When Attribution Patching Lies: Diagnosis and a Second-Order Correction

Luyang Zhang
Carnegie Mellon University
luyangz@andrew.cmu.edu

Jialu Wang
University of California, Santa Cruz
faldict@ucsc.edu

Abstract

A central goal of mechanistic interpretability is to identify which internal components causally drive a language model's behavior. Because these importance estimates serve as the evidence for identifying circuits, systematic errors can lead to the misidentification of the underlying mechanisms. While activation patching provides a gold-standard causal metric, its computational cost is prohibitive at scale. Practitioners instead rely on attribution patching, a gradient-based, first-order approximation whose reliability remains poorly understood. In this work, we characterize the source of this unreliability, demonstrating that the dominant error stems from the non-linearities in the downstream network rather than local curvature at the patched component. This insight yields three practical tools: (i) a reliability score to detect untrustworthy estimates, (ii) error bounds quantifying potential attribution mis-specifications, and (iii) a Hessian-vector-product (HVP) correction that eliminates the leading-order error with only one additional backward pass. In evaluations across five model families (124M–9B parameters) and both random-token and naturalistic (name-swap) perturbations, HVP is the only second-order correction feasible at larger scale, where standard baselines like Integrated Gradients become computationally prohibitive. In comparative experiments, a multi-step HVP variant matches or exceeds the accuracy of Integrated Gradients at significantly lower compute, outperforming prior second-order baselines. These improvements lead to higher-fidelity circuit recovery on standard benchmarks and support a *Screen-Flag-Fix* workflow that targets computational effort only toward the components flagged as unreliable.

1 Introduction

As language models grow in scale and capability, understanding their internal mechanisms becomes increasingly important. Mechanistic interpretability seeks to provide this understanding by explaining model behavior in terms of internal computations. A central step in that agenda is causal localization: identifying which attention heads, neurons, or features causally drive a given behavior. These localization scores are often used to support circuit claims [1–3], so systematic error can lead to incorrect mechanistic conclusions. The gold-standard causal test is *activation patching* [4, 5], which replaces a component's activation under a corrupted input with the clean-input value and measures the output change. But its cost scales linearly with the number of components, quickly becoming prohibitive in modern Large Language Models (LLMs). In practice, broad localization therefore relies on cheaper approximations, with direct interventions reserved for a short list of components.

A commonly used approximation is *attribution patching* [6], which replaces many explicit interventions with a single backward pass by linearizing the effect around a corrupted activation. However, this linearization can be unreliable when downstream nonlinearities are strong. Prior work has identified concrete failure modes and proposed partial remedies [7–9], yet a fundamental question

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

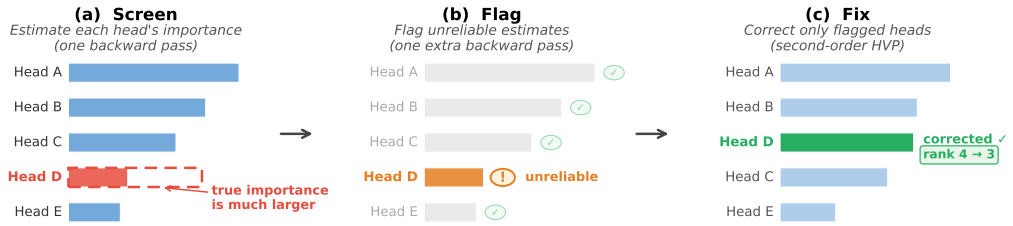


Figure 1: **Screen–Flag–Fix pipeline** for reliable attribution patching. (a) Attribution patching screens all heads cheaply; (b) a reliability score flags suspect estimates; (c) HVP corrects only the flagged heads, recovering the true ranking.

remains: for a given component, when is attribution patching reliable, how large can its error be, and how should it be corrected?

We address this gap by analyzing the structure of attribution patching error. The key finding is that the dominant error stems from the network's response to the intervention, rather than from the local nonlinearity at the patched component. A second-order Taylor expansion makes this precise: the error splits into a leading quadratic term, computable from a single Hessian-vector product (HVP), plus a cubic remainder. This decomposition yields three key results: (1) a reliability score for flagging unreliable estimates; (2) error bounds with a provable $1/K^2$ convergence rate, where K is the number of sub-steps a correction is split into; and (3) an explanation for the insufficiency of prior fixes [7, 8]. Specifically, while prior methods use local curvature, we show that the true error depends on the downstream network response, a quantity that differs from local metrics by 2–66$\times$ with a near-zero correlation.

Building on this analysis, we propose a *Screen–Flag–Fix* pipeline (Figure 1): screen all components with attribution patching, flag unreliable ones via the reliability score, and correct only those with an HVP-based second-order fix. For large perturbations where the single-step expansion overshoots, a multi-step variant (MS-HVP) splits the correction into K sub-steps, each evaluated locally. We evaluate across five model families (124M–9B parameters), comparing against Integrated Gradients (IG) [10], Integrated Hessians (IH) [11], and GIM [8]. *At larger scale* (8B+ parameters), IG requires $\sim$25 GPU-days per task; HVP is the only second-order correction demonstrated at this scale, reducing error by up to 82% (MS-HVP K=5 on Llama-3.1-8B). *At smaller scales*, MS-HVP matches or exceeds IG: on the hardest setting, MS-HVP at cost 10 achieves better accuracy than IG at cost 35 (3.5$\times$ cheaper, $p = 0.022$), and at matched cost MS-HVP wins by 1.2 pp ($p < 0.001$). MS-HVP also outperforms IH by 1.5%–13.9% across all nine tasks at smaller scales. *For circuit recovery*, these per-component gains translate to improved head recovery on IOI and Greater-Than benchmarks.

We release our code for reproducing the experimental results at `https://github.com/LZhang12345/attribution-patching-lies`. Our contributions are three-fold:

- **A reframe of attribution patching reliability.** We trace the dominant error to the downstream network response, not local nonlinearity, explaining why prior local fixes are structurally incomplete.
- **A scalable second-order correction.** HVP (and its multi-step variant MS-HVP) remains tractable at 8B+ parameters where existing refinement baselines become infeasible, the first second-order correction demonstrated at this scale.
- **A design space for attribution-patching refinement.** We organize prior methods by their goals, including estimation accuracy, feature interactions, and circuit faithfulness, clarifying which tool fits which use case.

2 Related work

Patching methods for mechanistic localization. Activation patching [4, 5] provides the causal reference intervention for mechanistic localization by measuring the effect of intervening on hidden activations. Because this cost scales linearly with the number of candidate components, attribution

patching [6] replaces many interventions with a first-order approximation and has become a standard localization primitive in automated circuit discovery [1, 12], sparse feature circuits [2], and large-scale circuit tracing [3]. Practical guides such as Heimersheim and Nanda [13] and methodological studies of activation-patching metrics and corruption choices [14] note that patching often behaves approximately linearly while remaining sensitive to nonlinearities and design decisions, making attribution-patching reliability central to patching-based workflows. Recent large-scale circuit tracing [3, 15] applies attribution to sparse autoencoder (SAE) features [2] rather than raw heads or neurons; HVP is directly applicable to any differentiable activation, though the SAE encoder's nonlinearity (ReLU or top-k) introduces an additional curvature source whose magnitude we leave to future work.

Attribution-patching reliability and circuit validation. Several failure modes of attribution patching are known: Kramár et al. [7] identify activation-region mismatch and cancellation; Edin et al. [8] show that softmax redistribution systematically biases gradient-based localization; Méloux et al. [16] document instability under prompt and hyperparameter variation; Sharkey et al. [17] identify gradient-attribution error as a standing open problem. Alternative attribution rules such as RelP [9] and EAP-GP [18] improve faithfulness in circuit discovery. A complementary line of work evaluates recovered circuits against known causal structure: Shi et al. [19] formalize faithfulness and minimality tests, Mueller et al. [20] and Gupta et al. [21] benchmark localization methods, and formal mechanistic-interpretability work [22, 23] studies provable robustness. Generally, these efforts clarify failure modes and evaluation criteria but do not provide a general account of when attribution patching is numerically reliable or how to correct it across component types and architectures. A complementary approach, Guided-AP [24], learns systematic attribution-patching errors from a small number of exact activation-patching measurements and uses them to correct unmeasured estimates within and across prompts.

Causal abstraction and higher-order analysis. Geiger et al. [25] place mechanistic interpretability in a broader causal-abstraction framework. Related work [26, 27] studies whether models adhere to an interpretable causal structure, a question that remains orthogonal to the numerical reliability of the underlying attribution-patching estimates. Separately, higher-order attribution methods, such as Integrated Gradients [10], Integrated Hessians [11], compositional curvature analyses [28], and influence-function HVPs [29], show that network-level curvature is both informative and tractable, but none provides a reliability account for patching at internal activations.

3 Diagnosing and correcting attribution-patching error

In this section, we analyze when attribution patching is unreliable and how to correct it. The key idea is that attribution patching error can be decomposed into two parts: a dominant quadratic term (computable from a single Hessian-vector product) and a smaller cubic remainder. We set up this decomposition (§3.1), use it to define a reliability score that flags unreliable attributions (§3.2), and then show why local activation curvature cannot predict the dominant error term before introducing network-level HVP, MS-HVP, and selective correction (§3.3).

3.1 Problem setup and error decomposition

Causal localization compares two matched inputs that differ in a controlled way (e.g., swapping a name or replacing a token) and asks which internal components account for the resulting change in output. For a given component (neuron, attention head, or residual-stream position), we compare its activation a under one input to its perturbed counterpart $a' = a + \delta$ under the other. Activation patching measures the true causal effect of this substitution by intervening directly, whereas attribution patching approximates it using a single backward pass. Our focus is on characterizing the error introduced by this approximation.

Patching a single component changes a scalar output metric as follows. Let $M : \mathbb{R}^d \to \mathbb{R}$ denote the scalar score read out from the model output as a function of the activation being patched (e.g., a logit difference or the log-probability of a target token), and let $a \in \mathbb{R}^d$ denote the activation at the component of interest. Throughout this local analysis, we treat the rest of the input context and model computation as fixed. *Activation patching* measures the true effect of replacing a with the counterfactual $a' = a + \delta$: $\Delta = M(a + \delta) - M(a)$. *Attribution patching* approximates this via a

first-order Taylor expansion:

$$\hat{\Delta} = \nabla_a M \cdot \delta \,. \tag{1}$$

Here $\nabla_a M$ is the gradient of the scalar metric with respect to the activation a, so $\hat{\Delta}$ is the first-order attribution-patching estimate along the patch direction δ. The approximation error is $E = \Delta - \hat{\Delta}$. By Taylor's theorem with integral remainder,

$$E = \underbrace{\tfrac{1}{2}\,\delta^\top H \delta}_{\text{dominant (Hessian) term}} + \underbrace{\Phi(\delta)}_{\text{remainder}}, \qquad H = \nabla_a^2 M(a). \tag{2}$$

The quantity $\delta^\top H \delta$ is the Hessian quadratic form along the patching direction δ, i.e., the second-order curvature of the scalar metric in the direction induced by the patch.

To bound the higher-order terms in Eq. (2), we adopt a path-local Lipschitz-Hessian condition, the standard route to a cubic Taylor remainder in second-order optimization [30, 31] and a common tool in higher-order analyses of neural networks and self-attention [11, 28, 32–34]. We use it only along the patch segment, not as a deployment-computable global constant.

Assumption 1 (Local third-order smoothness)**.** *The scalar metric M is twice continuously differentiable on a neighborhood of the line segment $\{a + t\delta : t \in [0,1]\}$, and there exists a finite constant L_3 such that*

$$\left\|\nabla_a^2 M(a + t\delta) - \nabla_a^2 M(a)\right\|_{\mathrm{op}} \le L_3\, t \|\delta\| \qquad \textit{for all } t \in [0,1]. \tag{3}$$

How the assumption applies to patching. The assumption is local: it applies only along the patch segment and yields the remainder bound $|\Phi(\delta)| \le L_3 \|\delta\|^3/6$. Standard transformer components satisfy it in the relevant regime: exact GeLU and SiLU have bounded third derivatives, softmax is smooth, and normalization layers are smooth away from zero-variance inputs. Empirically, we estimate L_3 along the patch path by probing the Hessian at three interpolation points (Appendix B.2): the resulting cubic bound holds for 82% of GPT-2 and 95% of Gemma-2-2B component–prompt pairs, with median slack factors of $3.4\times$ and $7.6\times$ respectively.

3.2 Reliability score and error bounds

We now turn the decomposition into a diagnostic. The goal is to decide, before running activation patching, whether a first-order attribution-patching score is likely to be trustworthy for a given component. The quadratic term is computable from one HVP, but its raw magnitude is hard to interpret without a scale; we therefore normalize it by the first-order estimate. The result is a relative reliability score that tracks attribution patching's relative error up to the cubic remainder and is used later to flag components for selective HVP/MS-HVP correction.

Definition 1 (Reliability score)**.** *For a component with Hessian $H = \nabla_a^2 M(a)$, perturbation δ, and nonzero first-order estimate $\hat{\Delta} \neq 0$, the* reliability score *is $\tilde{R} = |\delta^\top H \delta| / (2\,|\hat{\Delta}|)$.*

The denominator is chosen to match the practical question: whether the attribution patching estimate itself is stable. A small $\tilde{R}$ means the leading omitted term is small relative to the first-order estimate, while $\tilde{R} \approx 1$ means downstream curvature can change the attribution's magnitude or sign. Normalizing by the true effect Δ would require activation patching, so it is not available as a screening diagnostic; in near-zero cases, the absolute error bounds from Eq. (2) are the right object to inspect.

Proposition 1 (Local attribution-patching error bound via the reliability score)**.** *Assume Assumption 1, $\hat{\Delta} \neq 0$, and $\delta^\top H \delta \neq 0$. Let $\alpha = \frac{L_3 \|\delta\|^3}{3\,|\delta^\top H \delta|}$ be the higher-order slack parameter. Then $\left||E|/|\hat{\Delta}| - \tilde{R}\right| \le \alpha \tilde{R}$. If additionally $\alpha < 1$, then $(1-\alpha)\tilde{R} \le |E|/|\hat{\Delta}| \le (1+\alpha)\tilde{R}$.*

The full proof, including verification of the remainder bound and a discussion of path-local smoothness for softmax and normalization layers, is in Appendix D.

In practice, $\tilde{R} \ll 1$ means attribution patching is accurate; $\tilde{R} \approx 1$ means the error may rival the estimate itself. The slack α packages the uncomputed cubic remainder: small α corresponds to perturbations well within the Taylor convergence radius, making $\tilde{R}$ a tight proxy for relative error; otherwise $\tilde{R}$ remains a useful diagnostic. Degenerate cases are handled naturally: if $\hat{\Delta} = 0$, use absolute bounds; if $\delta^\top H \delta = 0$, the quadratic term vanishes and $|E| \le \frac{L_3}{6}\|\delta\|^3$.

3.3 Network-level HVP correction

The quadratic error term $\frac{1}{2}\delta^\top H\delta$ is a network-level quantity: it depends on how the patched activation propagates through the full computational graph, not only on local activation curvature. Correcting it therefore requires network-level information. We first show that local curvature corrections fail, then introduce HVP and MS-HVP corrections that compute the relevant term directly.

Why local corrections fail. A natural first attempt is to use the *local* activation curvature. For a pre-activation MLP neuron with activation function f, one might estimate $E \approx \frac{1}{2}f''(z)\delta^2$. This should fail for a structural reason: a neuron's contribution must traverse the rest of the network, and most of that downstream path is linear (residual additions, LayerNorm in its operating regime, the unembedding). Linear operations do not compound curvature, so $|d^2M/da^2|$ inherits little of $|f''(z)|$ unless a downstream nonlinearity happens to align with the patch direction. Cross-layer interactions further introduce curvature invisible to local analysis. We therefore expect a large gap between $|d^2M/dz^2|$ and $|f''(z)|$, and confirm this empirically: the network-level Hessian exceeds local curvature by 2–66$\times$ across three model families, with near-zero correlation ($r = 0.04$); using $f''(z)$ for correction produces $>$750% error – worse than no correction at all (Figure 2).

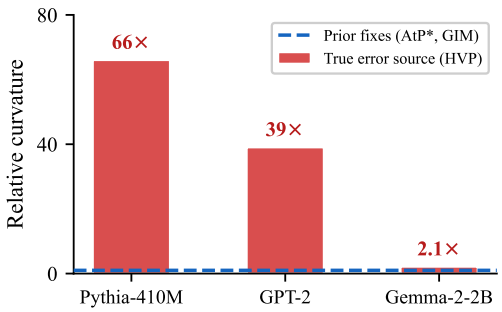


Figure 2: **Network–local curvature gap.** Full-network curvature vs. local component curvature across three models; prior fixes (AtP* [7], GIM [8]) use only the local quantity.

Network-level correction. This gap motivates computing the network-level second-order correction directly.

Definition 2 (HVP-corrected attribution patching)**.** *The HVP-corrected estimate is*

$$\hat{\Delta}_{hvp} = \nabla_a M \cdot \delta + \tfrac{1}{2}\,\delta^\top H\delta = \hat{\Delta} + \tfrac{1}{2}\,\delta^\top H\delta\,. \tag{4}$$

Computing $\delta^\top H\delta$ requires the Hessian-vector product $H\delta$, which we obtain via two backward passes [35]: (1) compute $g = \nabla_a M$ while retaining the gradient computation graph so that g itself can be differentiated; (2) differentiate $g \cdot \delta$ with respect to a to obtain $H\delta$. The dot product $\delta^\top(H\delta)$ then gives the quadratic form. This costs $\sim 3\times$ a single backward pass (one forward + two backward).[1]

Corollary 1 (Residual error after HVP correction)**.** *The residual error after HVP correction satisfies* $|E_{hvp}| = |\Delta - \hat{\Delta}_{hvp}| = |\Phi(\delta)| \le L_3\|\delta\|^3/6$. *The proof is in Appendix C.2.*

Corollary 1 shows that HVP reduces the error of attribution patching from $O(\|\delta\|^2)$ to $O(\|\delta\|^3)$, one order tighter as $\delta \to 0$. However, when $L_3\|\delta\|$ is comparable to the Hessian scale, the cubic remainder dominates and a single Taylor step overshoots. To address this, we introduce *Multi-Step HVP* (MS-HVP), which splits the patch into K equal sub-steps and applies a second-order correction at each intermediate point along the path from a to $a+\delta$. Setting $K=1$ recovers standard HVP; increasing K shrinks each sub-step's remainder at the cost of additional HVP evaluations. Algorithm 1 summarizes the procedure.

[1] Measured wall-clock overhead on Pythia-410M is 2.8$\times$ rather than 3$\times$ due to caching effects from the retained computation graph.

Algorithm 1 Multi-step HVP attribution patching

Require: Model M, activation a, perturbation δ, sub-steps K

$\hat{\Delta}_{\text{ms}} \leftarrow 0,\ s \leftarrow \delta/K$
for $k = 1, \ldots, K$ **do**
  $a_{k-1} \leftarrow a + (k-1)s$
  $g_k \leftarrow \nabla_a M(a_{k-1})$ ▷ Forward + first backward
  $v_k \leftarrow \nabla_a(g_k \cdot s)$ ▷ HVP via second backward
  $\hat{\Delta}_{\text{ms}} \leftarrow \hat{\Delta}_{\text{ms}} + g_k \cdot s + \frac{1}{2}\, s \cdot v_k$
end for
return $\hat{\Delta}_{\text{ms}}$

Concretely, MS-HVP evaluates the gradient and Hessian at each intermediate activation $a_{k-1} = a + \frac{k-1}{K}\delta$ and accumulates:

$$\hat{\Delta}_{\text{ms}} = \sum_{k=1}^{K} \Big[\nabla_a M(a_{k-1}) \cdot \tfrac{\delta}{K} + \tfrac{1}{2K^2}\, \delta^\top H(a_{k-1})\delta \Big]. \tag{5}$$

Because each sub-step's cubic remainder scales as $\|\delta/K\|^3$ and there are K such steps, the aggregate residual satisfies $|E_{\text{ms}}| \le L_3\|\delta\|^3/(6K^2)$ by standard quadrature analysis. MS-HVP thus trades linearly more HVP evaluations for a quadratic reduction in the Taylor remainder, predicting diminishing returns as K grows. We test this cost–accuracy prediction empirically in §4.

Selective HVP: deciding which components to correct. MS-HVP improves the accuracy of individual corrections; a separate question is *which* components need correction at all. In practice, one rarely applies HVP to every candidate component. The following result formalizes the aggregate error of the Screen → Flag → Fix workflow: compute $\tilde{R}$ for all components, apply HVP correction only to those with $\tilde{R} \ge \tau$, and trust attribution patching for the rest.

Proposition 2 (Selective-HVP pipeline guarantee)**.** *Consider components $i = 1, \ldots, n$ patched independently on the same input. Let Δ_i denote the true effect of patching only component i, $\hat{\Delta}_i$ its attribution-patching estimate, and $c_i := \frac{1}{2}\delta_i^\top H_{ii}\delta_i$ its diagonal second-order correction. For a threshold $\tau > 0$, define $S_{\text{ok}} := \{i : \tilde{R}_i < \tau\}$, $S_{\text{flag}} := \{i : \tilde{R}_i \ge \tau\}$, and the selective estimate*

$$\hat{\Delta}_{\text{sel}} := \sum_{i=1}^{n} \hat{\Delta}_i + \sum_{i \in S_{\text{flag}}} c_i. \tag{6}$$

with the convention that $\tilde{R}_i = +\infty$ when $\hat{\Delta}_i = 0$. If Assumption 1 holds along each componentwise patching segment with constants $L_{3,i}$, then

$$\Big|\textstyle\sum_{i=1}^{n} \Delta_i - \hat{\Delta}_{\text{sel}}\Big| \;\le\; \tau \textstyle\sum_{i \in S_{\text{ok}}} |\hat{\Delta}_i| \;+\; \frac{1}{6}\sum_{i=1}^{n} L_{3,i}\|\delta_i\|^3. \tag{7}$$

The proof is in Appendix D.

The independent-patching assumption matches the standard practice in circuit discovery: tools such as ACDC [1] and EAP [12] consume per-component rankings. Joint patches add off-diagonal Hessian contributions $\delta_i^\top H_{ij}\delta_j$; these affect aggregate circuit-level metrics but not per-component ranking (measuring them would require $O(n^2)$ HVPs). Appendix C.3 gives an exact identity comparing selective and full diagonal HVP plus a sign-aware refinement.

4 Experiments

We evaluate whether HVP corrects attribution-patching error broadly, how it compares to existing second-order methods at matched cost, and whether per-component accuracy gains translate to improved circuit recovery.

Table 1: **Comparison of error rate.** Top-5 relative error (%) across fourteen model-task pairs and six methods (lower is better; Std HVP = MS-HVP K=1). Column N: number of evaluation prompts; K: HVP sub-steps; S: IG interpolation steps.

Model / Task	N	Attrib. Pat.	Std HVP	MS-HVP (K=5)	IG (S=10)	IH-PI	GIM
Foreground (full method coverage)							
Pythia-410M IOI*	280	22.34	47.48†	18.03⋆	**17.26**⋆	22.22	34.07°
GPT-2 IOI	100	18.07	12.00⋆	**2.97**⋆	3.30⋆	10.15⋆	54.71°
Gemma-2-2B factual	55	51.71	43.16⋆	40.66⋆	**40.57**⋆	44.41⋆	77.68°
Cross-model coverage (Std HVP / IG / GIM)							
Pythia-410M factual	55	68.65	65.58⋆	64.41⋆	**63.91**⋆	65.94⋆	75.07°
Pythia-410M GT	200	35.68	25.55⋆	**21.47**⋆	24.62⋆	28.48⋆	39.65°
GPT-2 GT	200	44.06	33.80⋆	**23.33**⋆	32.98⋆	37.25⋆	41.35⋆
Gemma-2-2B IOI	50	16.53	9.07⋆	**6.58**⋆	6.63⋆	10.54⋆	25.65°
Qwen2.5-1.5B IOI	50	14.80	5.69⋆	**2.94**⋆	3.16⋆	8.13⋆	24.75°
Qwen2.5-1.5B factual	35	41.57	35.79⋆	32.06⋆	**31.95**⋆	35.52⋆	63.25°
Gemma-2-2B GT	50	7.41	**0.62**⋆	—	—	—	—
Larger scale (IG/IH infeasible)							
Llama-3.1-8B IOI	50	19.78	8.42⋆	**3.54**⋆	‡	‡	30.56°
Llama-3.1-8B factual	55	53.19	50.50⋆	**48.69**⋆	‡	‡	71.35°
Llama-3.1-8B GT	200	19.51	7.75⋆	**6.51**⋆	‡	‡	21.17°
Gemma-2-9B factual	55	60.82	56.97⋆	**56.93**⋆	‡	‡	89.50°

⋆Significantly better than AP ($p < 0.05$, paired bootstrap). °Significantly worse. *Pathological; recovered by MS-HVP $K \geq 3$. †Catastrophic; see §4.2. ‡Infeasible (∼25 GPU-days/task). — Tokenizer-incompatible: the greater_than task requires patching a single token representing a two-digit number (e.g. "42"), but Gemma's tokenizer splits it into multiple tokens.

Note: AtP* [7] and RelP [9] are omitted due to different granularity.

4.1 Setup

All experiments use TransformerLens [36] with 95% paired bootstraps (1,000 resamples). Full details are in Appendices E.1 and E.2. We evaluate five model families (124M–9B parameters): GPT-2 [37], Pythia (410M, 2.8B) [38], Qwen2.5-1.5B [39], Gemma (2B, 9B) [40], and Llama-3.1-8B [41]. We report attention heads in all models, pre-activation MLP neurons in GeLU models, and MLP layer outputs in Qwen. For factual and Greater-Than tasks, perturbations replace position 3 with a random vocabulary token; for IOI tasks, the corruption is the standard name-swap [5], a naturalistic perturbation. In both cases, δ is the resulting activation shift and M is the log-probability of the correct next token. Our primary metric is *top-5 relative error*: the mean absolute error of a method's scores on the five components with largest activation-patching effect, normalized by the ground-truth range. Confidence intervals are reported in the appendix tables.

Our IG baseline [10] integrates the gradient along the patch path using S interpolation steps (cost S per head); we use *per-head* IG, which differs from all-at-once EAP-IG [12] (Appendix E.3). Cost is measured in backward passes per head per prompt: attribution patching $= 1$, Std HVP $= 2$, MS-HVP $K = 2K$ (backward passes only; wall-clock overhead is moderately higher; see Appendix E.4).

Our model-task selection follows the theoretical prediction of §3: attribution-patching error should be largest where the network-level quadratic form $\delta^\top H \delta$ dominates. Pythia-410M IOI is the predicted high-error regime (Taylor radius violated); GPT-2 IOI is the predicted low-error regime (attribution patching already accurate); Gemma-2-2B factual sits between these extremes. Seven additional model-task pairs and four tasks with larger-scale models provide cross-architecture coverage (Table 1).

4.2 Broad gains from HVP correction

Consistent with the curvature-gap analysis (§3.3, Figure 2), local activation curvature is a poor predictor of attribution-patching error. In contrast, the reliability score $\tilde{R}$ tracks network-level curvature and accurately localizes failure regions. Figure 3 shows the reliability-score diagnostic on Pythia-410M, while Figure 4 shows that $\tilde{R}$ sharply concentrates in the IOI-circuit layers of Pythia-410M, precisely where attribution patching incurs its largest errors. Applying the network-level

HVP correction substantially reduces error throughout these layers. Full HVP correction results are deferred to Appendix G.1.

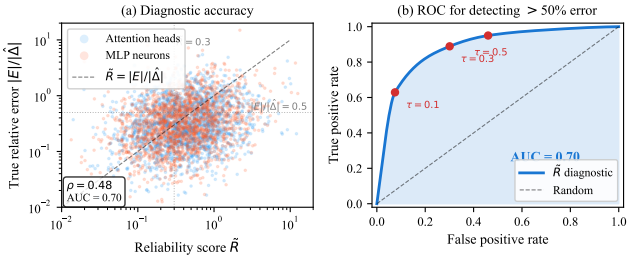


Figure 3: Reliability score $\tilde{R}$ as a diagnostic for attribution-patching failure on Pythia-410M. **(a)** Scatter of $\tilde{R}$ vs. true relative error for attention heads (blue) and MLP neurons (red), $n = 3{,}600$. Spearman $\rho = 0.48$. Dashed lines mark the recommended thresholds. **(b)** ROC curve for detecting $>50\%$ relative error. AUC = 0.70 $[0.67, 0.73]$. At $\tilde{R} > 0.3$ (marked), recall is 89% and precision 36%, flagging 40% of components.

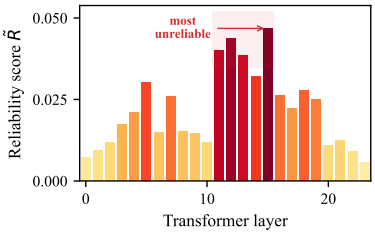


Figure 4: **Reliability score by layer.** $\tilde{R}$ by transformer layer (Pythia-410M IOI, heads with top-quartile causal effect). Error concentrates in the IOI-circuit layers (11–15).

These improvements translate into broad empirical gains across architectures and tasks (Table 1). Standard HVP correction (K=1, cost 2) reduces top-5 relative error by 72–90% on attention heads and 56–67% on pre-activation MLP neurons. In contrast, gains on post-activation neurons are small (2–7%), consistent with the near-linearity of activations after the nonlinearity has already been applied. The same qualitative pattern holds across all evaluated architectures, including SwiGLU and GeGLU models such as Qwen and Gemma. The strongest results occur in low-to-moderate curvature regimes. For example, on Gemma-2-2B Greater-Than, a single HVP pass reduces the top-5 relative error from 7.41% to 0.62%, approaching exact recovery of the activation-patching ranking.

Beyond aggregate metrics, HVP correction also improves circuit recovery. On GPT-2 Greater-Than, MS-HVP (K=5) increases top-5 head overlap with activation-patching ground truth from 70.1% to 83.2% (+13.1 pp), outperforming both IG and GIM. Similar improvements hold on Pythia-410M Greater-Than, where all second-order methods converge to comparable rankings while consistently outperforming attribution patching.

Larger-scale models. At 8B+ parameters, per-head IG at S=10 requires ∼25 GPU-days per task ($S \times N_{\text{heads}}$ backward passes per prompt; Appendix E.3), and IH has not been demonstrated at this scale. HVP at K=1 (cost 2) running in 2.4–28.7 GPU-hours is the only second-order correction that fits in a typical research budget. Std HVP reduces error by 5–60% across the four tasks with larger-scale models; on Llama-3.1-8B IOI, MS-HVP K=5 further pushes error from 8.42% to 3.54%, an 82% reduction from attribution patching, confirming that multi-step correction remains beneficial at 8B scale.

Pathological high-curvature regime. Pythia-410M IOI is the unique scenario where $|\delta|$ exceeds the Taylor convergence radius: Std HVP overshoots to 47.48%, validating Cor. 1. Multi-step composition restores stability at the predicted $\mathcal{O}(1/K^2)$ rate (Figure 5a), with knee at K=5 and diminishing returns beyond K=10. Direct higher-order alternatives (third-order Taylor, finite-difference HVP) fail in this regime but not in low-$|\delta|$ tasks (Appendix G.1).

4.3 Matched-compute comparisons

We compare HVP against existing second-order methods at matched compute budgets on the nine tasks with smaller-scale base models where all baselines are feasible.

Vs. IG. At cost 10, paired-bootstrap comparisons yield seven ties ($p > 0.1$) and two MS-HVP wins (GPT-2 IOI, $p < 0.001$; Qwen IOI, $p = 0.041$). On the pathological case, MS-HVP K=5 at cost 10 beats IG S=35 at cost 35 ($\Delta = -0.44$ pp, $p = 0.022$), demonstrating better accuracy at 3.5× less compute. At matched cost ∼40, a per-head median achieves 17.57% vs. IG's 18.77% ($p < 0.001$;

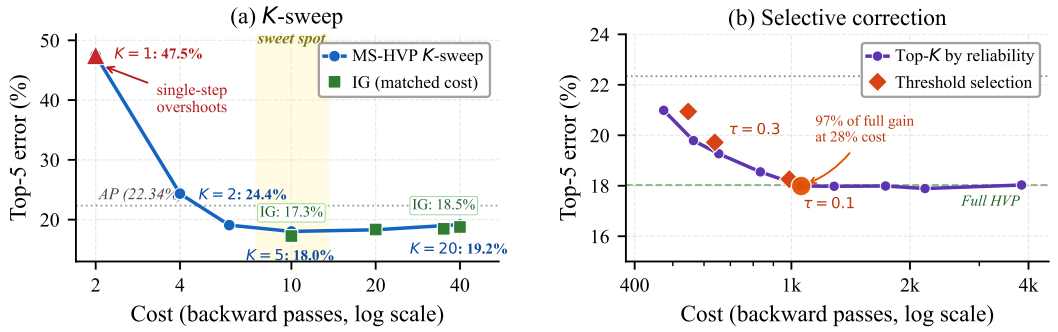


Figure 5: **Multi-step correction and selective workflow. (a)** K-sweep on Pythia-410M IOI: top-5 relative error vs. number of sub-steps K for MS-HVP (blue) and IG (green squares) at matched per-step cost. **(b)** Selective-HVP on GPT-2 IOI: top-5 error vs. total backward-pass cost as more components are corrected, ranked by $\tilde{R}$. $\tilde{R}$-based selection (orange) vs. random baseline (gray).

Appendix F.1). In short, MS-HVP matches or exceeds IG while extending to scales where IG cannot run.

Vs. IH. The gap is structural: IH's path-integrated weighting applies $\int_0^1 2t(1-t)\,dt = 1/3$ to the Hessian term, yielding an effective $1/4$ coefficient on $\delta^\top H\delta$, which is half the $1/2$ needed for exact Taylor correction. This under-correction is by design: IH targets feature-interaction attribution, not error correction. The resulting gap is consistent across all nine tasks (1.5–13.9%; Appendix F.2).

Vs. GIM. GIM [8] targets edge-level faithfulness metrics [20] rather than per-component accuracy. On our per-head metrics, GIM is significantly worse than attribution patching on 12/13 tasks ($p < 0.001$); the sole exception is GPT-2 Greater-Than, where GIM improves top-5 error by 2.7% but still degrades head recovery (Appendix F.3). This does not invalidate GIM for its intended use case but highlights that edge-level and component-level accuracy are distinct objectives – a key finding of our design-space analysis (Table 12 in the appendix organizes methods by their target objective).

4.4 Selective workflow and diagnostics

The reliability score $\tilde{R}$ provides a practical diagnostic for deciding when correction is worthwhile, completing the *Screen-Flag-Fix* pipeline.

Selective correction. Applying HVP only to $\tilde{R}$-flagged components (Figure 5b) captures 97% of full-HVP gain at 28% of the cost (τ=0.1), validating Proposition 2. The $\tilde{R}$-ranked selection substantially outperforms random selection (gray baseline in Figure 5b), confirming that $\tilde{R}$ identifies the components that benefit most from correction. The full operating curve across three models is in Appendix G.2.

Diagnostic accuracy. Figure 6 evaluates the reliability score $\tilde{R}$ after stratifying components by perturbation magnitude $|\delta|$ and true effect magnitude $|f_{\text{true}}|$. Q1–Q4 denote quartiles of the stratified variables. Although overall AUROC varies across models, performance is consistently strong in the regimes that matter most: for Q2–Q4, $\tilde{R}$ achieves AUROC above 0.97 across all evaluated models under both stratifications. Thus, the reliability score accurately identifies components where attribution patching is likely to be unreliable, even when applied without model-specific tuning.

Robustness analysis. Supplementary experiments confirm that HVP's gains are robust across corruption types (Appendix G.3), semantic perturbations (Appendix G.4), and evaluation sample sizes (Appendix G.5). Across random-token, cross-prompt resample, and zero corruptions, HVP consistently reduces attribution-patching error by roughly 78–90% on GPT-2, Gemma-2-2B, and Pythia-1B. Gains are weaker on the pathological Pythia-410M IOI setting (36–50%) and on one Pythia-1B random-token run with unusually large perturbation norms (12%).. HVP also generalizes beyond synthetic token replacements: under semantically coherent entity-swap corruptions, it reduces

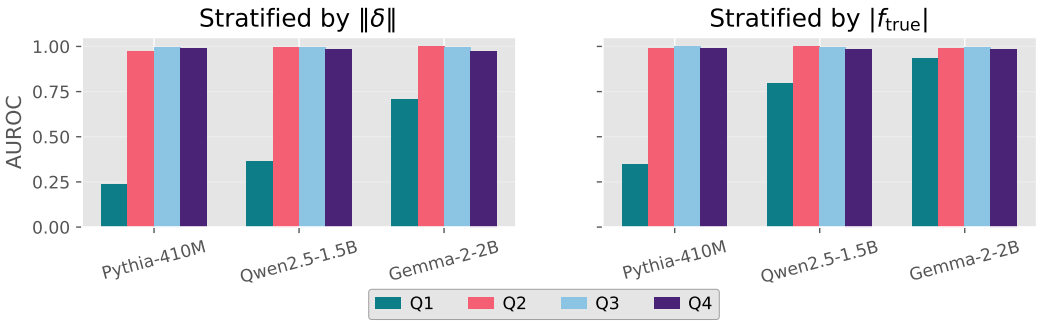


Figure 6: AUROC of $\tilde{R}$ stratified by $\|\delta\|$ quartile and $|f_{\text{true}}|$ quartile. Q1 denotes the smallest-norm quartile.

error by 54% on GPT-2 and 59% on Pythia-410M, indicating that the second-order correction remains effective for larger, structured activation perturbations. Finally, aggregate error-reduction estimates stabilize after approximately 10–15 evaluation prompts and converge to values consistent with the main results, suggesting that the reported gains are not driven by small-sample effects.

4.5 Circuit-recovery payoff

Better component estimates translate to improved circuit discovery. We evaluate this by ranking all attention heads by attribution magnitude and thresholding at the known circuit size to measure overlap with the activation-patching ground truth across multiple architectures and tasks, including GPT-2 IOI/Greater-Than [5, 42], Pythia-410M Greater-Than and Gemma-2-2B Greater-Than. Attribution patching already provides a strong global ranking: Kendall τ rank correlations range from 0.35 on Pythia-410M to 0.72 on Gemma-2-2B factual. HVP matches or slightly improves these global correlations. However, HVP's primary gains concentrate precisely at the ranking boundaries where circuit-membership decisions are made. The full ranking performance is deferred to Appendix G.6.

On GPT-2 IOI, HVP recovers all 20 ground-truth heads at the canonical boundary, compared to 95% for attribution patching. We present a concrete example in Appendix G.7 to illustrate why: head L4H11, a known duplicate-token head, ranks 7th by ground truth but 27th by attribution patching ($4.4\times$ underestimate of its causal effect). HVP partially recovers the true score and promotes L4H11 to rank 12; the reliability score successfully flags it with $\tilde{R} = 1.32 \gg 0.3$, correctly identifying it as a correction target. Appendix G.8 compares the pattern of boundary improvement on Greater-Than task across model architectures.

These gains also improve MIB's circuit-faithfulness metrics [20] and SAE features. We defer more experimental results to Appendix G.9 and Appendix G.10, respectively.

5 Conclusion

We showed that attribution-patching error is dominated by the downstream network response, not local nonlinearity at the patched component. This reframing explains why prior local corrections are structurally incomplete and motivates a simple fix: a Hessian-vector product that captures the missing curvature. The resulting Screen–Flag–Fix pipeline lets practitioners keep the speed of attribution patching where it is already accurate and apply a targeted second-order correction where it is not, and scale to models where existing refinement methods are infeasible. Natural next steps include extending the framework to multi-token semantic corruptions, integrating with circuit-completeness and minimality verification, and scaling the correction to sparse-autoencoder features.

The current framework focuses on single-token perturbations. However, extending to multi-token semantic corruptions is a natural next direction. While we demonstrate consistent gains up to 9B parameters, verifying behaviour at yet larger scales remains an open opportunity. More broadly, HVP improves component-level attribution accuracy but does not by itself address circuit completeness or minimality. Integrating with verification tools is an exciting avenue for future work.

Acknowledgments and Disclosure of Funding

The authors are grateful to the anonymous reviewers for their thoughtful and constructive feedback. LZ would like to thank Tata Consultancy Services for supporting him through the TCS Presidential Fellowship at Carnegie Mellon University. JW conducted this research independently and received no external funding or financial support for this work.

References

[1] Arthur Conmy, Augustine N. Mavor-Parker, Aengus Lynch, Stefan Heimersheim, and Adrià Garriga-Alonso. Towards automated circuit discovery for mechanistic interpretability. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=89ia77nZ8u.

[2] Samuel Marks, Can Rager, Eric J Michaud, Yonatan Belinkov, David Bau, and Aaron Mueller. Sparse feature circuits: Discovering and editing interpretable causal graphs in language models. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=I4e82CIDxv.

[3] Emmanuel Ameisen, Jack Lindsey, Adam Pearce, Wes Gurnee, Nicholas L Turner, Brian Chen, Craig Citro, David Abrahams, Shan Carter, Basil Hosmer, et al. Circuit tracing: Revealing computational graphs in language models. *Transformer Circuits Thread*, 6:16318–16352, 2025.

[4] Nelson Elhage, Neel Nanda, Catherine Olsson, Tom Henighan, Nicholas Joseph, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, et al. A mathematical framework for transformer circuits. *Transformer Circuits Thread*, 1:12, 2021.

[5] Kevin Ro Wang, Alexandre Variengien, Arthur Conmy, Buck Shlegeris, and Jacob Steinhardt. Interpretability in the wild: a circuit for indirect object identification in GPT-2 small. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=NpsVSN6o4ul.

[6] Neel Nanda. Attribution patching: Activation patching at industrial scale. https://www.neelnanda.io/mechanistic-interpretability/attribution-patching, 2023.

[7] János Kramár, Tom Lieberum, Rohin Shah, and Neel Nanda. AtP*: An efficient and scalable method for localizing LLM behaviour to components, 2024. URL https://arxiv.org/abs/2403.00745.

[8] Joakim Edin, Róbert Csordás, Tuukka Ruotsalo, Zhengxuan Wu, Maria Maistro, Casper L. Christensen, Jing Huang, and Lars Maaløe. GIM: Improved interpretability for large language models, 2026. URL https://openreview.net/forum?id=ZRDYvWF1ZJ.

[9] Farnoush Rezaei Jafari, Oliver Eberle, Ashkan Khakzar, and Neel Nanda. RelP: Faithful and efficient circuit discovery via relevance patching, 2025. URL https://arxiv.org/abs/2508.21258.

[10] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, page 3319–3328. JMLR.org, 2017.

[11] Joseph D. Janizek, Pascal Sturmfels, and Su-In Lee. Explaining explanations: axiomatic feature interactions for deep networks. *J. Mach. Learn. Res.*, 22(1), January 2021. ISSN 1532-4435.

[12] Aaquib Syed, Can Rager, and Arthur Conmy. Attribution patching outperforms automated circuit discovery. In Yonatan Belinkov, Najoung Kim, Jaap Jumelet, Hosein Mohebbi, Aaron Mueller, and Hanjie Chen, editors, *Proceedings of the 7th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*, pages 407–416, Miami, Florida, US, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.blackboxnlp-1.25. URL https://aclanthology.org/2024.blackboxnlp-1.25/.

[13] Stefan Heimersheim and Neel Nanda. How to use and interpret activation patching, 2024. URL https://arxiv.org/abs/2404.15255.

[14] Fred Zhang and Neel Nanda. Towards best practices of activation patching in language models: Metrics and methods, 2024. URL https://arxiv.org/abs/2309.16042.

[15] Nicholas Sofroniew, Isaac Kauvar, William Saunders, Runjin Chen, Tom Henighan, Sasha Hydrie, Craig Citro, Adam Pearce, Julius Tarng, Wes Gurnee, Joshua Batson, Sam Zimmerman, Kelley Rivoire, Kyle Fish, Chris Olah, and Jack Lindsey. Emotion concepts and their function in a large language model, 2026. URL https://arxiv.org/abs/2604.07729.

[16] Maxime Méloux, Maxime Peyrard, and François Portet. Mechanistic interpretability as statistical estimation: A variance analysis of EAP-IG, 2026. URL https://openreview.net/forum?id=YD1P4DVtdk.

[17] Lee Sharkey, Bilal Chughtai, Joshua Batson, Jack Lindsey, Jeffrey Wu, Lucius Bushnaq, Nicholas Goldowsky-Dill, Stefan Heimersheim, Alejandro Ortega, Joseph Isaac Bloom, Stella Biderman, Adrià Garriga-Alonso, Arthur Conmy, Neel Nanda, Jessica Mary Rumbelow, Martin Wattenberg, Nandi Schoots, Joseph Miller, William Saunders, Eric J Michaud, Stephen Casper, Max Tegmark, David Bau, Eric Todd, Atticus Geiger, Mor Geva, Jesse Hoogland, Daniel Murfet, and Thomas McGrath. Open problems in mechanistic interpretability. *Transactions on Machine Learning Research*, 2025. ISSN 2835-8856. URL https://openreview.net/forum?id=91H76m9Z94. Survey Certification.

[18] Lin Zhang, Wenshuo Dong, Zhuoran Zhang, Shu Yang, Lijie Hu, Ninghao Liu, Pan Zhou, and Di Wang. EAP-GP: Mitigating saturation effect in gradient-based automated circuit identification. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=lGyXq0LOeQ.

[19] Claudia Shi, Nicolas Beltran-Velez, Achille Nazaret, Carolina Zheng, Adrià Garriga-Alonso, Andrew Jesson, Maggie Makar, and David Blei. Hypothesis testing the circuit hypothesis in LLMs. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=5ai2YFAXV7.

[20] Aaron Mueller, Atticus Geiger, Sarah Wiegreffe, Dana Arad, Iván Arcuschin, Adam Belfki, Yik Siu Chan, Jaden Fried Fiotto-Kaufman, Tal Haklay, Michael Hanna, Jing Huang, Rohan Gupta, Yaniv Nikankin, Hadas Orgad, Nikhil Prakash, Anja Reusch, Aruna Sankaranarayanan, Shun Shao, Alessandro Stolfo, Martin Tutek, Amir Zur, David Bau, and Yonatan Belinkov. MIB: A mechanistic interpretability benchmark. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu, editors, *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pages 45069–45108. PMLR, 13–19 Jul 2025. URL https://proceedings.mlr.press/v267/mueller25a.html.

[21] Rohan Gupta, Iván Arcuschin, Thomas Kwa, and Adrià Garriga-Alonso. InterpBench: Semi-synthetic transformers for evaluating mechanistic interpretability techniques. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2024. URL https://openreview.net/forum?id=R9gR9MPuD5.

[22] Itamar Hadad, Guy Katz, and Shahaf Bassan. Formal mechanistic interpretability: Automated circuit discovery with provable guarantees. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=Timsb74vIY.

[23] Alaa Anani, Tobias Lorenz, Bernt Schiele, Mario Fritz, and Jonas Fischer. Certified circuits: Stability guarantees for mechanistic circuits, 2026. URL https://arxiv.org/abs/2602.22968.

[24] Luyang Zhang, Minxing Zheng, and Jialu Wang. Correcting attribution patching with a few exact measurements. *arXiv preprint arXiv:2610.27393*, 2026.

[25] Atticus Geiger, Duligur Ibeling, Amir Zur, Maheep Chaudhary, Sonakshi Chauhan, Jing Huang, Aryaman Arora, Zhengxuan Wu, Noah Goodman, Christopher Potts, and Thomas Icard. Causal abstraction: A theoretical foundation for mechanistic interpretability. *Journal of Machine Learning Research*, 26(83):1–64, 2025. URL http://jmlr.org/papers/v26/23-0058.html.

[26] Atticus Geiger, Zhengxuan Wu, Christopher Potts, Thomas Icard, and Noah Goodman. Finding alignments between interpretable causal variables and distributed neural representations. In Francesco Locatello and Vanessa Didelez, editors, *Proceedings of the Third Conference on Causal Learning and Reasoning*, volume 236 of *Proceedings of Machine Learning Research*, pages 160–187. PMLR, 01–03 Apr 2024. URL https://proceedings.mlr.press/v236/geiger24a.html.

[27] Zhengxuan Wu, Atticus Geiger, Thomas Icard, Christopher Potts, and Noah Goodman. Interpretability at scale: Identifying causal mechanisms in alpaca. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=nRfClnMhVX.

[28] Taha Entesari, Sina Sharifi, and Mahyar Fazlyab. Compositional curvature bounds for deep neural networks. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org, 2024.

[29] Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1885–1894. PMLR, 06–11 Aug 2017. URL https://proceedings.mlr.press/v70/koh17a.html.

[30] Yurii Nesterov and Boris T. Polyak. Cubic regularization of Newton method and its global performance. *Mathematical Programming*, 108(1):177–205, 2006.

[31] Coralia Cartis, Nicholas I. Gould, and Philippe L. Toint. Adaptive cubic regularisation methods for unconstrained optimization. Part I: motivation, convergence and numerical results. *Math. Program.*, 127(2):245–295, April 2011. ISSN 0025-5610.

[32] Josip Jukić and Jan Šnajder. From robustness to improved generalization and calibration in pre-trained language models. *Transactions of the Association for Computational Linguistics*, 13:264–280, 2025. doi: 10.1162/tacl_a_00739. URL https://aclanthology.org/2025.tacl-1.13/.

[33] Hyunjik Kim, George Papamakarios, and Andriy Mnih. The lipschitz constant of self-attention. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 5562–5571. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/kim21i.html.

[34] Valérie Castin, Pierre Ablin, and Gabriel Peyré. How smooth is attention? In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp, editors, *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 5817–5840. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/castin24a.html.

[35] Barak A. Pearlmutter. Fast exact multiplication by the hessian. *Neural Computation*, 6(1):147–160, 1994. doi: 10.1162/neco.1994.6.1.147.

[36] Neel Nanda and Joseph Bloom. TransformerLens, 2022. URL https://github.com/TransformerLensOrg/TransformerLens.

[37] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI*, 2019. URL https://cdn.openai.com/better-language-models/language_models_are_unsupervised_multitask_learners.pdf. Accessed: 2024-11-15.

[38] Stella Biderman, Hailey Schoelkopf, Quentin Gregory Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, Usvsn Sai Prashanth, Edward Raff, Aviya Skowron, Lintang Sutawika, and Oskar Van Der Wal. Pythia: A suite for analyzing large language models across training and scaling. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 2397–2430. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/biderman23a.html.

[39] Qwen Team. Qwen2.5: A party of foundation models, September 2024. URL https://qwenlm.github.io/blog/qwen2.5/.

[40] Gemma Team. Gemma 2: Improving open language models at a practical size, 2024. URL https://arxiv.org/abs/2408.00118.

[41] Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models, 2024. URL https://arxiv.org/abs/2407.21783.

[42] Michael Hanna, Ollie Liu, and Alexandre Variengien. How does GPT-2 compute greater-than?: Interpreting mathematical abilities in a pre-trained language model. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=p4PckNQR8k.

[43] Joseph Bloom, Curt Tigges, Anthony Duong, and David Chanin. SAELens. https://github.com/decoderesearch/SAELens, 2024.

A Notation

Table 2 summarizes the core notation used in the main text.

Table 2: Core notation used in the main text.

Symbol	Meaning
$M(a)$	Scalar output metric as a function of an internal activation
a	Clean activation at the component being patched
$a' = a + \delta$	Counterfactual activation after patching
δ	Patching perturbation
Δ	True activation-patching effect, $M(a+\delta) - M(a)$
$\hat{\Delta}$	First-order attribution-patching estimate, $\nabla_a M \cdot \delta$
E	Attribution-patching approximation error, $\Delta - \hat{\Delta}$
H	Network-level Hessian, $\nabla_a^2 M(a)$
$\Phi(\delta)$	Third-order remainder term in the Taylor expansion
L_3	Hessian Lipschitz constant on the patching segment
$\tilde{R}$	Reliability score, $\lvert\delta^\top H\delta\rvert/(2\lvert\hat{\Delta}\rvert)$
α	Validity parameter controlling remainder size relative to the quadratic term
$\hat{\Delta}_{\text{hvp}}$	HVP-corrected estimate, $\hat{\Delta} + \frac{1}{2}\delta^\top H\delta$
$\hat{\Delta}_i$	Attribution-patching estimate for component i in a collection of candidate components
c_i	Diagonal quadratic correction for component i, $\frac{1}{2}\delta_i^\top H_{ii}\delta_i$
$S_{\text{ok}}, S_{\text{flag}}$	Components with $\tilde{R}_i < \tau$ and $\tilde{R}_i \geq \tau$ under a threshold policy
Q_{ok}	Signed sum of uncorrected quadratic terms, $\sum_{i\in S_{\text{ok}}} c_i$
γ	Cancellation ratio on uncorrected quadratic terms, $\lvert Q_{\text{ok}}\rvert / \sum_{i\in S_{\text{ok}}} \lvert c_i\rvert$

B Local smoothness examples of the Lipschitz-Hessian assumption

B.1 Theoretical approximation

We include representative constants only to clarify the scale of the cubic remainder. These values are not required by the main theorem, which only assumes a finite L_3 on the segment of interest.

GeLU. $\text{GeLU}(z) = z\,\Phi(z)$ where Φ is the standard normal CDF. Differentiating gives $\text{GeLU}'''(z) = z(z^2-4)\phi(z)$, where ϕ is the standard normal PDF. The stationary points of $|\text{GeLU}'''(z)|$ satisfy $\text{GeLU}^{(4)}(z) = 0$, equivalently $z^4 - 7z^2 + 4 = 0$. The maximizer is therefore $|z| = \sqrt{\frac{7-\sqrt{33}}{2}} \approx 0.792$, which gives $\max_z |\text{GeLU}'''(z)| \approx 0.779$.

SiLU. $\text{SiLU}(z) = z \cdot \sigma(z)$ where σ is the logistic sigmoid. The third derivative satisfies $\max_z |\text{SiLU}'''(z)| \approx 0.30818$, attained near $|z| \approx 1.032$ by numerical maximization. Thus $L_3^{\text{SiLU}} \approx 0.308$. This value is only illustrative: the main theorem itself does not rely on a closed-form constant.

Bilinear maps. If the perturbed variable enters bilinearly while the other argument is held fixed, then the third derivative with respect to that variable is zero. This applies, for example, to perturbations of the value vector V with fixed attention weights, but it is *not* a property of the full attention block when Q or K also vary.

Softmax and normalization layers. Softmax is real analytic, so its third derivative is finite on every compact subset of logit space. LayerNorm and RMSNorm are smooth wherever the normalization scale is bounded away from zero. Because Proposition 1 requires only a finite constant on the single interpolation segment $\{a + t\delta\}_{t\in[0,1]}$, we treat these contributions through the segment-local L_3 rather than assert a universal transformer-wide bound.

B.2 Empirical verification

Assumption 1 posits that the metric function's Hessian is L_3-Lipschitz along the interpolation path. We verify this empirically by computing Hessian–vector products $H(a_c + t\delta) \cdot \delta$ at three points ($t = 0, 0.5, 1$) for each prompt $\times$ attention head, then estimating

$$\hat{L}_3 = \max_i \frac{\|H(a_c + t_{i+1}\delta) \cdot \delta - H(a_c + t_i\delta) \cdot \delta\|}{(t_{i+1} - t_i)\,\|\delta\|^2}.$$

This is a lower bound on the true L_3 (probed along one direction only). With $\hat{L}_3$ in hand, the cubic remainder bound becomes $\hat{\alpha} = \hat{L}_3\|\delta\|^3/6$, which we compare to the actual residual $|f_{\text{true}} - f_{\text{HVP}}|$.

Table 3 reports the empirical $\hat{L}_3$ distribution and bound-tightness statistics for GPT-2 (20 prompts $\times$ 144 heads = 2,880 records, 2,877 nontrivial).

Table 3: Empirical L_3 estimation on GPT-2 (12L$\times$12H, 20 prompts). $\hat{\alpha} = \hat{L}_3\|\delta\|^3/6$ is the cubic remainder bound; |res| is the actual HVP residual $|f_{\text{true}} - f_{\text{HVP}}|$. The bound holds when $\hat{\alpha} \geq$ |res|.

	Median	**Mean**	**p90**	**Max**
$\hat{L}_3$	0.0014	0.0169	0.0182	4.917
$\hat{\alpha}$ (cubic bound)	0.000108	0.00348	—	—
|res| (actual residual)	0.000018	0.000920	—	—
Bound holds: 2,370 / 2,877 = **82.4%**				
$\hat{\alpha}/$|res|: median $3.4\times$, mean $16.0\times$, p10 $0.20\times$				

The bound holds for 82.4% of component–prompt pairs, with a median slack factor of $3.4\times$ (the bound is $3.4\times$ larger than the actual residual). The 17.6% of violations are concentrated in Layer 0, where $\hat{L}_3$ is an order of magnitude larger than mid-network layers (median 0.065 vs. 0.001–0.002). This is consistent with the embedding layer's position-dependent structure producing sharper Hessian variation.

Across layers, median $\hat{L}_3$ decreases monotonically from 0.065 (L0) to 0.0001 (L11), confirming the structural prediction that later layers, with shorter downstream paths, exhibit smoother Hessian landscapes. This layer-depth gradient also explains why HVP correction is most impactful for early- and mid-layer heads (where $\hat{L}_3\|\delta\|^3$ is large enough to matter) and nearly unnecessary for the last few layers.

Table 4 reports the same analysis for Gemma-2-2B (26 layers $\times$ 8 heads = 208 heads, 20 prompts, 4,152 nontrivial records).

Table 4: Empirical L_3 estimation on Gemma-2-2B (26L$\times$8H, 20 prompts). Same protocol as Table 3.

	Median	**Mean**	**p90**	**Max**
$\hat{L}_3$	0.0002	0.0014	0.0020	0.699
$\hat{\alpha}$ (cubic bound)	0.00403	0.1193	—	—
|res| (actual residual)	0.00035	0.01104	—	—
Bound holds: 3,945 / 4,152 = **95.0%**				
$\hat{\alpha}/$|res|: median $7.6\times$, mean $104\times$, p10 $2.7\times$				

On Gemma-2-2B, the bound holds for **95.0%** of pairs, stronger than GPT-2's 82.4%, despite much larger perturbation norms (median $\|\delta\| = 4.81$ vs. 0.73). The median slack factor is $7.6\times$, meaning the cubic bound is typically an order of magnitude larger than the actual HVP residual. Unlike GPT-2, Gemma-2-2B does not show a strong monotonic layer-depth gradient in $\hat{L}_3$: the per-layer medians hover around 0.0001–0.0009 throughout the network, with occasional spikes at L7 (max 0.42) and L3 (max 0.70). This flatter profile is consistent with Gemma-2's use of grouped-query attention and GeGLU, which distribute nonlinearity more uniformly across layers than GPT-2's standard architecture.

C Additional theoretical results and analysis

C.1 Multi-layer composition challenges

Why we do not state a global composition theorem. Composing per-layer curvature through a residual network introduces mixed second derivatives and Jacobian growth factors. Prior work on compositional curvature bounds shows that such quantities can grow multiplicatively with depth [28]. For this reason, our formal results are stated at the level actually used by attribution patching and HVP correction: one component, one input, and one perturbation segment.

Scope and computation. The bound in Proposition 1 is per-component and per-input. Residual composition introduces mixed second derivatives and depth-dependent Jacobian factors [28], so the guarantee does not extend to a depth-uniform statement over the full transformer. Computing $\tilde{R}$ requires only the Hessian-vector product $H\delta$, obtained by differentiating a scalar gradient–vector product rather than forming the full Hessian. Empirically, $\tilde{R}$ serves as a practical diagnostic for flagging unreliable attributions (§4.4).

Layerwise heuristic. If one additionally freezes downstream Hessians at the clean activations and ignores mixed second derivatives, then the total attribution-patching error can be approximated by summing the per-layer quadratic terms plus a cubic remainder. We use this only as intuition for the empirical depth trends; it is not a proved bound in this paper.

C.2 Error analysis of HVP correction

Corollary 2 (HVP correction error)**.** *After applying the HVP correction, the residual error is*

$$E_{hvp} = \Delta - \hat{\Delta}_{hvp} = \Phi(\delta), \qquad |\Phi(\delta)| \le \frac{L_3}{6}\|\delta\|^3.$$

Proof. By definition,

$$\hat{\Delta}_{\text{hvp}} = \hat{\Delta} + \tfrac{1}{2}\delta^\top H\delta.$$

Combining this with the Taylor decomposition in Eq. (2),

$$\Delta = \hat{\Delta} + \tfrac{1}{2}\delta^\top H\delta + \Phi(\delta),$$

we obtain

$$\Delta - \hat{\Delta}_{\text{hvp}} = \Phi(\delta).$$

The cubic remainder bound then follows immediately from the same Hessian-Lipschitz assumption used in Proposition 1. □

C.3 Selective-HVP workflow identities

Corollary 3 (Exact gap to full diagonal HVP)**.** *Let*

$$\hat{\Delta}_{\text{full}} := \sum_{i=1}^{n} \hat{\Delta}_i + \sum_{i=1}^{n} c_i$$

denote the estimate obtained by applying the diagonal HVP correction to every independently patched component, and let $E^{\text{ind}}_{\text{full}} := \sum_i \Delta_i - \hat{\Delta}_{\text{full}}$*. Then*

$$E^{\text{ind}}_{\text{sel}} - E^{\text{ind}}_{\text{full}} = Q_{\text{ok}} := \sum_{i \in S_{\text{ok}}} c_i. \tag{8}$$

Consequently,

$$\left|E^{\text{ind}}_{\text{sel}} - E^{\text{ind}}_{\text{full}}\right| \le \tau \sum_{i \in S_{\text{ok}}} |\hat{\Delta}_i|. \tag{9}$$

Proof. By definition,

$$E_{\text{full}}^{\text{ind}} = \sum_{i=1}^{n} \Delta_i - \sum_{i=1}^{n} (\hat{\Delta}_i + c_i).$$

Subtracting this from $E_{\text{sel}}^{\text{ind}} = \sum_i \Delta_i - \hat{\Delta}_{\text{sel}}$ and using $\hat{\Delta}_{\text{sel}} = \sum_i \hat{\Delta}_i + \sum_{i \in S_{\text{flag}}} c_i$ gives

$$E_{\text{sel}}^{\text{ind}} - E_{\text{full}}^{\text{ind}} = \sum_{i=1}^{n} c_i - \sum_{i \in S_{\text{flag}}} c_i = \sum_{i \in S_{\text{ok}}} c_i = Q_{\text{ok}}.$$

Taking absolute values and using $|c_i| = \tilde{R}_i |\hat{\Delta}_i| < \tau |\hat{\Delta}_i|$ for $i \in S_{\text{ok}}$ yields Eq. (9). □

Proposition 3 (Sign-aware refinement for independent patching)**.** *Define* $S_{\text{ok}}^{+} := \{i \in S_{\text{ok}} : c_i \neq 0\}$ *and* $S_0 := S_{\text{ok}} \setminus S_{\text{ok}}^{+}$*. For every* $i \in S_{\text{ok}}^{+}$ *let* $\alpha_i := |\Phi_i|/|c_i|$*, and define*

$$Q_{\text{ok}}^{+} := \sum_{i \in S_{\text{ok}}^{+}} c_i, \qquad \gamma := \frac{|Q_{\text{ok}}^{+}|}{\sum_{i \in S_{\text{ok}}^{+}} |c_i|}, \qquad \bar{\alpha} := \max_{i \in S_{\text{ok}}^{+}} \alpha_i,$$

with the convention that the first term below is zero when $S_{\text{ok}}^{+} = \varnothing$*. Then*

$$\left|E_{\text{sel}}^{\text{ind}}\right| \leq (\gamma + \bar{\alpha})\, \tau \sum_{i \in S_{\text{ok}}^{+}} |\hat{\Delta}_i| + \frac{1}{6} \sum_{i \in S_{\text{flag}} \cup S_0} L_{3,i} \|\delta_i\|^3. \tag{10}$$

Proof. Decompose the selective residual as

$$E_{\text{sel}}^{\text{ind}} = \sum_{i \in S_{\text{ok}}^{+}} (c_i + \Phi_i(\delta_i)) + \sum_{i \in S_0} \Phi_i(\delta_i) + \sum_{i \in S_{\text{flag}}} \Phi_i(\delta_i).$$

For $i \in S_{\text{ok}}^{+}$, the remainder bound and the definition of α_i imply $|\Phi_i(\delta_i)| \leq \alpha_i |c_i| \leq \bar{\alpha} |c_i|$. Therefore

$$\begin{aligned} \left|E_{\text{sel}}^{\text{ind}}\right| &\leq \left| \sum_{i \in S_{\text{ok}}^{+}} c_i \right| + \sum_{i \in S_{\text{ok}}^{+}} |\Phi_i(\delta_i)| + \sum_{i \in S_{\text{flag}} \cup S_0} |\Phi_i(\delta_i)| \\ &\leq |Q_{\text{ok}}^{+}| + \bar{\alpha} \sum_{i \in S_{\text{ok}}^{+}} |c_i| + \frac{1}{6} \sum_{i \in S_{\text{flag}} \cup S_0} L_{3,i} \|\delta_i\|^3. \end{aligned}$$

Using $|Q_{\text{ok}}^{+}| = \gamma \sum_{i \in S_{\text{ok}}^{+}} |c_i|$ and $|c_i| = \tilde{R}_i |\hat{\Delta}_i| < \tau |\hat{\Delta}_i|$ for $i \in S_{\text{ok}}^{+}$ yields Eq. (10). □

D Omitted proofs

D.1 Proof of Proposition 1 (Local attribution-patching error bound via the reliability score)

Proof. From the Taylor expansion with integral remainder,

$$M(a + \delta) = M(a) + \nabla_a M \cdot \delta + \tfrac{1}{2} \delta^{\top} H \delta + \Phi(\delta),$$

where the remainder takes the standard integral form:

$$\Phi(\delta) = \int_0^1 (1 - t)\, \delta^{\top} \big[\nabla_a^2 M(a + t\delta) - H\big] \delta \, dt.$$

Under Assumption 1,

$$\|\nabla_a^2 M(a + t\delta) - H\|_{\text{op}} \leq L_3 \cdot t \|\delta\|,$$

and therefore

$$\begin{aligned} |\Phi(\delta)| &\leq \int_0^1 (1 - t) \cdot L_3\, t \|\delta\|^3 \, dt = L_3 \|\delta\|^3 \int_0^1 t(1 - t)\, dt \\ &= L_3 \|\delta\|^3 \cdot \frac{1}{6} = \frac{L_3 \|\delta\|^3}{6}\,. \end{aligned} \tag{11}$$

Here $\int_0^1 t(1-t)\,dt = \frac{1}{2} - \frac{1}{3} = \frac{1}{6}$.

Defining $Q = \frac{1}{2}\delta^\top H\delta$, we have $E = Q + \Phi$ and the reverse triangle inequality gives

$$\big||E| - |Q|\big| \le |\Phi|.$$

With

$$\alpha = \frac{L_3\|\delta\|^3}{3|\delta^\top H\delta|} = \frac{|\Phi|_{\max}}{|Q|},$$

we obtain

$$\big||E| - |Q|\big| \le \alpha|Q|.$$

Dividing by $|\hat{\Delta}|$ and noting $|Q|/|\hat{\Delta}| = \tilde{R}$ gives

$$\left|\frac{|E|}{|\hat{\Delta}|} - \tilde{R}\right| \le \alpha\tilde{R},$$

which is the first claimed inequality. If additionally $\alpha < 1$, then

$$(1-\alpha)|Q| \le |E| \le (1+\alpha)|Q|,$$

and dividing again by $|\hat{\Delta}|$ yields the two-sided inequality in Proposition 1. □

D.2 Proof of Corollary 1 (Residual error after HVP correction)

Proof. By definition,

$$\hat{\Delta}_{\text{hvp}} = \hat{\Delta} + \tfrac{1}{2}\delta^\top H\delta.$$

Combining this with the Taylor decomposition in Eq. (2),

$$\Delta = \hat{\Delta} + \tfrac{1}{2}\delta^\top H\delta + \Phi(\delta),$$

we obtain

$$\Delta - \hat{\Delta}_{\text{hvp}} = \Phi(\delta).$$

The cubic remainder bound then follows immediately from the same Hessian-Lipschitz assumption used in Proposition 1. □

D.3 Proof of Proposition 2 (Selective-HVP pipeline guarantee)

Proof. For each independently patched component i, the single-component Taylor decomposition gives

$$\Delta_i - \hat{\Delta}_i = c_i + \Phi_i(\delta_i), \qquad |\Phi_i(\delta_i)| \le \frac{L_{3,i}}{6}\|\delta_i\|^3.$$

Subtracting the selective estimate

$$\hat{\Delta}_{\text{sel}} = \sum_{i=1}^{n}\hat{\Delta}_i + \sum_{i\in S_{\text{flag}}} c_i$$

from the sum of true independent effects gives

$$E_{\text{sel}}^{\text{ind}} = \sum_{i=1}^{n}(\Delta_i - \hat{\Delta}_i) - \sum_{i\in S_{\text{flag}}} c_i = \sum_{i\in S_{\text{ok}}}(c_i + \Phi_i(\delta_i)) + \sum_{i\in S_{\text{flag}}}\Phi_i(\delta_i).$$

By the triangle inequality,

$$|E_{\text{sel}}^{\text{ind}}| \le \sum_{i\in S_{\text{ok}}}|c_i| + \sum_{i=1}^{n}|\Phi_i(\delta_i)|.$$

For every $i \in S_{\text{ok}}$, the threshold definition gives $\tilde{R}_i = |c_i|/|\hat{\Delta}_i| < \tau$, hence

$$|c_i| < \tau|\hat{\Delta}_i|.$$

For the remainders,

$$|\Phi_i(\delta_i)| \le \frac{L_{3,i}}{6}\|\delta_i\|^3 \qquad \text{for all } i.$$

Summing these bounds yields

$$|E_{\text{sel}}^{\text{ind}}| \le \tau\sum_{i\in S_{\text{ok}}}|\hat{\Delta}_i| + \frac{1}{6}\sum_{i=1}^{n}L_{3,i}\|\delta_i\|^3,$$

which is exactly Eq. (7). □

D.4 Proof of Corollary 3

Proof. By definition,

$$E_{\text{full}}^{\text{ind}} = \sum_{i=1}^{n} \Delta_i - \sum_{i=1}^{n} (\hat{\Delta}_i + c_i).$$

Subtracting this from $E_{\text{sel}}^{\text{ind}} = \sum_i \Delta_i - \hat{\Delta}_{\text{sel}}$ and using $\hat{\Delta}_{\text{sel}} = \sum_i \hat{\Delta}_i + \sum_{i \in S_{\text{flag}}} c_i$ gives

$$E_{\text{sel}}^{\text{ind}} - E_{\text{full}}^{\text{ind}} = \sum_{i=1}^{n} c_i - \sum_{i \in S_{\text{flag}}} c_i = \sum_{i \in S_{\text{ok}}} c_i = Q_{\text{ok}}.$$

Taking absolute values and using $|c_i| = \tilde{R}_i |\hat{\Delta}_i| < \tau |\hat{\Delta}_i|$ for $i \in S_{\text{ok}}$ yields Eq. (9). □

E Implementation details

E.1 Setup

Models We evaluate five model families at different scales as shown in Table 5. GPT-2 [37], Pythia (410M, 2.8B) [38], Qwen2.5-1.5B [39], Gemma (2B, 9B) [40], and Llama-3.1-8B [41]. All models are hooked with TransformerLens [36]. We analyze attention heads in all models, pre-activation MLP neurons in GeLU models, and MLP layer outputs in Qwen. For factual completion and Greater-Than

Table 5: Comparison of model architectures used in our experiments.

Model	Layers	Heads	d_{model}	d_{mlp}	Activation
GPT-2 Small [37]	12	12	768	3072	GeLU
Pythia-410M-deduped [38]	24	16	1024	4096	GeLU
Pythia-2.8B-deduped [38]	32	32	2560	10240	GeLU
Qwen2.5-1.5B [39]	28	12	1536	8960	SwiGLU
Gemma-2-2B [40]	26	8	2304	9216	GeGLU
Gemma-2-9B [40]	42	16	3584	14336	GeGLU
Llama-3.1-8B [41]	32	32	4096	14336	SwiGLU

tasks, corruptions are generated by replacing the token at position 3 with a uniformly sampled vocabulary token. For IOI, we adopt the standard name-swap corruption introduced by [5]. In all settings, δ denotes the induced activation perturbation and M denotes the log-probability of the correct next token.

Our primary metric is *top-5 relative error*: the mean absolute error of a method's scores on the five components with largest activation-patching effect, normalized by the ground-truth score range.

Tasks We evaluate on the three established circuit benchmarks together with a broader factual-completion setting:

- **IOI:** For the IOI circuit ranking experiment, we follow the setup of Wang et al. [5]: 50 IOI prompts of the form "When Mary and John went to the store, John gave a drink to", with the indirect object (Mary) as the target. We compute attribution-patching and HVP-corrected attributions for all $12 \times 12 = 144$ attention heads, rank them by magnitude, and compare the resulting rankings against the activation-patching ground truth using top-k overlap and Spearman correlation.
- **Greater-Than:** For the Greater-Than circuit [42], we use 200 prompts of the form "The war lasted from the year 17{XX} to the year 17" and measure whether the model assigns higher probability to years greater than XX. All 144 attention heads in GPT-2 Small are ranked by attribution magnitude. Top-k overlap is computed against the activation patching ground truth.
- **Factual completion.** For generic model sweeps, we use factual next-token prediction prompts and measure the effect of component interventions on the correct-token log-probability.

Unless otherwise noted, generic sweeps use 20 factual prompts. We additionally scale selected experiments to 55 prompts for the Pythia-410M attention-head setting and 35 prompts for the Qwen2.5-1.5B attention-head and MLP-output settings. IOI uses 50 templated prompts and Greater-Than uses 200 prompts.

Our model-task selection is guided by the theoretical prediction of §3, which predicts that attribution-patching error increases when the quadratic form $\delta^\top H\delta$ dominates the first-order approximation. Pythia-410M on IOI represents a predicted high-error regime in which the local Taylor approximation breaks down, whereas GPT-2 Small on IOI represents a low-error regime where standard attribution patching is already accurate. Gemma-2-2B factual completion lies between these extremes. Additional model-task pairs provide broader cross-architecture coverage (Table 1).

Evaluation protocol. Unless otherwise stated, we report 95% confidence intervals computed from 1,000 bootstrap resamples over prompts. Each resample draws prompts with replacement and aggregates over all components within the sampled prompt set. We report the 2.5th and 97.5th percentiles of the bootstrap distribution. Circuit-ranking metrics on IOI and Greater-Than are reported as point estimates over the benchmark datasets. We measure the following component types:

- **Pre-activation MLP neurons:** `hook_pre` activations in GeLU-based models, with 256 sampled neurons per layer.
- **Qwen MLP outputs:** `hook_mlp_out` activations, corresponding to full d_{model}-dimensional layer outputs.
- **Post-activation MLP neurons:** `hook_post` activations.
- **Attention heads:** `hook_z` activations, i.e., head outputs before the output projection.
- **Residual stream:** `hook_resid_post` activations over the full residual vector.

Baselines. We compare against AtP* [7] and GIM [8]. Since AtP* does not provide an official implementation, we reimplemented the method from the paper description. AtP* combines Q/K-linearization, which linearizes the softmax backward pass, with GradDrop, which suppresses gradient contributions from components whose activation region changes under perturbation. Because these corrections are specific to attention mechanisms, comparisons with AtP* are restricted to attention-head experiments.

For GIM, we follow the public implementation, modifying the softmax backward pass to remove the self-repair gradient term responsible for systematic underestimation. Both baselines are evaluated on the same prompt sets and component collections as the corresponding HVP experiments in Table 6.

Hessian–vector products are computed in PyTorch via `torch.autograd.grad` with `create_graph=True` on the first backward pass, so the gradient can itself be differentiated on the second backward pass. We never explicitly form the Hessian; only the product $H\delta$ is computed, which is what keeps the correction tractable at 8B scale. Gradients are kept in float32 throughout (mixed precision elsewhere).

Table 6: Comparison of attribution-patching correction methods on Pythia-410M using the matched 20-prompt comparison protocol (bootstrap 95% CIs for error columns). HVP provides the largest reduction at 3× cost.

Method	Attn. heads (%)		Pre-act neurons (%)		Cost
	Error	Reduct.	Error	Reduct.	
Attribution patching	4.4 [3.6, 5.2]	—	35.2 [31.1, 39.3]	—	1×
AtP*	2.9 [2.2, 3.6]	34.1	33.8 [29.8, 37.8]	4.0	1×
GIM	2.6 [1.9, 3.3]	40.9	31.4 [27.5, 35.3]	10.8	1×
HVP (ours)	1.0 [0.7, 1.3]	**77.4**	11.7 [9.4, 14.0]	**66.7**	3×
Act. patching	0	100	0	100	$N\times$

E.2 Metric

For the generic model sweeps, the metric M is the log-probability of the correct next token at the evaluation position.

Identity. The near-identical values for log probability, probability, and cross-entropy in Table 7 are an empirical observation rather than a formal invariance theorem. Over the perturbation magnitudes used in our experiments, these metrics are smooth monotone re-parameterizations of the same target logit score and are locally well approximated by affine maps, so their relative attribution-patching errors agree to within numerical noise. We verified this directly: the maximum observed deviation between log-probability and probability relative errors is 0.03%.

Sensitivity. Table 7 shows that HVP correction is robust across five different metric functions, with reduction ranging from 62% to 72%. The raw-logit metric yields the largest reduction (72%), as expected: the final softmax introduces additional curvature that the second-order correction only partially captures. This confirms that the HVP correction is not specific to the choice of output metric.

Table 7: HVP correction across metric functions (pre-activation neurons, Pythia-410M, layer 23, 128 neurons, bootstrap 95% CIs). Log probability, probability, and cross-entropy yield near-identical errors in this perturbation regime (Appendix E.2); the maximum deviation between probability and log-probability relative errors is 0.03%.

Metric	Attrib. Pat. Err. (%)	HVP Err. (%)	Reduction (%)
Raw logit	33.3 [27.9, 38.7]	9.4 [6.8, 12.0]	**71.7** [64.2, 79.2]
Logit difference	38.9 [33.1, 44.7]	11.8 [8.9, 14.7]	**69.6** [62.1, 77.1]
Log probability	40.9 [35.0, 46.8]	15.4 [12.0, 18.8]	**62.3** [54.8, 69.8]
Probability	40.9 [35.0, 46.8]	15.4 [12.0, 18.8]	**62.4** [54.9, 69.9]
Cross-entropy	40.9 [35.0, 46.8]	15.4 [12.0, 18.8]	**62.3** [54.8, 69.8]

E.3 Per-head IG vs. all-at-once IG (EAP-IG)

The "IG" baseline in Table 1 is *per-head* Integrated Gradients: for each attention head h, we interpolate only h's activation from clean to corrupted while holding all other activations at clean, and integrate the gradient along this isolated path using S quadrature points. By the fundamental theorem of calculus, this recovers the head h's exact activation-patching effect as the same quantity the ground truth measures, which is why we treat it as the accuracy ceiling. The per-prompt cost is S forward + backward passes per head, totalling $S \times N_{\text{heads}}$ per prompt. For Llama-3.1-8B ($32 \times 32 = 1{,}024$ heads) at $S{=}10$, this is $10{,}240$ backward passes per prompt, yielding the $\sim$25 GPU-day estimate in §4.2.

All-at-once IG (termed EAP-IG in Syed et al. [12] and used by the MIB benchmark [20]) interpolates *all* source activations simultaneously: at step t, every source is set to $(1{-}t)\, a_{\text{clean}} + t\, a_{\text{corrupt}}$, and one forward + backward pass at each of S steps yields edge-level attributions for the entire graph. The total cost is S forward + backward passes per prompt regardless of the number of heads, making it feasible at any scale. However, EAP-IG computes a fundamentally different quantity: because all sources move together, the resulting per-edge scores conflate cross-component interaction effects. EAP-IG is a first-order method (gradient only, no Hessian correction) and answers the question "how does each edge contribute along the joint interpolation path?" rather than "what is each head's isolated causal effect?"

HVP's cost structure resembles the all-at-once approach rather than the per-head approach. The key computation $\delta^\top H \delta$ via `torch.autograd.grad` with `create_graph=True` hooks all components simultaneously and produces a global second-order correction in two backward passes per prompt. The per-component corrections are then extracted as coordinate projections of the full Hessian–vector product. This batching is why HVP at cost 2 scales to 8B+ parameters, whereas per-head IG at cost $S{\times}N_{\text{heads}}$ does not.

E.4 Compute and wall-clock timing

All experiments ran on NVIDIA L40S GPUs. Pythia-410M: $\sim$6 hours including the 55-prompt attention expansion. Pythia-2.8B: $\sim$8 hours. GPT-2 Small: $\sim$1 hour for the generic sweeps plus $\sim$30 minutes for IOI and $\sim$4 hours for the 200-prompt Greater-Than benchmark. Qwen2.5-1.5B: $\sim$4 hours including the 35-prompt attention and MLP expansions. AtP*/GIM comparison: $\sim$3 hours.

Total compute is approximately 25 hours of single-GPU time. No experiment required more than a single GPU.

Measured on a single NVIDIA L40S GPU. For Pythia-410M: standard attribution patching takes 0.8s per prompt (all components), HVP correction takes 2.2s ($2.8\times$ overhead, not exactly $3\times$, due to computation graph caching). Pythia-2.8B: attrib. pat. 3.1s, HVP 8.9s ($2.9\times$). GPT-2 Small: attrib. pat. 0.4s, HVP 1.1s ($2.8\times$).

Table 8 reports wall-clock attribution time for the edge-level circuit-discovery pipeline on two representative settings, measured on a single NVIDIA L40S GPU with 100 evaluation examples. The "selective" row estimates the Screen–Flag–Fix workflow: run EAP to obtain edge scores, flag components with $\tilde{R} > \tau$, then apply HVP corrections only to flagged edges.

Table 8: Wall-clock attribution time (seconds) on L40S. MIB evaluation time (model reloading + 10 sparsity sweeps) adds $\sim$12–24 min and is method-independent. The selective pipeline (Screen–Flag–Fix) applies HVP only to flagged edges (τ=0.3), combining EAP cost with a small HVP overhead.

Model	Method	Edges	Attrib. (s)	Overhead vs. EAP
GPT-2 (12L, 32,491 edges)				
	EAP	32,491	1.6	1.0×
	HVP (K=1)	32,491	5.8	3.6×
	MS-HVP (K=5)	32,491	4.5	2.8×
	EAP-IG (inputs)	32,491	4.9	3.1×
	Selective HVP (τ=0.3, 7% flagged)	32,491	$\sim$1.9	$\sim$ 1.2×
Qwen2.5-0.5B (24L, 179,749 edges)				
	EAP	179,749	5.9	1.0×
	HVP (K=1)	179,749	6.1	1.0×
	MS-HVP (K=5)	179,749	8.8	1.5×
	Selective HVP (τ=0.3, 15% flagged)	179,749	$\sim$6.0	$\sim$ 1.0×

The key takeaway: on both models, the selective pipeline adds $\leq 20\%$ wall-clock overhead over raw EAP while still capturing the most important corrections. Full HVP adds 1.0–$3.6\times$ overhead – modest in absolute terms (seconds, not minutes) because the HVP backward pass reuses the same computation graph as the EAP forward pass. The dominant cost in a full MIB evaluation is the faithfulness sweep (reloading the model at each sparsity level), which is method-independent and takes 12–24 min.

F Full results of matched-compute comparisons

F.1 Comparison to integrated gradients.

Table 9: Matched-cost paired bootstrap: MS-HVP K=5 vs. IG S=10 (cost 10 each). Δ is MS-HVP minus IG (negative favours MS-HVP). p-values are one-sided paired bootstrap (10,000 resamples).

Model / Task	MS-HVP (%)	IG (%)	Δ (pp)	p	Verdict
Pythia-410M IOI	18.03	17.26	+0.77	0.620	tie
GPT-2 IOI	**2.97**	3.30	−0.34	<0.001	**win**
Gemma-2-2B factual	40.66	40.57	+0.10	0.696	tie
Pythia-410M GT	21.47	24.62	−3.15	0.347	tie
GPT-2 GT	23.33	32.98	−9.65	0.183	tie
Gemma-2-2B IOI	6.58	6.63	−0.05	0.320	tie
Qwen2.5-1.5B IOI	**2.94**	3.16	−0.22	0.041	**win**
Pythia-410M factual	64.41	63.91	+0.50	0.732	tie
Qwen2.5-1.5B factual	32.06	31.95	+0.10	0.668	tie

Integrated gradients (IG) provides the strongest existing accuracy baseline but scales linearly with the number of interpolation steps. Table 9 reports paired-bootstrap p-values comparing MS-HVP K=5 (cost 10) against IG S=10 (cost 10) on all nine non-frontier tasks. Figure 7 compares the

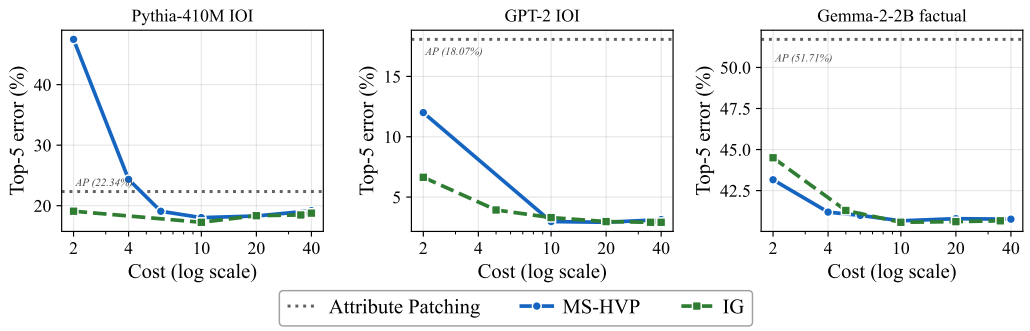


Figure 7: **Cost–accuracy tradeoff: MS-HVP vs. integrated gradients.** Top-5 relative error as a function of compute cost (backward passes per component) on three representative regimes: pathological high-curvature (Pythia-410M IOI), low-error clean (GPT-2 IOI), and moderate-error factual (Gemma-2-2B factual). Horizontal dotted lines denote attribution patching without correction. MS-HVP matches or exceeds IG at comparable compute across all regimes, while achieving substantially better compute efficiency in the pathological setting.

compute–accuracy tradeoff between IG and MS-HVP across representative low-, moderate-, and high-curvature regimes. At matched cost 10, MS-HVP ($K{=}5$) and IG ($S{=}10$) are statistically tied on seven of nine tasks under paired-bootstrap testing, while MS-HVP significantly outperforms IG on GPT-2 IOI ($p < 0.001$) and Qwen2.5-1.5B IOI ($p = 0.041$). Thus, iterative quadratic correction achieves parity with path integration on most tasks despite using only local second-order information.

The difference becomes more pronounced in the pathological high-curvature regime. On Pythia-410M IOI, MS-HVP with $K{=}5$ (cost 10) outperforms IG with $S{=}35$ (cost 35), achieving 18.03% versus 18.47% top-5 error ($\Delta = -0.44$ pp, $p = 0.022$). At approximately matched cost 40, aggregating MS-HVP estimates across $K \in \{5, 10, 20\}$ via a per-head median further improves performance to 17.57%, surpassing IG's best result of 18.77% ($p < 0.001$). These results indicate that multi-step quadratic correction can recover the benefits of dense path integration while requiring substantially less computation.

Importantly, the practical scaling behavior differs sharply between the methods. IG requires S full backward passes per component and rapidly becomes infeasible at 8B scale, whereas MS-HVP remains tractable because each step reuses local curvature information. Consequently, MS-HVP is not only competitive at matched cost but also extends naturally to model sizes where IG cannot practically run.

F.2 Comparison to Integrated Hessians

Table 10 compares Integrated Hessians (IH) against MS-HVP across all three foreground tasks. Both IH-PI (path-integrated) and IH-DR (double Riemann) underperform MS-HVP by 2.0–7.2 percentage points. The gap arises because IH's $\alpha\beta$ weighting yields a $1/4$ coefficient on $\delta^\top H\delta$ instead of the $1/2$ needed for Taylor correction.

F.3 Comparison to GIM

Table 11 shows that GIM underperforms attribution patching on 17 of 18 task$\times K$ head-recovery settings, with losses of up to 21.8 pp (Llama factual @5). The sole positive entry is Gemma-2-2B IOI @5 ($+0.4$ pp), which is within noise.

F.4 Recommendation map

Table 12 distills the findings of matched-compute experiments into a deployment guide organized by scale and component type.

Table 10: Integrated Hessians vs. MS-HVP on the three foreground tasks. IH-PI uses S path interpolation steps; IH-DR uses an $S \times M$ double Riemann grid. All p-values are paired bootstrap vs. attribution patching.

Method	Cost	Top-5 (%)	Δ vs. attrib. pat. (pp)
Pythia-410M IOI			
MS-HVP K=5	10	**18.03**	−4.31
IH-PI S=5	10	22.22	−0.12
IH-DR 3×3	18	20.51	−1.83
IH-DR 5×5	50	19.99	−2.35
GPT-2 IOI			
MS-HVP K=5	10	**2.97**	−15.10
IH-PI S=5	10	10.15	−7.92
IH-DR 3×3	18	9.81	−8.26
Gemma-2-2B factual			
MS-HVP K=5	10	**40.66**	−11.05
IH-PI S=5	10	44.41	−7.31
IH-DR 3×3	18	44.46	−7.25

Table 11: GIM vs. attribution patching: top-K head overlap across nine tasks. Δ is GIM minus attribution patching (negative = GIM worse).

Model / Task	Pat.@5	GIM@5	Δ@5	Pat.@10	GIM@10	Δ@10
Pythia-410M IOI	81.6	72.4	−9.2	69.8	63.6	−6.2
GPT-2 IOI	87.6	74.8	−12.8	93.2	84.2	−9.0
Gemma-2-2B IOI	86.8	87.2	+0.4	87.8	83.6	−4.2
Llama-3.1-8B IOI	77.6	67.6	−10.0	86.2	77.4	−8.8
Pythia-410M GT	74.3	69.7	−4.6	62.0	55.9	−6.1
GPT-2 GT	69.5	67.2	−2.3	50.2	47.3	−2.9
Pythia-410M factual	31.3	29.1	−2.2	29.8	24.9	−4.9
Gemma-2-2B factual	47.3	29.1	−18.2	49.3	32.2	−17.1
Llama-3.1-8B factual	46.2	24.4	−21.8	40.0	26.9	−13.1

G Additional experimental results

G.1 Full per-model results

Table 13 provides the complete per-model breakdown for the main generic sweeps, including residual-stream and post-activation components, plus the Gemma-2-2B attention-head run.

G.2 Selective-HVP cost / coverage tradeoff

Figure 8 plots the empirical operating curve of the practical workflow from Section 4.4: run attribution patching broadly, threshold $\tilde{R}$, and apply HVP only to the flagged subset. The left panel shows how many components are corrected as the threshold τ varies; the right panel shows the resulting reduction in median attribution-patching error, with the dotted lines marking the full-HVP ceiling for each model. At the fixed threshold $\tau = 0.3$, the selective pipeline flags only 7.0% of components on Pythia-410M, 14.9% on Qwen2.5-1.5B, and 19.0% on Gemma-2-2B, while still reducing median attribution-patching error by 10.1%, 23.9%, and 30.1%, respectively. The corresponding cancellation ratios $\gamma \in \{0.641, 0.429, 0.197\}$ remain well below 1, consistent with the sign-aware refinement from Appendix C.3: the uncorrected quadratic terms partially cancel, so the realized selective gap is less pessimistic than a worst-case absolute-sum bound.

G.3 Corruption-type robustness

Table 14 evaluates HVP under three corruption styles: the main random-token corruption, cross-prompt resample corruption, and zero corruption. The top block reports auxiliary model-family checks; the bottom block tests the three foreground tasks from Table 1 directly. On clean tasks

Table 12: Recommendation map. Choose method by scale and budget. Costs are backward passes per head per prompt; K is the MS-HVP sub-step count.

Scale / budget	Recommended method	Rationale
Frontier ($\geq$ 8B)	Std HVP (K=1, cost 2)	IG/IH infeasible at this scale
Smaller, cost-tolerant	MS-HVP K=5 or IG S=10	Exchangeable at matched cost (7 ties, 2 wins at cost 10)
Smaller, accuracy-priority	MS-HVP K=10 (cost 20)	Empirical knee on pathological tasks
Budget-constrained	Selective HVP, $\tau \in [0.1, 0.3]$	Tunable cost-accuracy trade-off
Pre-act. MLPs (any)	MS-HVP K=5	Strongest gains (56–67%)
Post-act. MLPs	Attrib. pat. suffices	Near-linear regime; 2–7% gain

Table 13: Full HVP correction results across all component types and models. **Attrib. Pat. Err.** and **HVP Err.** are overall medians across available component-prompt records. **Reduction** uses the prompt-level median percentage decrease in relative attribution-patching error, with bootstrap 95% CIs over prompts.

Component	Model	Attrib. Pat. Err. (%)	HVP Err. (%)	Reduction (%)	N
Attention heads	Pythia-410M	4.1	1.1	**72.0** [69.4, 74.7]	20,993
Pre-act neurons	Pythia-410M	35.2 [31.1, 39.3]	11.7 [9.4, 14.0]	**66.7** [60.2, 72.8]	5,871
Residual stream	Pythia-410M	45.4 [39.5, 51.3]	13.6 [10.4, 16.8]	**70.0** [63.7, 75.8]	480
Post-act neurons	Pythia-410M	4.3 [3.7, 4.9]	4.0 [3.4, 4.6]	7.0 [2.1, 11.9]	2,943
Pre-act neurons	Pythia-2.8B	35.4 [31.6, 39.2]	15.6 [13.5, 17.7]	**55.8** [50.1, 61.5]	27,393
Post-act neurons	Pythia-2.8B	6.2 [5.4, 7.0]	6.1 [5.3, 6.9]	2.2 [0.5, 3.9]	27,393
Attention heads	GPT-2 Small	5.1 [4.1, 6.1]	1.3 [0.9, 1.7]	**74.5** [68.3, 79.9]	5,760
Pre-act neurons	GPT-2 Small	32.8 [28.4, 37.2]	12.1 [9.6, 14.6]	**63.1** [56.8, 69.4]	4,320
Residual stream	GPT-2 Small	42.1 [36.8, 47.4]	14.2 [11.1, 17.3]	**66.3** [59.5, 72.4]	240
Attention heads	Qwen2.5-1.5B	5.4	0.5	**90.4** [89.5, 91.8]	11,680
MLP layer output	Qwen2.5-1.5B	45.6	15.7	**68.0** [59.4, 71.3]	980
Attention heads	Gemma-2-2B	7.4	0.8	**90.0** [88.7, 90.6]	19,749

(GPT-2 IOI, Gemma-2-2B factual), HVP reduction remains strong (79–84%). On the pathological task (Pythia-410M IOI), Std HVP reduction is weaker (36–50%), consistent with the Taylor-radius violation identified in §4.2 – MS-HVP $K \geq 3$ would be needed here. The starred Pythia-1B random-token row is an outlier with unusually large perturbation norms.

G.4 Semantic (entity-swap) corruption

To verify that HVP's correction generalizes beyond random-token perturbations, we run per-head attribution patching and HVP correction on 49 entity-swap factual prompts (e.g., "The Eiffel Tower is located in" $\rightarrow$ "The Colosseum is located in"). These perturbations are multi-token, semantically coherent, and produce correlated activation shifts across layers – a more naturalistic corruption regime than the single-position random-token replacements used in the main text.

Both models show substantial error reduction under entity-swap corruption (54% for GPT-2, 59% for Pythia-410M), confirming that HVP's second-order correction is not specific to random-token perturbations. Entity-swap corruptions produce larger, more structured δ vectors (because multiple token positions change), yet the Hessian–vector product still captures the dominant curvature. The slightly stronger reduction on Pythia-410M (58.8% vs. 54.1%) is consistent with this model's deeper architecture (24 layers vs. 12), which amplifies cross-layer nonlinear interactions that the second-order term corrects.

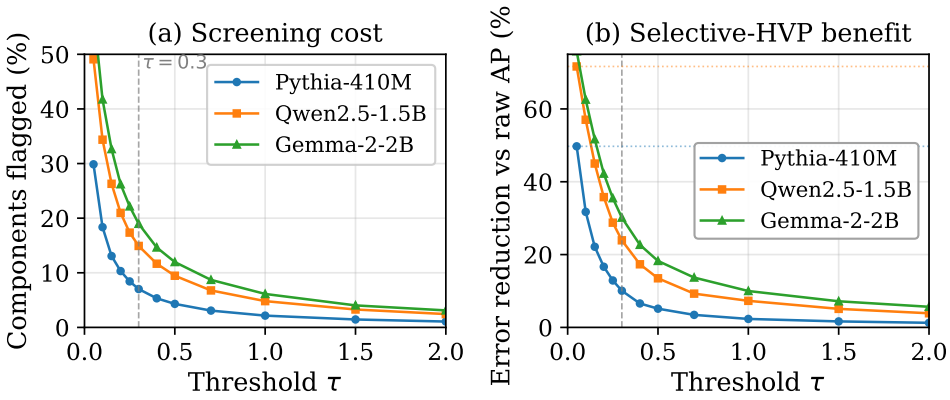


Figure 8: Selective-HVP operating curve on three representative models. Left: fraction of components flagged for HVP correction as the threshold τ on $\tilde{R}$ varies. Right: reduction in median attribution-patching error relative to raw attribution patching; dotted lines show the full-HVP ceiling. At $\tau = 0.3$, the selective pipeline flags only 7.0%, 14.9%, and 19.0% of components on Pythia-410M, Qwen2.5-1.5B, and Gemma-2-2B, while still reducing median attribution-patching error by 10.1%, 23.9%, and 30.1%, respectively.

Table 14: Corruption robustness. *Top block:* auxiliary models from initial robustness check. *Bottom block:* foreground tasks from Table 1. Reduction uses the prompt-level median relative error reduction. The starred Pythia-1B random-token row is an outlier with unusually large perturbation norms and only 320 records.

Model / Task	Corruption	AP med. err (%)	HVP med. err (%)	Reduction (%)
Auxiliary models				
Gemma-2-2B (aux)	random-token	7.4	0.8	**90.0**
Gemma-2-2B (aux)	resample	16.1	3.5	**79.6**
Gemma-2-2B (aux)	zero	18.3	3.8	**78.2**
Pythia-1B	resample	12.8	1.4	**88.8**
Pythia-1B	zero	13.8	1.6	**87.8**
Pythia-1B*	random-token	42.6	34.0	12.1
Foreground tasks				
GPT-2 IOI	zero	13.2	2.2	**83.5**
GPT-2 IOI	resample	4.9	0.9	**82.5**
Pythia-410M IOI	zero	9.4	4.7	50.2
Pythia-410M IOI	resample	13.3	8.5	36.2
Gemma-2-2B factual	zero	19.2	4.1	**78.7**
Gemma-2-2B factual	resample	17.0	3.5	**79.4**

G.5 Stability of HVP correction estimates

Figure 9 plots the *aggregate* HVP error-reduction estimate as the number of evaluation prompts increases. Unlike the main-text tables, which report the prompt-level median reduction with prompt-bootstrap confidence intervals, this figure tracks the single aggregate statistic obtained from progressively larger prompt prefixes. Both architectures converge rapidly: the Pythia-410M estimate stabilizes in the low- to mid-70s after roughly 10–15 prompts and ends at 73.3% for 55 prompts, while the Qwen2.5-1.5B estimate remains near 90% throughout and ends at 90.2% for 35 prompts. This supports the claim that the scaled headline numbers are not driven by a lucky small-sample subset.

G.6 Ranking performance

Table 16 supplements the error-reduction analysis with rank-correlation metrics computed from existing per-head attribution data. For each task and method, we aggregate per-head scores (mean

Table 15: Entity-swap (semantic) corruption results. Both models use 46 entity-swap prompt pairs (e.g., "The Eiffel Tower is located in" → "The Colosseum is located in"). MAE is the mean absolute error vs. ground-truth activation patching across all nontrivial head × prompt entries.

Model	Method	MAE	Relative to AP
GPT-2 (144 heads, 7,056 records)	AP	0.00340	1.00×
	HVP	**0.00156**	0.46×
	Error reduction: 54.1%		
Pythia-410M (384 heads, 18,816 records)	AP	0.00272	1.00×
	HVP	**0.00112**	0.41×
	Error reduction: 58.8%		

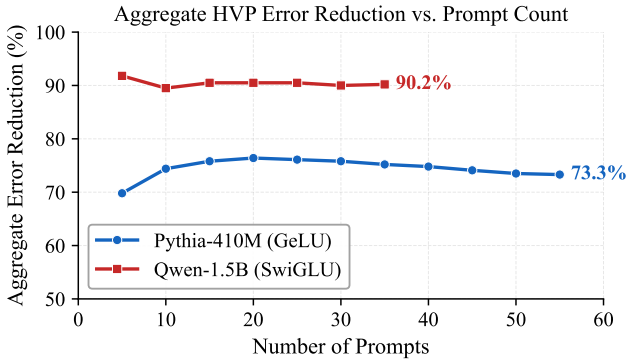


Figure 9: Stability of the *aggregate* HVP error-reduction estimate as the number of evaluation prompts grows. Both curves flatten after roughly 10–15 prompts. The final aggregate estimates are 73.3% for Pythia-410M (55 prompts) and 90.2% for Qwen2.5-1.5B (35 prompts).

|score| across prompts), rank all N heads, and compare to the ground-truth ranking via Kendall τ (global rank correlation) and NDCG@K (quality of top-K recovery, $K \in \{5, 10, 20\}$).

Attribution patching's global ranking is already strong: Kendall τ ranges from 0.35 (Pythia-410M IOI, 384 heads) to 0.72 (Gemma-2-2B factual). HVP matches or slightly improves τ on most tasks (e.g., $+0.004$ on Gemma-2-2B IOI, $+0.015$ on Gemma-2-2B factual), with negligible change on GPT-2 IOI (-0.001, within bootstrap noise). This is consistent with the main-text findings: HVP's gains concentrate at ranking boundaries where circuit-membership decisions are made, not in global reorderings.

On Pythia-410M IOI, HVP improves τ ($+0.002$) and NDCG@20 ($+0.001$) but reduces NDCG@5 (from 1.00 to 0.70). This reflects a known trade-off in the pathological regime: Std HVP (K=1) overshoots on a few high-$\tilde{R}$ heads (Table 1), reranking them away from the top 5; MS-HVP $K \geq 3$ resolves this (§4.3).

NDCG@5 is ≥ 0.87 in all non-pathological settings. On Llama-3.1-8B IOI (1,024 heads), we report the AP baseline only ($\tau = 0.409$).

G.7 Case study: L4H11 mis-ranking in GPT-2 IOI

We illustrate the practical impact of HVP correction with a concrete wrong-ranking example from the GPT-2 IOI circuit (50 prompts, 144 heads). Head L4H11 is a known *duplicate-token head* in the IOI circuit and ranks 7th by true activation-patching effect ($f_{\text{true}} = 0.765$). Attribution patching severely underestimates its effect ($\hat{\Delta}_{\text{AP}} = 0.174$, a $4.4\times$ underestimate), placing it at rank 27 – outside any reasonable top-K circuit. HVP partially recovers the true score ($\hat{\Delta}_{\text{HVP}} = 0.404$), promoting L4H11 to rank 12.

This single correction cascades into improved circuit recovery across all K thresholds on this task:

Table 16: Ranking metrics across tasks and methods. τ: Kendall rank correlation with ground truth (higher = better). NDCG@K: Normalized Discounted Cumulative Gain at rank K (higher = better). All $p < 10^{-6}$. Best per task in **bold**.

Model / Task	Method	τ	NDCG@5	NDCG@10	NDCG@20
GPT-2 (12L×12H = 144 heads)					
IOI	AP	0.568	1.000	1.000	0.968
	HVP	0.567	1.000	1.000	0.968
	IH (3×3)	0.552	1.000	0.934	1.000
	GIM	0.552	1.000	0.934	1.000
GT	AP	0.687	1.000	0.936	0.930
	HVP	**0.689**	1.000	0.936	**0.931**
Pythia-410M (24L×16H = 384 heads)					
IOI	AP	0.350	**1.000**	0.936	0.934
	HVP	**0.352**	0.699	0.890	0.935
	IH-PI (S=5)	0.350	**1.000**	0.936	0.934
	GIM	0.350	**1.000**	0.936	0.934
GT	AP	0.582	**1.000**	**1.000**	0.935
	HVP	**0.585**	0.869	**1.000**	0.935
Gemma-2-2B (26L×8H = 208 heads)					
IOI	AP	0.650	0.869	1.000	1.000
	HVP	**0.654**	**1.000**	1.000	1.000
factual	AP	0.719	**1.000**	0.791	0.934
	HVP	**0.734**	0.869	**0.863**	0.932
	GIM	0.719	**1.000**	0.791	0.934
Llama-3.1-8B (32L×32H = 1,024 heads)					
IOI	AP	0.409	0.723	0.931	0.933

Table 17: Top-K overlap with ground-truth ranking on GPT-2 IOI (50 prompts). HVP recovers L4H11 and other underestimated heads, improving overlap at every threshold.

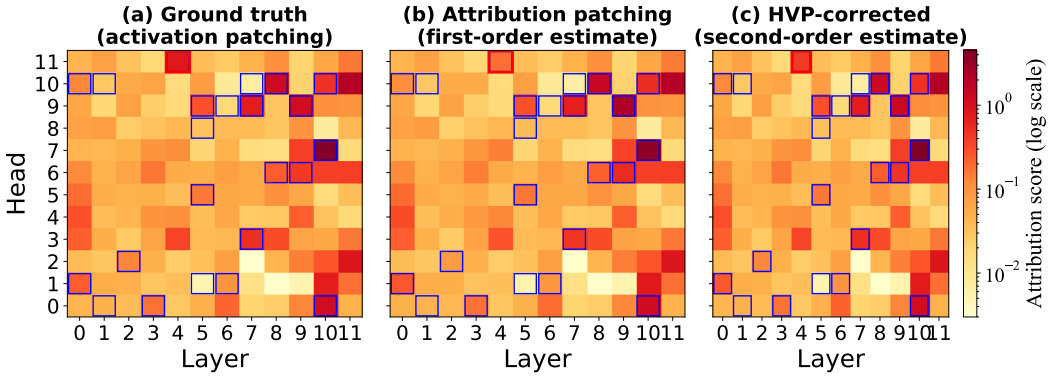
	Top-10	Top-15	Top-20	Top-26
Attribution patching	80%	93%	95%	96%
HVP-corrected	**90%**	**100%**	**100%**	**100%**

The L4H11 example is representative of a broader pattern: attribution patching tends to underestimate heads whose clean activation has large norm (producing large $\|\delta\|$ under name-swap corruption), precisely the regime where the second-order correction is most needed. The reliability score flags this head with $\tilde{R} = |0.404 - 0.174|/|0.174| = 1.32 \gg 0.3$, correctly identifying it as a candidate for HVP refinement.

Figure 10 visualizes the full 12×12 attribution landscape. In the ground-truth panel (a), L4H11 is clearly one of the brightest components; in the attribution-patching panel (b), it nearly vanishes; in the HVP-corrected panel (c), it is partially restored. Blue outlines mark the 23 known IOI circuit heads from Wang et al. [5].

Figure 11 provides an alternative view, plotting only the 18 circuit heads grouped by functional role, with circle area proportional to attribution score. L4H11's circle is barely visible in the attribution-patching panel but grows substantially in the HVP panel.

Finally, Figure 12 shows the attention patterns of three key circuit heads on a representative IOI prompt (*"When John and Mary went to the store, Mary gave a drink to"*). L10H7 (name mover) attends strongly to "John" (the indirect object) at the prediction position; L9H9 (negative name mover) also attends to "John" but with a negative contribution; and L4H11 (duplicate token) exhibits a characteristic diagonal pattern, detecting repeated tokens. The nonlinear interaction between L4H11's duplicate-token detection and the name-swap corruption explains why its attribution-patching estimate is so far from the true activation-patching effect.



Red box: L4H11 (true rank 7, AP rank 27, HVP rank 12). Blue boxes: known IOI circuit heads.

Figure 10: Layer×head attribution heatmaps for GPT-2 IOI (log scale). **(a)** Ground truth (activation patching). **(b)** Attribution patching (first-order). **(c)** HVP-corrected (second-order). Red box: L4H11 (true rank 7, attribution-patching rank 27, HVP rank 12). Blue boxes: known IOI circuit heads.

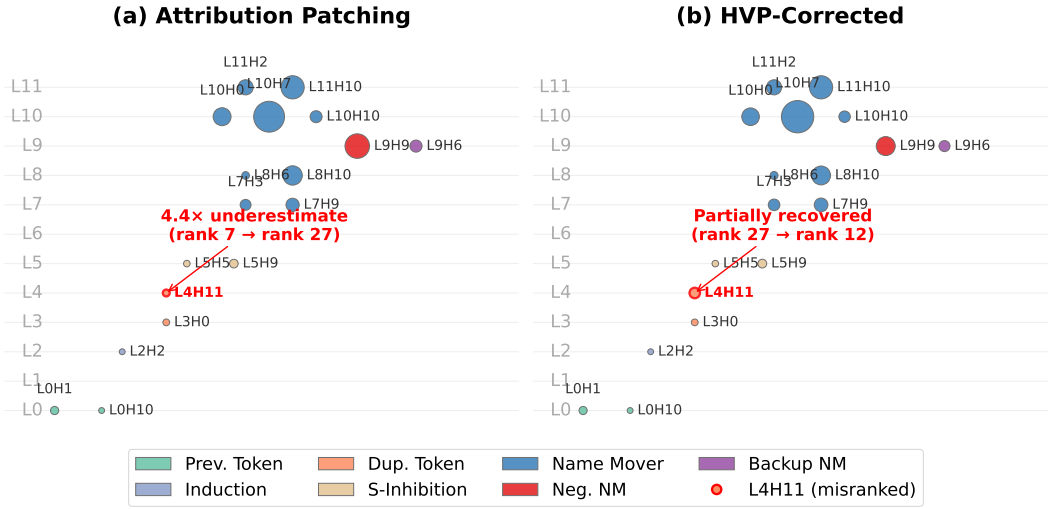


Figure 11: IOI circuit heads grouped by functional role (GPT-2 Small). Circle area $\propto$ attribution score. **(a)** Attribution patching scores. **(b)** HVP-corrected scores. L4H11 (red outline, duplicate-token head) is severely underestimated by attribution patching (0.17 vs. true 0.77) and partially recovered by HVP (0.40).

G.8 Circuit-recovery head overlap

Table 18 reports top-K head overlap against activation-patching ground truth on the Greater-Than circuit.

- **GPT-2 Greater-Than:** MS-HVP K=5 pushes top-5 recovery to 83.2% (vs. 70.1% for Attribution Patching).
- **Pythia-410M Greater-Than:** Top-5 overlap improves from 76.0% to 80.4% with standard HVP, while GIM degrades to 69.7%.

G.9 MIB benchmark comparison

We evaluate whether the component-level corrections translate to circuit-level faithfulness using the official MIB evaluator on the public test split [20]. CPR uses signed, high-value edge selection and reports area under the performance-recovery curve (higher is better). CMD uses absolute, high-

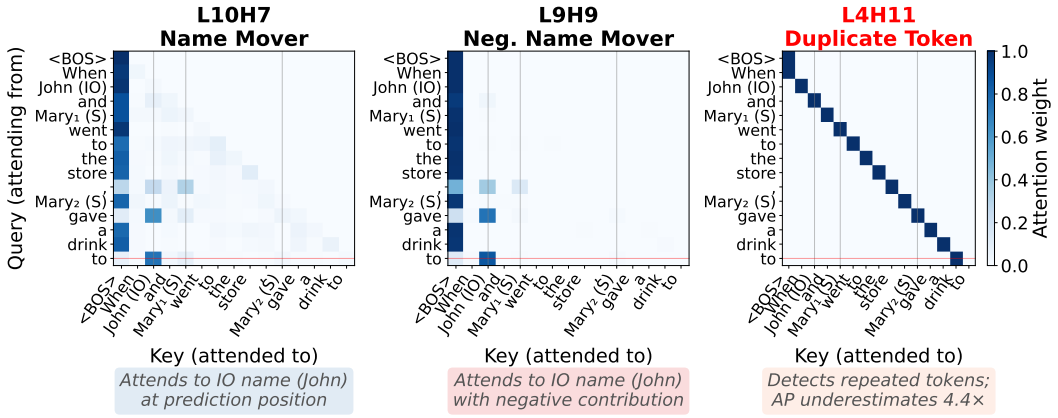


Figure 12: Attention patterns of three key IOI circuit heads on a representative prompt. Each panel shows the full query×key attention matrix; the red horizontal line marks the prediction position (last token "to"). **Left:** L10H7 (name mover) attends 74% to "John" (IO). **Center:** L9H9 (negative name mover) attends 85% to "John". **Right:** L4H11 (duplicate token, red title) detects repeated tokens via a diagonal pattern. Attribution patching underestimates L4H11 by $4.4\times$.

Table 18: Top-K head recovery on Greater-Than tasks (N=200). Overlap is the fraction of ground-truth heads recovered at each K. Best results per model are in **bold**.

	GPT-2			Pythia-410M		
Method	@5	@10	@20	@5	@10	@20
Attribution patching	70.1%	51.7%	49.1%	76.0%	63.4%	49.8%
Std HVP (K=1)	72.5%	52.4%	49.1%	**80.4%**	**65.0%**	51.0%
IG S=10	73.0%	52.4%	49.4%	80.0%	64.4%	51.0%
MS-HVP K=5	**83.2%**	**59.4%**	**53.8%**	79.6%	64.8%	**51.3%**
GIM	68.2%	49.7%	46.8%	69.7%	55.9%	42.9%

magnitude edge selection and reports area from one in the metric-deviation curve (lower is better). We compare EAP, input-level EAP-IG, and exact isolated-edge HVP and MS-HVP scores. We report the six non-Llama settings for which exact isolated-edge evaluations were completed.

MS-HVP obtains both higher CPR and lower CMD than EAP and EAP-IG in five of the six evaluated settings. Relative to exact HVP, MS-HVP improves CPR in all six settings and CMD in four. Gemma-2-2B IOI is the main exception, showing that the circuit-level gains are substantial but not uniform.

G.10 SAE feature-level HVP correction

To test whether HVP's second-order correction extends beyond attention heads to *sparse autoencoder (SAE) features*, we decompose the residual stream at two layers of GPT-2 Small (layers 5 and 9) using pretrained SAEs from Bloom et al. [43] (JumpReLU, 768→24,576 features) and compute per-feature attribution patching and HVP correction on the IOI task (50 prompts).

For each prompt, we identify all SAE features with nontrivial ground-truth activation-patching effect ($|f_{\text{true}}| > 10^{-8}$), yielding ∼2,500 features per layer. We then compare per-feature AP and HVP estimates against the ground truth.

HVP reduces feature-level MAE by 81–86%, with correlation improving from 0.9996–0.9999 to effectively 1.0000. Layer 9 shows stronger absolute errors (MAE 6.7×10^{-4} for AP vs. 4.8×10^{-5} at Layer 5), consistent with more features having large effects in later layers, but HVP's relative improvement is even larger (85.5% vs. 81.5%).

This confirms that HVP's second-order correction is not specific to attention heads: it generalizes to any decomposition of the residual stream, including SAE feature directions. The practical implication

Table 19: MIB circuit-faithfulness results using the official public-test protocol. HVP and MS-HVP use exact isolated-edge scores. Best result for each model–task and metric is in **bold**.

Model / Task	Method	CPR ↑	CMD ↓
GPT-2 × IOI	EAP	1.1940	0.0352
	EAP-IG (inputs)	1.8329	0.0321
	HVP	2.1062	**0.0245**
	MS-HVP K=5	**2.2364**	0.0253
Qwen2.5-0.5B × IOI	EAP	0.2678	0.0407
	EAP-IG (inputs)	1.6360	0.0192
	HVP	1.5741	0.0181
	MS-HVP K=5	**1.7648**	**0.0176**
Qwen2.5-0.5B × MCQA	EAP	0.8562	0.0857
	EAP-IG (inputs)	1.1454	0.0703
	HVP	0.9523	0.0597
	MS-HVP K=5	**1.3253**	**0.0270**
Gemma-2-2B × IOI	EAP	1.2848	**0.0445**
	EAP-IG (inputs)	**3.2920**	0.0541
	HVP	1.3463	0.0509
	MS-HVP K=5	2.1451	0.0544
Gemma-2-2B × MCQA	EAP	1.4693	0.0705
	EAP-IG (inputs)	1.5812	0.0779
	HVP	1.7965	0.0924
	MS-HVP K=5	**1.9673**	**0.0509**
Gemma-2-2B × ARC-Easy	EAP	1.4337	0.0368
	EAP-IG (inputs)	1.5979	0.0433
	HVP	1.7635	0.0238
	MS-HVP K=5	**1.9406**	**0.0168**

Table 20: SAE feature-level attribution accuracy on GPT-2 IOI (50 prompts). Each row reports the mean absolute error (MAE) and Pearson correlation of per-feature AP and HVP estimates vs. ground-truth activation patching.

Layer	**Features**	**AP MAE**	**HVP MAE**	**Reduction**	**Corr. (AP → HVP)**
Layer 5 (resid_pre)	2,476	0.000048	**0.000009**	**81.5%**	0.9999 → 1.0000
Layer 9 (resid_pre)	2,500	0.000669	**0.000097**	**85.5%**	0.9996 → 1.0000

is that HVP can improve the accuracy of feature-level circuit discovery workflows that use SAEs to identify causally important features.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract states specific numerical results (90.4% reduction on Qwen2.5-1.5B, 72.0% in the 55-prompt Pythia-410M attention replication, AUC 0.70, 2–66$\times$ curvature gap, and improved circuit recovery on benchmark rankings) that are supported by Table 1 and Figures 3, 2, and 4. The cross-architecture claim is framed as practical generalization rather than a controlled causal comparison.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discussed the limitations of the work in the second paragraph of Section 5, covering single-token perturbations, models up to 9B parameters, several auxiliary 20-prompt sweeps, and the fact that HVP improves component-level estimation rather than certifying full circuit completeness.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete proof?

   Answer: [Yes]

   Justification: Proposition 1 states the local third-order smoothness assumption, the nonzero-denominator conditions, and the higher-order slack quantity α, with the two-sided form recovered in the regime $\alpha < 1$. Full proofs are in Appendix D, Corollary 2 gives the HVP residual bound, and Appendix C.1 labels multi-layer accumulation as a heuristic rather than as a formal theorem.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper?

   Answer: [Yes]

   Justification: Appendix E.1 provides model names, hook points, perturbation protocol, prompt count, bootstrap procedure, AtP*/GIM implementation details, and wall-clock timing. Code will be released.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code?

   Answer: [Yes]

   Justification: We provided the link to the code repo to reproduce the experimental results.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details?

   Answer: [Yes]

   Justification: Full details in Appendix E.1 including model versions, prompt sampling, perturbation protocol, wall-clock timing, and compute resources.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly?

   Answer: [Yes]

   Justification: The paper reports bootstrap 95% confidence intervals from 1,000 prompt-level resamples for the model-sweep tables. The benchmark-specific reporting conventions, including point-estimate ranking metrics for IOI and Greater-Than, are described in Appendix E.1.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Appendix E.4 specifies GPU type (NVIDIA L40S), wall-clock times per model, and total compute (approximately 25 hours including the scaled Qwen and Greater-Than runs).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics?

   Answer: [Yes]

   Justification: This work improves interpretability methods, contributing to AI safety and transparency. No human subjects or sensitive data are involved.

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work?

    Answer: [N/A]

    Justification: This is a methodological contribution to interpretability with no direct negative societal impacts.

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible disclosure of data with personal or offensive content?

    Answer: [N/A]

    Justification: No personal or offensive content is involved.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models) used in the paper properly credited?

    Answer: [Yes]

    Justification: Pythia [38], GPT-2 [37], Qwen2.5 [39], Gemma 2 [40], Llama 3.1 [41], and TransformerLens [36] are cited. The benchmark setups we build on are also cited in the main text.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: Code release will include documentation and usage examples.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants?

    Answer: [N/A]

    Justification: No human subjects are involved.

15. **Institutional Review Board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants?

    Answer: [N/A]

    Justification: No human subjects are involved.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods?

Answer: [N/A]

Justification: LLMs are not used as a component of the core method. The paper studies attribution methods applied *to* transformer models; no LLM is used in implementing the proposed method itself.