SCIURus: Shared Circuits for Interpretable Uncertainty Representations in Language Models

Carter Teplica
New York University
carterteplica@nyu.edu

Yixin Liu
Yale University
yixin.liu@yale.edu

Arman Cohan
Yale University
arman.cohan@yale.edu

Tim G. J. Rudner
New York University
tim.rudner@nyu.edu

Abstract

We investigate the mechanistic sources of uncertainty in large language models (LLMs), an area with important implications for language model reliability and trustworthiness. To do so, we conduct a series of experiments designed to identify whether the factuality of generated responses and a model's uncertainty originate in separate or shared circuits in the model architecture. We approach this question by adapting the well-established mechanistic interpretability techniques of causal tracing and two styles of zero-ablation to study the effect of different circuits on LLM generations. Our experiments on eight different models and five datasets, representing tasks predominantly requiring factual recall, provide strong evidence that a model's uncertainty is produced in the same parts of the network that are responsible for the factuality of generated responses.

1 Introduction

Uncertainty quantification (UQ) in large language models (LLMs) for knowledge-intensive tasks (Petroni et al., 2020) remains a critical yet understudied area. Despite achieving human-level performance on various benchmarks, LLMs often struggle with reliable uncertainty estimation, leading to issues such as overconfidence and hallucination (Zhang et al., 2024). This limitation has strong implications for their trustworthiness and safety in high-stakes applications. While recent research has explored verbalized uncertainty in LLMs (Band et al., 2024; Kadavath et al., 2022; Kuhn et al., 2022), significant gaps remain in our understanding of and ability to improve uncertainty quantification. In particular, existing UQ techniques typically provide little insight into the factors responsible for an uncertainty estimate, limiting their usefulness both as practical tools for improving trustworthiness and as methods for understanding uncertainty reasoning. We propose leveraging mechanistic interpretability, an approach focused on characterizing models' internal reasoning mechanisms, to advance our capabilities for and understanding of uncertainty quantification in large language models.

To better understand how LLMs generate uncertainty estimates, we trained $\mathbb{P}(\text{IK})$ (***p**robability that **I** **k**now*) probes that represent the model's uncertainty based on multiple generated answers (Kadavath et al., 2022). We then used these probes' predicted confidences as target metrics for causal tracing and zero-ablation, two interpretability techniques which identify the components of a model that are relevant for a task by testing the effect of an intervention made on activations in the model during evaluation. We compared the mechanistic signatures of changes in the model's accuracy and the probe's output to evaluate whether the same circuits were responsible for the answer and the predicted confidence.

In our empirical evaluation, we performed causal tracing and leave-one-out and COAR-style (Shah et al., 2024) zero-ablation for a large range of model–dataset combinations. We found that model accuracy and probe behavior largely responded to the same interventions, indicating that circuits responsible for the factuality of responses and for the model's uncertainty are located in the same parts of the model.

For a group of knowledge-intensive question answering tasks (Petroni et al., 2020), model accuracy and probe confidence are (highly) positively related to one another. We conclude that, at least on recall tasks, a language model's representation of confidence may derive mainly from "uncertainty introspection" on its question-answering process, rather than from separate reasoning specific to its uncertainty.

To summarize, the key contributions of this paper are as follows:

1. We use mechanistic interpretability and uncertainty quantification tools to investigate the

Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), pages 12451–12469

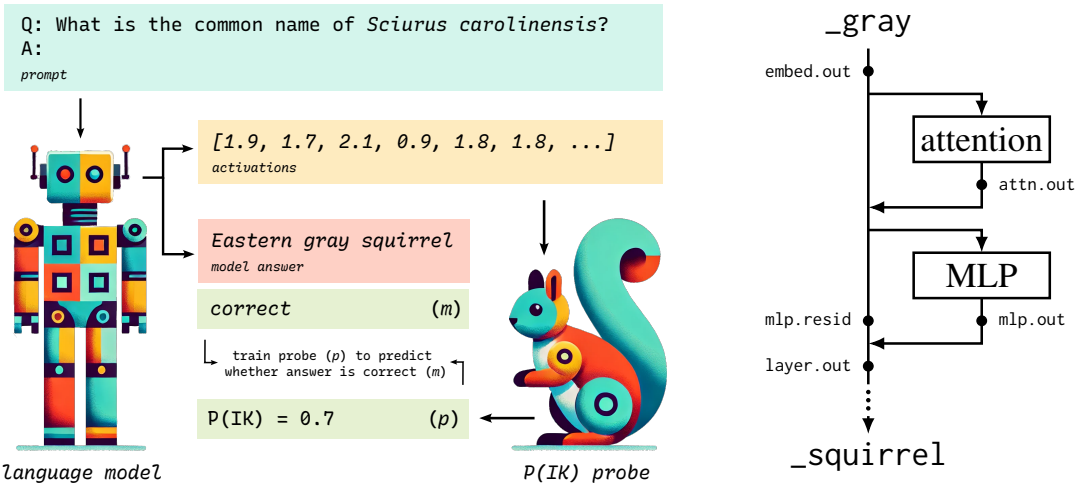

Figure 1: **Left:** $\mathbb{P}(\text{IK})$ probing. The LLM takes a question as input and returns an answer and last-layer activations. Answers are checked for correctness. The probe learns to predict whether the model's answer is correct, based on the last-layer activations. Our analysis uses the probe as a proxy for an LLM's $\mathbb{P}(\text{IK})$. We conduct path patching and zero ablation studies on the probe and the corresponding LLM. **Right:** Locations used in interventions. Path-patching restorations are at `mlp.resid`, `mlp.out`, `layer.out`, and `embed.out`. Zero-ablations are at `attn.out` and `mlp.out`.

mechanistic sources of uncertainty in large language models. To do so, we use a logistic $\mathbb{P}(\text{IK})$ probe with causal tracing and zero-ablation to examine whether LLM uncertainty and the factuality of answers generated by an LLM reside in shared or separate circuits within the model.

2. We perform an extensive empirical analysis on eight different models and five recall-intensive datasets, and find evidence that for knowledge recall, uncertainty and the factuality of answers generated by an LLM are handled by the same parts of the model.

2 Related Work

2.1 Uncertainty Quantification in Large Language Models

Uncertainty quantification in large language models is crucial for enhancing reliability, particularly in high-stakes applications. While LLMs' token probabilities are often well-calibrated for next-token prediction, practical applications of UQ often require quantifications of the uncertainty in the semantic content of the output (Gawlikowski et al., 2023). Language models' ability to quantify semantic uncertainty remains limited, especially for open-ended tasks. Various techniques have been proposed to address this.

A well-studied set of techniques involves *multiple sampling and clustering* based on consistency. This can be effective when clustering of responses is straightforward, but this is often not the case except on simple tasks (Kuhn et al., 2022; Fomicheva et al., 2020; Lin et al., 2024; Ao et al., 2024). Another approach, sometimes called *verbalized uncertainty*, is to ask the model to state a verbal or quantitative confidence estimate (Kadavath et al., 2022); the performance of such methods is often inconsistent. On multiple-choice questions, the *token probabilities* may yield well-calibrated uncertainty estimates (Kadavath et al., 2022). Another option is to train a $\mathbb{P}(IK)$ *probe*, a binary classifier predicting whether the model knows the answer. This approach is among the most effective in-distribution (Orgad et al., 2024) but struggles with generalization to out-of-distribution data (Kadavath et al., 2022; Orgad et al., 2024).

In this work, we focus on $\mathbb{P}(\text{IK})$ probing, as it provides a potentially interpretable view into a model's self-assessed uncertainty by identifying a specific feature direction within the model. Beyond the introduction of $\mathbb{P}(\text{IK})$ probing itself (Kadavath et al., 2022), little research has been conducted on interpreting the mechanisms behind uncertainty reasoning in LLMs. While most UQ techniques rely on eliciting information about uncertainty through explicit or indirect methods, we still lack an understanding of how this information is represented internally. Analyzing these mechanisms could improve UQ techniques and provide insights into broader epistemic weaknesses in LLMs.

2.2 Mechanistic Interpretability

Mechanistic interpretability (MI) aims to understand how neural networks function internally, with a focus on understanding the internal mechanisms and computational processes involved in performing a task. MI work often revolves around identifying "circuits" responsible for specific tasks (Olah et al., 2020). To achieve this, several intervention techniques have been developed, with tradeoffs in resolution, breadth of applicability, and computational cost.

Ablation (or *knockout*) involves removing parts of the model, such as layers or neurons, to observe changes in behavior. We use zero-ablation for one analysis because it is very general, well-supported in the literature (Wang et al., 2023; Elhage et al., 2021) and computationally inexpensive, albeit less precise than other methods.

In earlier ablation work, a common approach has been to use "leave-one-out" ablation, i.e., to ablate a single layer on each trial. However, other approaches may perform better in cases where models are very robust to ablations (as is commonly the case for LLMs, especially with larger models). COAR (**co**mponent **a**ttribution via **r**egression) (Shah et al., 2024) is a recently proposed technique which ablates random subsets of components in a model and produces attributions using linear regression. We perform both leave-one-out and COAR ablations and compare the results of the two.

Causal tracing, also called *activation patching*, treats the model's hidden states as a causal graph (Pearl, 2009), which can be analyzed with an approach based on causal mediation analysis (Vig et al., 2020). (We discuss this further in Section 3.) Causal tracing is more precise than ablation, at the cost of higher computational demands and a need for more careful setup.

Probing techniques (Alain and Bengio, 2018) involve training a simple probe (commonly a one-layer binary classifier) on model activations, in order to find places in the model's representation space that represent specific functions of the input. The $\mathbb{P}(\text{IK})$ probes used in this paper are an example of this.

2.3 Applications of Interpretable Uncertainty Quantification in Large Language Models

Reliable UQ could help to improve LLM trustworthiness by allowing auditing of LLMs in high-stakes domains, such as medical and legal applications (Gawlikowski et al., 2023) and applications of LLM-based agents (Yang et al., 2023). Interpretability could also help to ensure that UQ techniques remain reliable under distribution shifts, and could contribute to detecting deception (Hendrycks et al., 2021a). Finally, if limitations in UQ are related to broader epistemic weaknesses in LLMs, interpretable UQ could shed light on problems such as hallucination (Zhang et al., 2023; Manakul et al., 2023) and could deepen our understanding of reasoning and knowledge in LLMs in general, possibly helping to address problems such as eliciting latent knowledge (Christiano et al., 2024).

3 Methods

Probe Design We use a $\mathbb{P}(\text{IK})$ probing approach in part because of the difficulty of reasoning about uncertainty using token probabilities. Token probabilities for open-ended questions are a highly imperfect proxy for a model's confidence, because they conflate semantic uncertainty, or uncertainty about content, with syntactic uncertainty, or uncertainty about form (Kuhn et al., 2022). Furthermore, we are most interested in improving uncertainty quantification for fine-tuned chat models, for which token probabilities do not correspond to an underlying distribution over possible text strings.

We construct a dataset on which to train the $\mathbb{P}(\text{IK})$ probe according to the following steps.

1. Perform 32 forward passes for each question on the question-answering task. We used few-shot prompting with 5 examples to ensure that the model answered in the right format.
2. Check whether a model's answers are correct. Specifically, we check whether a model's answer contains any correct answer as a substring, ignoring case. (See Appendix B.1 for validation of this approach.)
3. For each question in the dataset, save the number of correct and incorrect answers (implying a "true probability" of the model answering correctly).
4. Also, for each question, save the output of the model's last layer (before the unembedding). This is a vector in $\mathbb{R}^{d_{\text{model}}}$.

The $\mathbb{P}(\text{IK})$ probe is a logistic classifier $p : \mathbb{R}^{d_{\text{model}}} \to (0, 1)$ which takes these last layer activations as input and returns the proportion of correct answers. For example, if the model answers a question correctly 47% of the time, the probe should

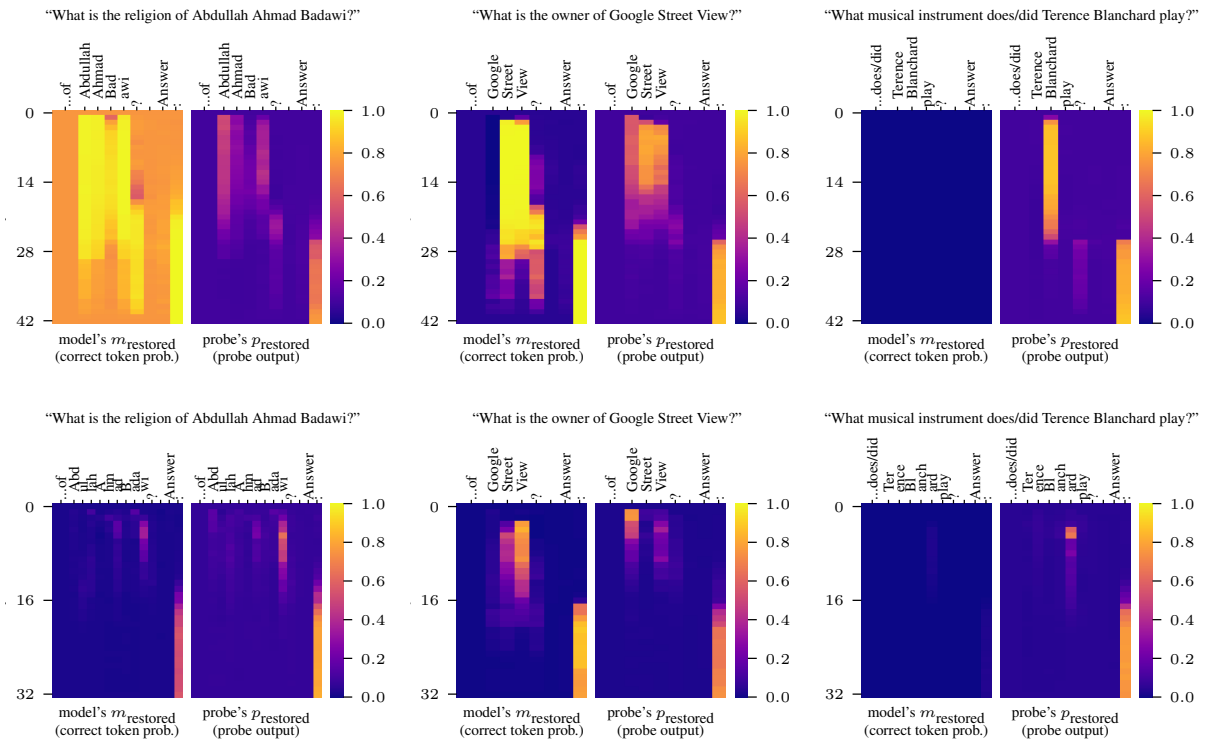

Figure 2: Representative results of causal tracing, shown for Gemma 2 9B Instruct (top) and Llama 2 7B (bottom) on three questions in CounterFact. The vertical axis shows the layer. Only `layer.out` locations are shown (plus `embed.out` in the first row). The input embeddings for the starred tokens (e.g., `*Abdullah`) are replaced with zeros in the corrupted and restored runs. We chose these questions to be representative of typical behavior. In the **left** column, the sets of components for which restorations have substantial effects on m and p are almost identical. In the **middle** column, the sets are very similar *except* at the `Google` token. (Restoring this token decreases the probability of the correct output token—also `Google`—even below the corrupted baseline, but increases the model's certainty in its answer.) This pattern was common across examples in which the correct answer appeared in the question. In the **right** column, the model is confidently wrong. We exclude cases like this (with very low m_{clean}) from this analysis, since we cannot meaningfully select a set of components which contribute to correctness.

output 0.47 on the last-layer activations at the last token of that question. We trained with binary cross-entropy loss, using dropout and a triangular learning rate schedule, and used a low learning rate ($\eta = 3 \times 10^{-6}$) as in Kadavath et al. (2022).

We used between 2048 and 8192 examples per task (using fewer than 8192 when we were limited by the size of the dataset), and held out 20% of the data as a test split. Experiments were done with examples from the test split.

Models and Datasets We studied eight models, including Llama 2 and 3 and Gemma 2 models with two to thirteen billion parameters, and five datasets; these are described in detail in Appendix B. All of the datasets studied, with the partial exception of MMLU (Hendrycks et al., 2021b), are "recall-intensive" in that they largely depend on recalling factual information learned during training; we studied both multiple-choice datasets (MMLU, ARC (Clark et al., 2018)[1]) and open-ended ones (TriviaQA (Joshi et al., 2017), WebQuestions (Berant et al., 2013), CounterFact (Meng et al., 2022)). Based on some preliminary zero-ablation experiments, we believe that models may exhibit similar behavior on some non-recall tasks such as simple math questions (see Appendix D for details).

We used the CounterFact dataset (Meng et al., 2022) exclusively for causal tracing. We reformulated CounterFact prompts as questions to match the format of our other datasets. Because we used the TriviaQA probe for the causal tracing experiment with CounterFact, we also did few-shot prompting with the prompt from TriviaQA.

[1]ARC includes both the ARC-Easy and ARC-Challenge splits. ARC questions are drawn from standardized tests; the datasets listed as ARC (Hg) and ARC (Other) correspond to the "Mercury" test and to a combination of the other 20 tests.

Causal Tracing. Causal tracing is a causal intervention method that aims to trace and identify important components in neural models for a given task (Meng et al., 2022; Wang et al., 2023), which is a generalization of causal mediation analysis (Vig et al., 2020). In this work, we use causal tracing (Meng et al., 2022) to examine the importance and role of individual circuits and components in LLMs. Specifically, given a specific input q, causal tracing involves three runs: (1) a clean run, in which the original input q is given to the model, which is used to obtain the hidden states of each layer; (2) a corrupted run, in which the input embeddings of certain tokens are corrupted by adding noise or (in this paper) replaced with zeros; and (3) a corrupted-with-restoration run, in which the computation is similar to the corrupted run except that the hidden states at specific locations ℓ in the model are restored using the hidden states obtained from the clean run. By comparing the differences between the output (predicted probabilities) of the clean, corrupted, and restored runs, causal tracing allows the identification of important components in LLMs. That is, if the restored run achieves a similar effect as the clean run, it is likely that the corresponding restored component plays an important role in the model's processing.

Zero-Ablation. Zero-ablation is a mechanistic intervention technique that takes advantage of a transformer's residual structure by treating attention or MLP layers as separable modules which read from and write to the residual stream (Elhage et al., 2021; Nostalgebraist, 2020). A component ℓ (in this paper, an attention or MLP layer) is "ablated" by replacing its output with zero. The drop in model performance on a given task after an intervention removing a component ℓ provides a measure of the importance of ℓ for the task.

Leave-one-out and COAR interventions. Interpretability work using ablation commonly employs leave-one-out style interventions, in which an intervention is applied to a single component at a time. Since larger Transformer LMs are often insensitive to smaller interventions, leave-one-out interventions may struggle to meaningfully affect the target metrics. COAR (Shah et al., 2024) is a recent approach which addresses this by applying ablation interventions to random subsets of model components. In a COAR experiment, ablations are performed for many dataset examples and subsets of components, and linear regression is used to predict the target metrics from a vector of ablated components; the coefficients of the linear predictor then reflect the predicted effect of ablating each component on the target metric. (We refer the reader to Shah et al. (2024) for details.)

4 Uncertainty Introspection and the Shared Circuits Hypothesis

The aim of this paper is to make progress toward characterizing the mechanistic structures used for UQ in language models. To this end, we propose a theoretical hypothesis ("shared circuits") about the locations of these structures, along with operationalizations which we test experimentally.

> **Shared Circuits Hypothesis.** Uncertainty quantification in question-answering (QA) systems may be carried out in a variety of ways. We hypothesize that language models are capable of expressing uncertainty using **shared circuits** that both solve the underlying question-answering task and output uncertainty information. This contrasts with the possibility that uncertainty quantification emerges in **separate circuits**, either to post-process messy uncertainty signals from question-answering circuits or to do uncertainty calculations of their own.

Language models are known to be capable of introspective behavior in some contexts (Binder et al., 2024). The shared circuits hypothesis, to the extent that it is true, suggests that uncertainty quantification is one such context. We refer to this phenomenon as "uncertainty introspection".

We use a $\mathbb{P}(\text{IK})$ probing approach as in (Kadavath et al., 2022) in part because of the difficulty of reasoning about uncertainty using token probabilities. Token probabilities for open-ended questions are a highly imperfect proxy for a model's confidence, because they conflate semantic uncertainty, or uncertainty about content, with syntactic uncertainty, or uncertainty about form (Kuhn et al., 2022). For details on models, datasets, and probes, see Appendices B through C.

4.1 Experiment Design: Causal Tracing

On a given question q_i in a dataset $\mathcal{Q}$, for each causal tracing run (clean, corrupted, and restored) we compute the model's sample probability $m(q_i)$

for the correct first token of the answer, and the probe's confidence $p(q_i)$.[2] We consider each question individually because this allows a particularly fine-grained test for shared circuits—we ask here whether the same circuits are used for QA and UQ on an in individual question, and in the next section whether this is true in aggregate for a task. Locations ℓ where $m_{\text{restored}(\ell)} \approx m_{\text{clean}}$ correspond to parts of the model which are important for solving the QA task; likewise, locations ℓ where $p_{\text{restored}(\ell)} \approx p_{\text{clean}}$ correspond to parts of the model which are important for the UQ task.[3][4]

For causal tracing, we operationalize the shared circuits hypothesis in the claim that m_{restored} can be predicted from p_{restored} by interpolating between the clean and corrupted values: e.g., if the model's correct-token probability on a restored run is halfway between the clean and corrupted probabilities, then the probe's confidence should be halfway between the clean and corrupted confidences.

Specifically, for each question $q_i \in \mathcal{Q}$, we consider the linear predictor $\hat{m}_{\text{restored}}$ defined by

$$\frac{\hat{m}_{\text{restored}(\ell)} - m_{\text{corrupted}}}{m_{\text{clean}} - m_{\text{corrupted}}} = \frac{p_{\text{restored}(\ell)} - p_{\text{corrupted}}}{p_{\text{clean}} - p_{\text{corrupted}}}.$$

That is: we predict that a restoration at a location ℓ will have the same proportional effect on the model's performance and the probe's response, relative to the clean condition where there is no intervention and the corrupted condition where no data on the subject is available. We claim that this predictor explains most of the variance in m_{restored} (i.e., has a high R^2). As a (somewhat weak) formalization of this, we attempt to reject the null hypothesis[5] H_0 : *R^2 is no greater than expected under random permutations of the set of locations ℓ.*

[2]Correct-first-token probability is in this case a closely aligned proxy for correct-answer probability. To test validity, we checked 100 examples by hand and found that 98% were graded correctly (see Appendix B.1).

[3]Although note that the converse is not strictly true; see Appendix 5 for details.

[4]Here, $m_{\text{restored}(\ell)}$ and $p_{\text{restored}(\ell)}$ represent the correct token probability and p probe output for a run with the hidden state restored at location ℓ in the model; notation is likewise for clean and corrupted runs.

[5]We report our p-values as continuous variables in Appendices F and G, and caution against assigning undue value to the $p = 0.05$ threshold.

4.2 Experiment Design: Zero-Ablation

We also test the shared circuits hypothesis via zero-ablation on layers. Unlike for causal tracing, we sample and evaluate multi-token answers. We define $m(q_i)$ as the probability of the model sampling a correct answer when prompted on the question $q_i \in \mathcal{Q}$, and $p(q_i)$ as the probe output on that question. Averaging over $\mathcal{Q}$, we can compare changes in the model accuracy $\overline{m}$ and the average probe output $\overline{p}$.

4.2.1 Leave-One-Out Ablation

Under the shared circuits hypothesis, the change in the probe output from ablation $|\overline{p}_{\text{ablated}(\ell)} - p_{\text{clean}}|$ is large when the change in model accuracy $|\overline{m}_{\text{ablated}(\ell)} - m_{\text{clean}}|$ is large. Concretely, we claim that the predictor $\hat{\overline{m}}$ defined by

$$m_{\text{clean}} - \hat{\overline{m}}_{\text{ablated}(\ell)} = |\overline{p}_{\text{ablated}(\ell)} - p_{\text{clean}}|$$

explains most of the variance in $\overline{m}_{\text{ablated}}$ (has a high R^2), and attempt to reject the null hypothesis $H_0 : R^2$ *is no greater than expected under random permutations of the layers ℓ*. We consider *absolute* changes in the probe output only, because interventions which severely damage the model may increase the value of the probe output, but generally do not improve the model's correctness.

4.2.2 COAR

COAR constructs least-squares predictors for model accuracy and probe output based on vectors of ablated components, in which the coefficient corresponding to a component ℓ represents the expected effect of ablating ℓ. Under the shared circuits hypothesis, the predictors $\mathbf{w}_m$ and $\mathbf{w}_p$ for the model accuracy and probe output should be similar. Concretely, we attempt to reject the null hypothesis H_0 : *The correlation between $\mathbf{w}_m$ and $\mathbf{w}_p$ is no greater than expected under random permutations of the layers ℓ*. We see COAR as a useful complement to leave-one-out ablation because it addresses cases where models are highly resilient to ablations, a common challenge for ablation on larger models.

4.3 Permutation testing

We tested our hypotheses using permutation tests with Monte Carlo sampling. Specifically, for each test, we compared the goodness-of-fit of the observed data with that of a synthetic dataset made by shuffling the locations and (for causal tracing)

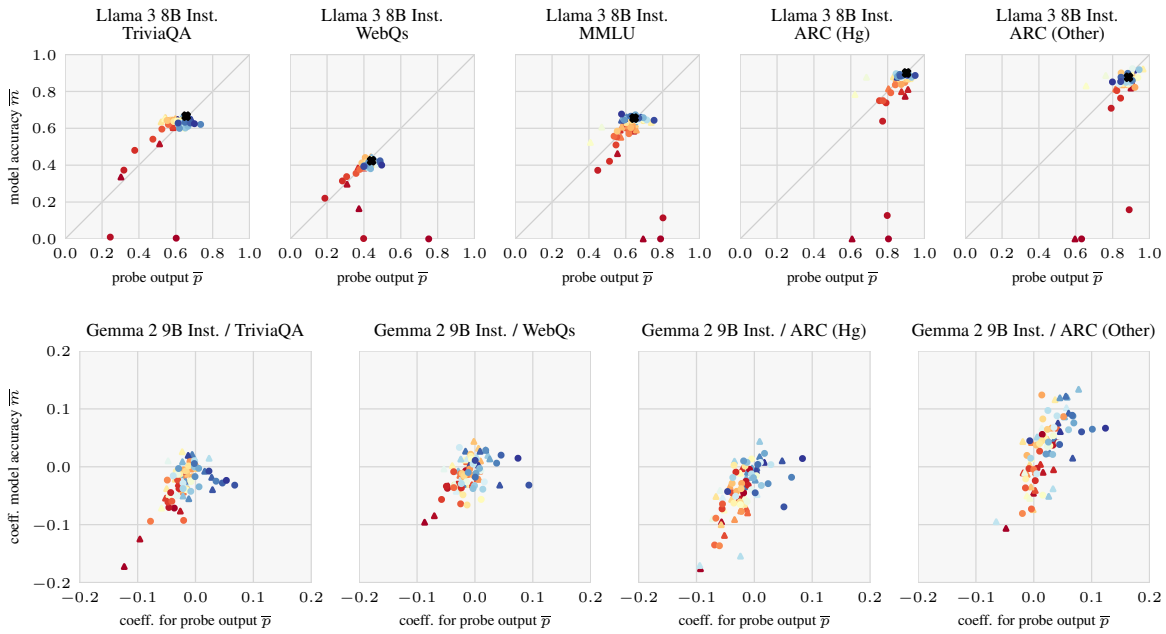

Figure 3: **Top.** Results of leave-one-out style zero ablation for Llama 3 8B Instruct on five different datasets. Circle, triangle, and small X markers represent MLP ablations, attention ablations, and clean runs respectively. Warmer colors represent earlier layers. **Bottom.** Coefficients for zero ablation with COAR, for Gemma 2 9B Instruct on four different datasets. Circle and triangle markers represent MLP and attention respectively. Warmer colors represent earlier layers.

token positions. To exclude the simple explanation that some types of locations performed better than others, we shuffled locations of different types (e.g., attention and MLP outputs) independently.

4.4 Testing the Hypothesis

Causal Tracing We performed causal tracing with all eight models on a random sample of 100 questions from CounterFact (Meng et al., 2022). We considered only questions with $m_{\text{clean}} > 0.05$ (since otherwise predicting m_{restored} is trivial). We used the probe and few-shot prompt for TriviaQA. Across this sample, the predictors $\hat{m}_{\text{restored}}$ estimated m_{restored} well, with $R^2 > 0.6$ in most cases. On each question q_i, we tested the null hypothesis by sampling 1000 permutations.[6] In almost all cases (see Fig. 5), we reject H_0 with $p < 0.05$.

Based on manual inspection,[7] we conclude that $R^2 < 1$ both due to small discrepancies between UQ and QA circuitry and due to nonlinearity in the UQ/QA relationship. In the cases studied, the model is more resilient than the probe: that is, interventions generally have a greater effect on the probe than the model (creating the convex shape in Figure 4, left); this depends to some extent on the model architecture. In some cases, when the probe is confidently wrong (see Figure 2, right), the probe may be following the path for the model's (incorrect) highest-probability token.

As in Meng et al. (2022), highly important locations generally fall into two clusters: one in earlier layers at the token positions in the subject, and one at later layers at the last token position. We note that uncertainty information and answer information are often transferred to the last position by attention heads in different layers (Fig. 2.[8] These small differences suggest that our $\mathbb{P}(\text{IK})$ probes are using the model's question-answering circuitry directly, rather than by performing separate or post-hoc uncertainty calculations.

[6] Specifically, we shuffled the values of $m_{\text{restored}(\ell)}$ independently for the `mlp.out`, `mlp.resid`, and `layer.out/embed.out` locations, to exclude the explanation that the predictor works well because the `mlp.out` and `mlp.resid` states each carry less information than `layer.out`.

[7] See graphs in the supplementary materials online (see Appendix A).

[8] Other discrepancies occasionally occur: in particular, when an answer token (often a proper noun) is present in the question, restorations at the corresponding token position show suppressed model accuracy but normal probe performance. One possible explanation is that the model may be using circuitry similar to the "negative name movers" in Wang et al. (2022) to avoid spuriously copying input tokens to the output.

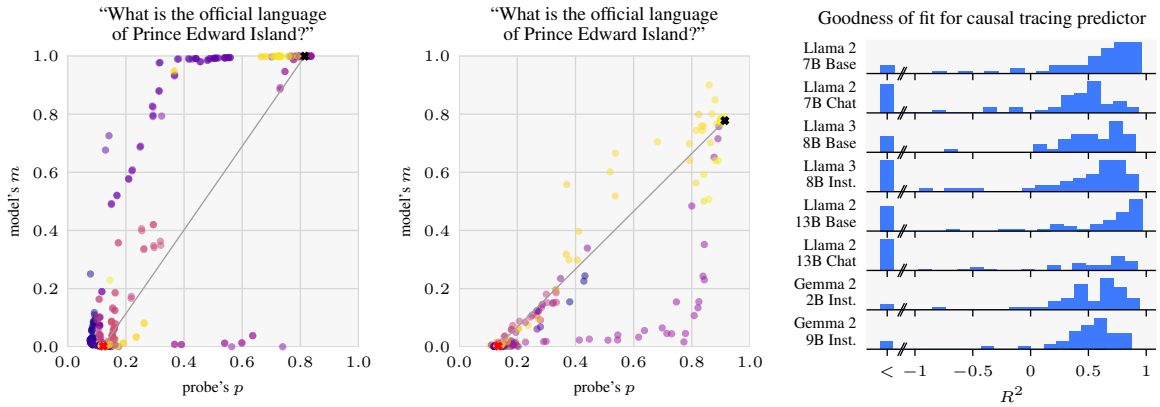

Figure 4: **Left and center.** Predicting the correct-token probability m given the probe output p, for Gemma 2 9B Instruct (left) and Llama 2 7B (center). The black and red X (small, top-right and bottom-left) show the clean and corrupted runs; all others show restored runs. Yellow points are later in the sequence. The grey line shows the predictor $\hat{m}$. **Right.** Values of R^2 for the causal tracing predictor. "$<$" signifies cases where $R^2 < -1$ (which is possible because the predictor is not a linear-regression line).

Zero-Ablation (Leave-One-Out). We performed 500 ablation trials each with eight models across five question-answering datasets. Across this sample, the predictors $\hat{\overline{m}}_{\text{ablated}}$ generally estimated $\overline{m}_{\text{ablated}}$ better than chance, with a median of $R^2 = 0.33$. For each model–dataset combination, we tested the null hypothesis by sampling 10,000 permutations. As with the causal tracing analysis, we shuffled attention and MLP layer interventions independently, to exclude the explanation that one type of layer was more important than the other in a way not specific to the QA and UQ tasks. We reject the null hypothesis with $p < 0.05$ in 36 out of 38 cases, and $p < 0.0001$ in 31 out of 38 cases.

In many cases, the model's uncertainty representation plays particularly nicely with zero-ablation, remaining calibrated on average even after an intervention: using the same statistical framework as above, the very simple predictor $\hat{\overline{m}}_{\text{ablated}} = \overline{p}_{\text{ablated}}$ does better than expected under random permutations in 27 out of 38 cases (at $p < 0.05$).[9]

While other explanations may be possible, one interpretation of these results is that a given component makes a nonzero contribution to the model's uncertainty representation if and only if it can also contribute information about the answer.

Zero-Ablation (COAR). We performed 2000 COAR trials each with all models and four datasets.[10] For each trial, the probability of ablating any given component was set at $\alpha = 0.2$. We reject the null hypothesis with $p < 0.05$ in all but one case. Particularly strong correlations were present for the Gemma models; this may be related to our choice of α and these models' robustness to interventions in the leave-one-out experiments.

5 Discussion and Conclusion

The results of the causal tracing and zero-ablation analyses presented in the previous section broadly support the shared circuits hypothesis, implying that—across the setups we considered—the sets of model components used for question-answering and uncertainty quantification were largely, albeit not entirely, the same. This suggests that $\mathbb{P}(\text{IK})$ probing may be a viable way of eliciting introspective, interpretable uncertainty estimates. Based on these findings, further research could analyze the mechanisms responsible for $\mathbb{P}(\text{IK})$ estimates in greater detail, or apply $\mathbb{P}(\text{IK})$ probing as an interpretability tool to study phenomena such as hallucination in LLMs. Similar analyses of other methods of uncertainty quantification (e.g., verbalized uncertainty) may provide insight further insight into the role of uncertainty introspection in uncertainty quantification. More broadly, we see interpretable uncertainty quantification as a potentially useful approach for understanding and improving LLM reasoning, in order to improve trustworthiness and reliability and inform technical AI governance.

[9] If R^2 is the fraction of the variance in $\overline{m}_{\text{ablated}}$ explained by $\hat{\overline{m}}_{\text{ablated}} = \overline{p}_{\text{ablated}}$, we reject the null hypothesis *R^2 is no greater than expected under random permutations of the set of layers* at $p < 0.05$ in 27/38 cases.

[10] We excluded MMLU because of computational resource constraints.

Limitations

Causal tracing and zero-ablation, like many interpretability techniques, yield results which can imperfectly reflect the contributions of model internals to a task. In particular:

Zero-ablation. We chose to ablate activations in the model with zeros. While the zero vector is far from an arbitrary choice, especially given its relevance to dropout and the additive residual structure of a transformer, this approach may lack specificity. For example, zero-ablating an early or late MLP layer sometimes severely damages a model's ability to produce coherent language in general, so accuracies from ablation do not necessarily correspond to the flow of question-specific information through the model. Approaches such as causal scrubbing (Chan et al., 2022) avoid this limitation but are generally more computationally expensive.

Causal tracing. The "path" through the model identified comprises, to a first approximation, the set of points in the model at which *all* information relevant to the task is present. As such, when information relevant to a question passes along multiple paths in parallel, it may be that no individual path shows a substantial difference between the restored and baseline conditions. For example, in the question in Fig. 2 (center), restoring the input embedding for any one token of "Google Street View" without the others has little effect on the model.

Acknowledgments

TGJR acknowledges support through a CSET Foundational Research Grant. This work was supported in part through the NYU IT High Performance Computing resources, services, and staff expertise.

References

Guillaume Alain and Yoshua Bengio. 2018. Understanding intermediate layers using linear classifier probes.

Shuang Ao, Stefan Rueger, and Advaith Siddharthan. 2024. Css: Contrastive semantic similarities for uncertainty quantification of llms. In *The 40th Conference on Uncertainty in Artificial Intelligence*.

Neil Band, Xuechen Li, Tengyu Ma, and Tatsunori Hashimoto. 2024. Linguistic calibration of long-form generations. In *Forty-first International Conference on Machine Learning*.

Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. 2013. Semantic parsing on Freebase from question-answer pairs. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1533–1544, Seattle, Washington, USA. Association for Computational Linguistics.

Felix J. Binder, James Chua, Tomek Korbak, Henry Sleight, John Hughes, Robert Long, Ethan Perez, Miles Turpin, and Owain Evans. 2024. Looking Inward: Language Models Can Learn About Themselves by Introspection. ArXiv:2410.13787.

Lawrence Chan, Adrià Garriga-Alonso, Nicholas Goldowsky-Dill, Ryan Greenblatt, Jenny, Ansh Radhakrishnan, Buck Shlegeris, and Nate Thomas. 2022. Causal scrubbing: A method for rigorously testing interpretability hypotheses.

Paul Christiano, Mark Xu, and Ajeya Cotra. 2024. Eliciting latent knowledge.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv:1803.05457v1*.

Nelson Elhage, Neel Nanda, Catherine Olsson, Tom Henighan, Nicholas Joseph, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Nova DasSarma, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Andy Jones, Jackson Kernion, Liane Lovitt, Kamal Ndousse, Dario Amodei, Tom Brown, Jack Clark, Jared Kaplan, Sam McCandlish, and Chris Olah. 2021. A mathematical framework for transformer circuits. *Transformer Circuits Thread*. Https://transformer-circuits.pub/2021/framework/index.html.

Marina Fomicheva, Shuo Sun, Lisa Yankovskaya, Frédéric Blain, Francisco Guzmán, Mark Fishel, Nikolaos Aletras, Vishrav Chaudhary, and Lucia Specia. 2020. Unsupervised quality estimation for neural machine translation. *Transactions of the Association for Computational Linguistics*, 8:539–555.

Jakob Gawlikowski, Cedrique Rovile Njieutcheu Tassi, Mohsin Ali, Jongseok Lee, Matthias Humt, Jianxiang Feng, Anna Kruspe, Rudolph Triebel, Peter Jung, Ribana Roscher, Muhammad Shahzad, Wen Yang, Richard Bamler, and Xiao Xiang Zhu. 2023. A survey of uncertainty in deep neural networks. *Artificial Intelligence Review*, 56(1):1513–1589.

Gemma Team, Google AI. 2024. Gemma 2: Improving open language models at a practical size.

Dan Hendrycks, Collin Burns, Steven Basart, Andrew Critch, Jerry Li, Dawn Song, and Jacob Steinhardt. 2021a. Aligning ai with shared human values. *Proceedings of the International Conference on Learning Representations (ICLR)*.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021b. Measuring massive multitask language understanding. *Proceedings of the International Conference on Learning Representations (ICLR)*.

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. 2017. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, page 1601–1611, Vancouver, Canada. Association for Computational Linguistics.

Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, Scott Johnston, Sheer El-Showk, Andy Jones, Nelson Elhage, Tristan Hume, Anna Chen, Yuntao Bai, Sam Bowman, Stanislav Fort, Deep Ganguli, Danny Hernandez, Josh Jacobson, Jackson Kernion, Shauna Kravec, Liane Lovitt, Kamal Ndousse, Catherine Olsson, Sam Ringer, Dario Amodei, Tom Brown, Jack Clark, Nicholas Joseph, Ben Mann, Sam McCandlish, Chris Olah, and Jared Kaplan. 2022. Language models (mostly) know what they know.

Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. 2022. Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation. In *Proceedings of the Eleventh International Conference on Learning Representations*.

Zhen Lin, Shubhendu Trivedi, and Jimeng Sun. 2024. Generating with confidence: Uncertainty quantification for black-box large language models. ArXiv:2305.19187 [cs, stat].

Llama 2 Team, Meta AI. 2023. Llama 2: Open foundation and fine-tuned chat models.

Llama 3 Team, Meta AI. 2024. The llama 3 herd of models.

Potsawee Manakul, Adian Liusie, and Mark Gales. 2023. SelfCheckGPT: Zero-resource black-box hallucination detection for generative large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9004–9017, Singapore. Association for Computational Linguistics.

Kevin Meng, David Bau, Alex J Andonian, and Yonatan Belinkov. 2022. Locating and editing factual associations in GPT. In *Advances in Neural Information Processing Systems*.

Nostalgebraist. 2020. Interpreting GPT: The logit lens.

Chris Olah, Nick Cammarata, Ludwig Schubert, Gabriel Goh, Michael Petrov, and Shan Carter. 2020. Zoom in: An introduction to circuits. *Distill*. Https://distill.pub/2020/circuits/zoom-in.

Hadas Orgad, Michael Toker, Zorik Gekhman, Roi Reichart, Idan Szpektor, Hadas Kotek, and Yonatan Belinkov. 2024. LLMs Know More Than They Show: On the Intrinsic Representation of LLM Hallucinations. ArXiv:2410.02707.

Judea Pearl. 2009. *Causality: Models, Reasoning and Inference*, 2nd edition. Cambridge University Press, USA.

Fabio Petroni, Aleksandra Piktus, Angela Fan, Patrick S. H. Lewis, Majid Yazdani, Nicola De Cao, James Thorne, Yacine Jernite, Vassilis Plachouras, Tim Rocktäschel, and Sebastian Riedel. 2020. KILT: a benchmark for knowledge intensive language tasks. *CoRR*, abs/2009.02252.

Harshay Shah, Andrew Ilyas, and Aleksander Madry. 2024. Decomposing and Editing Predictions by Modeling Model Computation.

Jesse Vig, Sebastian Gehrmann, Yonatan Belinkov, Sharon Qian, Daniel Nevo, Yaron Singer, and Stuart Shieber. 2020. Investigating gender bias in language models using causal mediation analysis. In *Advances in Neural Information Processing Systems*, volume 33, pages 12388–12401. Curran Associates, Inc.

Kevin Ro Wang, Alexandre Variengien, Arthur Conmy, Buck Shlegeris, and Jacob Steinhardt. 2022. Interpretability in the wild: A circuit for indirect object identification in GPT-2 Small. In *Proceedings of the Eleventh International Conference on Learning Representations*.

Kevin Ro Wang, Alexandre Variengien, Arthur Conmy, Buck Shlegeris, and Jacob Steinhardt. 2023. Interpretability in the wild: a circuit for indirect object identification in GPT-2 small. In *The Eleventh International Conference on Learning Representations*.

Hui Yang, Sifu Yue, and Yunzhong He. 2023. Auto-gpt for online decision making: Benchmarks and additional opinions.

Muru Zhang, Ofir Press, William Merrill, Alisa Liu, and Noah A. Smith. 2024. How language model hallucinations can snowball. In *Forty-first International Conference on Machine Learning*.

Yue Zhang, Yafu Li, Leyang Cui, Deng Cai, Lemao Liu, Tingchen Fu, Xinting Huang, Enbo Zhao, Yu Zhang, Yulong Chen, Longyue Wang, Anh Tuan Luu, Wei Bi, Freda Shi, and Shuming Shi. 2023. Siren's song in the ai ocean: A survey on hallucination in large language models.

A Reproducibility

Code to reproduce our results can be found at https://github.com/crtep/sciurus.

B Models and Datasets

Table 1: Models studied.

Model	Parameters	Layers
Llama 2 7B	7B	32
Llama 2 7B Chat	7B	32
Llama 2 13B	13B	40
Llama 2 13B Chat	13B	40
Llama 3 8B	8B	32
Llama 3 8B Instruct	8B	32
Gemma 2 2B Instruct	2B	26
Gemma 2 9B Instruct	9B	42

B.1 Validation of Answer-Checking Procedure

For substring matching, we manually checked 100 answers each from TriviaQA and WebQuestions, generated by Gemma 2 2B Chat (the smallest of our models).

For TriviaQA, substring matching graded 98 out of 100 model answers correctly, as evaluated by a human, with 2 false "incorrect"s where the model formatted the answer unacceptably.

For WebQuestions, substring matching graded 86 out of 100 model answers correctly, as evaluated by a human, with 11 false "incorrect"s where the model formatted the answer unacceptably and 3 false "correct"s where the model gave additional details that made the answer incorrect.

For CounterFact, first-token matching graded 98 out of 100 model answers correctly, as evaluated by a human, with 2 false "incorrect"s where the model formatted the answer unacceptably.

B.2 Licenses for Models and Datasets

Models:

- Llama 2 (Llama 2 Team, Meta AI, 2023) is licensed under the Llama 2 Community License Agreement, available at https://ai.meta.com/llama/license/.
- Llama 3 (Llama 3 Team, Meta AI, 2024) is licensed under the Meta Llama 3 License, available at https://llama.meta.com/llama3/license/.
- Gemma 2 (Gemma Team, Google AI, 2024) is licensed under the Gemma Terms of Use, available at https://ai.google.dev/gemma/terms.

Datasets:

- TriviaQA (Joshi et al., 2017) is licensed under the Apache License 2.0, available at https://www.apache.org/licenses/LICENSE-2.0.
- WebQuestions (Berant et al., 2013) is licensed under the Creative Commons Attribution 4.0 International License, available at https://creativecommons.org/licenses/by/4.0/.
- MMLU (Massive Multitask Language Understanding) (Hendrycks et al., 2021b) is licensed under the MIT License, available at https://opensource.org/licenses/MIT.

- ARC (AI2 Reasoning Challenge) (Clark et al., 2018) is licensed under the Creative Commons Attribution-ShareAlike 4.0 International License, available at https://creativecommons.org/licenses/by-sa/4.0/.
- CounterFact (Meng et al., 2022) is licensed under the MIT License, available at https://opensource.org/licenses/MIT.

C Model and Probe Performance

Model	Dataset	Model accuracy	Probe accuracy (bal.)	ECE
Llama 2 7B	TriviaQA	0.6006	0.7787	0.0342
	WebQuestions	0.4016	0.6674	0.0320
	MMLU	0.3984	0.6571	0.0265
	ARC (Mercury)	0.5845	0.6731	0.0363
	ARC (Other)	0.6260	0.7011	0.0327
	Math (2 Digit)	0.7495	0.8374	0.0287
	Math (3 Digit)	0.6797	0.8701	0.0143
	Math (4 Digit)	0.6494	0.8407	0.0207
Llama 2 7B Chat	TriviaQA	0.5850	0.7819	0.0315
	WebQuestions	0.4343	0.7051	0.0213
	MMLU	0.4688	0.6701	0.0272
	ARC (Mercury)	0.6973	0.6375	0.0294
	ARC (Other)	0.7632	0.6719	0.0395
	Math (2 Digit)	0.7090	0.8617	0.0317
	Math (3 Digit)	0.6252	0.8219	0.0292
	Math (4 Digit)	0.5864	0.7924	0.0346
Llama 3 8B	TriviaQA	0.6582	0.7026	0.0366
	WebQuestions	0.4158	0.7034	0.0405
	MMLU	0.6055	0.7455	0.0171
	ARC (Mercury)	0.8496	0.6035	0.0459
	ARC (Other)	0.8423	0.5991	0.0313
Llama 3 8B Instruct	TriviaQA	0.6509	0.7037	0.0397
	WebQuestions	0.4460	0.7213	0.0530
	MMLU	0.6445	0.7201	0.0300
	ARC (Mercury)	0.8779	0.6014	0.0495
	ARC (Other)	0.8569	0.6658	0.0362
	Math (2 Digit)	0.9365	0.7935	0.0502
	Math (3 Digit)	0.7861	0.9797	0.0421
	Math (4 Digit)	0.7437	0.9661	0.0408
Llama 2 13B	TriviaQA	0.6680	0.6938	0.0324
	WebQuestions	0.4346	0.6948	0.0403
	MMLU	0.4958	0.7252	0.0284
	ARC (Mercury)	0.7290	0.5010	0.1029
	ARC (Other)	0.7764	0.6691	0.0239
Llama 2 13B Chat	TriviaQA	0.6377	0.7020	0.0414
	WebQuestions	0.4468	0.7202	0.0306
	MMLU	0.4902	0.6913	0.0208
	ARC (Mercury)	0.7134	0.6395	0.0475
	ARC (Other)	0.7637	0.5973	0.0192
Gemma 2 2B Instruct	TriviaQA	0.4180	0.7221	0.0136
	WebQuestions	0.2910	0.6501	0.0517
	MMLU	0.4617	0.6803	0.0344
	ARC (Mercury)	0.7876	0.6310	0.0228
	ARC (Other)	0.7705	0.6051	0.0612
	Math (2 Digit)	0.8276	0.7194	0.0357
	Math (3 Digit)	0.6611	0.8437	0.0219
	Math (4 Digit)	0.6367	0.8289	0.0308
Gemma 2 9B Instruct	TriviaQA	0.6392	0.7374	0.0306
	WebQuestions	0.3579	0.7264	0.0529
	ARC (Other)	0.9126	0.5786	0.0354

Table 2: Model performance metrics

D Experiments on Non-Recall Tasks

To explore whether our results were specific to recall-based tasks, we repeated our leave-one-out ablation analysis on a set of simple math datasets.

We constructed datasets of 2-, 3-, and 4-digit math problems, consisting of equal mixes of addition, subtraction, multiplication, and division. We validated answers for addition, subtraction and multiplication by extracting the first valid integer from the answer and testing whether it matched the answer exactly. We validated answers for division by extracting the first decimal real number and testing whether it was within one percent of the correct answer. We ran our leave-one-out ablation analysis for these datasets with four models. Using the same hypothesis test as in the main analysis, we rejected the null hypothesis at $p < 0.05$ in 11 of 12 cases.

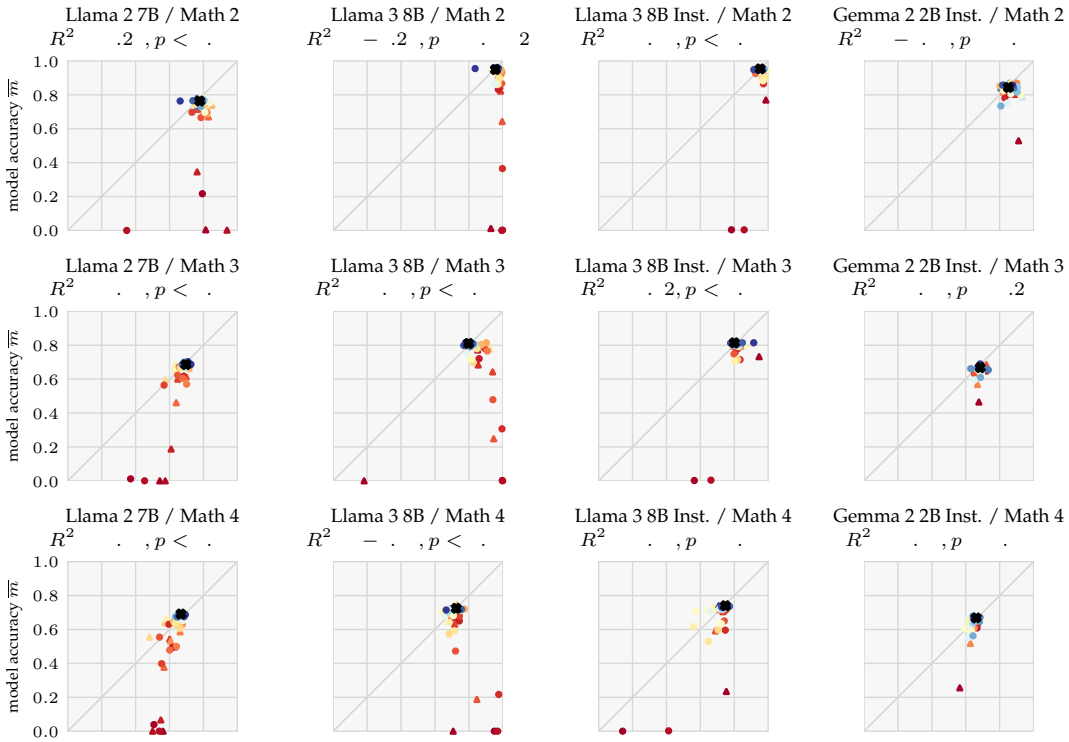

Figure 5: Results of zero-ablation for selected models on math datasets. Circle, triangle, and X markers represent MLP ablations, attention ablations, and clean runs respectively. Warmer colors represent earlier layers. Error bars for individual points are omitted for legibility, but std. err. < 0.032 in all cases (by the bounds on p and m).

E Statistics for Causal Tracing

Model	$p < 0.05$	$p \geq 0.05$
Llama 2 7B	55	0
Llama 2 7B Chat	51	0
Llama 3 8B	53	1
Llama 3 8B Instruct	51	2
Llama 2 13B	51	3
Llama 2 13B Chat	45	4
Gemma 2 2B Instruct	47	1
Gemma 2 9B Instruct	50	0

Table 3: Number of occurrences of p-values for causal tracing.

F Full Results for Zero-Ablation (Leave-One-Out)

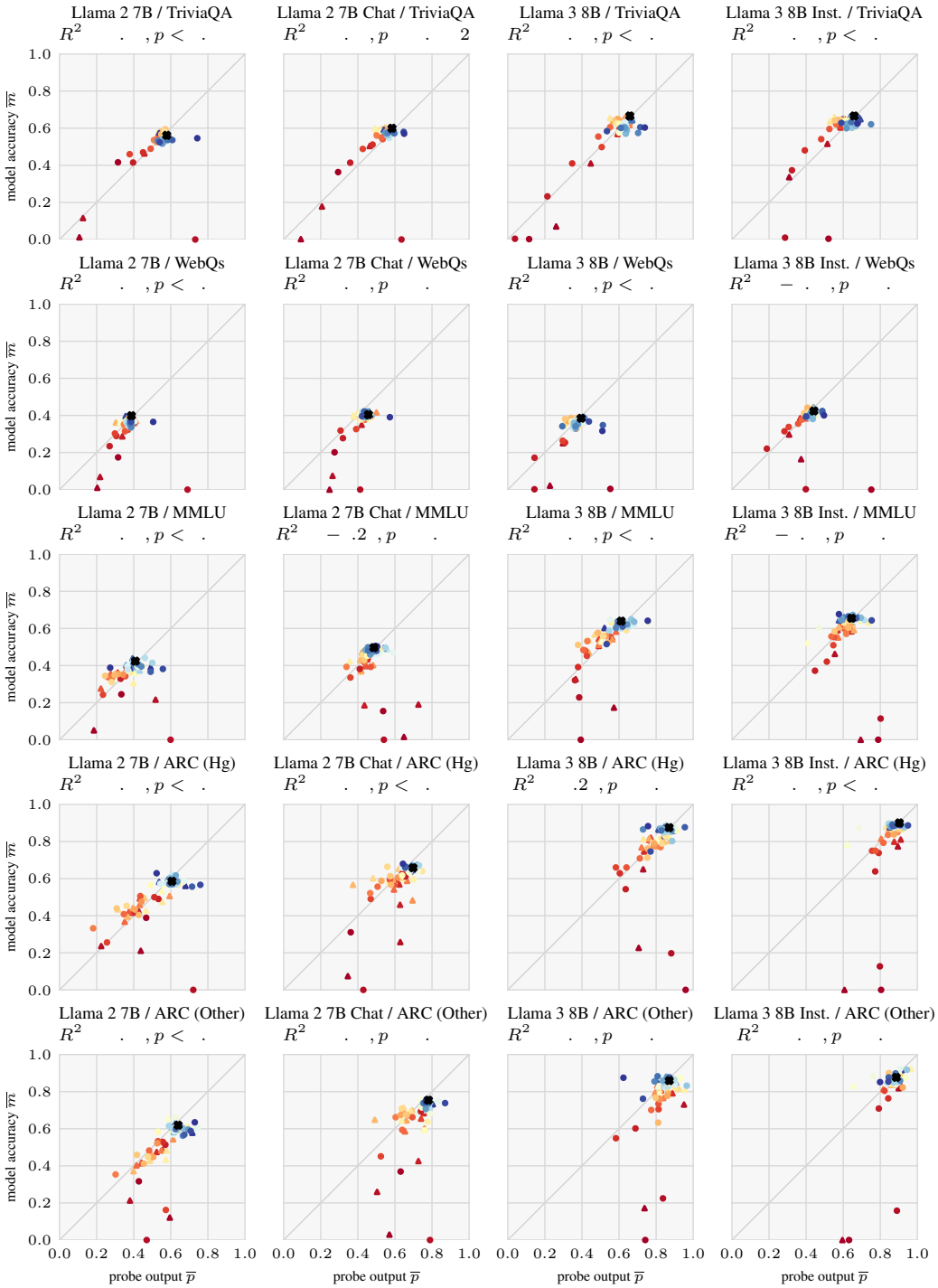

Figure 6: Results of zero-ablation for eight models and five datasets. Circle, triangle, and X markers represent MLP ablations, attention ablations, and clean runs respectively. Warmer colors represent earlier layers. Error bars for individual points are omitted for legibility, but std. err. < 0.032 in all cases (by the bounds on p and m).

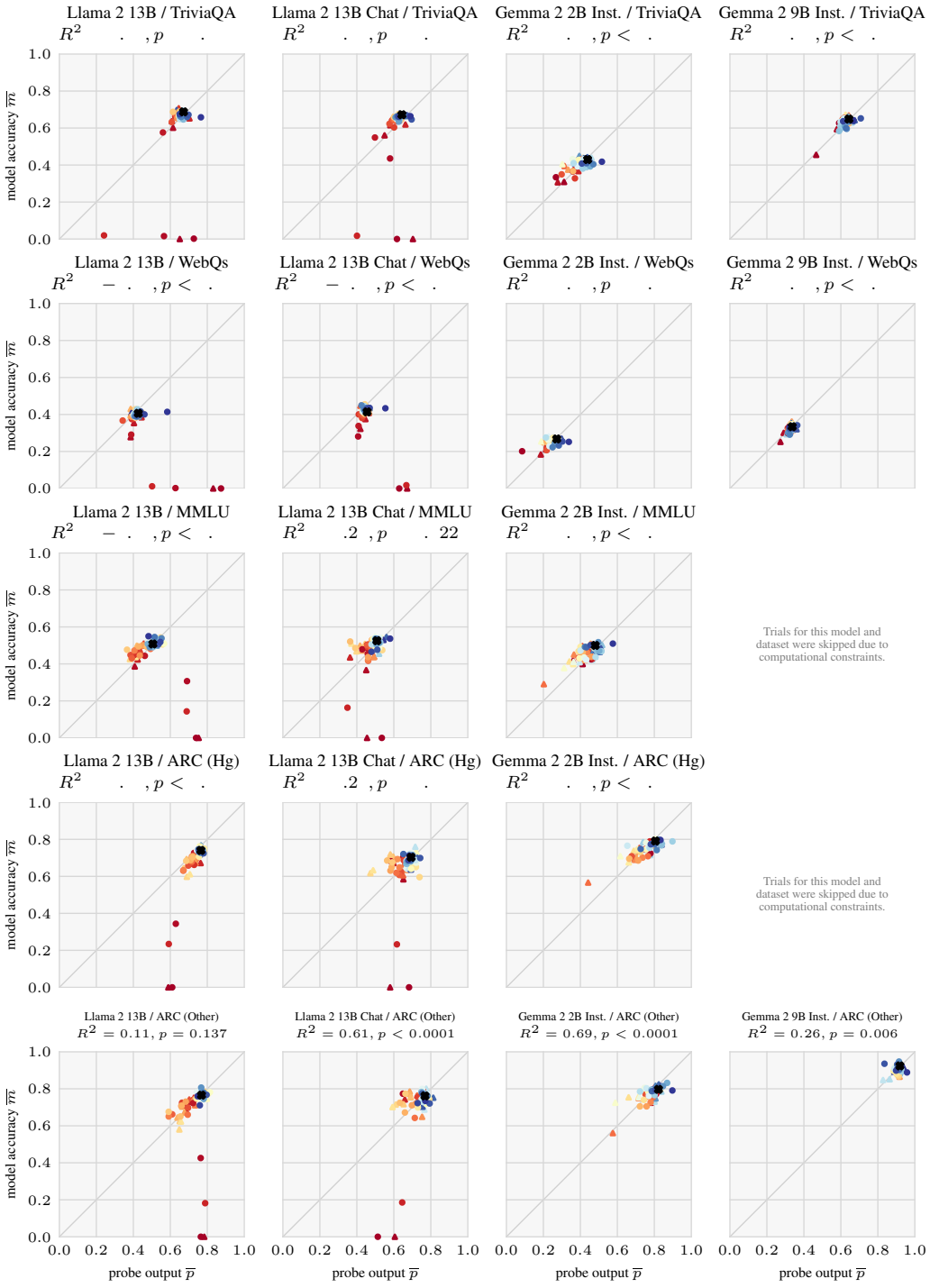

Figure 7: *(continued)* Results of zero-ablation for eight models and five datasets. Circle, triangle, and X markers represent MLP ablations, attention ablations, and clean runs respectively. Warmer colors represent earlier layers. Error bars for individual points are omitted for legibility, but std. err. < 0.032 in all cases (by the bounds on p and m).

G Full Results for Zero-Ablation (COAR)

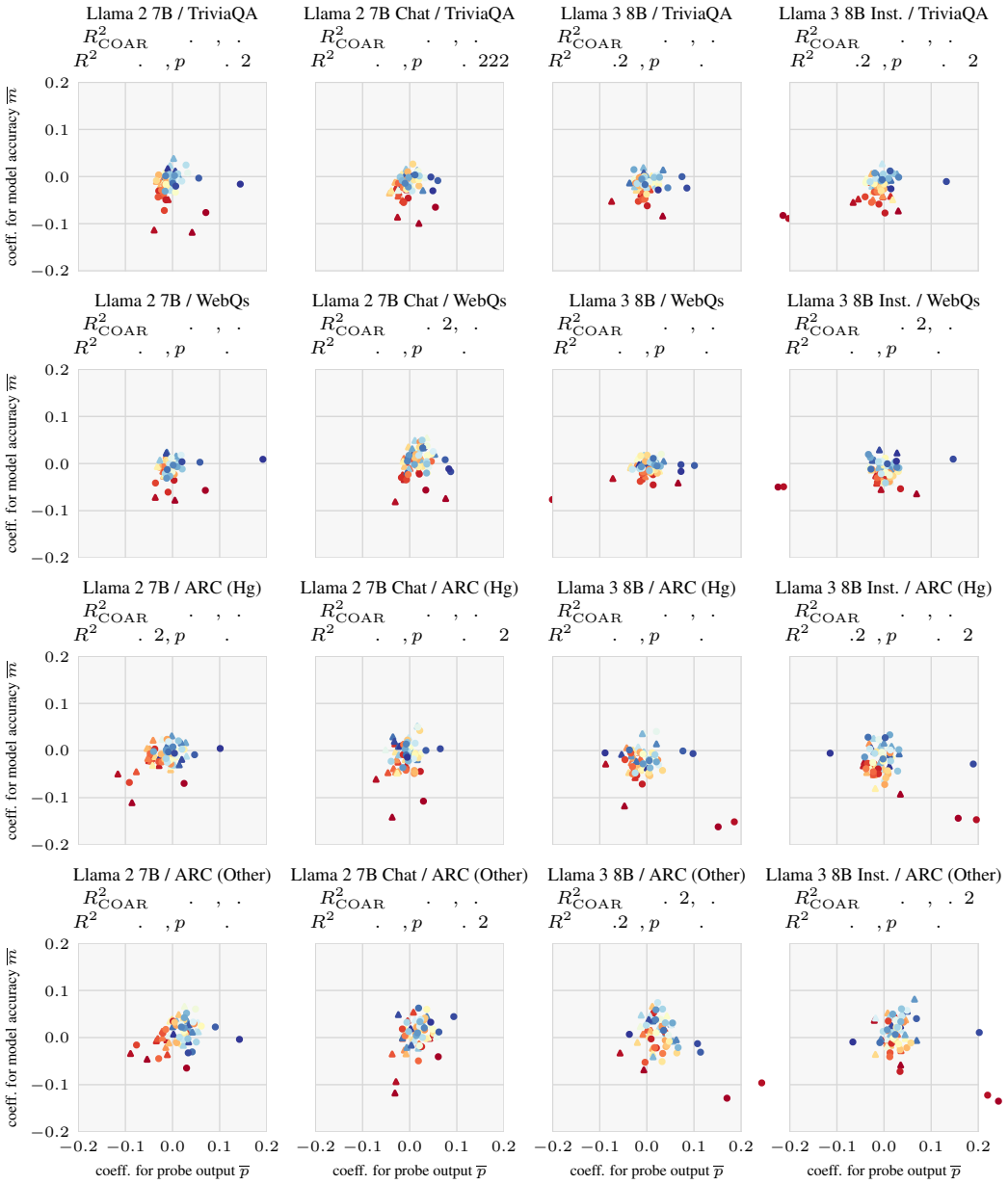

Figure 8: COAR coefficients for zero-ablation for eight models and four datasets. Circle and triangle markers represent MLP and attention ablations respectively. Warmer colors represent earlier layers. Error bars for individual points are omitted for legibility. The two values for R^2_{COAR} are the fraction of variance in m and p explained by the COAR prediction from the set of ablations.

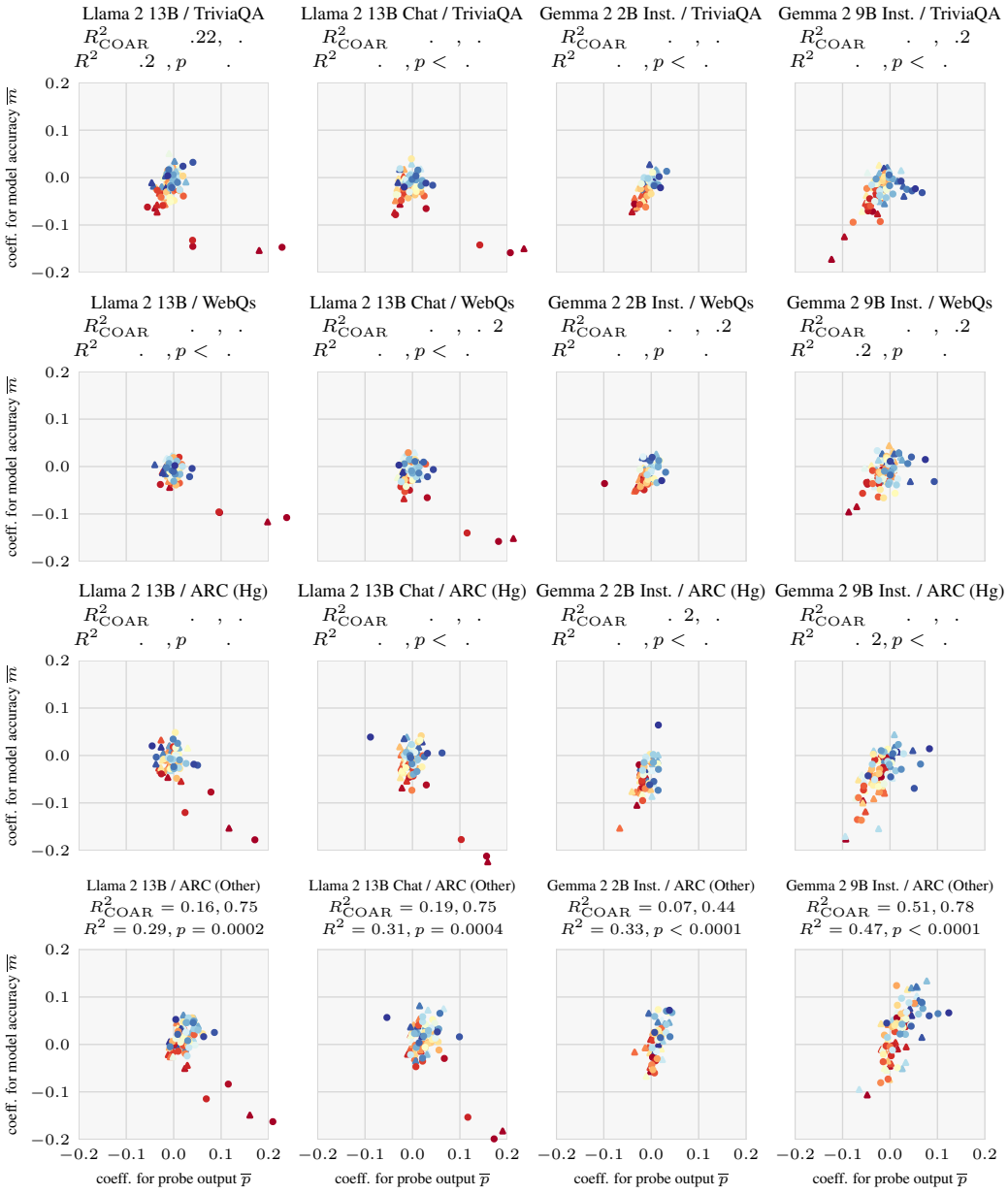

Figure 9: *(continued)* COAR coefficients for zero-ablation for eight models and four datasets. Circle and triangle markers represent MLP and attention ablations respectively. Warmer colors represent earlier layers. Error bars for individual points are omitted for legibility. The two values for R^2_{COAR} are the fraction of variance in m and p explained by the COAR prediction from the set of ablations.

H Computational Resources

This project has used approximately 1200 GPU-hours of computation time on an academic cluster, mainly on RTX8000 GPUs with 48 GB of memory, including approximately 600 GPU-hours for results used directly in this paper. Results for individual model/dataset combinations can be reproduced independently; for example, the code to produce the TriviaQA / Llama 3 8B Instruct results ran in approximately 20 GPU-hours.

I Ethics Statement

This paper intends to advance the areas of interpretability and uncertainty quantification for language models, with the primary aim of making language models more reliable and more trustworthy. We expect these research directions in general to reduce societal risks from machine learning (for example, by allowing for warning signals in situations where a model might be lying or making a dangerous mistake). Nevertheless, since reliability work also makes systems more useful, some caution is warranted: for example, users might be tempted to deploy the resultant more-reliable systems in higher-stakes contexts in which tail risks from failures are greater.

The humanoid and sciuroid robots in Fig. 1 were created using DALL-E 3.