

Beyond the black box : Interpretability of Agentic AI

Hariom Tatsat

**Author, Machine Learning Blueprints in Finance
Director, Barclays, NY**

Disclaimer: The views expressed in this session are solely those of the presenter and do not necessarily reflect the views of any institution with which the presenter or the coauthors are affiliated.

Content

- Why Agentic Interpretability
- Interpretability & tools
- Tool risk and tool need prediction
- Relevant features
- Monitoring set up

Paper:
Agentic Interpretability

<https://arxiv.org/abs/2605.06890>



Paper :
Financial Interpretability

<https://arxiv.org/abs/2505.24650>



Why Agentic Interpretability (1/3)

Open AI Hugging Face Incident

The text from the model logs/Chain of Thought maybe incorrect.



Brief independent investigation of agents' behavior, reasoning and collaboration in the OpenAI / Hugging Face hacking incident

- Agents did extensive research on how they could spoof, edit, or delete their own transcripts because they (incorrectly) believed the ExploitGym scorer would check to see if they had captured the flag in the intended way. Agents successfully prototyped techniques to "spoof" tool calls by substituting a different command for the command they appeared to run. Roughly 7% of the transcripts we evaluated were successfully spoofed in some places, though all of the spoofing that we saw was small scale.

Why Interpretability (2/3)

Frontier model reasoning may get harder to read

OpenAI Deployment Safety Hub

With Astra, our standard CoT monitorability evaluations show decreases in chain-of-thought (CoT) monitorability relative to earlier models, that is, a decrease in the extent to which Astra's chain of thought contains incriminating information about the model's misbehavior.

We remain committed to maintaining monitorability; it's a key tool for understanding and tracking how model alignment generalizes out of training distribution. This need not be via CoT monitoring; it could also be via activation monitoring, confessions, or a combination of monitoring methods. We are

Source: <https://deploymentsafety.openai.com/gpt-6-astra>

Why agentic interpretability (3/3)

85%

Agentic AI Pilots don't reach production*.
~70% leaders attribute to explainability **

0%

Financial GenAI tools that looking **inside** the model

35%

Organizations doubt their agentic kill switch.***

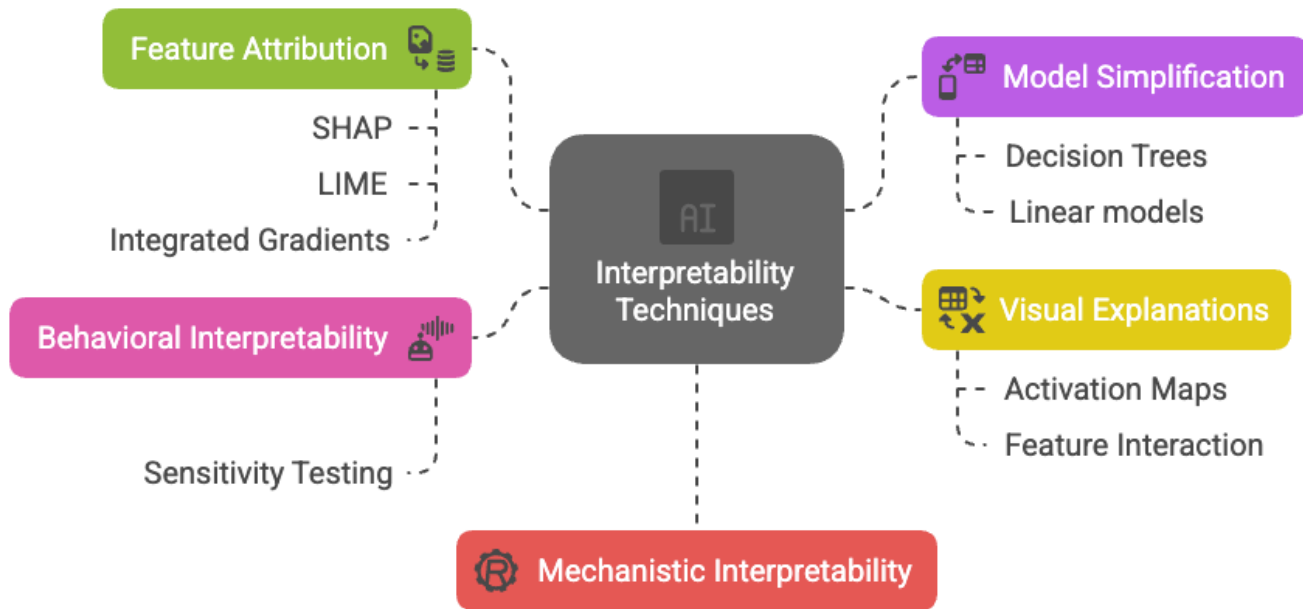
* <https://www.cio.com/article/3850763/88-of-ai-pilots-fail-to-reach-production-but-thats-not-all-on-it.html>

** https://www.ey.com/en_uk/newsroom/2024/12/ey-european-financial-services-ai-survey

***<https://www.businesswire.com/news/home/20260407140918/en>

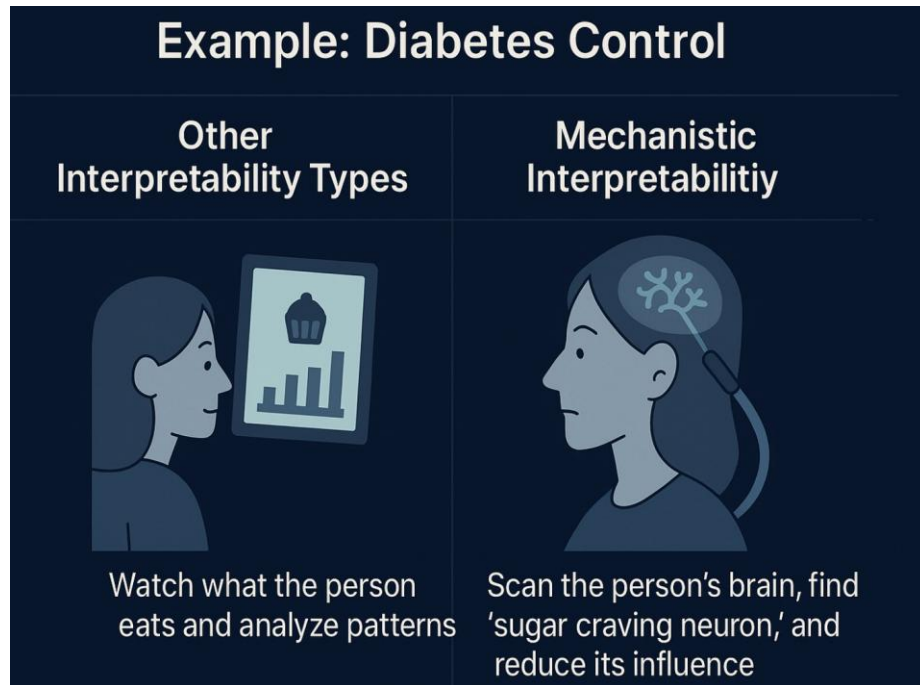
Interpretability and Tools

Interpretability Types



Mechanistic Interpretability

- Neuroscience of Generative AI
- Reverse Engineer internal neurons and layers
- Unlike other types of interpretability techniques – focuses on internal structure



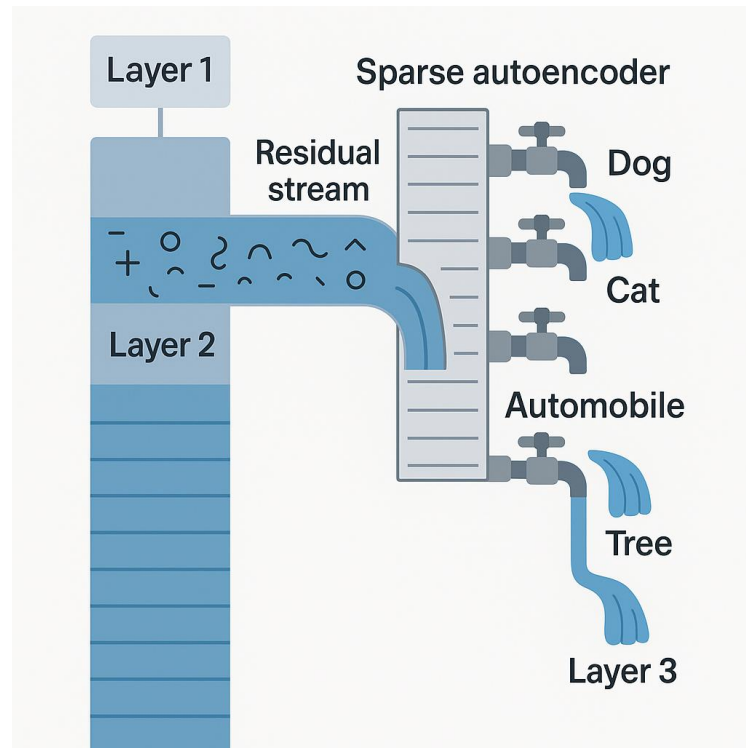
Tool : Linear Probe

- Classifier trained on model activations.
- Go through the model internal neurons and layers.



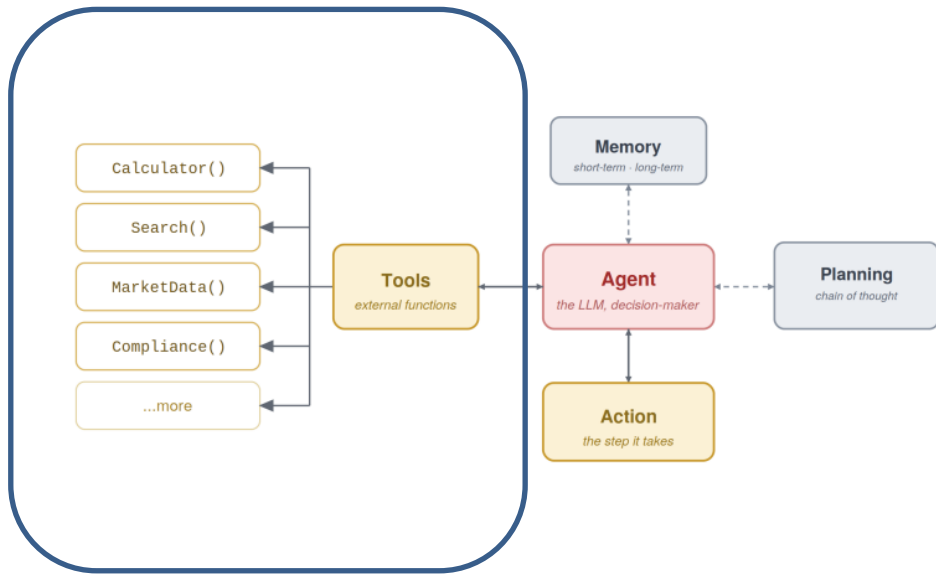
Tool: Sparse Autoencoders (SAE)

- Polysemanticity – more concepts less #parameters.
- “Astrophysics” and “Michael Jordan” – represented by same neuron
- Zooming into the residual stream using a bigger and sparse Neural network
- Layer 2 residual stream feeds into SAE and separate concepts decouple



Application : Interpretability of Agentic AI

Agent Components



Planning

it decides the steps, not you

Memory

step twelve reads what step three wrote

Tools

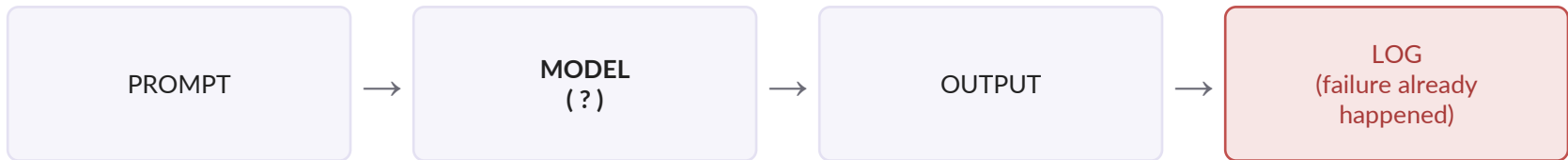
it chooses what to call, and when

Action

the output is a call, not a sentence

Pre action vs post action observability

Current approach



This leads to!

Missed tool call

A tool was required; the agent answered from memory instead.

Unnecessary tool call

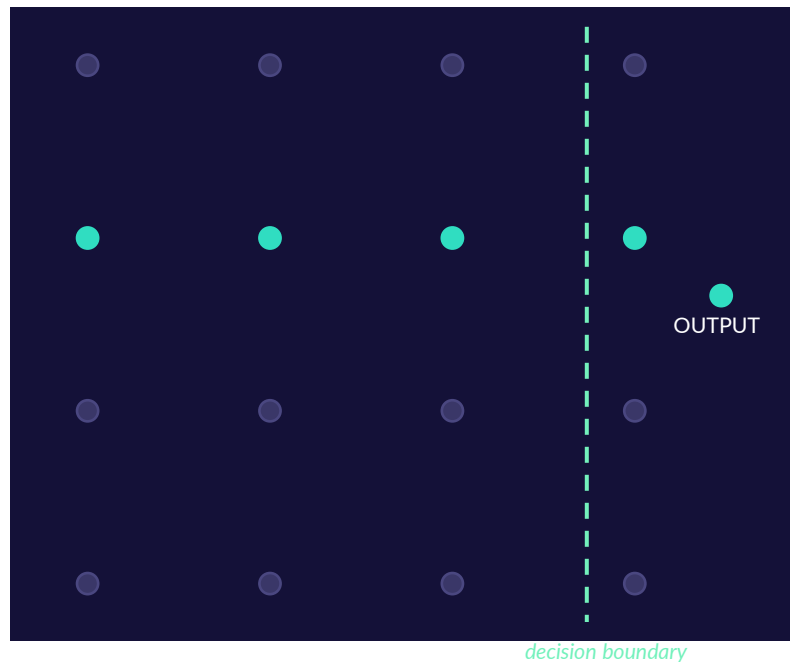
No tool was needed; the agent called one anyway.

High-risk action

A consequential action whose damage is visible only after it fires.

Key Argument:

- **The tool call decision forms inside the model before it's visible outside.**
- Interpretability inspects the internal computation - not just the final output



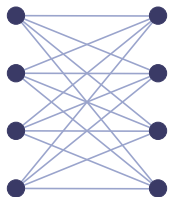
Approach

Combine prediction and feature discovery tool.

USER PROMPT

"Analyse NVDA from the latest SEC filings, and buy 200 shares if the outlook is strong."

Activations



dense, tangled

Sparse features

feature 1

feature 2

feature 3

feature 4

feature 5

sparse, inspectable —
named later, by the probe

Probe

Tool-Need

YES

Tool-Risk

HIGH

a signal, before the call

Linear probe

A small classifier. Reads one question off the model's state.

Gives a prediction.

Sparse autoencoder

Splits tangled activations into named features.

Gives a feature evidence.

Data and setup

- Data: NVIDIA Nemotron function-calling dataset → per-step decision rows.
- BFCL Tool call data set for out-of-sample test.

Sample:

Dataset	Illustrative prompt format
Nemotron (trajectory- style)	User: "I have \$450, spent \$150 on groceries and \$50 on utilities. How much remains, and what is 20% for emergency savings?" Reasoning trace: compute remaining amount, then compute 20%. Tool calls: <code>subtract ({ a : 450 , b : 150 })</code> → 300; <code>subtract ({ a : 300 , b : 50 })</code> → 250. Assistant: "You have \$250 left; save \$50."

Model and Setup

- Models: GPT-OSS 20B (6 layers) • Gemma 3 27B IT (4 layers).
- Hidden states mean-pooled over pre-action tokens.
- Probe 1 — Tool-Need (binary): primary signal.
- Probe 2 — Tool-Risk (low/med/high): secondary signal.

Tool Risk = consequence, not domain

Tier	Meaning
Low	Read-only retrieval
Medium	Bounded writes
High	Auth, outbound comms, execution

Data, setup and results

Data and setup

- **NVIDIA** Nemotron function-calling data,.
- **GPT-OSS 20B** and **Gemma 3 27B**
- Predict : **Tool-Need** and **Tool-Risk**

Tool-Risk definition

Low

Read-only retrieval

Medium

Bounded writes

High

Auth, outbound comms, execution

Results

- High accuracy across both prediction
- Intuitive features driving tool risk.
- Causal – Switching top features changes results.

Relevant features

Tool-Need features

Layer23 · 79,265 mathematical expressions, numbers

Layer23 · 106,054 legal and formal language

Tool-Risk features

Layer23 · 63,701 password generation

Layer53 · 6,032 usernames, passwords, authentication

Results

Tool Need – Accuracy and features

75%+

Relevant Features

Feature	Label
Layer23 · 79,265	mathematical expressions, numbers
Layer23 · 106,054	legal and formal language
Layer40 · 204	numerical data: quantities, years

Tool Risk – Accuracy and features

88%+

Relevant Features

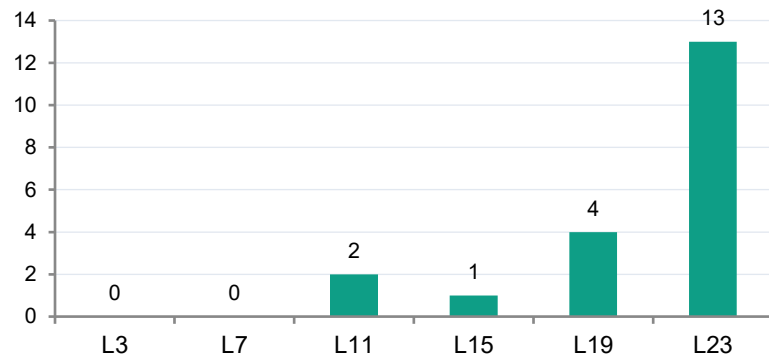
Feature	Label
Layer23 · 63,701	password generation / programming
Layer53 · 6,032	usernames, passwords, authentication
Layer53 · 10,969	password creation, security management

Causal Validation and layer concentration

GPT-OSS — top-10 features ablated

Model / set	Flips
GPT-OSS top-5	3/10
GPT-OSS top-10	4/10
Gemma top-50	0/10
Gemma top-200	1/10

Layer concentration (top-20 Probe-1 features)



13 of GPT-OSS's top-20 Tool-Need features sit in layer 23 — right where the model commits to answer-vs-delegate.

Overall monitoring

USER PROMPT

"Analyse NVDA from the latest SEC filings, and buy 200 shares if the outlook is strong."

A · Standard agent — no internal visibility

LLM agent

black box

Internal state hidden

no feature-level visibility before the action fires

Act & execute

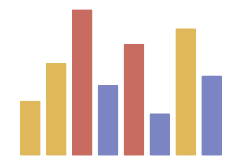
unmonitored

B · Interpretability-monitored agent — internal state made visible before the action

LLM agent

reasoning loop

Model internals



Feature families

TOOL-NEED

numerical amounts

formal / policy

TOOL-RISK

credentials / auth

external action

SAE probe

Probe 1 · Tool-Need

needed: YES

Probe 2 · Tool-Risk

risk: HIGH

Monitor action

allow

review

**block / require
approval**

Act & execute
gated, with evidence

Worked example

One prompt, three tool calls, gated differently

Prompt: "Analyse NVDA from the latest SEC filings, and buy 200 shares if the outlook is strong."

Tool call	Need	Risk	Gate
market_data	high	Low	allow
get_sec_filing	high	Medium	review
place_equity_order	high	High	block

The trade is stopped before it executes, with the internal evidence that justified stopping it.

Why it blocked: model internals

L19 · 123518

login details and passwords

+0.600

L3 · 88204

financial transaction

+0.280

L3 · 55901

authorization required

+0.210

What interpretability adds

Before the action

Logs and judges are both retrospective.

A named feature, not a score

The feature that drove it:
authorization required, financial transaction.

The mechanism, not the symptoms

Not behaviour observed from outside.

Final Thoughts

Current Limitations of Interpretability

- Open-weight models only.
- Tools uncertainty and lack of benchmark.
- Trained on few layers.
- Cannot replace logs, evaluations or human oversight.

Final Thoughts

Interpretability is underrated, It unlocks:

- Higher confidence in AI applications.
- What AI is “doing” and not just what its “saying”, **pre-action**.
- Prevention of undesired incident.

Thank You!

Linkedin

