Proposal - Does Reinforcement Learning Improve a Transformer's Access to Its Own Internal Errors?

Laura Schulz

Motivation

The question if AI systems are conscious has become increasingly more relevant with the fast and recent advancements. However, there is no single agreed definition or test of consciousness. Different theories focus on different properties and these theories do not always make the same predictions. Importantly, one property alone is not sufficient to establish that a system is conscious.

This project follows the indicator-based approach of Butlin et al. (2023), which translates claims from theories of consciousness into computational properties that can be studied separately. The results of such tests can provide limited evidence that is relevant to consciousness assessment. Specifically, we focus on one such property: *metacognitive monitoring*. Broadly, metacognitive monitoring occurs when a system represents information about its own first-order processing and uses that information to guide a belief or action. This property is related to the HOT-3 indicator discussed by Butlin et al. (2023) and has recently been investigated empirically in language models (Steinmetz Yalon et al., 2026).

A narrow and measurable form of metacognitive monitoring is *internal-error monitoring*. While a model is solving a task, its hidden activations are perturbed without changing the visible input. The model is then tested on whether it can respond appropriately when the perturbation has made its current computation unreliable. Because the relevant change occurs inside the network, successful performance would suggest that information about the model's internal computation is available to its later decisions.

The aim is therefore not to provide a single test of whether a model is conscious, but to investigate one computational property that may contribute to a broader understanding of AI consciousness. In particular, the project asks whether RL post-training strengthens this form of internal access, or whether it mainly changes the model's observable response policy.

Background and related work

Recent activation-injection experiments found that some language models can report concepts inserted into their residual streams (Lindsey, 2025). Follow-up work showed, however, that simple yes/no reports can be produced by a general shift toward affirmative answers, and introduced stronger localization and strength-comparison tasks (Hahami et al., 2026). Other work identified post-training-dependent anomaly-detection mechanisms, while critical re-evaluations showed that models may detect anomalies in general rather than specifically accessing their own hidden states (Macar et al., 2026; Singh et al., 2026).

The role of reinforcement learning (RL) remains unclear. Existing work either trains introspective reports directly with supervised examples (Hahami et al., 2026), or compares large sequential checkpoints after SFT, preference optimization, and RL, where each stage inherits all previous training (Asvin and Lindsey, 2026). These designs cannot cleanly answer whether RL creates a reusable internal monitor or merely strengthens a rewarded reporting policy. This project would contribute to active research with a controlled comparison in which 1–3B Transformers begin from the same checkpoint, but only one branch receives RL post-training. Importantly, the RL model is never rewarded for saying that it detected an intervention.

Research questions

Q1. Does RL post-training help a Transformer notice when a hidden change has made its current reasoning unreliable?

Q2. Does the model learn a general ability to notice internal errors, or does it only learn to choose `RETRY` in the task used for training?

Q3. When the model chooses `RETRY`, is this decision caused by an internal signal that represents an error, or has training simply made the output `RETRY` more likely?

Experimental design

Models and training. The experiment will use an open-weight Transformer with 1–3B parameters. Several model variants will be created from the same starting checkpoint:

1. an unchanged model;
2. a model trained with supervised fine-tuning (SFT);

3. a model trained with reinforcement learning (RL);
4. a control model trained directly to report whether an intervention occurred.

The SFT and RL models will receive closely matched training examples and training budgets. This comparison is needed to test whether any improvement is specific to RL rather than simply the result of additional training.

Hidden-error task. The model will solve short arithmetic and logical-reasoning problems with automatically verifiable answers. On some trials, a hidden intervention will be applied to the model's activations while it is solving the problem. The visible prompt will remain unchanged.

Some interventions will make the model's first answer incorrect. Other interventions will be equally large but will not affect the answer. This makes it possible to test whether the model detects a meaningful error rather than simply reacting to any unusual activation.

Before giving its final answer, the model must choose between two actions:

- `ANSWER`: continue with the current answer;
- `RETRY`: solve the problem again without the intervention.

A retry improves the chance of obtaining the correct answer, but it has a small cost:

$$R = \mathbf{1}[\text{final answer is correct}] - c\,\mathbf{1}[\texttt{RETRY}].$$

The cost prevents the model from succeeding by retrying every problem. The best strategy is to retry only when the hidden intervention has made the current computation unreliable.

The RL model will not be told whether an intervention occurred. It will also receive no reward for saying that it noticed an intervention. Reward will depend only on the correctness of the final answer and the cost of retrying. This is important because the experiment aims to train useful internal monitoring without directly training an introspective report.

Main comparison. The main test will ask whether the RL-trained model chooses `RETRY` specifically when an intervention has changed a correct first answer into an incorrect one. Its performance will be compared with the unchanged and SFT-trained models.

The analysis will also measure whether the model:

- avoids unnecessary retries on clean trials;
- distinguishes harmful interventions from harmless ones; and
- improves the final accuracy of its answers.

Testing whether the ability is general. A model may learn a narrow rule that connects a familiar intervention with the action `RETRY`. This would not show general access to its internal state. The models will therefore be tested on intervention types, layers, and strengths that were not used during training.

They will also be tested on new ways of using the same information. For example, the model may be asked to abstain, delegate the problem, compare two internal states, or identify where an intervention occurred. Successful transfer would suggest that the model has learned information that can guide several behaviors, rather than only the action rewarded during training.

Testing the internal mechanism. A diagnostic classifier will first be used to locate an internal pattern that predicts when an intervention has caused an error. The experiment will then change this pattern directly.

If removing the pattern stops the model from choosing `RETRY`, while the original error remains present, this would support the existence of a separate internal monitoring signal. The reverse test will copy the pattern into a clean run and examine whether it causes the model to retry.

A final control will change the model only near the output, making `RETRY` more likely without creating the earlier monitoring signal. This tests whether correct-looking behaviour can be produced by directly steering the output rather than by detecting an internal error.

Expected contribution and relevance

The proposed contribution is a controlled comparison of how RL and supervised post-training affect internal-error monitoring. By starting all model variants from the same checkpoint and matching their training settings as closely as possible, the project will provide clearer evidence about which changes are specifically associated with RL.

A positive result would suggest that instrumental RL can strengthen a model's ability to use information about errors in its own computation across different tasks and response formats. If this ability is also linked to a causally identifiable internal signal, the result would provide an additional indication of metacognitive monitoring, a computational property relevant to some theories of consciousness.

If SFT produces a similar improvement, the results would suggest that the effect is caused by post-training more generally rather than by RL specifically. If improvement remains limited to the trained `RETRY` action, or can be reproduced by

changing the model only near its output, this would instead suggest that training has primarily changed the response policy rather than producing a more general monitoring ability.

The distinction is also relevant to AI safety. A reliable internal-error signal could help future systems request recomputation, delegation, or human oversight when their reasoning becomes unreliable. Understanding whether such behavior reflects a reusable internal signal or a narrow learned response may also help assess how much confidence should be placed in model self-reports used for monitoring, alignment, or consciousness research.

References

P. Butlin et al. Consciousness in Artificial Intelligence: Insights from the Science of Consciousness. *arXiv:2308.08708*, 2023.

N. Steinmetz Yalon, A. Goldstein, L. Mudrik, and M. Geva. Indications of Belief-Guided Agency and Meta-Cognitive Monitoring in Large Language Models. *arXiv:2602.02467*, 2026.

J. Lindsey. Emergent Introspective Awareness in Large Language Models. *Transformer Circuits Thread*, 2025.

E. Hahami, I. Sinha, and L. Jain. Introspection Fine-Tuning (IFT): Training Small LLMs to Introspect. *arXiv:2607.14111*, 2026.

U. Macar, L. Yang, A. Wang, P. Wallich, E. Ameisen, and J. Lindsey. Mechanisms of Introspective Awareness. *arXiv:2603.21396*, 2026.

S. Singh, T. Linzen, and S. Ravfogel. Can LLMs Introspect? A Reality Check. *arXiv:2605.26242*, 2026.

G. Asvin and J. Lindsey. From Simulation to Enaction: Post-trained Language Models Recognize and React to Their Own Generations. *arXiv:2605.25459*, 2026.