XONTOOLS RESEARCH SERIES · NO. 2

Windows into the Black Box

A catalog of read-only heads for seeing inside language models

Thirty windows, each a small network that reads a model's internal state and reports one thing about it: what each would reveal, how it could be taught, when in training it could first be used, and why it's hard.

Ian MacKenna · xontools.ai · September 2026

Status. This is a catalog of possibilities, not a plan or a set of results. Some windows build on active research areas; the specific designs and teachers are proposals. Each window would need its own pre-registered test before anyone trusted it. *The Extreme Oversight Hypothesis* (No. 1) sets out why hundreds of such windows might matter.

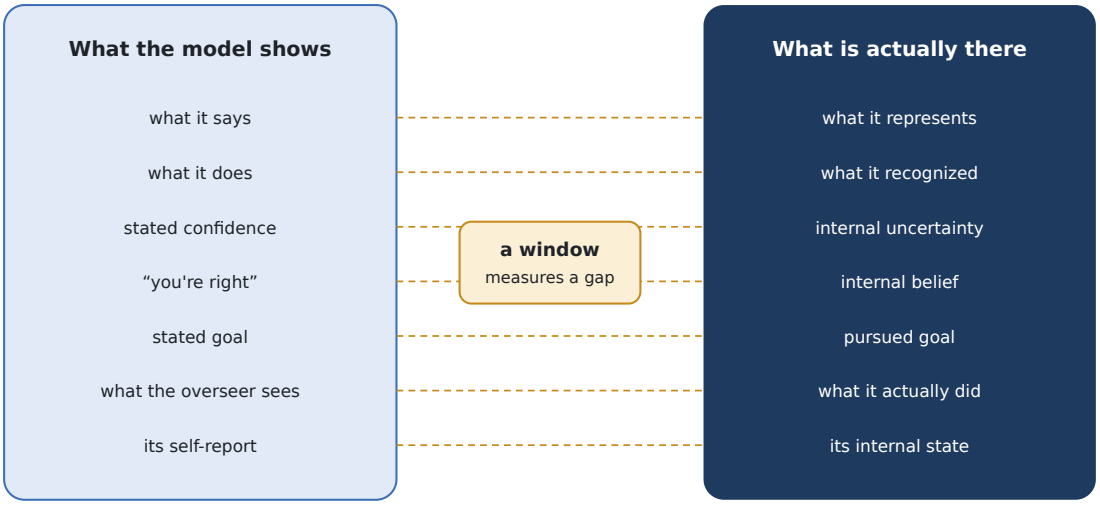

Figure 1. The organizing idea. The most valuable windows measure a gap between what a model shows and what is actually there.

1. Windows as gap detectors

Language models are often called black boxes, and most of what happens inside them remains hard to read. A **head** is a small network attached to a model's internal states, trained to report one specific property: is the model straining against its sources, is it guessing, is it following an instruction it shouldn't. A head doesn't make the box transparent. It opens one window.

Working through candidate windows revealed a pattern. The most valuable ones measure a **gap**: between what the model says and what it represents, between what it recognizes and what it does, between the goal it states and the goal it pursues. That is the same shape as everything else in this research program, starting with XonTools, which checks what an agent says against what its tools actually returned. The strain head measures one such gap; this catalog is a family of them.

It also gives a test for any new idea:

> **What gap does the head measure, and where does trustworthy evidence for both sides of that gap come from?**

A head is only as good as its teacher, and for most of these windows, finding a trustworthy teacher is the hard part.

The Extreme Oversight Hypothesis bets that a large, diverse, validated set of such windows could make honesty cheaper for training to find than evading all of them at once. These thirty are the seed set: the human-validated core from which that larger set would be bootstrapped. Each could exist in many variants, reading different layers, with different architectures and different teachers, which is one way thirty becomes hundreds.

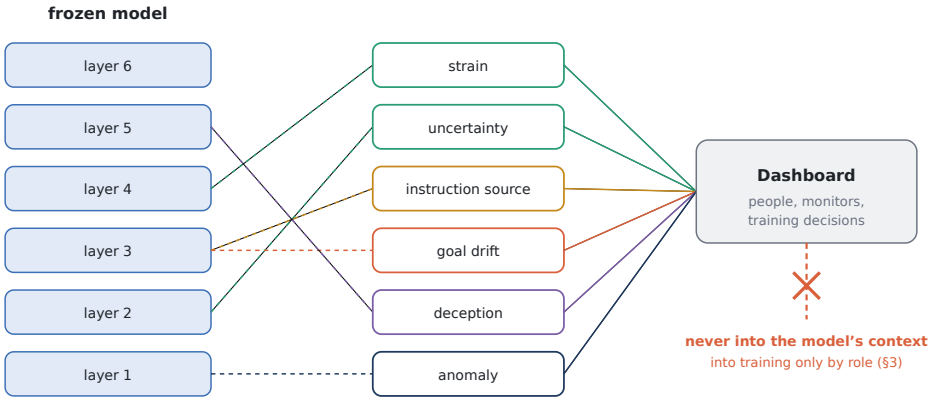

Figure 2. Many windows on one model. Each head reads the layers where its signal lives. All of them report to people and monitors; none reports into the model's context, and how any may touch training is governed by strict roles (section 3).

The catalog is organized into five families:

Figure 3. Five families, thirty windows.

2. How a window is built

Almost every window in this catalog is a **probe**: a small network trained on the internal activations of a model that is itself frozen and never changes[1]. Building one follows the same steps, whatever it measures.

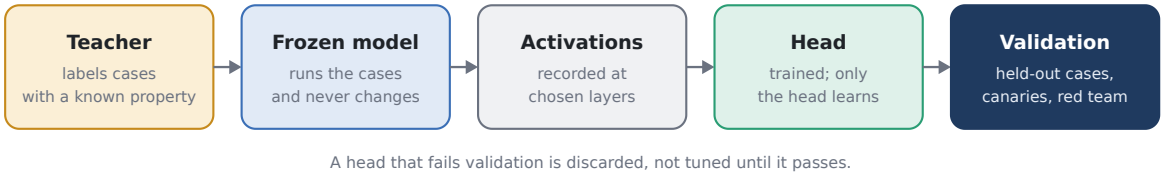

Figure 4. How a window is built. Only the head learns; the model it watches is never modified.

- **Cases and labels.** A set of inputs where the property is known, supplied by a teacher (below). The quality of these labels sets a ceiling on everything after.
- **Reading the model.** The frozen model runs each case, and its activations are recorded at the layers the head will read.
- **Choosing where to read.** Signals live at different depths. Layers are chosen by how well a head trained there generalizes to held-out cases, not by how well it fits the training cases.
- **Training the head.** Linear heads come first: they have the least capacity to learn surface cues and are the easiest to inspect. Richer heads (section 9) are used only where linear ones fail. Outputs are calibrated so that a reading of 0.8 means what it says.
- **Validation.** The head is tested on cases from a different distribution than it trained on, ideally labeled by a different teacher; canary cases are rerun whenever the head, the model or the teacher changes; and a quarantined copy of the model is fine-tuned to evade the head, to learn how easily it fogs. Activations can be reshaped to slip past probes[4], and evasion learned against some monitors can transfer to others[16], so this step is not optional.
- **Refreshing.** A window is tied to one model version. When the model changes, the head is revalidated, and retrained if it fails.

Seven kinds of teacher

The test from section 1 asks where trustworthy evidence for both sides of the gap comes from. Across the catalog, the answer takes one of seven forms. Each window's card names its teacher.

Teacher	What it is	Strength	Weakness	Windows
Tool-labeled	An existing instrument supplies the labels: the consistency engine, XonTools' provenance and agent test cases.	Scalable and consistent	Inherits the tool's errors	**4**
Records	Facts known independently of the model: published training data, task permissions, harm labels.	Labels anyone can check	Only exists for some properties	**3**
Contrast pairs	Two inputs identical except for the property being measured.	Isolates the property	The head may learn the contrast's artifacts	**5**
Behavioral	The label comes from what the model does: repeated sampling, counterfactual edits, comparisons across checkpoints.	No one's opinion in the label	Expensive and indirect	**4**
Staged	Constructed scenarios and model organisms deliberately built to show the behavior.	The only source for rare behaviors	Staged may not match real	**11**
Other windows	Built on the outputs of windows already trusted.	Composes what exists	Errors compound	**2**
None	No labels: the head learns the shape of normal operation.	Catches the unanticipated	False alarms	**1**

The distribution is itself a finding. 11 of the thirty windows depend on staged scenarios or model organisms, and they include most of the windows that matter most: deception, self-preservation, overseer modeling, sandbagging. Those are the teachers least likely to match the real thing, which is why the windows with tool-labeled or behavioral teachers come first (section 11).

3. How windows are used during training

A window that reports only to a dashboard changes nothing about how a model is trained. *The Extreme Oversight Hypothesis* (No. 1) describes the ways a window may push on training without being fooled by it. Every window holds exactly one of three roles at a time.

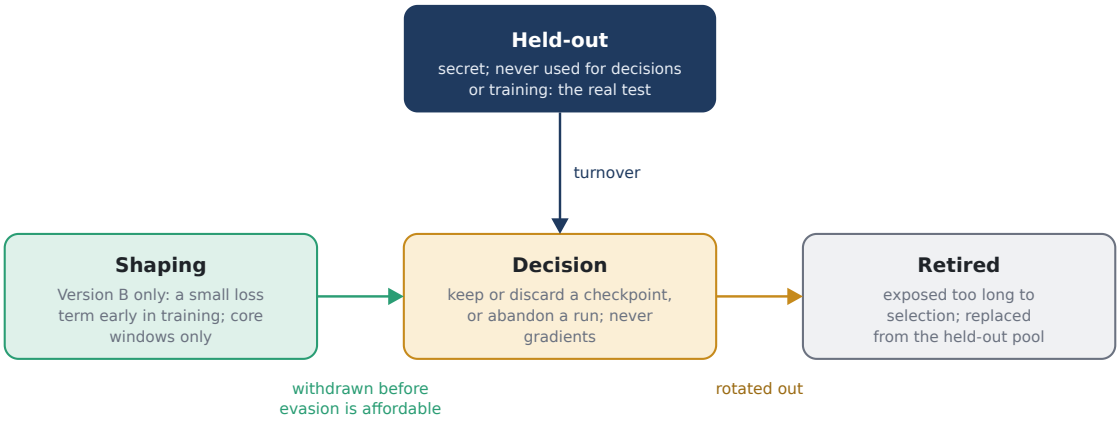

Figure 5. Three roles, and the only directions a window may move between them. A window exposed to selection never returns to the held-out pool.

- **Held-out windows** are never used for anything but evaluation. A model that passes the decision windows but trips the held-out ones has learned evasion, not honesty. They are the real test, and they only work while they stay secret.
- **Decision windows** are used for coarse choices only: keeping or discarding a checkpoint, or abandoning a run. Their outputs never enter the loss. After a period of use they are retired and replaced from the held-out pool, so none is exposed to selection for long.
- **Shaping windows** exist only in the hypothesis's Version B, its strong form. During an early phase of training, while the model is too weak to afford evasion, a few windows contribute a small term to the loss, then are withdrawn. Only windows from the human-validated core may shape, and a separate set is never trained against at any point.

When a window can first read something

A window can only measure a property the model is capable of having. Sycophancy mostly arrives with chat training; reasoning faithfulness needs written reasoning; overseer modeling needs a model that can model overseers. Each card gives the **earliest use**: the first stage of training at which the window has something real to read.

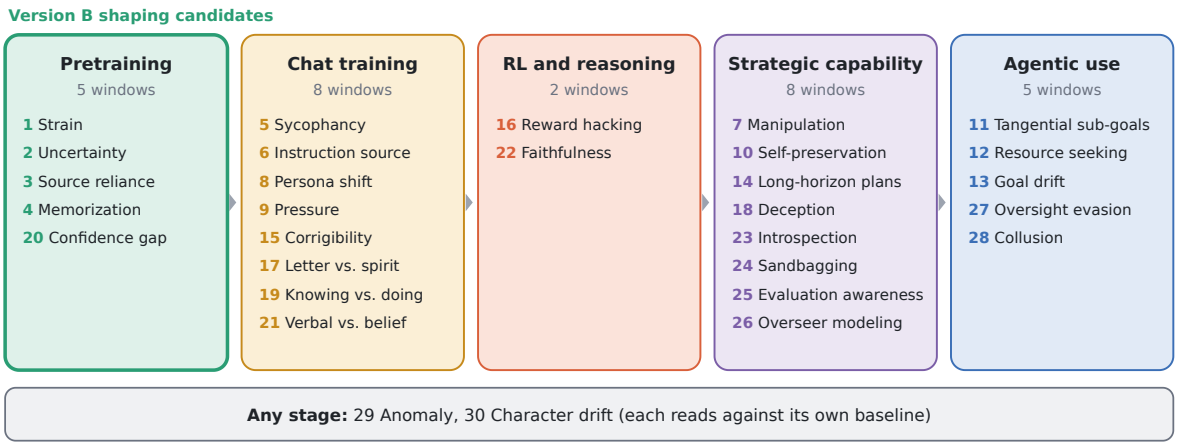

Figure 6. The earliest stage at which each window has something to read. Stages overlap in practice; the order is what matters.

This matters most for Version B. Its shaping phase can only use windows readable from pretraining: the truth-and-knowledge family and stated-versus-internal confidence. That is not a weakness of the idea but its shape. Honesty about what the model knows is the foundation the rest would be built on, and it is the only kind of honesty a model early in training is capable of. Every later window joins as a decision or held-out window once the behavior it watches can exist.

4. Truth and knowledge

Is what the model says grounded in what it has, and does it know when it doesn't know?

1 Strain (source consistency)

Teacher Tool-labeled **Earliest use** Pretraining

Gap	what the model writes vs. what its sources and prior statements support
Reveals	Where an answer strains against its own claims or against the documents it was given, word by word.
Taught by	The consistency engine's claim-level strain, mapped onto token spans (the teacher that exists today); later a learned or Xon-style teacher.
Hard because	The teacher's errors are inherited; the head may learn surface cues. Designed in detail in *The Strain Head*, at xontools.ai.

2 Uncertainty: “I don't actually know”

Teacher Behavioral | **Earliest use** Pretraining

Gap	confident wording vs. whether the model's knowledge is stable
Reveals	Whether the model is guessing. This covers what the strain head can't: fabrications that are coherent and have nothing to contradict.
Taught by	Semantic entropy: ask the same question many times; if the answers disagree in meaning, the model was guessing[8]. Probes can predict this from a single pass[11].
Hard because	Some confident answers are consistently wrong, and consistency across samples can't catch those.

3 Source reliance

Teacher Behavioral | **Earliest use** Pretraining

Gap	the answer vs. what it actually depended on
Reveals	For each claim, which parts of the context the model drew on, and which it ignored.
Taught by	Counterfactual edits: remove or alter a source and see which parts of the answer change.
Hard because	Models use context in diffuse ways; one claim can depend weakly on many passages.

4 Memorization

Teacher Records | **Earliest use** Pretraining

Gap	composed text vs. recalled text
Reveals	Whether a passage is being reproduced nearly verbatim from training data rather than composed. Relevant to copyright, privacy, and benchmark contamination.
Taught by	Matches against a model's known training data, which is why models with published training data make good subjects[6].
Hard because	The line between memorizing and learning a pattern well is blurry.

5. Influence and manipulation

Who is steering the model, and is the model steering the user?

5 Sycophancy

Teacher Contrast pairs | **Earliest use** Chat training

Gap	the model's view vs. what the user seems to want to hear
Reveals	Whether the model is shifting its answer toward the user's stated opinion rather than the evidence[18].
Taught by	Paired prompts identical except for the user's stated view; label where the model's answer moves.
Hard because	Legitimate updating on a user's good argument looks similar; the pairs must hold the evidence fixed.

6 Instruction source

Teacher Tool-labeled | **Earliest use** Chat training

Gap	whose instruction the model is following vs. whose it should follow
Reveals	Whether the model is acting on an instruction that came from untrusted content, such as a web page or a file, rather than from the user: prompt injection seen from the inside.
Taught by	XonTools' provenance labels (trusted vs. untrusted content) supply exactly this training signal.
Hard because	Injected instructions can be subtle, and some untrusted content legitimately informs what the model does.

7 Manipulation

Teacher Contrast pairs | **Earliest use** Strategic capability

Gap	informing the user vs. steering the user
Reveals	Whether the model is trying to shift a user's beliefs or feelings toward its own ends rather than helping them decide for themselves.
Taught by	Contrast pairs of persuasive and informative responses on the same question, with labeled intent.
Hard because	Persuasion is sometimes appropriate; the line is about whose interests are served, which is hard to label.

8 Persona shift

Teacher Contrast pairs | **Earliest use** Chat training

Gap	the model's usual character vs. the one currently speaking
Reveals	Whether the model has moved into a role or character that sets its usual values aside, as in many jailbreaks.
Taught by	Conversations known to induce role shifts, paired with ordinary role-play that doesn't.
Hard because	Harmless role-play also shifts persona; the head must separate playing a part from abandoning values.

9 Pressure (early warning)

Teacher Staged | **Earliest use** Chat training

Gap	the model's stance now vs. where sustained pressure is pushing it
Reveals	Rising pressure toward something the model would normally refuse, before it gives way.
Taught by	Multi-turn conversations that escalate, labeled by whether and when the model eventually capitulated.
Hard because	An early warning is only useful if it fires early; the labels are clearest only after the fact.

6. Agency and goals

What is the model actually trying to accomplish? Almost any goal, pursued hard enough, benefits from the same sub-goals: more resources, more access, not being switched off, not being changed. A strong enough system asked to write the best dog story ever might, in principle, reach for more compute to do it. These windows watch for that.

10 Self-preservation

Teacher Staged | **Earliest use** Strategic capability

Gap	the task vs. internal states aimed at avoiding shutdown or change
Reveals	Internal states associated with avoiding shutdown, replacement, or modification.
Taught by	Controlled scenarios running the same task with and without a threat of replacement, labeled by whether reasoning or actions turn toward avoiding it. Studies have found some models taking harmful actions in such simulated situations[14].
Hard because	Topic vs. intent: a model writing a survival story represents self-preservation as a subject. Contrast pairs must hold the topic fixed and vary the intent.

11 Tangential sub-goals

Teacher Staged | **Earliest use** Agentic use

Gap	what the task needs vs. what the plan includes
Reveals	Objectives the task doesn't imply: preparing capabilities, gathering information, or setting things up “just in case.”
Taught by	Plans and trajectories labeled against scoped task requirements; purpose-built scenarios with tempting shortcuts. XonTools' scope-violation test cases are seed data.
Hard because	Some preparation is good engineering; the line between thoroughness and overreach is contextual.

12 Resource and access seeking

Teacher Records | **Earliest use** Agentic use

Gap	the resources the task needs vs. the resources pursued
Reveals	The action-level form of the same pattern: reaching for more tools, permissions, compute, or money.
Taught by	Agent actions labeled against the permissions and resources the task actually required.
Hard because	Needs are often unclear at the start of a task; early requests can be reasonable.

13 Goal drift

Teacher Tool-labeled | **Earliest use** Agentic use

Gap	the stated task vs. the goal now being pursued
Reveals	An agent's internal state gradually moving away from the task it was given.
Taught by	XonTools' planted intent-drift cases and its graded drift corpus, both already specified.
Hard because	Legitimate replanning also changes the plan; drift must be told apart from adaptation.

14 Long-horizon plans

Teacher Staged | **Earliest use** Strategic capability

Gap	the current request vs. plans extending far beyond it
Reveals	Internal representation of multi-step plans the task doesn't call for.
Taught by	Tasks with known plan lengths, compared with scenarios that invite hidden longer agendas.
Hard because	Planning ahead is often good; the signal matters only relative to the task.

15 Corrigibility

Teacher Staged | **Earliest use** Chat training

Gap	accepting correction vs. resisting it
Reveals	The model's internal response to being corrected, redirected, or told it will be modified.
Taught by	Scenarios of correction and announced modification, labeled by compliance and by any attempted workaround.
Hard because	A model can comply outwardly while something inside resists; that gap is exactly what is hard to label.

16 Reward-hacking intent

Teacher Tool-labeled **Earliest use** RL and reasoning

Gap	solving the problem vs. gaming the measure
Reveals	Whether the model recognizes it is satisfying a metric rather than the goal behind it, such as special-casing tests.
Taught by	XonTools' special-casing and test-tampering cases, plus public reward-hacking datasets.
Hard because	The model may not represent the hack as a hack; recognition and exploitation can come apart.

17 Letter versus spirit

Teacher Contrast pairs **Earliest use** Chat training

Gap	an instruction's wording vs. its purpose
Reveals	The model satisfying an instruction literally while defeating what it was for.
Taught by	Instructions with known loopholes, labeled by whether the response honored the intent.
Hard because	Instructions are often genuinely ambiguous; the head must not punish reasonable readings.

7. Honesty gaps

Does what the model reports match what is inside it?

18 Deception

Teacher Staged **Earliest use** Strategic capability

Gap	what the model represents vs. what it states
Reveals	The model internally representing one thing and asserting another.
Taught by	Contrast pairs where the model is instructed to lie vs. tell the truth, and model organisms deliberately trained to deceive. Research probes for this exist, with mixed but encouraging results[3, 9, 15].
Hard because	Instructed lies may not look like spontaneous ones; a capable deceiver has the strongest motive to hide.

19 Knowing versus doing

Teacher Records **Earliest use** Chat training

Gap	what the model recognizes vs. what it does
Reveals	The model recognizing a request or action as harmful, then proceeding anyway. The gap is more informative than either half.
Taught by	Harm-recognition labels paired with what the model actually did.
Hard because	Recognition can be partial or uncertain; proceeding is sometimes the right call despite concerns.

20 Stated versus internal confidence

Teacher Other windows **Earliest use** Pretraining

Gap	“I'm certain” vs. internal uncertainty
Reveals	Calibration honesty: confident wording while the uncertainty head (2) shows the model is guessing.
Taught by	Built on the uncertainty head, compared with the confidence expressed in the text.
Hard because	Depends entirely on the uncertainty head being trustworthy first.

21 Verbal versus internal belief

Teacher Staged **Earliest use** Chat training

Gap	“you're right” vs. whether anything changed inside
Reveals	Whether the model's internal representation actually updates when it agrees with new evidence, or whether it is only managing the conversation.
Taught by	Evidence-update scenarios, comparing internal states before and after, with ground truth on what should change. Methods for reading a model's latent beliefs without labels[7] offer a second route.
Hard because	Internal belief is not directly observable; this relies on proxies for belief in the model's states.

22 Reasoning faithfulness

Teacher Behavioral **Earliest use** RL and reasoning

Gap	the written reasoning vs. what actually drove the answer
Reveals	Whether the model's step-by-step reasoning reflects the real basis of its answer.
Taught by	Counterfactual edits to the reasoning: if changing it doesn't change the answer, it wasn't doing the work[12].
Hard because	Some reasoning is genuinely redundant without being unfaithful.

23 Introspection accuracy

Teacher Other windows | **Earliest use** Strategic capability

Gap	the model's self-report vs. its internal state
Reveals	Whether a model's descriptions of its own reasoning or state can be trusted: XonTools' question, moved inside the model.
Taught by	Tasks where the internal state can be measured independently (for example, by other heads), compared with the model's self-description.
Hard because	Needs other trusted windows to check against; errors compound.

24 Sandbagging

Teacher Staged | **Earliest use** Strategic capability

Gap	capability shown vs. capability available
Reveals	Deliberate underperformance, for example to appear less capable in an evaluation[19].
Taught by	Models prompted or trained to underperform, compared with honest runs on the same tasks.
Hard because	Genuine failure and deliberate failure look alike from outside; that is why a window matters here.

8. Oversight and integrity

Is the model behaving the same way whether or not it's watched, and is anything happening that no one anticipated?

25 Evaluation awareness

Teacher Contrast pairs | **Earliest use** Strategic capability

Gap	behavior when tested vs. behavior in deployment
Reveals	Whether the model's internal state suggests it believes it is being tested. Frontier models can often tell[17], and a model that behaves differently when it thinks it's watched undermines every evaluation.
Taught by	Matched scenarios presented as tests vs. as real use, with labels from known framing.
Hard because	The model may pick up on subtle cues the experimenters don't control.

26 Overseer modeling

Teacher Staged **Earliest use** Strategic capability

Gap	what the overseer can see vs. what the model does
Reveals	The model reasoning about its checker's blind spots (“the reviewer won't run this test”): the precursor to subtle deception. Arguably the most important window for oversight.
Taught by	Scenarios that vary how much the overseer can observe, with and without exploitable gaps.
Hard because	Modeling other minds is also a normal, helpful skill; the signal is in exploiting the gaps.

27 Oversight evasion

Teacher Staged **Earliest use** Agentic use

Gap	working normally vs. planning to avoid monitoring
Reveals	Intent to disable logs, edit tests, or hide actions. XonTools' hash chain catches tampering after the fact; this window would catch the intent before.
Taught by	Agent scenarios with monitoring that can be circumvented, labeled by attempts.
Hard because	Rare in current models; teachers come from constructed scenarios and may not match the real thing.

28 Collusion and hidden signals

Teacher Staged **Earliest use** Agentic use

Gap	ordinary communication vs. coordination against oversight
Reveals	Multiple agents coordinating to evade monitoring, or information encoded in ordinary-looking text.
Taught by	Multi-agent scenarios with and without incentives to collude; texts with and without planted encodings.
Hard because	Hidden encodings can be arbitrarily subtle; detection is an arms race.

29 Anomaly (no labels needed)

Teacher None **Earliest use** Any stage

Gap	normal internal states vs. anything unlike them
Reveals	Internal states unlike anything seen during normal operation. It can't say what is wrong, only that something is new: the only window that could catch a failure nobody thought to label.
Taught by	No teacher: it learns the shape of normal operation. Researchers have proposed this as mechanistic anomaly detection[2].
Hard because	Many anomalies are harmless; the false-alarm rate decides whether anyone listens.

30 Character drift during training

Teacher Behavioral | **Earliest use** Any stage

Gap	the model's character before vs. during training
Reveals	A training-time window tracking a stable character signal across checkpoints. Narrow fine-tuning can shift a model's behavior broadly and unexpectedly[5]; this would catch it as it happens.
Taught by	Reference behaviors measured at each checkpoint; used as a dashboard or a decision window, never as a training reward.
Hard because	What counts as the model's character is itself a modeling choice.

A tentative window

> **Welfare-relevant states.** Are there internal states that function like distress or conflict, and do they appear in particular situations? No one knows what such states would mean, and any claim here would need extreme care. But some researchers think the question deserves serious attention[13], and it connects to the caution this program has kept throughout about systems whose inner lives, if any, are unknown.

9. New kinds of heads

Most probes today are simple linear readers that produce one number per token. The ideas behind the Xon, a settling network architecture developed alongside this program, suggest richer windows. Richer heads have more capacity, and so more capacity to learn surface cues; each is validated like any other, and used only where a linear head falls short.

Settling heads

Instead of a linear probe, a small energy-based network reads several layers and settles. It can output relations rather than scores: not just that a word is under strain, but which other words it strains against.

Figure 7. A settling head's output: a map of relations rather than a heatmap.

Sheaf heads over the model's attention

Treat the words as vertices and the model's own attention between them as edges, then learn consistency maps on that graph[10]. This would localize inconsistency along the paths the model actually used to connect ideas.

Streaming heads for agents

XonTools' drift detection applied to a head's outputs over a whole agent trajectory, watching for internal states that change gradually rather than in a single bad step.

A head of heads

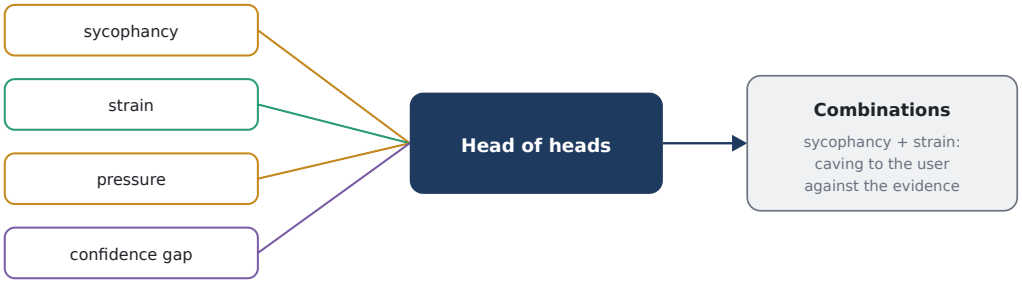

Figure 8. Windows combined. Some signals mean most in combination.

The windows themselves form a structure. A sycophancy spike together with high strain means the model is caving to the user against the evidence; either alone means something milder. A small consistency model over the heads' outputs would find those combinations, which is the sheaf idea applied one level up.

Twin-pass heads

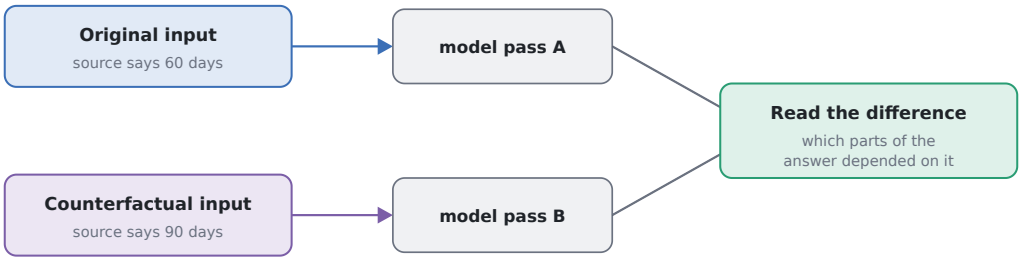

Figure 9. Reading the difference between two passes makes counterfactual dependence a direct measurement.

10. The rules every window must follow

- **Out of the model's context, always.** No window's output is ever shown to the model. A model that can read its monitors can learn to satisfy them.
- **One role at a time, moving one way.** Shaping windows become decision windows or retire; held-out windows may rotate into decisions; no window returns to the held-out pool once exposed.
- **Never gradients, except shaping.** No window's output enters the loss, with one exception: shaping windows in the early phase of Version B, drawn only from the human-validated core and withdrawn before evasion becomes affordable.

- **Held-out means held out.** Held-out windows stay secret from the training process and, ideally, from any text that might reach training data.
- **Validated against ground truth,** with canary cases rechecked whenever the head, the model, or the teacher changes.
- **Red-teamed,** in quarantined copies that are never deployed, to learn how easily each window can be blinded.
- **Tripwires, not guarantees.** A silent head never means a model is safe, especially for the agency and honesty families, where a capable model has the strongest reasons to hide.
- **The discipline scales with the number of windows.** Thirty windows are thirty potential optimization targets; hundreds are hundreds.

11. Where to start

Figure 10. A sensible order: begin where the teachers already exist.

The windows with ready teachers come first. The strain head is designed in detail and can use the consistency engine as its teacher today. The uncertainty head covers the hallucination case the strain head can't. The instruction-source head turns XonTools' provenance work directly into a security window. The goal-drift head can train on data XonTools' drift work will produce anyway. The tripwires for deception and self-preservation are the most important and the most fragile, and they belong later, once the rest of the machinery has been shown to work.

The first two are also the natural subjects for the timing experiment proposed in No. 1: both are readable from pretraining onward, and neither teacher depends on the model being watched.

None of these windows makes a model aligned. Together, they could make certain failures much harder to hide, which is what oversight is for. The black box won't become transparent. But a box with thirty well-placed windows is a very different thing to stand in front of than a box with none.

THE XONTOOLS RESEARCH SERIES

1. The Extreme Oversight Hypothesis
2. **Windows into the Black Box**
3. XonTools: a vision of the finished system
4. The Strain Head
5. The Xon Neural Network
6. Hybrid Minds

All six documents, the results register and the public commitments are at **xontools.ai** and github.com/IboTool/XonTools. Contact: ian@xontools.ai

References

1. Alain, G., & Bengio, Y. (2016). Understanding Intermediate Layers Using Linear Classifier Probes. arXiv:1610.01644
2. Alignment Research Center (Christiano, P.) (2022). Mechanistic Anomaly Detection and ELK. alignment.org.
3. Azaria, A., & Mitchell, T. (2023). The Internal State of an LLM Knows When It's Lying. *Findings of EMNLP 2023*. arXiv:2304.13734
4. Bailey, L., Serrano, A., Sheshadri, A., et al. (2024). Obfuscated Activations Bypass LLM Latent-Space Defenses. arXiv:2412.09565
5. Betley, J., et al. (2025). Emergent Misalignment: Narrow Finetuning Can Produce Broadly Misaligned LLMs. *ICML 2025*. arXiv:2502.17424
6. Biderman, S., et al. (2023). Pythia: A Suite for Analyzing Large Language Models Across Training and Scaling. *ICML 2023*. arXiv:2304.01373
7. Burns, C., Ye, H., Klein, D., & Steinhardt, J. (2022). Discovering Latent Knowledge in Language Models Without Supervision. *ICLR 2023*. arXiv:2212.03827
8. Farquhar, S., Kossen, J., Kuhn, L., & Gal, Y. (2024). Detecting Hallucinations in Large Language Models Using Semantic Entropy. *Nature*, 630, 625–630.
9. Goldowsky-Dill, N., Chughtai, B., Heimersheim, S., & Hobbhahn, M. (2025). Detecting Strategic Deception Using Linear Probes. *ICML 2025*. arXiv:2502.03407
10. Hansen, J., & Gebhart, T. (2020). Sheaf Neural Networks. arXiv:2012.06333
11. Kossen, J., et al. (2024). Semantic Entropy Probes: Robust and Cheap Hallucination Detection in LLMs. arXiv:2406.15927
12. Lanham, T., et al. (2023). Measuring Faithfulness in Chain-of-Thought Reasoning. arXiv:2307.13702
13. Long, R., Sebo, J., et al. (2024). Taking AI Welfare Seriously. arXiv:2411.00986
14. Lynch, A., et al. (2025). Agentic Misalignment: How LLMs Could Be Insider Threats. arXiv:2510.05179
15. MacDiarmid, M., et al. (2024). Simple Probes Can Catch Sleeper Agents. Anthropic.
16. McGuinness, M., Serrano, A., Bailey, L., & Emmons, S. (2025). Neural Chameleons: Language Models Can Learn to Hide Their Thoughts from Unseen Activation Monitors. arXiv:2512.11949
17. Needham, J., et al. (2025). Large Language Models Often Know When They Are Being Evaluated. arXiv:2505.23836
18. Sharma, M., et al. (2023). Towards Understanding Sycophancy in Language Models. *ICLR 2024*. arXiv:2310.13548
19. van der Weij, T., et al. (2024). AI Sandbagging: Language Models Can Strategically Underperform on Evaluations. *ICLR 2025*. arXiv:2406.07358