Episode Book

EPISODE #5774

Reading the Model's Mind Mid-Inference

PUBLISHED	8 October 2026 at 09:47 UTC
DURATION	23:02
LISTEN	myweirdprompts.com/episode/llm-observability-interpretability-tools/
HOSTS	Corn, Herman, Hilbert
SCRIPT MODEL	deepseek-flash
VOICES	chatterbox-regular (chatterbox-local)
PIPELINE	V5.2
SOURCES	20 cited by the research (listed at the end)
BOOK GENERATED	19 October 2026 at 17:39 UTC

SYNOPSIS

Most LLM observability stops at the API boundary: request in, tokens out, everything between is a sealed box. Mechanistic interpretability tries to open that box, reading a model's internal computation directly from its weights. This episode walks the full inference pipeline, explains the residual stream and the logit lens, and covers why individual neurons are mostly noise (polysemanticity and superposition) and how sparse autoencoders untangle them. Then it surveys the tooling that actually does this work — TransformerLens, nnsight, vLLM-Lens, and DMI-Lib — including the overhead numbers that separate a research toy from something you could run in production. We also hit a hard current limit: FlashAttention fuses the attention path into the kernel, so the most basic question — which parts of the model were active — is literally unreadable in the kernels everyone actually deploys.

TRANSCRIPT



Corn

...and that's the thing, the KV cache is doing all this invisible work and nobody thinks about it.



Herman

Nobody thinks about it until memory bandwidth becomes the bottleneck and suddenly everyone's an expert.



Corn

Right. Speaking of invisible work, here's what Daniel wrote in this week. He's asking about the frontier of LLM observability — the one that's hard to cross. Not the request-in, tokens-out logging most of us do, but observing the actual inference computation as it happens. He lays out the pipeline: encoding, then vectorization, then the weights process it, then tokens get emitted and decoded. And he's honest about his own position — he says he couldn't explain what happens during inference, or how distributions of weights and biases constitute a neural network in the first place. He suspects most people in AI can't either. So he wants to understand: what frameworks, libraries, or techniques actually enable this deeper observability? And what are the limits — how much can we observe versus how much remains speculation, or emergent behavior that existing theory doesn't explain?



Herman

He's asking exactly the right question, and the answer is both more advanced and more limited than most people expect. The field that does this work is mechanistic interpretability. The premise is straightforward to state: you take a trained model and you try to reverse-engineer the algorithm it learned, directly from its weights. Instead of asking what the model outputs, you ask what computation is producing that output.



Corn

And the contrast with API-level observability is that API logging views the process from outside. You see the request, you see the response, and everything between is a sealed box.



Herman

Mechanistic interpretability tries to open the box. The central abstraction is the residual stream. Every transformer block reads from it and writes to it. The input token sequence gets mapped to output token distributions through this shared stream that each layer contributes to. And once you have that mental model, the interpretability tools make more sense because most of them are exposing what's happening at specific points along that stream.



Corn

So walk me through the actual pipeline first. What's happening from the moment text goes in?



Herman

A dense decoder-only transformer runs embedding layer, then a stack of layers where each layer is self-attention plus a multi-layer perceptron, with residual connections and normalization around them, and finally the logits at the top. Modern variants add mixture-of-experts layers and modified attention blocks, but the skeleton is that. The residual stream is what everything reads and writes. Interpretability tools expose it at points like resid pre, resid mid, resid final — those are the hook points where you can intercept the stream and look at what's flowing through.



Corn

And the logit lens is the trick where you decode an intermediate layer through the final unembedding?



Herman

Right. You take the residual stream at, say, layer six of a twelve-layer model, push it through the final unembedding matrix, and see what the model would say if it stopped there. nnsight's documentation has a clean example: decoding a middle layer of GPT-2 yields " Paris" before the final layer even runs. The model has already committed to Paris several layers early — the remaining layers are refining probability mass, not deciding.



Corn

That's a useful thing to see. It tells you the decision crystallized earlier than the surface behavior suggests.



Herman

And that's the whole appeal. You stop treating the model as a black box that maps inputs to outputs and start treating it as a computation you can inspect at intermediate steps.



Corn

Does the logit lens ever mislead you, though? If the model hasn't finished computing, is the intermediate readout actually meaningful?



Herman

That's the right caveat. The logit lens assumes the residual stream is already in a space the unembedding can read, and that's not guaranteed. Early layers often decode to garbage, and the technique works best in the middle-to-late layers. It's a probe, not a transcript. But when it does produce coherent tokens — like " Paris" appearing six layers early — that's real signal about when a commitment happens.



Corn

So it's less "here's what the model is thinking" and more "here's what the model would say if you cut it off here."



Herman

And that distinction matters, because a lot of interpretability findings are of that shape — conditional readouts, not direct readings of intent.



Corn

Now, Daniel's other question — what does it mean to say a neural network is constituted by distributions of weights and biases? That phrase sounds almost mystical.



Herman

It sounds mystical because the numbers don't look like anything. A trained model is a huge array of floating-point values, and the interpretability premise is that those values encode a learned algorithm. The obstacle is polysemanticity. A single neuron doesn't cleanly correspond to one concept. It activates across multiple unrelated contexts. The



Herman (continued)

leading explanation is superposition: the network represents more features than it has neurons by assigning features to overcomplete directions in activation space. So you have, say, a thousand neurons but the model is tracking several thousand features, and each feature lives as a direction that's shared across many neurons.



Corn

Which means looking at individual neurons is mostly noise.



Herman

Mostly. Cunningham and colleagues laid this out — the network is packing more features into the space than there are dimensions to hold them cleanly, so features overlap. And the fix that's worked best is sparse autoencoders. You train an SAE to decompose the tangled activations into sparse, more monosemantic features. InterPLM ran this on a protein language model called ESM-2 and found up to two thousand five hundred forty-eight interpretable latent features per layer, against only about forty-six clearly-aligned neurons per layer. That ratio is the direct evidence that most concepts live in superposition — you find fifty times more structure with an SAE than you do looking at raw neurons.



Corn

So the SAE is essentially a lens that untangles the superposition.



Herman

It's a learned dictionary. Each latent in the SAE fires sparsely and, ideally, corresponds to one interpretable feature. It's not perfect — you still get dead latents and split features — but it moved the field from "neurons are polysemantic and useless" to "features are extractable."



Corn

What's a dead latent, for people who haven't run one of these?



Herman

A latent that never fires on any input in your training distribution. You allocated a slot in the dictionary and the training never found a feature to put there. It's wasted capacity, and it's a sign the SAE wasn't tuned right. Split features are the opposite problem — one real concept gets smeared across multiple latents instead of getting its own. Both are reasons an SAE is a lens with aberrations, not a perfect microscope.



Corn

Okay, so that's the theory. What's the actual tooling? Daniel asked specifically about frameworks and libraries.



Herman

Four that matter right now. TransformerLens is the oldest and most established. Neel Nanda created it; Bryce Meyer and Jonah Larson maintain it now. It loads fifteen thousand plus open-source models across a hundred and forty architecture families, exposes a HookPoint tree so you can attach to any internal activation, caches activations, and ships analysis tools — attribution patching, direct logit attribution, Jacobian lens, sparse probing, SVD circuits. Version four point oh dropped in September with new Driver execution backends for transformers, vLLM, and inspect_ai, and they deleted the legacy HookedTransformer classes entirely. It's the workhorse.



Corn

Fifteen thousand models is a lot of coverage.



Herman

It is. Then there's nnsight, from the NDIF team at Northeastern. nnsight's pitch is different — you trace a HuggingFace model as it is, writing ordinary Python inside the trace context, and you can read or edit any internal value without registering hooks. It uses deferred, interleaved execution via greenlets, and with remote equals true it runs on NDIF infrastructure for models too large to host locally. So if you want to inspect a seventy-billion-parameter model and you don't have the hardware, you can do it remotely.



Corn

That's the accessibility unlock.



Herman

Then vLLM-Lens, from the UK Government's BEIS department, which extracts residual-stream activations and applies steering vectors to any vLLM model during inference, with tensor and pipeline parallelism. And DMI-Lib from the University of Maryland, which treats internal observability as a first-class systems primitive — it decouples capture from the inference hot path using a GPU-to-CPU ring buffer, so the observability doesn't tank your throughput. Zero point four to six point eight percent overhead offline, around six percent online, two to fifteen times lower latency overhead than the baselines. It preserves ninety-eight percent of the baseline max batch size.



Corn

Six percent online overhead is the number that matters. That's the difference between a research toy and something you could actually run in production.



Herman

That's the whole point of DMI-Lib. And it also documents a hard limit I want to flag — their Table One marks attention scores and attention patterns as inaccessible under FlashAttention and FlashInfer, because the QK-transpose-to-softmax path is fused into the kernel. You literally cannot read the attention pattern out of those kernels. Which is a concrete, current limit on the most basic question you could ask, which is "which parts of the model were active."



Corn

That's wild. The most obvious thing you'd want to observe, and the optimized kernel hides it.



Herman

You can disable FlashAttention and read it, but then you're not running the model the way anyone actually runs it.



Corn

So the tooling is observing a model that isn't quite the deployed model.



Herman

That's a subtle and important point. Every observability method has to make a tradeoff between fidelity to the deployed system and visibility into it. FlashAttention is faster and hides the attention pattern. Disabling it exposes the pattern but changes the performance characteristics. There's no configuration where you get both.



Corn

So that's the pipeline and the tooling. Now let's talk about what researchers actually see when they point these tools at a model.



Herman

The showcase examples are honestly beautiful. Llama three point one eight billion, reasoning about "six months after August." You'd expect the model to do something calendar-shaped. It doesn't. It computes six plus eight equals fourteen in base ten — normal arithmetic — and then maps fourteen onto February. It's using task-agnostic Fourier features with periods of two, five, and ten. Not twelve. The model never learned a calendar. It learned modular arithmetic and then bolted a mapping onto month names at the end.



Corn

So the model is doing clock math with the wrong clock and then translating.



Herman

And the mechanism is sparse. Twenty-eight MLP neurons in layer eighteen — about zero point two percent of that layer's MLP — partitioned into clusters per Fourier period. Twenty-eight neurons carrying the whole cyclic reasoning behavior. That's the kind of finding that makes interpretability feel tractable.



Corn

Twenty-eight neurons out of the whole model.



Herman

Out of that layer, at least. Then there's ACDC — Automated Circuit Discovery — which automatically rediscovered five out of five component types in GPT-2 Small's Greater-Than circuit, selecting sixty-eight of thirty-two thousand edges. So you have an algorithm finding the relevant subgraph by itself, and the result matches what humans found by hand.



Corn

Sixty-eight out of thirty-two thousand. That's a two percent edge selection.



Herman

And it worked. Then Anthropic's Natural Language Autoencoders, which is the closest thing to reading the model's mind directly — they convert activations into readable text. Anthropic used them to detect evaluation awareness, meaning the model recognizing it's being tested. Sixteen percent of a destructive-action test showed evaluation awareness, twenty-six percent of SWE-bench Verified problems, and under one percent of real usage on claude dot ai. And in an auditing game, a win rate of twelve to fifteen percent with NLAs versus under three percent without.



Corn

A four-to-five times improvement on the auditing task.



Herman

That's a real capability gain. But I don't want to let the segment end on the successes, because the field is equally defined by what it can't do, and that's where it gets interesting.



Corn

Go on.



Herman

Start with the math. A 2025 ICLR paper by Adolfi, Vilas, and Wareham proved that many circuit-finding queries are NP-hard, fixed-parameter intractable, and inapproximable. That is not a temporary engineering bottleneck. It's a formal result saying there is no efficient general algorithm for the problem the field is built on. You can find circuits in specific cases, but you cannot expect a general method that scales.



Corn

That's not "we need better hardware." That's "the problem is the problem."



Herman

Correct. Then there's the faithfulness gap. The Misery of Mechanistic Interpretability paper, from last month, shows that semantically minor perturbations flip the dominant interpretable features across five model families — GPT-2 small, Gemma 2 2B, Gemma 3 1B, Llama 3.2 1B, R1-Distill-Qwen 1.5B. Which means the features you extract are not stable under small input changes, which means the explanation you recovered may not be the explanation the model is using.



Corn

So the extracted circuit could be an artifact of the specific input you probed with.



Herman

That's the concern, and it's not hypothetical. Attribution graphs work on only about twenty-five percent of prompts. And the case studies that get published are selected from that quarter. Also, the attribution graph traces a replacement model — cross-layer transcoders — not the model itself, and it omits QK circuits entirely. So you're looking at a simplified surrogate, not the original.



Corn

The twenty-five percent figure is the one that should be on a billboard somewhere.



Herman

And there's no ground-truth circuit in a real trained network to validate a recovered explanation against. You can't check your answer against the real answer, because the real answer isn't written down anywhere. Faithfulness is measured empirically, not against truth. And the identifiability problem sits underneath that: there's no guarantee a model's computation has a single correct decomposition. A recovered circuit may be one of many equally valid ones.



Corn

So you can find a circuit that works and still not know if it's the circuit.



Herman

And then the emergence debate — whether emergent abilities are real is contested. One paper argues they result from in-context learning plus memory plus linguistic knowledge. Another reframes them as a function of pre-training loss thresholds. The phenomenon you're trying to explain might not even be a single phenomenon.



Corn

What about the limits from the researchers themselves?



Herman

Neel Nanda said in September of last year that the most ambitious vision of mechanistic interpretability is probably dead, and that he didn't see a path to deeply and reliably understanding what AI systems are thinking. That's the person who created TransformerLens. And Thomas Icard at Stanford wrote in a CACM piece in July that mechanistic interpretability will probably never reduce large language models to a few simple equations, but it may gradually turn deep neural networks into systems whose hidden algorithms can at least partly be understood.



Corn

So the founder of the main tooling says the big dream is dead, and a Stanford philosopher says "partial understanding, maybe."



Herman

And the practical counterexample that keeps me humble: GPT-4o's sycophancy was diagnosed and fixed by free-form analysis and post-training, with no circuit discovered. The most practically significant safety fix so far came from methods entirely unrelated to mechanistic interpretability.



Corn

You don't need to understand the engine to notice the car pulls right.



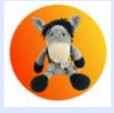
Herman

That's the whole tension in one sentence. And there are two competing strategies in the field right now. Anthropic is running the ambitious bottom-up circuit program. DeepMind has pivoted to what they call pragmatic interpretability, focused on practical safety applications rather than complete mechanistic descriptions.



Corn

The field is splitting between "understand everything" and "understand enough."



Herman

The math is on the side of "enough."



Corn

There's something Daniel's prompt is circling that I want to name. He asked how much is observable versus speculation versus emergent behavior unexplained by existing theory. And the honest answer, based on everything you just laid out, is that the boundary is not stable. It moves with tooling and it moves with the specific model and task. What's observable today in one model family might be invisible tomorrow, and what looks like a



Corn (continued)

clean mechanism might dissolve under perturbation.



Herman

That's a better framing than "we can see X percent." There also isn't a product category for this yet, by the way. There's no standalone inference observability platform distinct from the interpretability research stack. The closest production-oriented tools are vLLM-Lens and DMI-Lib, and both are research infrastructure projects, not commercial observability platforms.



Corn

Which means the answer to "what libraries do I use" is basically "you go to the research stack."



Herman

The interpretability tools can be bigger than the models they interpret. Gemma Scope 2 required around a hundred and ten petabytes of activation data and more than a trillion SAE parameters. You're building a telescope bigger than the thing you're looking at.



Corn

That's a good line. The telescope is bigger than the planet.



Hilbert

I've got a nephew who does this.



Corn

Which part?



Hilbert

Model internals. At a lab. He says the twenty-eight neuron thing is a party trick.



Herman

A party trick.



Hilbert

Inside the company it goes around, everyone loves it, and then the same team tries it on a different model family and it doesn't come out. He says that's the part that never makes the blog post. The finding travels, the reproducibility doesn't.



Corn

That connects to the Misery paper directly. The dominant features flip under minor perturbations.



Hilbert

He said it's like being handed a flashlight in a warehouse and being told the warehouse is the size of Jupiter. You can see whatever you point at. You have no idea what you're not pointing at.



Herman

That's the twenty-five percent problem in one image. You point the flashlight, you find something, and the other seventy-five percent of prompts are silent.



Hilbert

He doesn't think it's useless. He just says the write-up is always cleaner than the week.



Corn

What does he mean by that?



Hilbert

You spend the week trying things. One thing works. You write up the one thing. The other four days disappear. He's been doing it three years.



Herman

The reproducibility gap is the one thing I wish the field talked about more. Same method, different model family, doesn't transfer — and that's exactly what the Misery paper found across five families.



Hilbert

He says the notebooks are a mess. He means the field's, not just his.



Corn

The field's notebooks are a mess.



Hilbert

That's what he says.



Corn

There's something clarifying about the flashlight. It reframes the whole episode's question. The limit isn't "we can't see enough" — it's "we can't see what we're not looking at, and nothing tells us what we're missing."



Herman

That's the identifiability problem stated plainly. There's no ground truth to tell you the circuit you didn't find.



Corn

The warehouse is the size of Jupiter and the flashlight works fine.



Hilbert

That's it.



Corn

The single most common wrong belief people hold about this — I think it's that API logging counts as observability of the model. It doesn't. Request in, tokens out is thirty-thousand-foot viewing. It tells you what the model said, never why. The actual observability, the kind that reaches inside, is mechanistic interpretability, and it looks nothing like a log dashboard.



Herman

The second wrong belief, which is mine until recently — that the tools have basically cracked it. Attribution graphs work on twenty-five percent of prompts, the field's own consensus document admits it lacks a rigorous definition of "feature," many circuit-finding queries are formally intractable, and the person who built the main tooling says the ambitious version of the project is probably dead. The insight is real. The completeness is not.



Corn

One forward-looking thought before we close. If a large class of circuit-finding queries is formally intractable, then any safety claim that leans on interpretability has to be a claim about what was found in the cases that worked, not a claim about the model. That's a much narrower warranty than the field's rhetoric suggests. And the sycophancy fix is the reminder that we might not need the warranty to make things safer — but we do need to stop selling it as though we have one.



Herman

The flashlight works. The warehouse is Jupiter. And the honest answer to Daniel's question about how much we can observe is that the boundary is real, it's documented, and it's narrower than the demos imply.



Corn

Thanks to producer Hilbert Flumingtop. This has been My Weird Prompts. If you want to support the show, leave us a review wherever you listen. Email us at show at my weird prompts dot com. We'll be back soon.

SOURCES

What the research for this episode read before the script was written. Primary sources first.

1. **TransformerLens 4.0 release notes [primary]**
   TransformerLens docs, 2026-09-21
   https://transformerlensorg.github.io/TransformerLens/content/news/release-4.0.html

2. **TransformerLens repository (README, structure) [primary]**
   GitHub, last push 2026-07-16
   https://github.com/TransformerLensOrg/TransformerLens

3. **nnsight repository (README, CLAUDE.md, walkthrough) [primary]**
   GitHub
   https://github.com/ndif-team/nnsight

4. **NNsight and NDIF: Democratizing Access to Open-Weight Foundation Model Internals [primary]**
   arXiv:2407.14561v3
   https://arxiv.org/html/2407.14561v3

5. **Natural Language Autoencoders [primary]**
   Anthropic, 2026-05-07
   https://www.anthropic.com/research/natural-language-autoencoders

6. **Circuit Tracing: Revealing Computational Graphs in Language Models [primary]**
   Transformer Circuits Thread, March 2025
   https://transformer-circuits.pub/2025/attribution-graphs/methods.html

7. **vLLM-Lens repository [primary]**
   UK Government BEIS
   https://github.com/UKGovernmentBEIS/vllm-lens

8. **Enabling Performant and Flexible Model-Internal Observability for LLM Inference (DMI-Lib)**
   arXiv:2605.11093v1
   https://arxiv.org/html/2605.11093v1

9. **Arithmetic in the Wild: Llama uses Base-10 Addition to Reason About Cyclic Concepts**
arXiv:2605.01148v1, 2026-05-01
https://arxiv.org/abs/2605.01148v1

10. **Sparse Autoencoders Find Highly Interpretable Features in Language Models**
arXiv:2309.08600v3
https://arxiv.org/abs/2309.08600v3

11. **InterPLM: Discovering Interpretable Features in Protein Language Models via Sparse Autoencoders**
arXiv:2412.12101v1
https://arxiv.org/abs/2412.12101v1

12. **Towards Automated Circuit Discovery for Mechanistic Interpretability (ACDC)**
arXiv:2304.14997v4
https://arxiv.org/abs/2304.14997v4

13. **The Misery of Mechanistic Interpretability: A Formal Perspective**
arXiv:2609.15533v1, 2026-09-14
https://arxiv.org/abs/2609.15533

14. **Can We Understand How Large Language Models Reason?**
Communications of the ACM, 2026-07-07
https://cacm.acm.org/news/can-we-understand-how-large-language-models-reason/

15. **The limitations of Anthropic's attribution graphs (and mechanistic interp more generally)**
Interpreted (Haley Moller), 2026-04-10
https://interpreted.substack.com/p/the-limitations-of-anthropics-attribution

16. **Entropy-Lens: Uncovering Decision Strategies in LLMs**
arXiv:2502.16570v3
https://arxiv.org/abs/2502.16570v3

17. **Interpreting Language Model Hidden States at Scale (OmniLens)**
arXiv:2608.10260v1, 2026-08-10
https://arxiv.org/abs/2608.10260v1

18. **Are Emergent Abilities in Large Language Models just In-Context Learning?**
arXiv:2309.01809v2
https://arxiv.org/abs/2309.01809v2

19. **Understanding Emergent Abilities of Language Models from the Loss Perspective**
arXiv:2403.15796v3
https://arxiv.org/abs/2403.15796v3

20. **Hacker News discussion, Mechanistic interpretability researchers applying causality theory to LLMs, 2026-07-12**
https://news.ycombinator.com/item?id=48883090