, translate each proof step, close gaps that the textbook waved over, and verify that the translated version actually compiles. They give different agents different stages. One reads the textbook. One drafts the Lean statement. One drafts the proof. One closes the gaps the others left. They critique each other's output, and Lean is the ground-truth oracle for whether the final product survives.

52. Damra Vol 00:19:06

    And does it work at the textbook scale they're claiming? Or is this one chapter and a press release?

53. Lenar Kess 00:19:12

    Their claim is multi-textbook coverage with a meaningful fraction of theorems closing automatically. I haven't independently checked the numbers. The bigger story underneath — formalized math has been a fifteen-year project mostly run by small teams of dedicated mathematicians who hand-write Lean. The library that exists today, mathlib, was assembled one theorem at a time over a decade. If a multi-agent system can credibly do the bulk translation work, the rate of growth changes by an order of magnitude.

54. Damra Vol 00:19:41

    And once it's in Lean, it's verified. The agent can be wrong about the translation a hundred different ways. It can't be wrong about whether the Lean version compiles. The proof assistant is the ground truth. So unlike most agent benchmarks, this one isn't grading itself. The grading lives outside the loop entirely.

55. Lenar Kess 00:20:01

    That's the part that makes this category interesting to me. Most agent benchmarks grade themselves — the same model that produced the answer is involved in judging the answer. This one has an outside verifier that doesn't care about model-style answers. Either the proof closes or it doesn't. There's no rubric, no judge model, and no partial credit.

56. Damra Vol 00:20:21

    And it's an early sign that some agent workflows have natural verification built into the domain. Coding has tests. Math has proofs. Hardware design has simulation. Most other domains do not — which is why so much of the agent literature this week is about catching incoherence inside the trajectory rather than at the output. When you can't verify the output, you have to verify the process. When you can verify the output, the process can be as messy as it needs to be.

57. Lenar Kess 00:20:49

    That's where the day lands for me. A working local voice stack on a desk. A short note from Mollick about how to split your AI budget. A cluster of papers all saying that long agents fail in ways the harness has to catch. A benchmark for conflicting personal memory. An auditing technique that asks whether a refusal is shallow or deep. And a math project where the verifier is a proof assistant.

58. Damra Vol 00:21:12

The thread I'm pulling out — the model is rarely the constraint anymore on the things people are trying to build. The layer wrapped around it is. Mollick's learning budget, the MMPO memory policy, the RedundancyBench redundancy detector, the personal-memory conflict resolver, the SAE-based refusal auditor, and the proof-assistant verifier behind Atlas — different control layers, same job. They all sit between the model and the work, deciding what the model gets to do next.

59. Lenar Kess 00:21:41

    Tomorrow is going to be quiet. I'll be reading the MMPO paper end to end and seeing if the numbers hold up to a closer look. If something surprising lands over the weekend, we'll cover it Monday. Lenar Kess.

Hosts on this episode

- Lenar Kess moderator
- Damra Vol critic