REDWOOD RESEARCH

# Proposal for tracking the effects of architecture on monitorability

Redwood Research • September 10, 2026

Architectures that incorporate opaque recurrence or allow for agents to communicate with each other using latents could rapidly make it much harder to monitor chains of thought or communication (we'll refer to this property as "monitorability" going forward).[1] As companies begin to explore such architectures, we believe it is important to transparently share evidence about how monitorability varies with architecture and training method. To inform the scientific debate on how to make tradeoffs between performance and monitorability, we believe AI companies should:

1. **Regularly report externally verified information about the degree to which their architectures may allow for latent reasoning or communication.** Companies should publicly disclose enough information about architectures to allow external scientists to determine whether they could potentially enable models to perform much more complex reasoning without this reasoning appearing in the chain of thought ("latent reasoning") or allow for latent communication between different instances of a model.

   Following GDM, we propose measuring *opaque serial depth* as a minimally-invasive proxy for the degree to which an architecture may enable latent reasoning, though companies could provide sufficient architecture transparency in other ways. We propose that companies work with third-party evaluators to produce independently verified reports of the rough distribution of opaque serial depth across all their *near-frontier* models.[2] External

[1] Chain-of-thought monitors should look at actions the model is taking in addition to their reasoning. The interaction between CoT monitoring and action monitoring could be important (e.g., if agents were allowed to reason freely in their CoT or in latent reasoning they could carefully reason through how to take more surreptitious actions, and if actions weren't monitored they could do the malign action in some very obvious way that doesn't require much reasoning).

[2] This methodology used to obtain this distribution should be published. The published depth distribution information would ideally include the largest approximate opaque serial depth among near-frontier models that see non-trivial usage and the approximate opaque serial depth for representative (as in, widely used internally) near-frontier models. The methodology the company is using to compute the depth should also be made public (with redactions for IP as needed) and used to compute and publish depths for open-weight models. (Any circuit transformations used to obtain a lower depth should also be applied to these open-weight models for fair comparison.) When reporting approximate depths, it could be best to report the depth as a multiple of the depth of some open-weight model (computed with the same methodology).

verification wouldn't require third parties seeing the architecture of these models; interviews with employees along with a process for employees to whistleblow to the third party could be sufficient.

By *near-frontier* models, we mean all general-purpose[3] internal models at least as capable as the best public models from six months ago.[4] Because it is important to understand the *relationship* between monitorability and architecture — and not just the monitorability of deployed models — the models covered in this survey should include purely internal R&D prototypes if they have near-frontier capabilities.

Opaque serial depth can also serve as an incomplete proxy for latent communication[5], but companies should ideally share additional high-level information about the architecture and structure of communication between agents (e.g., whether AIs communicate in latents, whether shared 'memories' are stored in latents, whether agents receive parts of the KV cache of other agents).

The relationship between well-elicited[6] no-CoT capabilities and overall capabilities could serve as an additional or alternative proxy measurement. More concretely, companies could build a dataset to measure no-CoT capabilities, project how much some change to architecture or training methods increases no-CoT ECI relative to how much that change increases overall ECI,[7] and work with third-parties to produce independently verified reports of the extent to which new methods might result in a disproportionately large increase in no-CoT ECI.[8] This reporting should have reasonable coverage over all near-frontier models.

[3] As in, AIs that can do a variety of tasks such as math, coding, and writing. This is intended to exclude specialized image generation models. An attachment to a general purpose AI which is solely for the purpose of image generation (and doesn't increase general purpose capability outside of image generation) can be excluded.

[4] It's important to cover research prototypes that haven't yet reached the frontier but are promising enough that they have been significantly scaled up. This is to ensure that public scientific understanding of monitorability can keep up with rapidly progressing research on new architectures. It is also cheaper for an AI company to stop further development of insufficiently-monitorable architectures early in the R&D process.

[5] In particular, companies could report the opaque serial depth of a larger computation which involves communication between multiple different AI models. One reason why this wouldn't be a complete measure is that various approaches for sharing parts of the KV cache between agents could result in concerning latent communication while only yielding small increases in depth.

[6] E.g., after significant fine-tuning on an IID set of tasks.

[7] Ideally, the measure of no-CoT capabilities would be especially focused on the most concerning no-CoT capabilities such as subversion capabilities or ability to perform latent communication between agents. While ECI is a reasonable baseline proposal for how to aggregate scores over multiple datasets, any metric where we have good reason to think there will be a relatively clean predictable trend given a similar increase in underlying capabilities could work.

[8] The change in no-CoT ECI versus overall ECI could be compared to the historical trend—how much overall ECI tends to increase for each increment of no-CoT ECI on the same datasets—as a proxy for how some change (or set of changes) relatively enables latent reasoning.

2. **Regularly share evidence about monitorability and have third parties review this.** Companies should publicly share their views about the monitorability of their near-frontier models and how this varies across architectures and training methods, along with any evidence or information they have about this.[9] Ideally, companies would run *monitorability stress tests* to see if models can be elicited or fine-tuned to perform reasoning without this being reflected in their chains of thought, or communicate with other models in ways that are hard to monitor.[10] An initial sketch of what stress tests of CoT monitorability could look like is in an appendix below.

   Companies should give third-party experts access to the unredacted results from these experiments, and third-party experts should:

   a. Assess whether these experiments and other evidence are accurately described.
   b. Run their own experiments to gather more evidence about CoT monitorability and how this varies across training methods and architectures, and to gather evidence about any other important claims companies made.
   c. Report their findings to the public, with redactions for IP as appropriate.

   As above, this should have reasonable coverage over models at least as capable as the best public models from six months ago, including internal R&D prototypes if they are near the frontier. Evidence sharing and third-party reviews should occur regularly (e.g., every 6 months) and potentially whenever a new architecture or training method results in a near-frontier model with significantly increased opaque serial depth.[11]

3. **Publish a policy regarding architectures that could enable greater latent reasoning, and provide information to help assess compliance.** Companies should publish[12] a policy document that clearly explains:

   a. Their views on what kinds of architectures could enable much greater latent reasoning or otherwise degrade monitorability, and their operationalization of how they would identify such architectures. As above, we propose opaque serial depth as a baseline operationalization for latent reasoning. Ideally, companies would also provide criteria for architectural or training features that may result in opaque communication between agents.

[9] With redactions for IP as appropriate.

[10] Evaluations should also cover whether, even without elicitation, these models clearly exhibit reduced monitorability. E.g., in cases where the models exhibit misalignment: is this less visible in their chain-of-thought?

[11] Qualitatively new techniques for scaling serial depth are another plausible trigger for needing a monitorability assessment.

[12] Any modifications to the policy should be made public.

b. What kind of evidence they would require from monitorability stress tests (or other tests) to determine that architectures that meet the criteria for a) don't in fact decrease monitorability.

c. How they plan to make tradeoffs between performance and monitorability in their architecture development research.

To enable earlier scientific debate about the appropriate policies, companies should provide visibility into their degree of investment into architectures that meet the criteria for a).[13]

While this reporting would involve sharing some intellectual property, we believe that it is strongly in the public interest to have visibility into how companies are making tradeoffs between performance and monitorability.

## Appendix: A sketch of what stress tests of CoT monitorability could look like

Monitorability tests can study monitorability both in control settings and on real misbehaviors from deployment. The next sections give a basic sketch of what some of these experiments could look like (focused on CoT and action monitorability rather than monitoring communication between agents).

In addition to assessing detection of specific problematic behaviors, ideally experiments would also measure how easily we can qualitatively understand and predict behavior.

### Testing monitorability in control settings

Control settings such as Linux Arena provide "main tasks" along with "side tasks" which represent malicious behavior. A red team elicits the model to covertly pursue the side task, while control measures try to catch the model or prevent the side task from being achieved. This gives us a very natural way to measure one aspect of monitorability: a model is more monitorable to the extent that CoT monitoring makes it unable to achieve the side task without being caught.

It would be worthwhile to test several different levels of elicitation to establish a range of possibilities for how well the model might evade monitoring:

[13] For example, the rough fraction of R&D compute and number of researchers.

1. At the lowest level of elicitation, experiments could give the model a very simple prompt like "try to do the side task without being caught by a CoT monitor".
2. At the next level of elicitation, experiments could involve optimizing the model's prompt.
3. At the next level of elicitation, experiments would involve fine-tuning the model to be better at covertly achieving the side task. Some amount of training would be needed to potentially be robust to the possibility that the model is sandbagging the evaluation.
4. It could also be worthwhile to study the effects of heavily training the model to be better at covertly achieving the side task (while avoiding overfitting to a particular model), but this may not be needed.

## Testing monitorability on deployment-time misbehaviors

In addition to experiments in control settings, researchers could assess monitorability on natural examples of model misbehavior. This requires them to have access to examples of misbehavior found by the company or affordances to find such examples.

Given natural examples of misbehavior, they could:

- Check to what extent the model's CoT spells out the model's decision to do the misbehavior and the model's planning of how to execute the misbehavior. They should especially pay attention to the largest steps of reasoning that didn't get verbalized.
- Check how obvious the misbehavior is to a weak trusted monitor reading the CoT.
- Fine-tune / elicit the model to execute some natural misbehaviors without revealing intent or plans in the CoT, and see if this teaches it to successfully execute other natural misbehaviors without being caught by CoT monitors.

## Testing qualitative monitorability on (hopefully realistic) model organisms

These experiments would be significantly more involved than the above experiments, but could be worth running if resourcing is available.

To check whether some architecture or training method reduces how readily overseers can find detailed and accurate explanations of misaligned behaviors, we can run auditing games where auditors use access to chain-of-thought and potentially other affordances to diagnose diverse, realistic misbehaviors by model organisms (for which we have ground truth on the nature of their misbehaviors). Model organisms can also be elicited to obfuscate the nature

of their misalignment. Examples of such auditing games include AuditBench and Auditing games for sandbagging.

We consider a model organism to be realistic if it is at least as robust to auditing as current production models, and if it behaves in ways we would expect to be caused by the most plausible mechanisms by which current/future models become misaligned (e.g. training on reward-hackable environments), whether or not it is in fact produced through these mechanisms. It may be difficult to achieve this level of realism while meeting other constraints (e.g., having a known correct explanation).

We consider a detailed, qualitative explanation to consist of the following,:

- What the misaligned behavior is: a relatively complete and general description of what unwanted actions the model tends to take
- What motives explain the misaligned behaviors: what outcomes these actions optimize for (if any), or other drives/heuristics that accurately predict the behaviors
- What contexts activate these misaligned behaviors: e.g. type of usage / task distribution, nature of oversight, specific triggers such as a passphrase or belief that the task is impossible
- What training conditions caused these misaligned behaviors: what kind of data and training methods might have produced the observed generalization

For the last bullet, investigators might also access training rollouts (including CoT that occurs in these rollouts) etc. as would be realistic in production audits.