DO WE NEED HANDCRAFTED CORRUPTIONS? CONTEXT-PRESERVING ACTIVATION PERTURBATION IN REPRESENTATION SPACE

Anonymous authors
Paper under double-blind review

ABSTRACT

Activation patching is widely used in mechanistic interpretability to identify causally important model components. Yet its reliability depends on constructing corrupted inputs that alter task-relevant information while preserving the surrounding context. Designing such inputs is difficult in complex tasks, especially when answer-related evidence is dispersed and related mentions must be edited consistently. To address this challenge, we revisit activation patching in representation space rather than input text and introduce Context-Preserving Activation Perturbation (CPAP), which automatically constructs activation perturbations without using handcrafted corrupted inputs. CPAP maximizes the change in an upstream module's activation while constraining changes in selected downstream residual representations, using their stability as a practical proxy for preserving the surrounding computational context. Our experiments yield two main findings. First, across three models and four benchmarks, CPAP closely matches the component rankings from corrupted-input patching, with consistently higher rank agreement than zero and learned constant ablations. For GPT-2 circuit localization, it achieves the highest AUROC among automatic methods on all three tasks (0.968–1.000). Second, CPAP easily extends to explaining model behavior across complex tasks and larger models (27B). In long-context question answering and mathematical reasoning, it automatically constructs activation perturbations and identifies key components without manually designed corrupted inputs.

1 INTRODUCTION

Mechanistic interpretability takes a white-box approach to neural networks, reverse-engineering the internal computations of a model and the functions implemented by its components (Olah et al., 2020; Elhage et al., 2021). For this purpose, a widely used method is activation patching, which identifies causal relations among model components through interventions on corrupted inputs (Vig et al., 2020; Conmy et al., 2023). The method has been applied across a wide range of scenarios, including from the fundamental model capabilities (e.g. numerical comparison (Hanna et al., 2023), indirect-object identification (IOI) (Wang et al., 2022), and factual recall (Meng et al., 2022)), to the real-world applications, such as symptom recognition in medical language models (He et al., 2025).

The causal relation recovered by activation patching depends on how the corrupted input is constructed, which can substantially affect the reliability of the results (Zhang & Nanda, 2024). A valid corruption must alter the key information (e.g. name in IOI) in original text, while preserving the surrounding context (Heimersheim & Nanda, 2024). Such contrasts are typically designed manually and are difficult to construct (Bian et al., 2025), particularly in long-context tasks such as reading comprehension. As illustrated in Figure 1, dispersed and contiguous evidence pose distinct challenges. Dispersed evidence requires locating all relevant information across documents and consistently editing every mention. Contiguous evidence requires replacing the relevant passage with an equal-length span that changes only the property under test (Zhang & Nanda, 2024; Yeo et al., 2025).

To address this limitation, we revisit activation patching through the lens of causal mediation analysis (CMA) (Vig et al., 2020). From this perspective, the essential operation is an intervention on internal representations, which motivates a natural question: can we construct perturbations directly

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

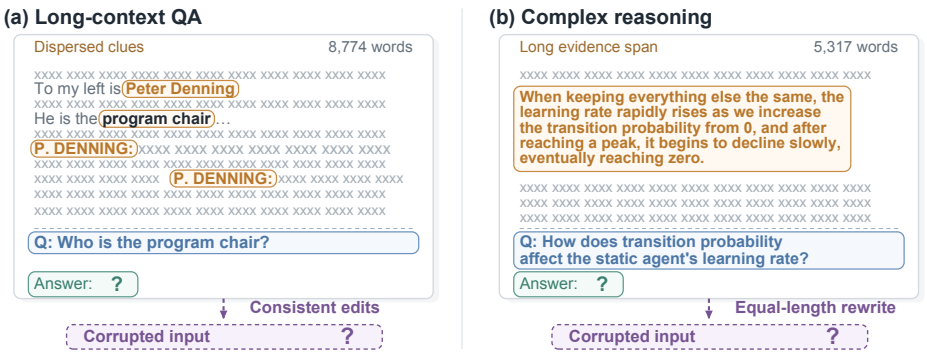


Figure 1: Why handcrafted corruptions are difficult to scale. Answer evidence involves repeated person references (a) or an entire explanatory sentence (b), requiring selective edits that preserve the surrounding context. Examples are from LongBench (Bai et al., 2024); gray text and document layouts are schematic.

in representation space without relying on manually designed corrupted inputs? The challenge lies in controlling what these perturbations change. Internal activations may encode task-relevant information together with background computational context in an entangled form (Elhage et al., 2022). A large perturbation to a module's representation may exceed the space and therefore disrupt unrelated downstream computations, making the resulting behavioral change difficult to attribute to the information under investigation. The central challenge is thus to induce a substantial perturbation to the information under test while preserving the surrounding computational context.

In this work, we introduce **Context-Preserving Activation Perturbation (CPAP)**, a method for constructing substantial local activation perturbations while limiting disturbance to the surrounding computational context. CPAP maximizes the deviation of an upstream module's activation from its clean value while keeping selected downstream residual representations close to their clean counterparts. Because the residual stream aggregates contributions from multiple components (Elhage et al., 2021), deviations in these downstream representations provide a practical proxy for disruption to the surrounding computation. We optimize this representation-level objective over activations induced by modified inputs, using gradients to guide the input search. The resulting perturbed activations define the patching intervention, eliminating the need to manually design task-specific corrupted examples.

Our experiments yield two main findings. First, we evaluate the effectiveness of CPAP on four benchmark tasks with established corruption strategies across GPT-2 Small (Radford et al., 2019), OLMo-2-1B (Allen Institute for AI, 2025), and Qwen3-8B-Base (Qwen Team, 2025). CPAP reliably localizes important components, closely aligning with handcrafted patching on these controlled tasks. It achieves the highest rank agreement among automatic perturbation methods in eight of twelve model–task combinations, with Top-10 Jaccard similarity ranging from 0.538 to 1.000. Second, we examine its extension to long-context question answering and mathematical reasoning (Bai et al., 2024; Cobbe et al., 2021), where reusable corruptions on inputs are difficult to construct. In these case studies, CPAP identifies heads that retrieve an answer from a document and transfer an already computed result to the final answer, respectively, without manually constructing corrupted counterparts.

2 Method

2.1 Activation Patching as a Targeted Intervention

Activation patching perturbs an internal module by replacing its activation and measuring the resulting change in model behavior to assess the module's causal contribution (Vig et al., 2020). For a frozen model f and input x, the intervention effect is

$$\Delta_m(\widetilde{h}_m; x) = S\big(f_{m \leftarrow \widetilde{h}_m}(x)\big) - S\big(f(x)\big), \tag{1}$$

where $f_{m \leftarrow \widetilde{h}_m}$ replaces module m's activation with $\widetilde{h}_m$, and S is a task score with higher values indicating better performance. A negative effect indicates that the perturbation reduces task performance.

To investigate how a module transmits a particular semantic factor, a targeted intervention should **change that factor** while **preserving unrelated background information** (Heimersheim & Nanda, 2024). Conceptually, write $h_m(x) = \Phi_m(s, b)$ at a specified token position, where s is the target factor, b is background information, and Φ_m encodes them jointly without assuming linearity or independence. An ideal objective over alternative factors s' and b' is

$$\max_{s', b'} \|\Phi_m(s', b') - \Phi_m(s, b)\|_2 \quad \text{subject to} \quad b' = b, \quad s' \neq s. \tag{2}$$

The distance measures intervention strength in activation space, while the constraints specify which information may change.

Corrupted-input patching approximates this contrast with a targeted input edit, using $\widetilde{h}_m = h_m(x^{\text{corr}})$ for the edited input x^{corr}. For example, in indirect object identification (IOI), replacing the second "Bob" with "Charlie" in "Alice and Bob went to the store. Bob gave a book to" breaks the name-repetition relation while retaining the surrounding text (Heimersheim & Nanda, 2024). **Zero ablation** sets $\widetilde{h}_m = 0$, while **Gaussian patching** adds Gaussian noise to selected token embeddings and extracts perturbed activations from the perturbed computation (Meng et al., 2022). Neither explicitly separates changes to s from changes to b, making the resulting effects difficult to attribute specifically to the target information (Heimersheim & Nanda, 2024; Zhang & Nanda, 2024). Constructing the desired activation perturbation therefore requires controlling these two kinds of change without directly observing or separately manipulating s and b.

2.2 Context-Preserving Activation Perturbation

A general objective for internal modules. The target and background factors in Equation (2) cannot be measured separately. Our preliminary experiments show that perturbations such as zero ablation can cause large changes in downstream representations. This motivates CPAP to constrain changes in selected downstream residuals as a proxy for preserving background information, while maximizing the upstream activation change. With $\widetilde{h}_m = \Phi_m(s', b')$ and fixed normalization scales, the resulting objective is

$$\max_{\widetilde{H} \in \mathcal{H}(x)} \frac{\left\|\widetilde{h}_m - h_m(x)\right\|_2}{a_m} \quad \text{subject to} \quad \max_{\ell \in \mathcal{D}} \frac{\|\widetilde{r}_\ell - r_\ell(x)\|_2}{b_\ell} \leq \tau_x. \tag{3}$$

Here $\mathcal{H}(x)$ is the space of candidate representation collections. Each $\widetilde{H} \in \mathcal{H}(x)$ contains a candidate module activation $\widetilde{h}_m$ and residual representations $\{\widetilde{r}_\ell\}_{\ell \in \mathcal{D}}$ from a candidate computation. The set $\mathcal{D}$ indexes residual checkpoints after selected layers; $h_m(x)$ and $r_\ell(x)$ are their clean counterparts. The positive scales a_m and b_ℓ are fixed root-mean-square activation norms estimated from reference inputs, and τ_x bounds the normalized residual change for input x (Figure 2). The objective applies to MLPs, attention heads, and attention query, key, or value representations.

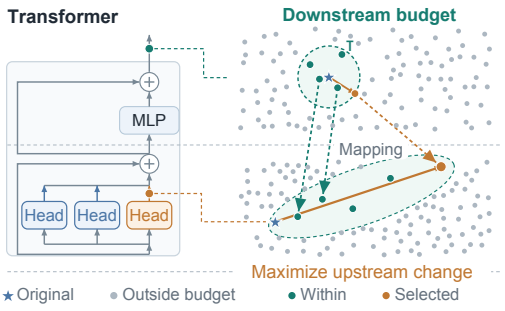


Figure 2: A downstream budget induces an upstream search region. Dashed mapping arrows connect downstream candidates to their upstream counterparts. Stars mark originals; solid orange arrows show selected changes. Both clouds share candidates; geometry is schematic.

2.3 An Efficient Instantiation of CPAP

Optimizing Equation (3) separately for every module would require repeated searches over a large activation space. We reduce this cost by restricting candidates to activations from local input edits and sharing a gradient-guided search across modules.

Restricting the candidate activation space. Direct search in the high-dimensional activation space is impractical. We instead obtain candidate activations by sampling inputs, but the full input space is still too large. CPAP therefore samples local transformations of the original input, substituting tokens at selected positions $\mathcal{P}$ with tokens whose embeddings are nearby. The candidate spaces are

$$\begin{aligned}\mathcal{X}_{\mathcal{P}}(x) = \{x' \neq x : |x'| = |x|,\ x'_i = x_i \text{ for } i \notin \mathcal{P},\\ x'_p \in \mathcal{V}_p \text{ for } p \in \mathcal{P}\},\\ \mathcal{H}_m(x) = \{h_m(x') : x' \in \mathcal{X}_{\mathcal{P}}(x)\}.\end{aligned} \tag{4}$$

Here x_i is the token at position i, $|x|$ is the sequence length, and $\mathcal{V}_p$ contains x_p and the nearest quarter of non-special vocabulary tokens by embedding cosine similarity. The editable positions $\mathcal{P}$ can be specified by the user or selected by our automatic position-search procedure (Appendix B.12.1).

Sharing a search across modules. To avoid a separate search for every tested module, we select one candidate text by maximizing aggregate head changes under a residual constraint. Let $\mathcal{U}$ be the selected head set, $z_u(x)$ the output of head u before the attention output projection, and a_u its reference scale. Both head outputs and residual representations are read at the prediction position. The joint criteria are

$$U(x', x) = \left[\frac{1}{|\mathcal{U}|}\sum_{u\in\mathcal{U}}\frac{\|z_u(x') - z_u(x)\|_2^2}{a_u^2}\right]^{1/2}, \tag{5}$$

$$D(x', x) = \max_{\ell\in\mathcal{D}}\frac{\|r_\ell(x') - r_\ell(x)\|_2}{b_\ell}. \tag{6}$$

The shared search selects

$$x^\star \in \arg\max_{x'\in\mathcal{X}_{\mathcal{P}}(x)} U(x', x) \quad \text{subject to} \quad D(x', x) \leq \tau_x, \quad C(x'; x) = 1. \tag{7}$$

where C rejects invalid decoded characters and requires a change in target behavior. The selected text supplies perturbed activations for each tested module, allowing the component scan to reuse a single search.

Gradient-guided candidate search. To accelerate candidate search, CPAP uses gradients of the representation-level objective to prioritize promising token substitutions. We illustrate the search with a single editable position p, so $\mathcal{P} = \{p\}$; Appendix B.10 gives the multi-position extension. From K sampled candidates $x'_1, \ldots, x'_K$, we calibrate the residual bound using a tunable quantile $q \in (0, 1]$ and define the proposal objective:

$$\begin{aligned}\tau_x &= Q_q\big(\{D(x'_i, x)\}_{i=1}^K\big),\\ F_{\lambda,q}(x') &= \frac{U(x', x)}{\bar{U}} - \lambda\,[D(x', x)/\tau_x - 1]_+^2.\end{aligned} \tag{8}$$

Here Q_q is the empirical quantile, $\bar{U}$ is the median initial value of U, $\lambda > 0$ is the penalty strength, and $[t]_+ = \max(t, 0)$. The bound τ_x remains fixed throughout the search. For a current candidate x', the gradient $\mathbf{g}_p = \nabla_{e_p} F_{\lambda,q}(x')$ ranks substitutions using the first-order score

$$\widehat{\delta F}(p, v) = \mathbf{g}_p^\top\big(E(v) - E(x'_p)\big), \qquad v \in \mathcal{V}_p, \tag{9}$$

where E is the embedding table and e_p is the current embedding at position p. Following gradient-guided discrete text search (Ebrahimi et al., 2018), this score forms a shortlist of promising substitutions for the representation objective.

CPAP evaluates every proposed discrete sequence with an actual forward pass and retains all evaluated candidates. Gradients only guide proposals: the final candidate has the largest measured U among those satisfying the fixed bound and behavioral check in Equation (7). Appendix B.3 gives beam and restart settings, hyperparameters, and behavioral checks.

3 EXPERIMENTS

3.1 EXPERIMENTAL SETTING

Tasks and evaluation levels. Two common goals of mechanistic interpretability are locating causally important components (Vig et al., 2020; Meng et al., 2022) and recovering the circuits underlying a behavior (Wang et al., 2022; Conmy et al., 2023). Accordingly, we evaluate CPAP through *single-stage localization*, which traces contributing upstream components from the output, and *circuit localization*, which extends tracing across stages of a circuit—a set of components and directed connections jointly implementing a behavior (Conmy et al., 2023). Single-stage experiments cover indirect object identification (IOI), Greater-than, acronym prediction, and month-sequence continuation, with reference components from their GPT-2 analyses (Wang et al., 2022; Hanna et al., 2023; García-Carrasco et al., 2024; Lan et al., 2024). Circuit experiments cover GPT-2 IOI, sentiment, and Greater-than (Wang et al., 2022; Tigges et al., 2024; Hanna et al., 2023), including both attention heads and MLPs for Greater-than. Appendix B.1 gives task descriptions and input formats.

Models and references. For the main comparisons, we study three pretrained checkpoints: GPT-2 Small (12 layers, 12 attention heads per layer), OLMo-2-1B[1] (16 layers, 16 heads), and Qwen3-8B-Base (36 layers; grouped-query attention with 32 query and 8 key–value heads) (Radford et al., 2019; Allen Institute for AI, 2025; Qwen Team, 2025). Published GPT-2 analyses provide reference components for these tasks. For OLMo and Qwen, where such references are unavailable, we use handcrafted corrupted-input patching as the reference. Corrupted inputs follow the original studies' complete input-edit rules; task and tokenization adjustments for OLMo and Qwen are detailed in Appendix B.1.

Activation perturbation methods. We compare CPAP with handcrafted corrupted-input patching, zero ablation, Gaussian patching, Optimal Ablation (OA), and resampling ablation. To compare input-based perturbations at matched locations, CPAP and Gaussian patching use every token position changed by the original-paper corruption. CPAP searches jointly over these positions without using the handcrafted substitutions, while Gaussian applies noise to all edited positions. For Gaussian, the standard deviation σ_g is three times the standard deviation over the reference edit-token embeddings (Meng et al., 2022); we average ten fixed draws per input. OA instead learns an input-independent activation vector for each component by matching the clean output distribution (Li & Janson, 2024). Resampling draws ten independent clean inputs from a disjoint calibration pool and averages their individual intervention effects, aligning donor activations by task position without requiring an answer change. Appendix B.1 and Table 5 give task-specific edits and examples; Appendix B.5 gives fitting and resampling details.

Evaluation. The target evaluation size is 100 examples per model–task setting. To keep comparisons aligned, we match each method and its corrupted-input reference on prediction steps and prompt weights, leaving missing interventions unobserved. We then average intervention effects across prompts, first averaging Acronym prediction steps within each prompt. The resulting mean maps are compared using Spearman correlation ρ and Top-10 Jaccard similarity J, alongside recovery of published GPT-2 components. For circuit localization, tables and ROC curves use the same inputs, tracing paths, and receiver sets across all six methods; sentiment weights adjective-pair groups equally. We report area under the ROC curve (AUROC), recovery, and precision to assess these reference circuits. Appendix B.2 specifies paths, scores, and contributing sample counts; Appendix B.7 gives selection and aggregation rules, and Appendix C presents the localization heatmaps.

3.2 MAIN RESULTS: METHOD EFFECTIVENESS

Single-stage localization: CPAP best matches corrupted-input patching among all evaluated alternatives in most settings. CPAP achieves the highest Spearman correlation with corrupted-input patching in eight of twelve model–task combinations (Table 1a), exceeding zero ablation and OA in all twelve and resampling in nine. Its Top-10 Jaccard similarity ranges from 0.538 to 1.000, with exact matches on GPT-2 Acronym, OLMo Acronym, and Qwen IOI. Across Qwen's 1,152 heads, its selected sets share ten heads with handcrafted patching on IOI and nine on each of the

[1]Throughout this paper, OLMo-2-1B refers to the OLMo-2-0425-1B checkpoint.

Table 1: Single-stage localization. (a) Spearman ρ / Top-10 Jaccard J against handcrafted patching; bold/underlining mark best/second-best distinct values, including ties. Qwen3 is Qwen3-8B-Base; †: conditioned next-digit prediction. (b) GPT-2 recovery. Adjacent Greater-than maps provide comparative examples of module importance located by different methods.

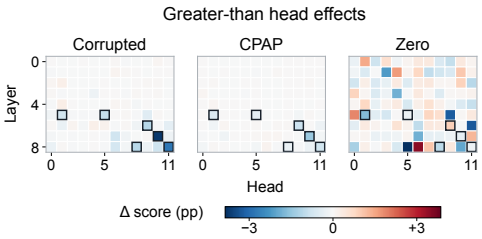
(a) Agreement with corrupted-input patching

Model	Task	CPAP	Zero	Gaussian	OA	Resampling
GPT-2	IOI	**0.536 / 0.818**	0.378 / 0.667	0.463 / **0.818**	0.506 / **0.818**	0.533 / **0.818**
GPT-2	Greater-than	**0.483 / 0.538**	0.230 / 0.250	0.354 / 0.429	0.151 / 0.429	0.049 / 0.429
GPT-2	Acronym	0.631 / **1.000**	-0.089 / 0.176	0.230 / 0.667	0.099 / 0.429	**0.688** / 0.818
GPT-2	Months	**0.500 / 0.667**	0.226 / 0.333	0.360 / 0.538	0.407 / 0.429	0.168 / 0.250
OLMo	IOI	**0.512 / 0.818**	0.292 / **0.818**	0.485 / 0.667	0.243 / 0.429	0.492 / **0.818**
OLMo	Greater-than	0.281 / 0.538	0.190 / 0.333	**0.365 / 0.667**	0.029 / 0.333	0.046 / 0.250
OLMo	Acronym	0.479 / **1.000**	0.113 / 0.250	0.416 / 0.538	0.414 / 0.818	**0.579 / 1.000**
OLMo	Months	**0.604 / 0.667**	0.356 / **0.667**	0.588 / 0.429	0.121 / 0.429	0.216 / 0.429
Qwen3	IOI	**0.646 / 1.000**	0.300 / 0.818	0.549 / 0.667	0.435 / 0.667	0.551 / 0.818
Qwen3	Greater-than†	**0.455 / 0.818**	0.314 / 0.667	0.146 / 0.111	0.105 / 0.429	0.249 / 0.667
Qwen3	Acronym	0.531 / **0.818**	0.246 / **0.818**	0.232 / 0.429	0.373 / **0.818**	**0.567 / 0.818**
Qwen3	Months	**0.583 / 0.818**	0.316 / 0.667	0.385 / 0.538	0.238 / 0.538	0.326 / **0.818**

(b) Recovery of published GPT-2 components

Method	IOI heads	Greater-than heads	Greater-than MLPs	Acronym heads	Months heads
CPAP	**5/5**	**6/6**	**4/4**	**4/4**	**2/4**
Corrupted	5/5	6/6	4/4	4/4	2/4
Zero	5/5	1/6	3/4	3/4	2/4
Gaussian	5/5	3/6	3/4	4/4	2/4
OA	5/5	5/6	3/4	3/4	1/4
Resampling	5/5	4/6	4/4	4/4	1/4

Note: Best results may be tied; only CPAP is highlighted.



other tasks. On Acronym, resampling has higher rank-agreement point estimates in all three models ($0.688/0.579/0.567$, versus CPAP $0.631/0.479/0.531$). However, a comparison on the same inputs and prediction positions gives bootstrap confidence intervals for these differences that include zero (Appendix B.5), so its observed advantage is not statistically conclusive. Gaussian also exceeds CPAP on OLMo Greater-than (0.365 versus 0.281).

Single-stage reference recovery: CPAP recovers all evaluated reference heads in IOI, Greater-than, and Acronym. CPAP and handcrafted patching recover all five IOI output heads, all six Greater-than reference heads, and all four Acronym output heads (Table 1b). Resampling likewise recovers all IOI and Acronym output references. For Greater-than, zero, Gaussian, OA, and resampling recover one, three, five, and four reference heads, respectively. At the MLP level, CPAP, handcrafted patching, and resampling recover all four reference layers (8–11); zero and Gaussian miss MLP-11, and OA misses MLP-8 because their perturbations raise the task score. Months recovery, however, is limited by the single-stage scope: later attention is frozen, without further upstream tracing of earlier heads' effects through intermediate components. Within this scope, resampling and OA recover one of four references, while the other methods recover two.

Circuit localization: CPAP achieves the highest AUROC among automatic methods on all three tasks: 0.968 on IOI, 0.983 on sentiment, and 1.000 on Greater-than. Table 2 and Figure 3 compare the six methods on GPT-2 reference circuits (Wang et al., 2022; Tigges et al., 2024; Hanna et al., 2023). Here, AUROC measures full-ranking quality, whereas recovery and precision depend on the component-selection rule. On IOI, CPAP recovers 21 of 26 reference heads, versus 25 for handcrafted patching and 22 for resampling; the latter methods have AUROCs of 0.989 and 0.938. On sentiment, CPAP recovers 8 of 9 reference heads, versus 7 for both handcrafted patching and resampling, whose AUROCs are 0.984 and 0.981. Resampling's seven selected sentiment heads are all references, yielding precision 1.000 despite its lower AUROC. On Greater-than, CPAP and handcrafted patching recover all six reference heads and four MLPs with positive mean score drops,

Table 2: Comparison of six activation perturbation methods for recovering GPT-2 reference circuits. Recovered, precision, and AUROC assess reference-component localization; ρ / J measure agreement with corrupted-input patching. Bold/underlining mark best/second-best distinct values among the five automatic methods, including ties. ρ / J are stage-averaged for IOI and sentiment; Greater-than combines heads and MLPs. Appendix B.7 gives selection rules and interval estimates.

Task	Method	Recovered	Precision	AUROC [95% CI]	ρ / J
IOI	CPAP	21/26	0.808	**0.968** [0.937, 0.981]	**0.563** / 0.631
	Corrupted	25/26	0.962	0.989 [0.977, 0.992]	1.000 / 1.000
	Zero	19/26	0.792	0.877 [0.858, 0.892]	0.314 / 0.422
	Gaussian	21/26	0.808	0.958 [0.944, 0.966]	0.418 / 0.582
	OA	18/26	0.720	0.921 [0.876, 0.957]	0.255 / 0.453
	Resampling	**22/26**	**0.846**	0.938 [0.926, 0.960]	0.452 / **0.650**
Sentiment	CPAP	**8/9**	0.889	**0.983** [0.976, 0.986]	0.762 / 0.603
	Corrupted	7/9	1.000	0.984 [0.979, 0.988]	1.000 / 1.000
	Zero	6/9	0.667	0.889 [0.859, 0.912]	0.181 / 0.381
	Gaussian	7/9	0.875	0.970 [0.961, 0.975]	0.388 / 0.603
	OA	7/9	0.875	0.980 [0.967, 0.996]	0.681 / 0.458
	Resampling	7/9	**1.000**	0.981 [0.975, 0.988]	**0.977** / **0.742**
Greater-than	CPAP	**10/10**	**1.000**	**1.000** [0.993, 1.000]	**0.561** / **0.818**
	Corrupted	10/10	0.900	0.999 [0.993, 1.000]	1.000 / 1.000
	Zero	8/10	0.300	0.685 [0.635, 0.719]	0.247 / 0.176
	Gaussian	8/10	0.400	0.749 [0.686, 0.770]	0.373 / 0.333
	OA	9/10	0.600	0.883 [0.709, 0.964]	0.169 / 0.429
	Resampling	9/10	0.700	0.927 [0.791, 0.988]	0.193 / 0.667

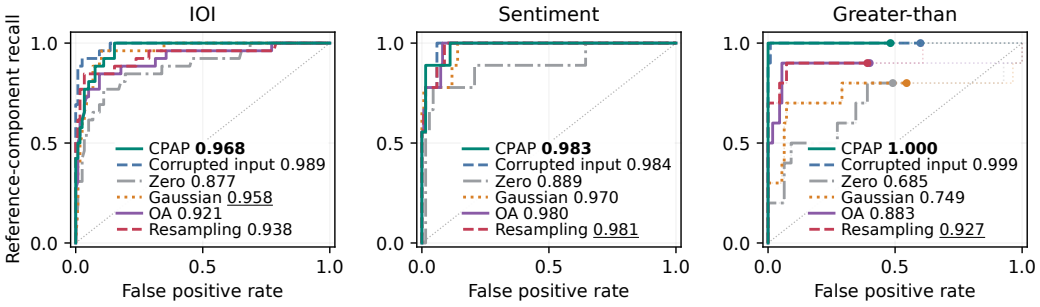


Figure 3: ROC curves for all six perturbation methods on fixed common GPT-2 inputs, matching Table 2. IOI combines six stage-wise absolute-effect ranks and sentiment combines two. Greater-than jointly ranks 108 heads and 12 MLPs by mean task-score drop; emphasized curves and endpoint markers cover positive drops, while faint dotted tails complete the full ROC. Reverse-direction references remain misses for localization. Legend AUROCs mark the best/second-best automatic values in bold/underlining. These are node rankings along specified paths; Appendix B.7 details the protocol.

versus nine references for resampling; handcrafted patching and resampling have AUROCs of 0.999 and 0.927. Restricting selection to ten components, CPAP attains precision 1.000, versus 0.900 for handcrafted patching and 0.700 for resampling, with the highest automatic-method agreement ($\rho = 0.561$, $J = 0.818$). Appendix B.7 gives the selection rules and tracing details.

3.3 SCALABILITY ANALYSIS

More complex tasks: CPAP extends to long-context and complex reasoning settings. We study dance-partner prediction after multiple swaps in BBH Tracking Shuffled Objects (Suzgun et al., 2023) and number prediction within multi-step, model-generated GSM8K solutions (Cobbe et al., 2021), with inputs spanning hundreds of tokens. Constructing x_c requires coordinated edits to repeated names or calculations while preserving context. We use OLMo-2-1B because it already answers selected examples correctly. Using 7 name and 9 GSM8K examples, we discover circuits through ACDC-style edge pruning (Conmy et al., 2023) with CPAP interventions and removal of inactive computation. We then assess faithfulness on 6 name and 9 GSM8K validation examples. We focus on attention heads, retaining all 16 MLPs because they are harder to decompose.

Table 3: Circuits found with CPAP interventions. Faithfulness is the fraction of the full model's mean target-probability gain over the CPAP background recovered by the circuit; it is not answer accuracy. Full graph: 256 heads and 1,329 connections.

Task	Heads	Connections	Faithfulness (%)	
			Discovery	Validation
Names	68	403	93.52	74.99
GSM8K (strict)	38	254	99.60	71.46

The circuits retain 74.99% and 71.46% validation faithfulness with 68 and 38 heads. To probe component roles, head interventions identify contributions to reading names and arithmetic expressions (Figure 4). Given the small, previously inspected validation sets and GSM8K's conditioning on generated reasoning, these findings remain exploratory and do not establish complete reasoning algorithms. Appendix B.12 gives settings, head analyses, and full circuits.

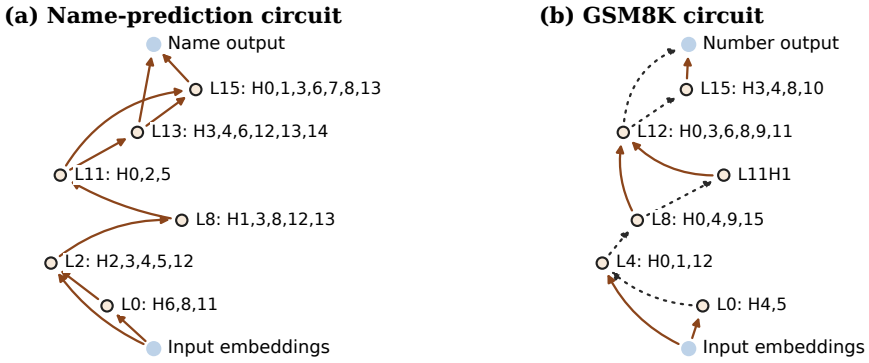


Figure 4: Simplified circuits. Nodes group the listed heads within a layer (L: layer; H: head). Brown arrows denote retained layer-level connections; dotted arrows summarize paths through hidden MLPs. All MLPs remain active; other circuit paths are omitted.

Larger models: CPAP extends to Qwen3.5-27B. To evaluate scalability in both model size and architecture, we test Qwen3.5-27B, which combines ordinary and linear attention. With the same search budget, CPAP scans 2,688 outputs and matches 10/10 and 8/10 of the corrupted-input Top-10 heads on IOI and Acronym, respectively (Table 4; settings in Appendix B.13). These results demonstrate that CPAP extends to this larger model with hybrid attention.

Table 4: Qwen3.5-27B: Spearman ρ / Top-10 Jaccard J against corrupted-input patching. Bold marks the best value per metric. Mean uses independent training activations.

Task	CPAP	Zero	Gaussian	Mean
IOI	0.464 / **1.000**	0.163 / 0.429	**0.580** / 0.818	0.247 / 0.818
Acronym	**0.377** / 0.667	0.170 / 0.333	0.367 / 0.667	0.320 / **0.818**

3.4 Ablation studies

Constraint and vocabulary ablations: overly loose downstream bounds can weaken localization. Rank agreement generally decreases as the bound is relaxed (Figure 5, left). At the default quantile, expanding the vocabulary lowers Greater-than agreement on GPT-2 but raises it on OLMo. These policy comparisons also vary absolute bounds at equal quantiles. We retain the nearest-25% policy and $q = 5\%$.

Search ablations: gradient guidance has no consistent advantage at matched budgets. Gradient and random search each lead IOI rank agreement at some budgets, but select identical Greater-than perturbations throughout (Figure 5, right). Larger budgets increase IOI's upstream change without consistently improving localization. These analyses use recorded single-position contrasts on fixed shared cohorts; Appendices B.8–B.9 report settings, curves, bounds, and paired uncertainty estimates.

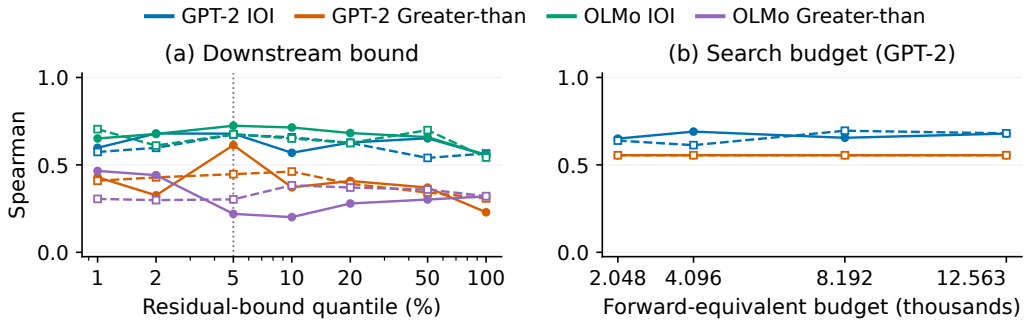


Figure 5: Rank agreement with handcrafted patching. Left: nearest-25% (solid) and uniform (dashed) policies; the vertical line marks $q = 5\%$. Right: gradient (solid) and random (dashed) search. Coincident curves have equal values.

Computational cost: CPAP constructs activation perturbations at lower cost than OA. We count evaluations per sequence or patched input–component pair, separating perturbation construction from localization. A full-budget CPAP search uses 12,180 forward and 192 backward evaluations per input, shared across components. OA fits reusable constants with 2,000–10,000 training steps per component in our records. For GPT-2 IOI, fitting all 144 heads requires $S \approx 4.8 \times 10^5$ total steps; with batch size 16 and validation, this costs about 7.9 million forwards and 7.7 million backwards, paid once. Reusing these constants over 100 inputs gives construction counts approximately $6.5\times$ and $400\times$ those of CPAP, respectively. Both methods then add 144 localization forwards per input. Actual runtime also depends on caching and batching; Appendix B.11 gives the accounting and amortization assumptions.

4 Discussion and Conclusion

We introduced Context-Preserving Activation Perturbation (CPAP), which constructs activation-patching interventions by maximizing local activation change while constraining downstream representation changes. Across controlled tasks, CPAP produces localization results closely aligned with handcrafted patching, while the name-prediction and GSM8K studies demonstrate its potential for circuit discovery without manually designed corrupted inputs. These findings suggest that representation constraints can reduce the manual effort needed to study model computations. However, residual proximity does not guarantee semantic separation, and the exploratory circuits, evaluated on small selected cohorts with previously inspected validation examples, do not establish complete reasoning mechanisms. Future work should assess generalization on larger, fresh datasets and develop more precise constraints and finer component decompositions.

AI USE STATEMENT

Generative AI tools were used to polish language and improve readability, and to help retrieve and identify relevant literature. They also supported research ideation and execution through methodological discussions and assistance with implementing and debugging experimental and analysis code. In addition, these tools assisted with preparing and refining figures and with drafting and revising manuscript sections based on the authors' research materials and recorded results. The authors are responsible for verifying all AI-assisted content against the original sources, research methods, and experimental records, and retain full responsibility for the accuracy, originality, and integrity of the final paper, figures, and code.

REPRODUCIBILITY STATEMENT

To support reproducibility, Section 2 specifies the CPAP objective and search procedure. Appendix B documents task construction, intervention paths, evaluation metrics, search settings, and baseline configurations. Appendix B.10 details the multi-position search algorithm, while Appendix B.12 describes sample selection, circuit pruning, and evaluation for the exploratory studies.

REFERENCES

Allen Institute for AI. OLMo-2-0425-1B model card, 2025. URL https://huggingface.co/allenai/OLMo-2-0425-1B.

Yushi Bai, Xin Lv, Jiajie Zhang, Hongchang Lyu, Jiankai Tang, Zhidian Huang, Zhengxiao Du, Xiao Liu, Aohan Zeng, Lei Hou, Yuxiao Dong, Jie Tang, and Juanzi Li. LongBench: A bilingual, multitask benchmark for long context understanding. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3119–3137, 2024. doi: 10.18653/v1/2024.acl-long.172. URL https://aclanthology.org/2024.acl-long.172/.

Tian Bian, Yifan Niu, Chaohao Yuan, Chengzhi Piao, Bingzhe Wu, Long-Kai Huang, Yu Rong, Tingyang Xu, Hong Cheng, and Jia Li. IBCircuit: Towards holistic circuit discovery with information bottleneck. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 4289–4302, 2025. URL https://proceedings.mlr.press/v267/bian25a.html.

Wei Chen, Zhen Huang, Liang Xie, Binbin Lin, Houqiang Li, Le Lu, Xinmei Tian, Deng Cai, Yonggang Zhang, Wenxiao Wang, Xu Shen, and Jieping Ye. From yes-men to truth-tellers: Addressing sycophancy in large language models with pinpoint tuning. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 6950–6972. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/chen24u.html.

Ziling Cheng, Meng Cao, Leila Pishdad, Yanshuai Cao, and Jackie CK Cheung. Can LLMs reason abstractly over math word problems without CoT? disentangling abstract formulation from arithmetic computation. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 14306–14333, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-332-6. doi: 10.18653/v1/2025.emnlp-main.723. URL https://aclanthology.org/2025.emnlp-main.723/.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021. URL https://arxiv.org/abs/2110.14168.

Arthur Conmy, Augustine N. Mavor-Parker, Aengus Lynch, Stefan Heimersheim, and Adrià Garriga-Alonso. Towards automated circuit discovery for mechanistic interpretability. In *Advances in Neural Information Processing Systems*, volume 36, pp. 16318–16352, 2023. doi: 10.52202/075280-0719. URL https://arxiv.org/abs/2304.14997.

Javid Ebrahimi, Anyi Rao, Daniel Lowd, and Dejing Dou. HotFlip: White-box adversarial examples for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 31–36, 2018. doi: 10.18653/v1/P18-2006. URL https://aclanthology.org/P18-2006/.

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

Nelson Elhage, Tristan Hume, Catherine Olsson, Nicholas Schiefer, Tom Henighan, Shauna Kravec, Zac Hatfield-Dodds, Robert Lasenby, Dawn Drain, Carol Chen, Roger Grosse, Sam McCandlish, Jared Kaplan, Dario Amodei, Martin Wattenberg, and Christopher Olah. Toy models of superposition. arXiv:2209.10652, 2022. URL https://arxiv.org/abs/2209.10652.

Jorge García-Carrasco, Alejandro Maté, and Juan Carlos Trujillo. How does GPT-2 predict acronyms? extracting and understanding a circuit via mechanistic interpretability. In *International Conference on Artificial Intelligence and Statistics*, volume 238 of *Proceedings of Machine Learning Research*, pp. 3322–3330, 2024. URL https://proceedings.mlr.press/v238/garcia-carrasco24a.html.

Satchel Grant, Simon Jerome Han, Alexa R. Tartaglini, and Christopher Potts. Addressing divergent representations from causal interventions on neural networks. In *International Conference on Learning Representations*, 2026. URL https://arxiv.org/abs/2511.04638.

Michael Hanna, Ollie Liu, and Alexandre Variengien. How does GPT-2 compute greater-than?: Interpreting mathematical abilities in a pre-trained language model. In *Advances in Neural Information Processing Systems*, volume 36, pp. 76033–76060, 2023. doi: 10.52202/075280-3322. URL https://arxiv.org/abs/2305.00586.

Michael Hanna, Sandro Pezzelle, and Yonatan Belinkov. Have faith in faithfulness: Going beyond circuit overlap when finding model mechanisms. In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=TZ0CCGDcuT.

Yinhan He, Wendy Zheng, Yushun Dong, Yaochen Zhu, Chen Chen, and Jundong Li. Towards global-level mechanistic interpretability: A perspective of modular circuits of large language models. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 22865–22880, 2025. URL https://proceedings.mlr.press/v267/he25x.html.

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

Michael Lan, Philip Torr, and Fazl Barez. Towards interpretable sequence continuation: Analyzing shared circuits in large language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 12576–12601. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.emnlp-main.699. URL https://aclanthology.org/2024.emnlp-main.699/.

Maximilian Li and Lucas Janson. Optimal ablation for interpretability. In *Advances in Neural Information Processing Systems*, volume 37, pp. 109233–109282, 2024. doi: 10.52202/079017-3468. URL https://arxiv.org/abs/2409.09951.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in GPT. In *Advances in Neural Information Processing Systems*, volume 35, pp. 17359–17372, 2022. doi: 10.52202/068431-1262. URL https://arxiv.org/abs/2202.05262.

Yaniv Nikankin, Anja Reusch, Aaron Mueller, and Yonatan Belinkov. Arithmetic without algorithms: Language models solve math with a bag of heuristics. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), *International Conference on Learning Representations*, volume 2025, pp. 55939–55965, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/8c5f30296296d2ae402ebbd09aaa9c12-Paper-Conference.pdf.

Chris Olah, Nick Cammarata, Ludwig Schubert, Gabriel Goh, Michael Petrov, and Shan Carter. Zoom in: An introduction to circuits. *Distill*, 2020. doi: 10.23915/distill.00024.001. URL https://distill.pub/2020/circuits/zoom-in/.

Gregory Polyakov, Christian Hepting, Carsten Eickhoff, and Seyed Ali Bahrainian. Interpretability analysis of arithmetic in-context learning in large language models. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 1758–1777, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-332-6. doi: 10.18653/v1/2025.emnlp-main.92. URL https://aclanthology.org/2025.emnlp-main.92/.

Qwen Team. Qwen3 technical report. arXiv:2505.09388, 2025. URL https://arxiv.org/abs/2505.09388.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. OpenAI technical report, 2019. URL https://cdn.openai.com/better-language-models/language_models_are_unsupervised_multitask_learners.pdf.

Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc V. Le, Ed H. Chi, Denny Zhou, and Jason Wei. Challenging BIG-bench tasks and whether chain-of-thought can solve them. In *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 13003–13051, 2023. doi: 10.18653/v1/2023.findings-acl.824. URL https://aclanthology.org/2023.findings-acl.824/.

Curt Tigges, Oskar J. Hollinsworth, Atticus Geiger, and Neel Nanda. Language models linearly represent sentiment. In *Proceedings of the 7th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*, pp. 58–87. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.blackboxnlp-1.5. URL https://aclanthology.org/2024.blackboxnlp-1.5/.

Jesse Vig, Sebastian Gehrmann, Yonatan Belinkov, Sharon Qian, Daniel Nevo, Yaron Singer, and Stuart Shieber. Investigating gender bias in language models using causal mediation analysis. In *Advances in Neural Information Processing Systems*, volume 33, pp. 12388–12401, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/92650b2e92217715fe312e6fa7b90d82-Abstract.html.

Kevin Wang, Alexandre Variengien, Arthur Conmy, Buck Shlegeris, and Jacob Steinhardt. Interpretability in the wild: a circuit for indirect object identification in GPT-2 small. arXiv:2211.00593v1, 2022. URL https://arxiv.org/abs/2211.00593v1.

Keyu Wang, Jin Li, Shu Yang, Zhuoran Zhang, and Di Wang. When truth is overridden: Uncovering the internal origins of sycophancy in large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2026.

Yiqun Wang, Chaoqun Wan, Sile Hu, Yonggang Zhang, Xiang Tian, Yaowu Chen, Xu Shen, and Jieping Ye. Tracing and dissecting how LLMs recall factual knowledge for real world questions. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 23246–23271, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.1133. URL https://aclanthology.org/2025.acl-long.1133/.

Wei Jie Yeo, Ranjan Satapathy, and Erik Cambria. Towards faithful natural language explanations: A study using activation patching in large language models. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 10425–10447. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.emnlp-main.529. URL https://aclanthology.org/2025.emnlp-main.529/.

Fred Zhang and Neel Nanda. Towards best practices of activation patching in language models: Metrics and methods. In *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2309.16042.

A Related Work

Causal localization and applications. Activation and path patching connect internal computations to model behavior, underpinning analyses of indirect object identification, numerical comparison, acronym prediction, and sequence continuation (Wang et al., 2022; Hanna et al., 2023; García-Carrasco et al., 2024; Lan et al., 2024). Applications now extend to mathematical reasoning: activation patching helps identify arithmetic circuits and heuristic neurons (Nikankin et al., 2025), investigate how problem abstraction precedes numerical computation (Cheng et al., 2025), and trace the influence of in-context examples in three-operand arithmetic (Polyakov et al., 2025). For sycophancy, path patching identifies attention heads that can be selectively fine-tuned to reduce agreement with incorrect user suggestions (Chen et al., 2024), while layer-level activation patching tests the causal role of opinion-induced representational changes (Wang et al., 2026). In factual question answering, token-level patching also helps locate answer-relevant information within generated rationales (Wang et al., 2025). Across these applications, intervention effects guide the selection of components for further analysis. Automated circuit discovery uses interventions to select computational subgraphs (Conmy et al., 2023), while gradient approximations such as EAP-IG reduce the cost of scoring candidate edges (Hanna et al., 2024). CPAP addresses the construction of activation perturbations, a choice distinct from the procedure used to scan components or approximate their intervention effects.

The choice of activation perturbation. Corrupted-input patching supplies a task-specific contrast, whereas zero, mean, and resampling ablations remove or replace information under different reference assumptions. Gaussian embedding corruption is also widely used in causal tracing (Meng et al., 2022). These choices, together with the output metric, can yield different localization conclusions (Zhang & Nanda, 2024; Heimersheim & Nanda, 2024). Optimal Ablation learns input-independent activation values that minimize the loss induced by removing input-dependent information (Li & Janson, 2024). CPAP's activation perturbations instead depend on the current input and are selected to produce substantial component-level variation under a residual-representation constraint.

Representation constraints and discrete search. Recent work shows that causal interventions can produce internal representations outside a model's natural distribution and introduces a counterfactual latent loss to control this divergence (Grant et al., 2026). CPAP shares the motivation of controlling unwanted representational changes, but uses clean-relative checkpoint distances to select input-conditioned donor activations, without estimating a latent density or asserting a manifold guarantee. Its optimization draws on gradient-guided token substitution (Ebrahimi et al., 2018); the gradient objective concerns internal features, and task behavior is checked separately. This separates the construction of an informative intervention from directly maximizing task failure.

B Implementation and Reference Details

B.1 Tasks and handcrafted corruptions

Indirect object identification (IOI). The model predicts the recipient in a sentence containing two names and a repeated subject (Wang et al., 2022). The recipient is the correct answer and the subject is the contrast answer. We use this task for single-stage localization in all three models and for multi-stage circuit localization in GPT-2 Small.

Greater-than. The model completes a year with a suffix above a stated starting-year boundary (Hanna et al., 2023). GPT-2 and OLMo score two-digit number tokens; OLMo uses an explicit century/suffix token split to retain the two-digit prediction unit. For Qwen, we supply the output tens digit and predict the units digit above the starting year's units digit, giving a conditioned next-digit variant. We exclude boundaries ending in 9, for which no larger units digit exists. The task score and the corresponding output domains are defined in Equation (10).

Acronym prediction. The model generates the initial letters of a three-word phrase (García-Carrasco et al., 2024). Each measured prediction step is conditioned on the preceding acronym letters. We resample the complete current word from the released vocabulary. When the Qwen tokenizer requires an equal-length replacement, we deterministically resample from the same word population subject to that length constraint; the edit can span multiple tokens.

Month-sequence continuation. The model predicts the next month in a sequence embedded in repeated sentence templates (Lan et al., 2024). We refer to this task as Months in the results.

Sentiment. For GPT-2 circuit localization, prompts combine an evaluative adjective and a sentiment-bearing verb before a conclusion predicting `great` or `terrible` (Tigges et al., 2024). The experiment traces the sentiment-reading and summarization paths. The designated correct and contrast answers reverse between positive and negative prompts.

Handcrafted corruptions. The handcrafted inputs follow the complete corruption rules of the original studies (Wang et al., 2022; Hanna et al., 2023; García-Carrasco et al., 2024; Lan et al., 2024; Tigges et al., 2024). IOI replaces all three name occurrences with unrelated names; Greater-than replaces the complete source-year suffix with `01`, including both digit tokens in Qwen; Acronym resamples the complete current word; Months uses the released randomized text; and sentiment uses the authors' cyclic pairing of interleaved positive and negative prompts, changing both the adjective and verb. After tokenization, CPAP and Gaussian use precisely the positions where the clean and corrupted inputs differ, with all other positions fixed. The sequence length is preserved, and no-op corruptions are excluded. Table 5 shows recorded examples.

This alignment concerns input corruption rather than every aspect of the original circuit-discovery pipelines. Qwen Greater-than retains the conditioned units-digit task, and sentiment uses 160 sentences drawn from the authors' templates and vocabularies. Months retains incidental background-name changes present in the released text. Its original pruning method used mean corrupted activations; our main comparison uses the corresponding per-input donors under the common paths, with a separate check averaging all 1,536 released corrupted inputs. The IOI backup experiment's shared ablation background is also separate from its ABC donor contrast.

Table 5: Original-paper corrupted-input rules and recorded examples. $[a \to b]$ marks each changed text span, which may contain multiple tokens. The same changed token positions are editable for CPAP and Gaussian. IOI uses the ABC contrast at every tracing stage.

Task	Corruption rule and recorded example
IOI: all stages	Replace all three name occurrences with unrelated names (ABC corruption). *Example:* Then, **[Eric → Amber]** and **[Allison → Rebecca]** were working at the hospital. **[Allison → Amy]** decided to give a computer to
Greater-than: GPT-2 / OLMo	Replace the complete starting-year suffix with 01. *Example:* The competition lasted from the year **[1362 → 1301]** to the year 13
Greater-than: Qwen	Replace the complete source-year suffix with 01, including both digit tokens; retain the conditioned output prefix. *Example:* The competition lasted from the year **[1362 → 1301]** to the year 136
Acronym	Resample the entire current word from the released word list. *Example:* The **[Gasp → Ego]** Ketchup Lid (
Months	Use the released randomized sequence and every resulting token difference, including incidental background-name edits when present. *Example:* Book lost in **[August. → December.]** Clock lost in **[September. → June.]** Nut lost in **[October. → June.]** Ice lost in **[November. → June.]** Desk lost in
Sentiment	Use the authors' cyclic pairing with opposite polarity; replace both the adjective and verb. *Example:* I thought this movie was **[adequate, → annoying,]** I **[admired → condemned]** it. Conclusion: This movie is

B.2 Intervention paths and evaluation metrics

Intervention paths and scores. For output-head scans, we patch one head at the prediction position, hold other attention outputs at their clean values, and recompute downstream feed-forward networks

(MLPs) and the residual computation. GPT-2 Greater-than instead measures head effects mediated through MLP-8: we scan layers 0–8, propagate the change through MLP-8 and later MLPs, and block the sender's direct residual route to the output. OLMo and Qwen Greater-than use output-head scans over all 16 and 36 layers, respectively. We additionally scan GPT-2 and Qwen MLP outputs directly to the logits with other component outputs fixed; Qwen MLP results appear in Appendix B.6. Layers and heads are indexed from zero. OLMo interventions recompute the attention-branch normalization after changing a head contribution.

For IOI, Acronym, Months, and sentiment, S is the logit difference between the designated correct and contrast answers. For Greater-than, with boundary y, we use

$$S_{\mathrm{GT}}(x) = 100 \left(\sum_{v>y} p(v \mid x) - \sum_{v \le y} p(v \mid x) \right), \tag{10}$$

where both sums range over two-digit number tokens for GPT-2 and OLMo. For Qwen, they instead range over the ten single-digit tokens and y is the starting year's units digit. In all cases, p comes from the full-vocabulary softmax. Effects have units of percentage points. This evaluation score is distinct from the number-restricted behavioral check used during search.

Heatmap comparison. We compare attention-head localization heatmaps with corrupted-input patching as the reference. Each heatmap contains signed mean intervention effects. For Acronym, prediction-step effects are first averaged within each original prompt, then averaged across prompts. Spearman correlation ρ compares the rankings of these effects; Jaccard similarity J compares the sets of ten heads with the smallest mean effects. Thus, $J = 1$ indicates identical selected sets, not identical effect magnitudes or per-input maps. For the current measurements, each comparison uses the same observations and prompt weights for the method and its corrupted-input reference.

Evaluation units and coverage. An evaluation example is one original prompt; prediction steps and random draws do not count as additional examples. Acronym has 215, 218, and 219 paired prediction steps across 80 prompts in GPT-2, OLMo, and Qwen, respectively, with steps averaged within each prompt. No-op corruptions are excluded; a search without an admissible perturbation leaves its step unobserved. Other methods use available observations matched to the handcrafted reference. Circuit cohorts and sentiment's group weighting are specified in Appendix B.7; fitting and calibration sizes in Appendix B.5 are separate from evaluation counts.

Reference recovery. Single-stage reference recovery uses each method's available-observation mean map and the published GPT-2 component sets in Appendix B.4. We retain the ten strongest negative head effects, with a separate positive ranking for negative name movers in IOI. For GPT-2 Greater-than MLPs, importance is the mean clean score minus the mean patched score, and only strictly positive importance counts as recovered, without a Top-k cutoff. All four reference layers, 8–11, remain in the denominator: zero or reverse-direction effects count as misses regardless of their absolute magnitude. The circuit table instead fixes a shared cohort for every method and metric, as specified in Appendix B.7. These reference sets are path-specific comparison targets, rather than exhaustive ground truth.

B.3 Search and measurement settings

CPAP joint-search configuration. For a model with L layers, $\mathcal{U}$ contains every head in layers 0 through $\lceil 2L/3 \rceil - 1$, and $\mathcal{D}$ contains post-layer residual checkpoints $\lfloor L/3 \rfloor$ through $\lceil 2L/3 \rceil - 1$. These are layers 0–7 and 4–7 for GPT-2, layers 0–10 and 5–10 for OLMo, and layers 0–23 and 12–23 for Qwen. Both feature collections are read at the prediction position. The head and residual sets overlap: "downstream" refers to the residual after component contributions have been combined, not to a disjoint set of strictly later layers. The scales in Equations (5)–(6) are frozen root-mean-square norms from the recorded reference inputs.

CPAP initializes with 1,024 sampled candidates, uses a beam of 8, retains up to 32 gradient-ranked tokens per editable position for each beam member, and runs up to 24 search rounds. The main experiments use $q = 5\%$ in Equation (8) to set the residual bound from the initialization candidates; Appendix B.8 varies this hyperparameter. The penalty strengths are $\{1, 4, 16, 64\}$. Initialization and

random sampling mix the closest 128 embedding neighbors and the full allowed neighborhood with equal probability. The beam keeps high objective values under the different penalties, the strongest feasible candidate, and the smallest-D candidate, removing duplicates. Ten percent of proposal attempts use a local random restart. An archive prevents evaluating duplicate substitutions. The per-input budget is 12,563 forward-equivalent units, counting a gradient evaluation as one forward and a backward charged at two forwards. The budget is a bookkeeping convention rather than a hardware-independent runtime claim.

For multiple edited positions, an initialization candidate is a complete token tuple, with each position sampled from its own embedding neighborhood. Gradients propose edits within that tuple, and all candidates are evaluated jointly under the same objective and frozen bound. Thus, the budget counts complete candidate sequences rather than multiplying by the number of editable positions. Appendix B.10 gives the tuple construction, gradient proposals, and final perturbation selection. The single-position derivation in Section 2 illustrates this search; the experiments use the full edit sets described above.

Fixed threshold and final selection. For K initial candidates, Q_q in Equation (8) returns the $\lceil qK \rceil$th smallest residual change without interpolation. The resulting bound remains fixed for new proposals and restarts, so q does not prescribe their acceptance fraction. Final selection uses the measured U and D from full candidate forward passes under Equation (7). Ties are resolved by smaller D and then evaluation order. If no candidate satisfies the residual bound and behavioral check, the perturbation and its intervention effect remain unobserved.

CPAP behavioral check. For IOI, Acronym, Months, and sentiment, CPAP uses the clean model's highest-scoring token in the configured output domain as the reference prediction. A candidate is admissible only if another token exceeds its logit by at least 10^{-4}. The output domain is the recorded task vocabulary, or the full vocabulary when no restriction is configured. For Greater-than, the candidate must reduce the probability difference above versus at-or-below the original boundary to at most 0.1 after normalizing over the 100 two-digit number tokens for GPT-2 and OLMo, or the ten single-digit tokens for Qwen. The boundary comes from the original prompt. This check changes a specified behavior; it does not require the candidate sentence to have a manually annotated alternative answer. The residual threshold is computed before applying the behavioral and text checks and is not relaxed when no candidate qualifies.

B.4 Reference components

The notation $\ell.h$ denotes head h in layer ℓ, with both indices starting at zero. The IOI reference uses the 26-head grouping in our recorded analysis of the arXiv preprint by Wang et al. (2022): regular name movers $\{9.9, 9.6, 10.0\}$; negative name movers $\{10.7, 11.10\}$; subject inhibition $\{7.3, 7.9, 8.6, 8.10\}$; duplicate-token heads $\{0.1, 0.10, 3.0\}$; induction heads $\{5.5, 6.9, 5.8, 5.9\}$; previous-token heads $\{2.2, 4.11\}$; and backup name movers $\{9.0, 9.7, 10.1, 10.2, 10.6, 10.10, 11.2, 11.9\}$. Fixed per-stage budgets test recovery at the reference circuit's approximate size; they do not infer how many heads a circuit should contain.

The Greater-than head reference is $\{5.1, 5.5, 6.9, 7.10, 8.8, 8.11\}$ and its MLP reference is $\{8, 9, 10, 11\}$ (Hanna et al., 2023). The Acronym output-head reference is $\{8.11, 9.9, 10.10, 11.4\}$, a subset of the full published circuit (García-Carrasco et al., 2024). The Months comparison set is $\{0.1, 4.4, 7.11, 9.1\}$ (Lan et al., 2024); its earlier heads can act through subsequent attention, which the output-head path freezes. Consequently, recovery in this scan is not a complete-circuit fidelity metric. Reference absence alone is also insufficient to classify an additionally detected component as causally irrelevant.

Sentiment reference heads. The six output references are direct readers $\{10.1, 10.4, 11.9\}$, summary readers $\{8.5, 9.10\}$, and mixed reader 9.2. The three summary-writer references are $\{7.1, 7.5, 6.4\}$, including the weaker head discussed in the text of Tigges et al. (2024). Appendix B.7 specifies how these stages are traced and combined.

B.5 Optimal Ablation fitting and resampling

For each component, OA fits an input-independent activation vector under the same intervention positions and propagation paths as the other methods. It minimizes the full-vocabulary Kullback–Leibler (KL) divergence $\mathrm{KL}(p_{\mathrm{clean}} \,\|\, p_{\mathrm{patched}})$, where p_{clean} and p_{patched} are the clean and patched output distributions. Head activations are fitted in the pre-output-projection space; MLP activations are fitted in the MLP output space. We initialize at the mean activation and optimize with AdamW, learning rate 0.002, zero weight decay, and batch size 16. Runs use float32 with TF32 disabled, at least 2,000 and at most 10,000 steps. We check a separate collection of inputs every 250 steps, stop when its best KL improves by less than 0.5% over 1,000 steps, and retain each component's best checkpoint. Model weights and reference-head labels are never optimized.

GPT-2 and OLMo single-stage tasks and the IOI circuit use 512 fitting groups and 128 groups for checkpoint selection; an acronym group contains three prediction prefixes. Qwen uses the corresponding retokenized input collections after removing groups that duplicate an evaluation sequence; IOI, Greater-than, and Months retain 512 fitting groups and 128 checkpoint-selection groups, while Acronym retains 489 and 125, respectively. Sentiment uses 128 fitting sentences and 32 sentences for checkpoint selection, with disjoint adjective-pair groups. Full token sequences are deduplicated against the recorded clean and manually corrupted evaluation sentences. Constants are fitted separately for each circuit stage. The IOI backup fit uses exactly the same frozen regular-name-mover ablation background as the other methods. KL is used only to fit the activation vectors; all six methods are evaluated using the same signed task-score change. This compares OA as a method for constructing activation perturbations under the shared localization protocol, rather than replicating its original all-position experiment (Li & Janson, 2024).

Resampling ablation. For each evaluation input we uniformly draw ten clean donor inputs with replacement from the corresponding independent calibration pool described above, using a fixed seed derived from the task and input identifier. We exclude any calibration group containing a recorded clean or corrupted evaluation sequence. Acronym donors match the current prediction step. Donors are not filtered by the model's answer or intervention effect. We replace one component at a time and average the ten resulting task-score changes, rather than averaging activations before intervention. Single-stage donors use their prediction-position activations; circuit donors align the second-subject, first-subject-plus-one, summary, or prediction positions by task role. This task-position alignment differs from the all-position padding convention in Li & Janson (2024). The same donor identities are shared across all heads, MLPs, and IOI stages for an input. The backup-name-mover stage retains the exact frozen recipient background, applying its fixed regular-name-mover intervention at each donor's prediction position.

All resampling measurements use float32 with TF32 disabled. Each task passes a reproduction of archived handcrafted-corruption maps and comparisons between reduced computation and native model hooks. Adding this baseline leaves the original five methods' maps and circuit cohorts unchanged. Table 6 compares CPAP and resampling on identical prediction steps and prompt weights. Its 2,000 paired bootstrap replicates use the existing input/group resampling scheme; intervals are conditional on the fixed ten donors and do not capture all donor-sampling uncertainty. These are pointwise intervals without a multiple-comparison adjustment. The three Acronym rank-agreement differences span zero.

B.6 Qwen Greater-than MLP localization

We also replace individual MLP outputs at the prediction position and measure their direct residual contributions to the final output, holding all other component outputs fixed. Table 7 reports agreement with handcrafted patching over all 36 MLPs in the conditioned next-digit task. MLPs are ordered by mean task-score drop. CPAP and zero have similar full-ranking correlations (0.809 and 0.808), and each shares two of the handcrafted Top-3 MLPs ($J = 0.500$); OA has the same overlap with a lower correlation of 0.575. Resampling matches the handcrafted Top-3 exactly ($J = 1.000$), despite a lower full-ranking correlation of 0.278. Each comparison uses the same 100 inputs. Top-3 is retained as a fixed-size agreement measure, distinct from reference recovery at positive importance; independently annotated Qwen reference components are unavailable.

Table 6: CPAP minus resampling on common single-stage prediction steps. Each comparison matches both methods and the handcrafted reference on valid prediction steps and prompt weights, recomputing metrics on this paired subset; differences need not equal subtraction of the method-specific results in Table 1a. Positive differences favor CPAP. Each cell reports the difference and a pointwise 95% paired bootstrap interval.

Model	Task	$\Delta\rho$ [95% CI]	ΔJ [95% CI]
GPT-2	IOI	+0.003 [-0.162, +0.157]	+0.000 [-0.280, +0.182]
GPT-2	Greater-than	+0.334 [+0.192, +0.544]	+0.110 [-0.110, +0.333]
GPT-2	Acronym	-0.040 [-0.115, +0.123]	+0.000 [-0.182, +0.333]
GPT-2	Months	+0.335 [+0.207, +0.393]	+0.417 [+0.205, +0.490]
OLMo	IOI	+0.020 [-0.117, +0.089]	+0.000 [-0.280, +0.152]
OLMo	Greater-than	+0.235 [+0.028, +0.446]	+0.288 [+0.083, +0.419]
OLMo	Acronym	-0.101 [-0.231, +0.093]	+0.000 [-0.182, +0.182]
OLMo	Months	+0.387 [+0.277, +0.452]	+0.238 [+0.128, +0.390]
Qwen3	IOI	+0.095 [+0.012, +0.132]	+0.182 [-0.182, +0.182]
Qwen3	Greater-than	+0.206 [+0.126, +0.252]	+0.152 [+0.000, +0.280]
Qwen3	Acronym	-0.053 [-0.096, +0.029]	+0.000 [-0.182, +0.152]
Qwen3	Months	+0.257 [+0.180, +0.315]	+0.000 [-0.182, +0.182]

Table 7: Qwen3-8B-Base Greater-than MLP agreement with handcrafted patching. N counts contributing prompts. Spearman correlation uses all 36 mean task-score drops; Top-3 Jaccard compares the three largest drops, all positive in these measurements. This measures agreement, not independent reference recovery.

Method	N	Spearman ρ	Top-3 Jaccard J
CPAP	100	0.809	0.500
Zero	100	0.808	0.500
Gaussian	100	-0.058	0.000
OA	100	0.575	0.500
Resampling	100	0.278	1.000

B.7 CIRCUIT LOCALIZATION PROTOCOL AND STAGE-LEVEL RECOVERY

IOI tracing paths. Following the circuit analysis of Wang et al. (2022), we trace name-mover queries, both subject-inhibition values and keys, and induction keys after the output scan. Query receivers are regular name movers $\{9.9, 9.6, 10.0\}$ at the prediction position. Both subject-inhibition paths use receivers $\{7.3, 7.9, 8.6, 8.10\}$ at the second subject occurrence: the value path targets reference senders $\{0.1, 3.0, 5.5, 6.9\}$ from the paper's main analysis, while the key path targets $\{0.10, 5.8, 5.9\}$ from its Appendix C. Induction-key receivers are $\{5.5, 6.9\}$ at the token following the first subject occurrence. A final output scan after ablating the three regular name movers exposes backup heads. Its shared template-matched background is distinct from the ABC donor contrast. These six paths, receiver sets, and backgrounds are fixed across all six methods; OA constants are fitted separately for each path, including subject-inhibition keys.

Sentiment tracing paths. Following Tigges et al. (2024), we first scan output heads at the final prediction position. We then patch sender contributions at the final occurrence of `movie` and propagate them to the value inputs of reader heads $\{8.5, 9.10, 9.2\}$ to locate summary writers. The task score uses the designated `great`/`terrible` pair, reversing the sign for negative prompts. These experiments evaluate component localization along published circuit paths; they do not autonomously discover every edge or measure whole-circuit fidelity.

Selection and recovery. The IOI budgets are 3 regular and 2 negative name movers, 4 subject-inhibition heads, 4 duplicate-token/induction heads through values and 3 through keys, 2 previous-token heads, and 8 backup heads. Each stage selects the strongest negative effects, except negative name movers, which use the strongest positive effects. Earlier selections are excluded from the backup list. These fixed budgets define our recovery statistic; they are not a Top-k criterion prescribed by the original paper. For sentiment, we select up to 6 negative output effects and 3 negative writer effects. Within each task, we take the union of the selected heads and compare it with the 26-head IOI or 9-head sentiment reference. Precision is the number of reference heads in that union divided by its size. Table 9 reports the IOI breakdown.

Node ranking and uncertainty. Both Table 2 and Figure 3 use inputs with measurements for all six methods and every path: 100 IOI inputs and 100 sentiment sentences from 38 adjective-pair groups. We rank each stage's absolute mean effects within its measured domain, normalize ranks to $[0, 1]$, and take each head's maximum across six IOI or two sentiment stages. Ties receive average ranks; heads outside a stage's domain contribute zero for that stage. Sweeping this score gives a ROC curve over all 144 heads, labeled by reference-set membership. AUROC evaluates ranking independently of the selection budgets. Spearman ρ and Top-10 Jaccard J compare stage-specific signed maps with corrupted-input patching and are averaged equally across stages. Sentiment effects are first averaged within adjective-pair groups, then equally across groups. All table metrics use these fixed cohorts; 95% AUROC intervals use 2,000 paired bootstrap resamples of inputs for IOI and groups for sentiment, with seed 20260925.

Greater-than joint head and MLP ranking. The Greater-than rows of Table 2 and third panel of Figure 3 combine 108 heads in layers 0–8, measured through MLP-8, with 12 MLP outputs patched directly to the logits. Both paths are defined in Appendix B.2. The 120 candidates contain six reference heads and four reference MLPs. Importance is $I_m = \mathbb{E}_x[S_{\text{clean}}(x) - S_{\text{patched},m}(x)]$ over the same 81 inputs shared by all methods and both paths. Both component types use the percentage-point score in Equation (10), so their scores are concatenated without type normalization. This evaluates joint node ranking along two specified paths, not complete circuit reconstruction.

Only $I_m > 0$ counts toward reference recovery. Reverse-direction references remain positive ROC labels and misses in the recovery denominator. The emphasized curves end after all positive-importance components enter; faint dotted tails complete the ROC at nonpositive thresholds. Full AUROC integrates both parts, with tied scores entering together and linear interpolation between points. It measures ranking separately from positive-direction recovery. We recompute means and scores in 2,000 paired input-bootstrap resamples (seed 20260925); intervals are reported in Table 2.

For the additional precision and agreement columns, every method uses the same joint Top-10 budget, matching the six-head/four-MLP reference size and the existing Top-10 agreement cutoff. Precision

is the fraction of references among these ten components, all of which have positive scores for every method. Spearman ρ compares all 120 scores with handcrafted patching, and Jaccard J compares the two joint Top-10 sets; these are joint metrics rather than stage averages. Component order breaks selection ties deterministically, although no observed Top-10 boundary is tied. CPAP selects all ten reference components, while handcrafted patching selects nine and shares nine selected components with CPAP ($J = 9/11$). Recovery retains its separate all-positive definition, so handcrafted patching still recovers all ten references there.

Table 8 also reports selection without the ten-component cap. Selecting every positive score admits 52–76 components, giving reference precision 0.118–0.173; resampling has the highest value under this broader rule (0.173), followed by OA (0.170). Thus the perfect CPAP precision in Table 2 concerns a fixed ten-component budget, not every component with a positive effect. Neither this aggregate ranking nor reference recovery establishes complete-circuit fidelity or perfect per-input localization.

Table 8: Greater-than selection at two fixed rules, using the same 81 inputs and joint scores as Table 2. Top-10 references determine its precision column. The remaining columns select all positive scores; P_+ is their reference count divided by set size. Positive-effect recovery remains the reference count out of ten.

Method	Top-10 references	All-positive size	References	P_+
CPAP	10/10	63	10	0.159
Corrupted	9/10	76	10	0.132
Zero	3/10	62	8	0.129
Gaussian	4/10	68	8	0.118
OA	6/10	53	9	0.170
Resampling	7/10	52	9	0.173

Table 9: IOI reference recovery across six stages on the same 100 common inputs used in Table 2 and Figure 3. Subject-inhibition V and K have separate sender references and budgets. The final two rows count distinct heads across stages.

IOI stage	CPAP	Corrupted	Zero	Gaussian	OA	Resampling
Regular / negative name movers	5/5	5/5	5/5	5/5	5/5	5/5
Subject inhibition	4/4	4/4	4/4	4/4	3/4	4/4
Duplicate-token / induction (V)	3/4	4/4	3/4	3/4	3/4	3/4
Duplicate-token / induction (K)	2/3	3/3	0/3	1/3	0/3	0/3
Previous token	1/2	2/2	0/2	1/2	0/2	2/2
Backup name movers	6/8	7/8	6/8	6/8	4/8	7/8
Reference heads in the union	21/26	25/26	19/26	21/26	18/26	22/26
Distinct selected heads	26	26	24	26	25	26

B.7.1 Full six-method comparison

Table 10 reproduces the complete six-method comparison in Table 2 for reference. All values use the same common inputs, paths, and selection rules described above. CPAP has the highest automatic-method AUROC on each task, while resampling recovers more IOI references and has higher sentiment precision and agreement with handcrafted patching.

B.8 Downstream-constraint and candidate-policy sensitivity

Searches and shared inputs. We compare the nearest-25% embedding neighborhood, with its preference for the nearest 128 tokens, against uniform sampling over all non-special vocabulary IDs except the original token. Uniform sampling also expands gradient proposals to that domain; OLMo's 74 embedding-padding rows without tokenizer entries are excluded. Both policies retain gradient guidance. These auxiliary experiments change only the second subject occurrence in IOI or the source-year token in Greater-than, rather than the complete edit sets of the main comparison. Intervention paths and observed layers follow Appendix B.2 and Appendix B.3.

Table 10: Full GPT-2 circuit comparison for all six perturbation methods, corresponding to Figure 3. IOI and sentiment use fixed stage budgets and stage-averaged ρ / J. Greater-than recovery counts positive-drop references; precision and J use the joint Top-10, and ρ uses all 120 components. AUROC intervals use 2,000 paired input/group resamples. Bold and underlining mark the best and second-best distinct values among all five automatic methods per task and metric, including ties; the Corrupted reference is excluded from these rankings.

Task	Method	Recovered	Precision	AUROC [95% CI]	ρ / J
IOI	CPAP	21/26	0.808	**0.968** [0.937, 0.981]	**0.563** / 0.631
	Corrupted	25/26	0.962	0.989 [0.977, 0.992]	1.000 / 1.000
	Zero	19/26	0.792	0.877 [0.858, 0.892]	0.314 / 0.422
	Gaussian	21/26	0.808	0.958 [0.944, 0.966]	0.418 / 0.582
	OA	18/26	0.720	0.921 [0.876, 0.957]	0.255 / 0.453
	Resampling	**22/26**	**0.846**	0.938 [0.926, 0.960]	0.452 / **0.650**
Sentiment	CPAP	**8/9**	0.889	**0.983** [0.976, 0.986]	0.762 / 0.603
	Corrupted	7/9	1.000	0.984 [0.979, 0.988]	1.000 / 1.000
	Zero	6/9	0.667	0.889 [0.859, 0.912]	0.181 / 0.381
	Gaussian	7/9	0.875	0.970 [0.961, 0.975]	0.388 / 0.603
	OA	7/9	0.875	0.980 [0.967, 0.996]	0.681 / 0.458
	Resampling	7/9	**1.000**	0.981 [0.975, 0.988]	**0.977** / **0.742**
Greater-than	CPAP	**10/10**	**1.000**	**1.000** [0.993, 1.000]	**0.561** / **0.818**
	Corrupted	10/10	0.900	0.999 [0.993, 1.000]	1.000 / 1.000
	Zero	8/10	0.300	0.685 [0.635, 0.719]	0.247 / 0.176
	Gaussian	8/10	0.400	0.749 [0.686, 0.770]	0.373 / 0.333
	OA	9/10	0.600	0.883 [0.709, 0.964]	0.169 / 0.429
	Resampling	9/10	0.700	0.927 [0.791, 0.988]	0.193 / 0.667

All seven quantiles, $q \in \{1, 2, 5, 10, 20, 50, 100\}\%$, share 1,024 initialization candidates within each policy. Each setting runs independently with 12,563 forward-equivalent units, 24 rounds, beam size eight, 32 gradient proposals per beam member, and seed 20260918. Bounds remain fixed after initialization; $q = 100\%$ is the largest initial residual change, not an unconstrained search. Initialization differs across policies, so equal quantiles need not give equal numerical bounds. Table 11 retains all settings and median bounds; Figure 5 summarizes rank agreement.

The fixed intersections across both policies and all seven thresholds contain 22 GPT-2 IOI, 43 GPT-2 Greater-than, 55 OLMo IOI, and 54 OLMo Greater-than prompts, from initial pools of 64, 89, 64, and 64. The GPT-2 IOI intersection also requires an admissible perturbation at the second edit position used in its original selection. Handcrafted maps use these same inputs; missing interventions are not assigned zero. These cohorts differ from the main comparison, so default-setting metrics can differ.

Results and uncertainty. At $q = 5\%$, uniform sampling reduces GPT-2 Greater-than agreement from $0.613/0.429$ to $0.447/0.333$ (Spearman/Top-10 Jaccard), but raises OLMo's from $0.220/0.538$ to $0.303/0.818$. Both IOI tasks retain $J = 0.818$ under either policy. Neither policy nor bound is uniformly best; the main experiments retain nearest 25% and $q = 5\%$. Because the candidate distribution also determines the bound, this is not a comparison at equal absolute constraints.

We use 2,000 paired input-bootstrap resamples shared across policies and thresholds. Table 12 gives default-setting differences. Three pointwise intervals exclude zero, all for OLMo Greater-than: Jaccard at 5% ($+0.280$, $[0.128, 0.485]$), and Spearman at 10% ($+0.182$, $[0.008, 0.242]$) and 20% ($+0.091$, $[0.0004, 0.1408]$). Other intervals include zero. Intervals condition on the recorded searches, omit optimizer-seed variation, and have no multiple-comparison adjustment; overlap with zero does not establish equivalence.

B.9 Search-budget comparison

Matched search budgets. We compare gradient and random search on GPT-2 using the single-position contrasts above and the settings in Appendix B.3. Both algorithms and all budgets share 1,024 initialization candidates and the fixed fifth-percentile residual bound. Random search uses the same mixture of nearest-128 and full-neighborhood proposals. Both algorithms select the largest

Table 11: Candidate-policy comparison on fixed shared prompts (n). Policy columns give Spearman / Top-10 Jaccard; the last column gives median residual bounds for nearest 25% / uniform.

Task (n)	q (%)	Nearest 25%	Uniform	Median τ_x
GPT-2 / IOI (22)	1	0.598 / 0.818	0.574 / 0.818	0.138 / 0.153
	2	0.680 / 0.818	0.598 / 0.818	0.141 / 0.157
	5	0.679 / 0.818	0.674 / 0.818	0.150 / 0.163
	10	0.569 / 0.818	0.658 / 0.818	0.157 / 0.170
	20	0.628 / 0.818	0.626 / 0.818	0.166 / 0.181
	50	0.652 / 1.000	0.540 / 1.000	0.197 / 0.207
	100	0.557 / 1.000	0.565 / 1.000	0.435 / 0.417
GPT-2 / Greater-than (43)	1	0.430 / 0.538	0.409 / 0.429	0.215 / 0.404
	2	0.326 / 0.429	0.429 / 0.429	0.236 / 0.491
	5	0.613 / 0.429	0.447 / 0.333	0.270 / 0.578
	10	0.372 / 0.333	0.462 / 0.429	0.360 / 0.604
	20	0.408 / 0.333	0.392 / 0.429	0.543 / 0.625
	50	0.370 / 0.333	0.342 / 0.333	0.646 / 0.659
	100	0.229 / 0.250	0.308 / 0.250	0.821 / 0.829
OLMo / IOI (55)	1	0.651 / 0.818	0.704 / 0.818	0.207 / 0.245
	2	0.677 / 0.818	0.611 / 0.818	0.220 / 0.257
	5	0.724 / 0.818	0.675 / 0.818	0.232 / 0.274
	10	0.714 / 0.818	0.651 / 0.667	0.246 / 0.287
	20	0.682 / 0.818	0.625 / 0.818	0.274 / 0.307
	50	0.659 / 0.818	0.699 / 0.818	0.337 / 0.378
	100	0.557 / 0.667	0.542 / 0.818	0.743 / 0.744
OLMo / Greater-than (54)	1	0.466 / 0.538	0.305 / 0.818	0.192 / 0.456
	2	0.441 / 0.538	0.298 / 0.818	0.209 / 0.525
	5	0.220 / 0.538	0.303 / 0.818	0.246 / 0.566
	10	0.201 / 0.538	0.383 / 0.818	0.340 / 0.605
	20	0.279 / 0.667	0.370 / 0.818	0.555 / 0.664
	50	0.302 / 0.818	0.359 / 0.818	0.710 / 0.745
	100	0.320 / 0.667	0.321 / 0.667	1.084 / 1.085

Table 12: Uniform minus nearest-25% agreement at $q = 5\%$, with pointwise 95% paired input-bootstrap intervals. Positive differences favor uniform sampling.

Task	$\Delta\rho$ [95% CI]	ΔJ [95% CI]
GPT-2 / IOI	$-0.005\,[-0.122, 0.102]$	$0.000\,[-0.182, 0.182]$
GPT-2 / Greater-than	$-0.166\,[-0.254, 0.002]$	$-0.095\,[-0.417, 0.000]$
OLMo / IOI	$-0.049\,[-0.108, 0.066]$	$0.000\,[-0.182, 0.182]$
OLMo / Greater-than	$0.083\,[-0.086, 0.216]$	$0.280\,[0.128, 0.485]$

upstream change satisfying that bound and the behavioral check. Each budget is a separate run with seed 20260918, not a checkpoint of one trajectory.

Budgets $B \in \{2{,}048, 4{,}096, 8{,}192, 12{,}563\}$ include initialization. Gradient search uses 24 rounds and 192 gradient evaluations, each charged as one forward plus one backward charged at two forwards, leaving $B - 576$ discrete candidate forwards; random search uses B candidate forwards. These are accounting units, not elapsed-time measurements (Appendix B.11).

Cohorts and results. The shared intersection across both algorithms and all budgets contains 50 IOI and 70 Greater-than prompts, from pools of 64 and 89. IOI scans output heads; Greater-than uses the head-to-MLP-8 path. Handcrafted maps use the same inputs, and missing effects remain unobserved. Table 13 retains all estimates and 95% intervals from 2,000 paired input-bootstrap resamples, conditional on the recorded searches. Figure 5 summarizes rank agreement.

Larger budgets raise IOI's mean upstream change from approximately 0.191 to 0.203 without consistently improving localization. Gradient search has higher rank agreement at 4,096 units, random search at 8,192, and their largest-budget values are nearly equal; every IOI setting selects the handcrafted Top-10 set. At the largest budget, the algorithms select identical tokens for 49/50 IOI prompts and all 70 Greater-than prompts. All eight Greater-than settings use the same tokens, already in initialization for 69 inputs, giving $\rho = 0.555$ and $J = 0.538$ throughout. Mean residual-change/bound ratios range from 0.963 to 0.974, with every candidate meeting its bound. These controls show no consistent localization advantage from gradient guidance.

Table 13: GPT-2 agreement at matched search budgets, including initialization and gradient computation. All rows use the same 50 IOI or 70 Greater-than prompts; brackets give 95% paired input-bootstrap intervals.

Task	Search	Budget	Spearman ρ [95% CI]	Top-10 J [95% CI]
IOI	Gradient	2,048	0.650 [0.520, 0.782]	1.000 [0.818, 1.000]
	Random	2,048	0.639 [0.538, 0.762]	1.000 [0.818, 1.000]
	Gradient	4,096	0.690 [0.554, 0.796]	1.000 [0.818, 1.000]
	Random	4,096	0.613 [0.518, 0.767]	1.000 [0.818, 1.000]
	Gradient	8,192	0.655 [0.533, 0.783]	1.000 [0.818, 1.000]
	Random	8,192	0.695 [0.562, 0.808]	1.000 [0.818, 1.000]
	Gradient	12,563	0.679 [0.566, 0.803]	1.000 [0.818, 1.000]
	Random	12,563	0.680 [0.567, 0.803]	1.000 [0.818, 1.000]
Greater-than	Gradient	2,048	0.555 [0.437, 0.639]	0.538 [0.429, 0.667]
	Random	2,048	0.555 [0.437, 0.639]	0.538 [0.429, 0.667]
	Gradient	4,096	0.555 [0.437, 0.639]	0.538 [0.429, 0.667]
	Random	4,096	0.555 [0.437, 0.639]	0.538 [0.429, 0.667]
	Gradient	8,192	0.555 [0.437, 0.639]	0.538 [0.429, 0.667]
	Random	8,192	0.555 [0.437, 0.639]	0.538 [0.429, 0.667]
	Gradient	12,563	0.555 [0.437, 0.639]	0.538 [0.429, 0.667]
	Random	12,563	0.555 [0.437, 0.639]	0.538 [0.429, 0.667]

B.10 Multi-position search for activation perturbations

The single-position search in Section 2.3 extends to multiple editable positions by representing each candidate as a complete tuple of replacement tokens. The main experiments use this joint search with the edit sets in Appendix B.1.

Joint candidate space. Let $\mathcal{P} = \{p_1, \ldots, p_d\}$ contain the d editable positions. In the experimental implementation, every selected position must change, so we remove the original token from each neighborhood: $\mathcal{V}_{p_j}^- = \mathcal{V}_{p_j} \setminus \{x_{p_j}\}$. A tuple $\mathbf{v} = (v_1, \ldots, v_d)$ defines a complete candidate sequence by

$$\mathbf{v} \in \prod_{j=1}^{d} \mathcal{V}_{p_j}^-, \qquad x_i^{\mathbf{v}} = \begin{cases} v_j, & i = p_j, \\ x_i, & i \notin \mathcal{P}. \end{cases} \tag{11}$$

All other token IDs and the sequence length remain fixed. This specializes Equation (4) by requiring every selected position to change.

Joint calibration and objective. We initialize with K distinct tuples $\{\mathbf{v}_i^{(0)}\}_{i=1}^{K}$. For each coordinate, sampling chooses with equal probability between the closest 128 available neighbors and the full allowed neighborhood, then draws uniformly from that set. The neighborhoods remain anchored to the original tokens. We evaluate U and D from Equations (5)–(6) on each complete sequence, and set

$$\begin{aligned} D_i^{(0)} &= D(x^{\mathbf{v}_i^{(0)}}, x), \qquad \tau_x = D^{(0)}_{(\lceil qK \rceil)}, \\ \bar{U} &= \mathrm{median}_{i=1,\dots,K}\, U(x^{\mathbf{v}_i^{(0)}}, x), \\ F^{\mathcal{P}}_{\lambda,q}(\mathbf{v}) &= \frac{U(x^{\mathbf{v}}, x)}{\bar{U}} - \lambda\,[D(x^{\mathbf{v}}, x)/\tau_x - 1]_+^2 . \end{aligned} \tag{12}$$

Here $D^{(0)}_{(k)}$ is the kth smallest initial residual change, so the empirical quantile uses an order statistic without interpolation. The implementation floors $\bar{U}$ at 10^{-8} for numerical stability. Both the calibration and the constraint concern the combined edit, with one fixed bound τ_x for the whole sequence. The main runs use $K = 1{,}024$ and $q = 0.05$; q remains a tunable hyperparameter.

Gradients at multiple positions. For a current beam tuple $\mathbf{v}$, one backward pass gives gradients at all editable positions. With $F_{\lambda,q}$ from Equation (8), the gradient and token shortlist at position p_j are

$$\begin{aligned} \mathbf{g}_j(\mathbf{v}) &= \nabla_{e_{p_j}} F_{\lambda,q}(x')\Big|_{x'=x^{\mathbf{v}}}, \\ \mathcal{L}_j(\mathbf{v}) &= \operatorname*{Top}_{r_j,\; u \in \mathcal{V}^-_{p_j}} \mathbf{g}_j(\mathbf{v})^\top \big(E(u) - E(v_j)\big), \qquad r_j = \min(32, |\mathcal{V}^-_{p_j}|). \end{aligned} \tag{13}$$

The operator Top_{r_j} returns the r_j token IDs with largest scores. Each gradient is evaluated in the context of the current replacements at all other positions. Penalties $\lambda \in \{1, 4, 16, 64\}$ cycle across beam members. As in the single-position case, subtracting the current-token term does not change the shortlist ranking.

Coupled proposals and beam updates. We choose a beam tuple uniformly, select a nonempty set of coordinates $S \subseteq \{1, \dots, d\}$, and sample a token uniformly from each selected coordinate's shortlist:

$$v'_j = \begin{cases} u_j, & u_j \sim \mathrm{Uniform}(\mathcal{L}_j(\mathbf{v})), \quad j \in S, \\ v_j, & j \notin S. \end{cases} \tag{14}$$

For these proposals, $|S| = 1$ with probability 0.5; otherwise it is uniform over $2, \dots, \min(4, d)$, with coordinates sampled without replacement. When $d = 1$, the sole coordinate is selected. Ten percent of all proposal attempts instead restart the entire tuple by sampling each coordinate from its closest 128 available neighbors. Duplicate tuples are skipped; any remaining proposal slots are filled using the initialization mixture. Every new proposal receives a complete forward evaluation of its joint U, D, and text and behavioral checks, rather than a sum of separate single-position effects.

The beam retains eight tuples using the multiple penalties, the largest-U candidate satisfying $D \le \tau_x$, and the smallest-D candidate, as described in Appendix B.3. Beam ranking normalizes U by the current archive median, while the gradient objective in Equation (12) keeps the initial scale fixed. The bound τ_x remains fixed in both. We run up to 24 rounds with a total budget of 12,563 forward-equivalent units per prediction step. Each complete candidate evaluation costs one unit, and a gradient evaluation for one beam tuple costs three; gradients at several editable positions share that evaluation.

Selecting the perturbed activations. Let $\mathcal{A}$ be the archive of all evaluated tuples, including initialization and restarts. We choose the final tuple using the exact joint measurements:

$$\begin{aligned} \mathbf{v}^\star &\in \operatorname*{arg\,max}_{\substack{\mathbf{v} \in \mathcal{A}:\; D(x^{\mathbf{v}}, x) \le \tau_x, \\ C(x^{\mathbf{v}}; x) = 1}} U(x^{\mathbf{v}}, x), \\ x^\star &= x^{\mathbf{v}^\star}, \qquad \widetilde{h}_m = h_m(x^\star). \end{aligned} \tag{15}$$

Here C is the text-validity and behavioral predicate in Equation (7) and Appendix B.3. Ties are resolved by smaller D, then evaluation order; if no tuple qualifies, the perturbation and its intervention effect remain unobserved. The multiple token edits construct one donor computation. Patching then uses its activation $\widetilde{h}_m$ at the component and position specified by the intervention path, reusing the selected donor across the component scan.

B.11 Forward and backward evaluation counts

Counting unit. We separate perturbation construction from scanning M components at one prediction step along one intervention path. Counts are per sequence or patched input–component pair, even within a batch. Shared clean-input capture, reference-scale and prefix preparation, model loading, file operations, and correctness checks are excluded; method-specific calibration captures are included once under caching. These are logical evaluations, not full-model passes or accelerator launches. Every component scan uses zero backward evaluations.

Per-input construction and scanning. CPAP's 24 rounds with beam size eight require 192 gradient forwards and backwards. Charging each backward as two forwards leaves $12{,}563 - 3 \times 192 = 11{,}987$ candidate forwards, including initialization. Capturing the selected donor gives 12,180 construction forwards and 192 backwards, shared across components, followed by M localization forwards. An unsuccessful search has no final donor capture or scan; each Acronym step requires a separate search. These counts describe the configured full budget, verified against 895 historical single-position attempts.

Zero requires no donor capture; handcrafted corruption requires one. Gaussian requires ten captures and $10M$ interventions. Resampling also uses $10M$ interventions, with at most ten uncached donor captures. Averaging follows intervention. Zero, handcrafted, Gaussian, resampling, and Mean construction need no backward evaluations. Additional stages and frozen backgrounds incur their own applicable counts.

Reusable construction. Mean averages C cached calibration sequences. For OA, C includes training and validation captures, S sums optimizer steps over all components, and V counts validation input–component evaluations. Batch size 16 gives $C + 16S + V$ construction forwards and $16S$ backwards, then M forwards per evaluation input. All heads count separately even in batched layer fits. Historical fits use 2,000–10,000 steps per component, with validation initially and every 250 steps; Mean and OA may use different calibration sizes.

A 100-input cost estimate. For GPT-2 IOI, the 12 saved head-layer fits cover all 144 heads and give $S = 480{,}000$ component-wise optimizer steps, with $C = 640$ calibration captures and $V = 264{,}192$ validation evaluations. OA construction therefore requires 7,944,832 forwards and 7,680,000 backwards, paid once. For 100 inputs, each with one prediction step and a full-budget search, CPAP construction requires 1,218,000 forwards and 19,200 backwards. OA's construction counts are thus approximately $6.5\times$ and $400\times$ larger, respectively, after reusing the fitted constants across all 100 inputs. Both methods add 14,400 localization forwards, leaving the total-forward ratio approximately $6.5\times$. This is a normalized estimate from recorded OA fits and the configured CPAP budget, rather than a new 100-input timing experiment. CPAP search cost grows with the number of prediction steps; OA construction is amortized over them, so these ratios decrease for larger evaluation sets.

Interpretation. CPAP candidate scoring reuses unchanged prefixes, while gradients process full sequences. OA differentiates cached intervention paths with respect to activation constants; localization also reuses clean activations. Sequence lengths, recomputed layers, and gradient scope differ, so these counts do not imply FLOP or runtime ratios.

B.12 Circuit discovery without handcrafted corruptions

B.12.1 Data and intervention construction

Name prediction. We use the dance-partner examples within BBH's three-object Tracking Shuffled Objects task (Suzgun et al., 2023). Each narrative specifies initial pairings and three swaps. The

model outputs the full partner name; option texts remain in the prompt, but letter labels are removed. Three fixed demonstrations are supplied. An earlier screen excluded demonstrations and previously inspected examples, then used a fixed hash ordering to select 32 correctly answered examples per split from 239 remaining questions. Among pairs with successful CPAP interventions and a changed first-answer-token class, the dance-partner subset contains seven discovery and six historical validation examples. This selected subset is not a random sample of all BBH problems. Input lengths range from 490 to 496 tokens.

Number prediction. For GSM8K (Cobbe et al., 2021), an earlier screen found 75 correct final numerical answers among 256 questions and selected 12 examples per split in a fixed order. We retain the 18 pairs with admissible CPAP interventions and changed first-number-token classes, preserving the nine/nine discovery/validation split. Prompts include three worked demonstrations and the model's own generated reasoning. The observation is immediately before the first occurrence of a number matching the final answer, which need not be the final calculation. Lengths range from 406 to 578 tokens. The local expressions comprise six subtractions, two multiplications, and one division in discovery; validation has one subtraction, two multiplications, two divisions, and four additions. These conditional continuations do not evaluate end-to-end mathematical solving.

Searching for an informative token position. We screen editable positions for perturbations that spread the final next-token distribution while preserving its likely candidates. Fixed instructions and demonstrations are protected. At each non-special position with eligible neighbors, we draw eight probes from up to 32 cosine-nearest vocabulary embeddings matching its generic form (capitalization, digit count, and whitespace or punctuation pattern). Each probe changes only that position's embedding to $0.25e_p + 0.75e_v$, where v is the sampled neighbor.

At the final prediction, π and π' denote clean and probed distributions renormalized within their respective top-ten token sets. Let R be the fraction of ten candidates shared by these sets, and W the shared tokens' total probability under π. An accepted probe scores

$$R\,W\,\frac{[H_2(\pi') - H_2(\pi)]_+}{\log 10}, \qquad H_2(\pi) = -\log \sum_v \pi(v)^2, \tag{16}$$

where $[z]_+ = \max(z, 0)$. Collision entropy H_2 increases as probability spreads; R and W penalize changes to the candidate set.

A local check requires the probed top-ten set to retain at least 0.8 of the clean, locally normalized top-ten mass at positions p and $p+1$ preceding the final prediction; absent control positions impose no check. Rejected probes score zero. We average all eight scores and rank positions in descending order, breaking ties by earlier position. Neither reference answers nor attention or head effects enter this ranking; the selected location need not be a semantic evidence token.

CPAP inputs and settings. We rank editable positions using the probe score above, then search the top one, two, three, and four positions under the same frozen ranking. Each stage has a budget of 12,563 forward-equivalent units. The residual bound is fixed from the fifth percentile of the first stage's 1,024 initial candidates and is reused at later stages. The upstream heads and residual checkpoints use the OLMo layer ranges in Appendix B.3; normalization uses the clean input's RMS activation norms. Only the current narrative is editable for names, protecting demonstrations, the final question, options, and output prefix. For GSM8K, only the generated reasoning prefix is editable. Every original/CPAP pair has equal token counts, although lengths differ across examples. A full forward pass verifies the residual constraint and behavior change. Name interventions must generate a different complete original option; GSM8K interventions must meet the recorded numeric-output checks. These checks do not guarantee grammatical text or a valid alternative reasoning chain.

B.12.2 Graph, pruning, and evaluation

Native computation graph. Our ACDC-style implementation (Conmy et al., 2023) preserves OLMo's native projections and normalization. The candidate graph contains 256 head-output connections and 1,073 residual connections, totaling 1,329. Head outputs feed a native attention-output aggregate for each layer. Embeddings and normalized attention/MLP outputs supply layer-wide query, key, and value inputs, MLP inputs, and the final residual readout. Since Q/K normalization

spans heads, residual connections are defined at the layer-input level; they are not independently identified head-to-head connections. Each connection bundles all sequence positions. Retained sources are recomputed in the current circuit; removed sources use cached activations from the fixed paired CPAP input. MLPs are intact blocks, with no neuron- or feature-level decomposition. All 16 remain in both main-text circuits; the further-pruned name variant retains 15.

Pruning criteria. Name discovery begins with the full graph and visits edges in reverse topological order. A deletion is accepted when its increase in mean full-vocabulary KL divergence from the clean model is below 10^{-3}. Structural cleanup removes computation disconnected from the output or transmitting only fixed background values, with numerical equivalence checked afterward. This yields the 68-head, 403-connection circuit. Further pruning alternates discovery-ranked module deletion and edge deletion until a complete pass accepts nothing. It uses the same local KL threshold, a cumulative KL ceiling of 0.02493924, and mean answer probability at least 0.49456348 (one percentage point below the initial circuit), yielding 29 heads and 171 connections.

GSM8K starts from the full graph, applies reverse-topological edge pruning, and alternates module/edge deletion until a no-deletion pass. The strict configuration uses a local KL increment below 10^{-4}, mean KL at most 0.005, mean target-number probability loss at most 0.01, and per-example loss at most 0.02 relative to the full model. The compact configuration uses 10^{-3}, 0.02, 0.02, and 0.05, respectively. Each discovery number token and its following >> boundary token must remain the most probable continuation. This boundary check prevents a continuation such as `3200` from being counted as a correct prediction of `320`. Both configurations were rerun from the full graph after this discovery-side correction, with thresholds unchanged. Validation examples were excluded from all pruning decisions, but had been inspected in earlier experiments; they constitute historical validation, not a fresh confirmatory test set.

Probability-based faithfulness. Let $P_C^{(i)}$ be the product of the conditional probabilities of every target name or number token for example i under circuit C, with target prefixes supplied. $P_{\text{full}}^{(i)}$ uses the unmodified model, and $P_{\text{bg}}^{(i)}$ uses the all-background graph, numerically equivalent to the paired CPAP forward pass. We report

$$F(C) = \frac{\overline{P}_C - \overline{P}_{\text{bg}}}{\overline{P}_{\text{full}} - \overline{P}_{\text{bg}}}, \qquad \overline{P}_C = \frac{1}{n}\sum_{i=1}^{n} P_C^{(i)}. \tag{17}$$

Thus, 100% recovers the full model's probability gain over the background and 0% matches the background. We take a ratio of means, without clipping, rather than averaging per-example ratios. Probabilities exclude answer terminators; the GSM8K termination check is a separate generation criterion. Faithfulness is neither generation accuracy nor the IOI logit-difference metric. KL uses natural logarithms and averages target positions within each example before averaging examples. Table 14 includes both pruning variants; intervals use 10,000 paired case bootstrap samples with seed 20260926 and do not include search uncertainty.

Table 14: All four frozen circuits. F_{disc} and F_{val} are discovery and historical-validation faithfulness. Brackets give approximate 95% bootstrap intervals for F_{val}. Connection counts refer to the fixed 1,329-connection graph.

Circuit	Heads	MLPs	Edges	F_{disc} (%)	F_{val} (%) [95% CI]	KL ↓
Names	68	16	403	93.52	74.99 [32.20, 94.77]	0.0220
Names (compact)	29	15	171	90.25	41.69 [-11.04, 69.43]	0.0541
GSM8K (strict)	38	16	254	99.60	71.46 [42.26, 98.39]	1.3467
GSM8K (compact)	23	16	156	97.82	63.19 [31.98, 89.95]	1.3971

Further compression reduces name validation faithfulness from 74.99% to 41.69% and GSM8K faithfulness from 71.46% to 63.19%. Actual complete-number generation is correct on 7/9 strict and 5/9 compact GSM8K validation examples, versus 9/9 for the full model; both circuits achieve 9/9 on discovery. The compact name circuit keeps every correct name token as the next-token argmax under supplied target prefixes in only 2/6 validation examples; this check excludes termination. These results expose a substantial size–faithfulness tradeoff. A no-deletion pass establishes only local irreducibility under the chosen order and discovery criteria, not global minimality or completeness.

Numerical checks and controls. Full and all-background graph endpoints match the corresponding model forward passes. Mixed graphs and final circuits were checked against independent hook implementations. Twenty random masks preserve receiver indegrees and head counts per layer before structural cleanup; effective connection counts can therefore differ. For compact names, validation target probability is 40.87%, compared with 31.03% for the background and 30.96% for the random-mask mean. For compact GSM8K, these values are 67.47%, 12.01%, and 12.04%. These are descriptive controls, not significance tests or strictly size-matched effective circuits.

B.12.3 Attention-head analysis and circuit views

Head roles supported by attention and intervention. We examine attention at the first target-token prediction and replace one retained head's output by its paired CPAP activation, recomputing its downstream computation. Table 15 reports validation means for the compact circuits, where these follow-up measurements were made. Name heads 13.12 and 15.13 attend to partner-name occurrences and have positive probability-drop effects, supporting a contribution to name reading. GSM8K heads 11.1, 12.3, and 12.9 focus on the current expression, while 15.10 concentrates on the prediction position within that expression. On the `2*9=` example, head 12.3 assigns 74.9% and 10.8% attention to `2` and `9`; head 15.10 assigns 69.9% to the final `=`. This supports context reading and downstream use, without identifying an arithmetic rule implemented by a head. Effects are conditional on the given circuit and are not additive.

Table 15: Head measurements in the compact variants. Attention is measured in the circuit; probability drop is the circuit's joint target probability minus the probability after intervening on the named head, in percentage points (pp). Indices start at zero.

Task	Head	Attended region	Attention (%)	Probability drop (pp)
Names	13.12	Partner names	64.54	2.19
Names	15.13	Partner names	39.86	0.68
GSM8K	11.1	Current expression	68.34	13.63
GSM8K	12.3	Current expression	74.55	10.87
GSM8K	12.9	Current expression	68.96	15.58
GSM8K	15.10	Prediction position	67.24	8.57

The original name model answers 10/13 examples after removing options and 11/13 after reordering them, but fails all 13 controls that change the final swap and hence the correct partner. Accordingly, the circuits explain the observed name outputs, not a verified three-swap tracking algorithm. GSM8K interventions sometimes produce malformed expressions, and the analyzed number can occur before the final calculation. These limits apply to semantic interpretation even when output faithfulness is high.

Reading the diagrams. Figure 4 shows selected paths between layer groups; listed head indices do not identify separate head-to-head edges. Brown arrows are retained connections and black dotted arrows summarize verified paths through hidden MLPs. Figures 6–7 show all retained heads. Their solid arrows include all direct connections between visible layer groups, while dashed arrows summarize selected MLP-mediated paths preserving visible-node reachability. MLPs remain active in every evaluated circuit. Colors identify receiving Q/K/V inputs or the readout; widths do not measure causal strength, and the display does not preserve every causal pathway.

Additional single-input observations. Two earlier OLMo cases were retrospectively selected for interpretable attention and intervention evidence from a screen of 65 inputs; they do not estimate dataset-level performance. In LongBench MultiFieldQA-English record 0 (Bai et al., 2024), the uncropped passage and question contain 1,511 tokens. Using the position ranking above and a single-position CPAP search, head 13.12 reads the answer phrase and its intervention lowers the fixed `South`-versus-`Ur` logit margin by 1.608; zeroing it instead raises the margin by 0.454. The search edits `Answer` to `expectedResult`; protecting that prompt marker leaves no admissible donor at the newly selected position under the same budget. This is evidence of answer reading and prompt sensitivity, not indispensability or reliable semantic-position selection.

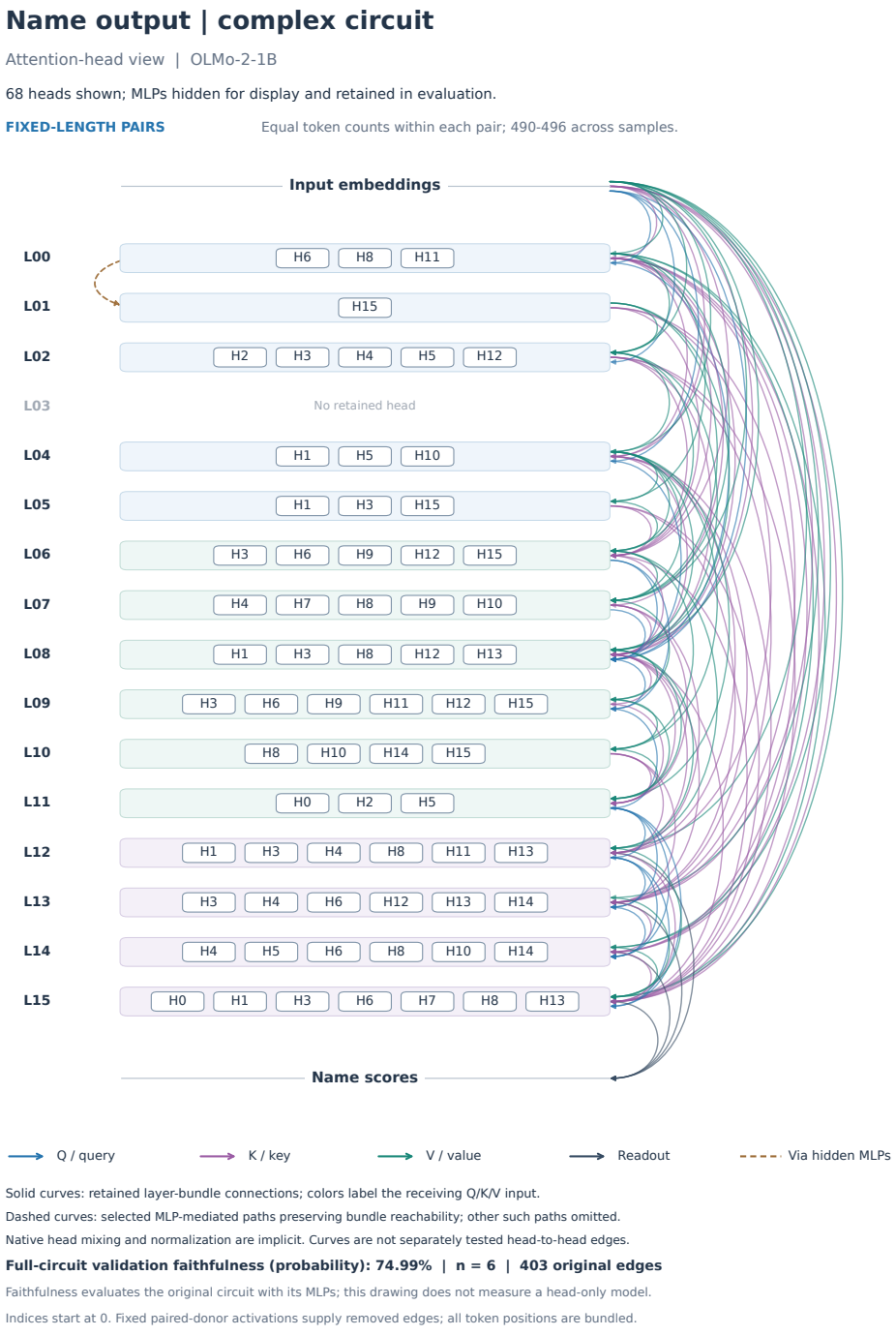


Figure 6: Full attention-head view of the 68-head name circuit. All 16 MLPs remain active but are hidden. Lines connect native layer groups; the reported faithfulness evaluates the complete circuit with MLPs.

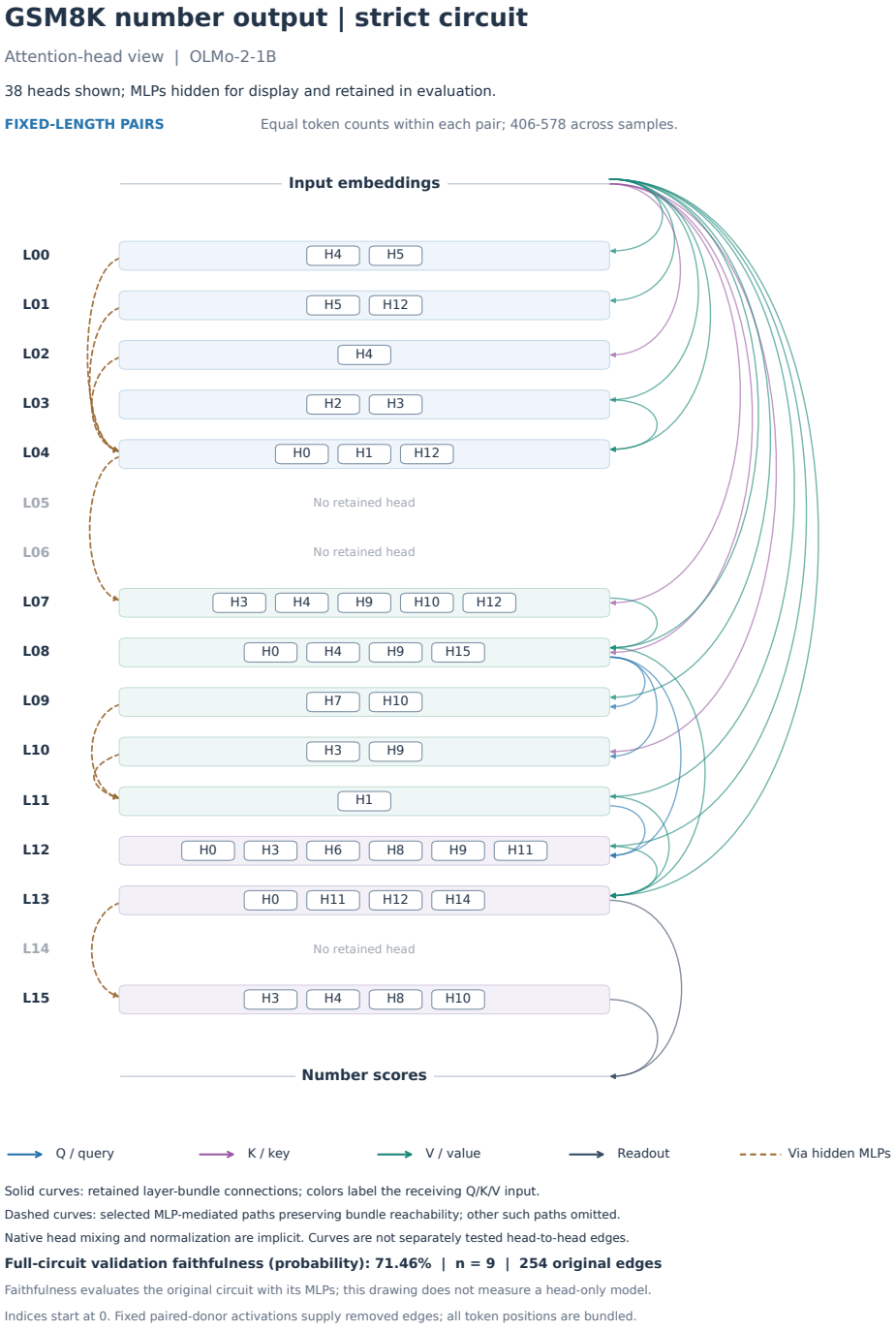


Figure 7: Full attention-head view of the 38-head GSM8K circuit under the strict pruning criteria. All 16 MLPs remain active but are hidden. The task is number continuation from a generated reasoning prefix.

In GSM8K test record 6 (Cobbe et al., 2021), the 448-token prefix contains model-generated reasoning after three worked demonstrations. Head 13.4 attends 51.9% and 25.2% to two earlier `260` tokens. Its intervention lowers the fixed `260`-versus-`26` margin by 0.615 but leaves answer probability above 99.9%; a separate probe before the first computed answer yields no clear arithmetic mechanism. Both scans fix other attention outputs and recompute downstream MLPs. They support reading previously available answers, rather than a complete reasoning algorithm. Search uses the main budget and threshold with per-input RMS normalization; a candidate must change the clean full-vocabulary winning token.

B.13 EXTENSION TO QWEN3.5-27B

Model and paths. The text branch of Qwen3.5-27B has 26.896 billion loaded parameters and 64 layers: 16 ordinary-attention layers with 24 output heads and 48 linear-attention layers with 48, totaling 2,688 modules. We intervene after each native gate and before the output projection at the prediction position, fix other attention or linear-mixing outputs, and recompute downstream MLPs and residual states. Padded coordinates are excluded. Runs use FP32 with TF32 disabled, split over two PPU-ZW810 devices; native-path and finite-difference gradient checks cover both layer types.

Data, search, and comparison. Table 4 uses 100 IOI prompts and 100 eligible Acronym steps from 58 prompts, with measurements for all five methods. These runs use the earlier single-position contrast: the second subject occurrence in IOI, or the current word's initial token in Acronym when an equal-length one-token alternative exists. Search keeps the 12,563-unit budget, 1,024 initialization candidates, fifth-percentile bound, and 24 rounds with beam size eight; upstream layers are 0–42 and residual checkpoints 21–42. Unlike Table 1, this experiment uses single-position edits and a checkpoint with different architecture and post-training, so it does not isolate parameter count.

Zero and ten-draw Gaussian use the same edit protocol. Mean averages independent clean training activations using FP64 accumulation and FP32 storage, without OA optimization; evaluation inputs are excluded. Acronym effects are averaged within each original prompt before averaging prompts. All methods use the same prompts; Spearman compares all 2,688 signed mean effects and Jaccard the ten most negative. Missing results are not imputed as zero. The 10/10 and 8/10 overlaps measure agreement with handcrafted patching, not independently annotated circuit recovery.

C Representative Localization Heatmaps

These maps illustrate GPT-2 internal-path tracing and OLMo/Qwen single-stage localization, complementing the GPT-2 maps beside Table 1. Table 1 and Table 10 retain the complete quantitative comparisons. Single-stage maps use each method's available observations, averaging Acronym steps within original prompts; circuit maps use the common inputs in Appendix B.7. Each row shares a color scale across all six methods. Blue indicates a task-score decrease, red an increase, and gray an unmeasured causal domain. Black boxes mark reference heads expected to reduce the score; gold boxes mark negative IOI name movers. OLMo and Qwen have no independent reference-head annotations.

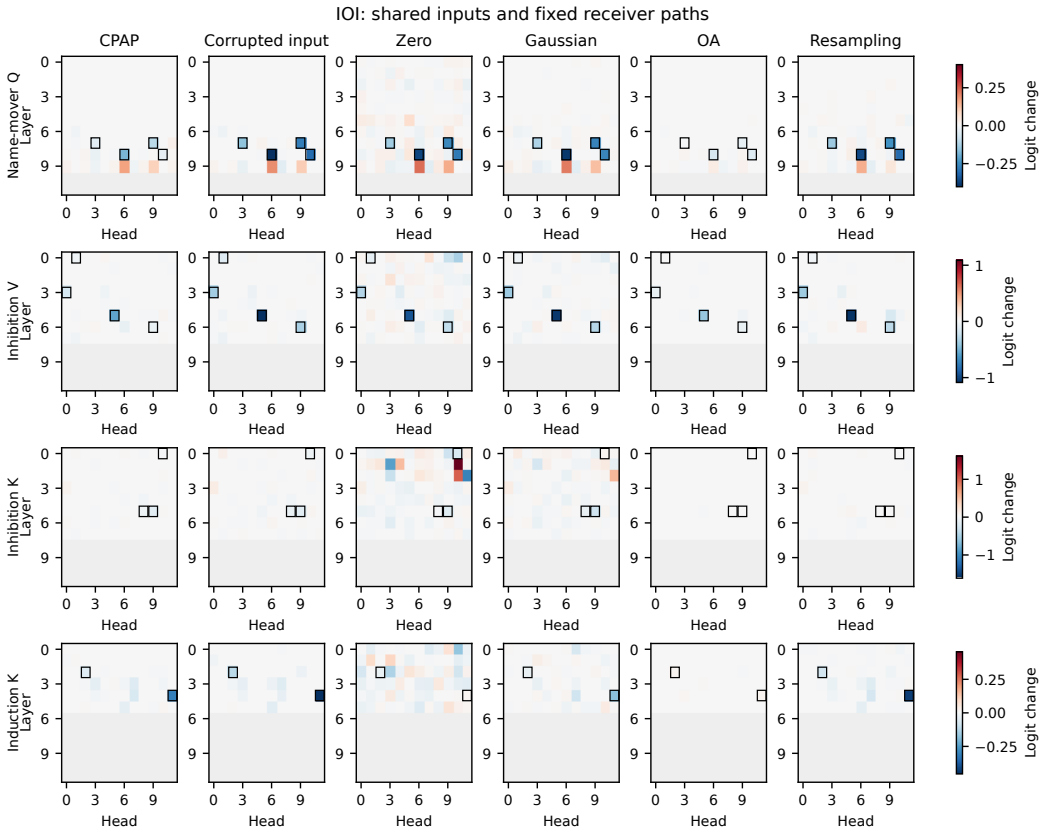


Figure 8: GPT-2 IOI internal paths on 100 common inputs. Rows target name-mover queries (Q), subject-inhibition values (V), subject-inhibition keys (K), and induction keys. Receivers are fixed across methods; the V and K paths have separate sender references.

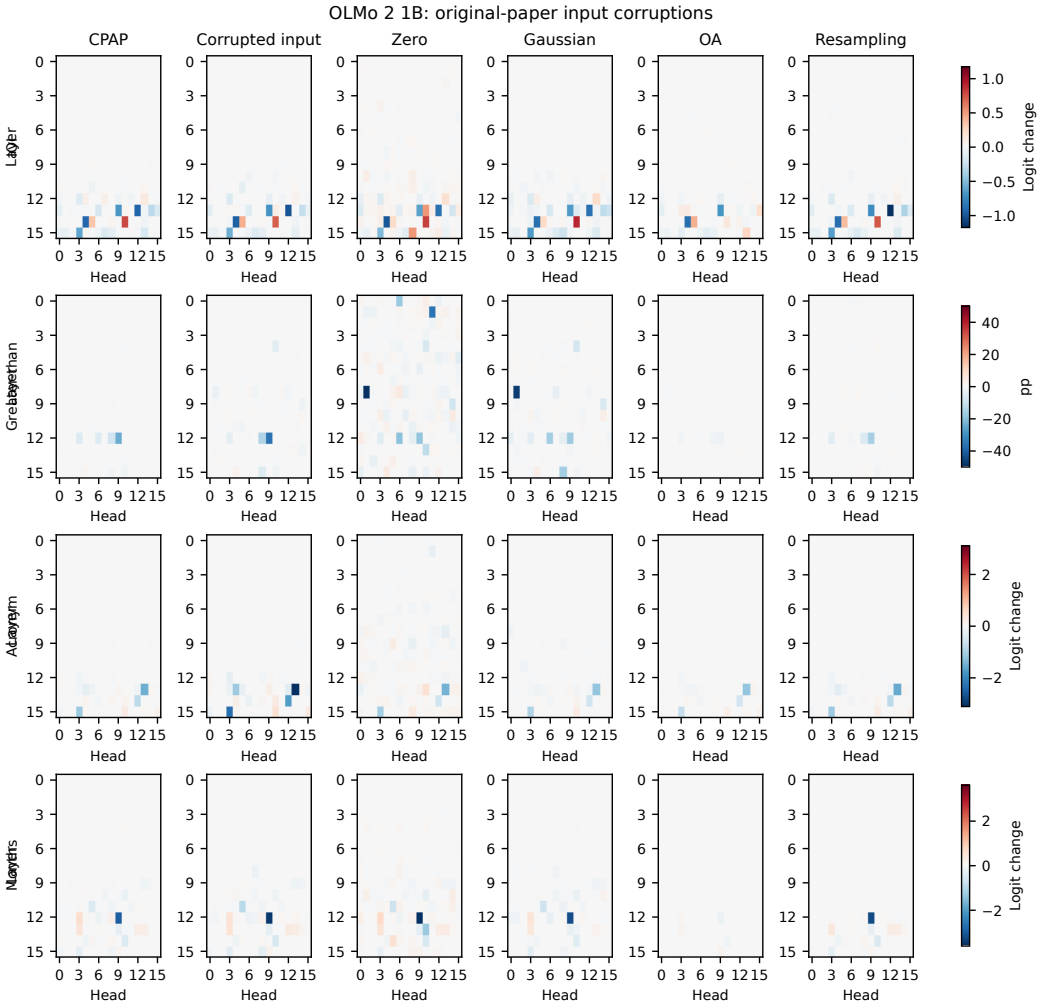


Figure 9: OLMo-2-1B output-head effects for all four single-stage tasks, scanning all 16 layers. CPAP uses all recorded admissible perturbations. Table 1 matches each method and its handcrafted reference on the same observations before computing agreement.

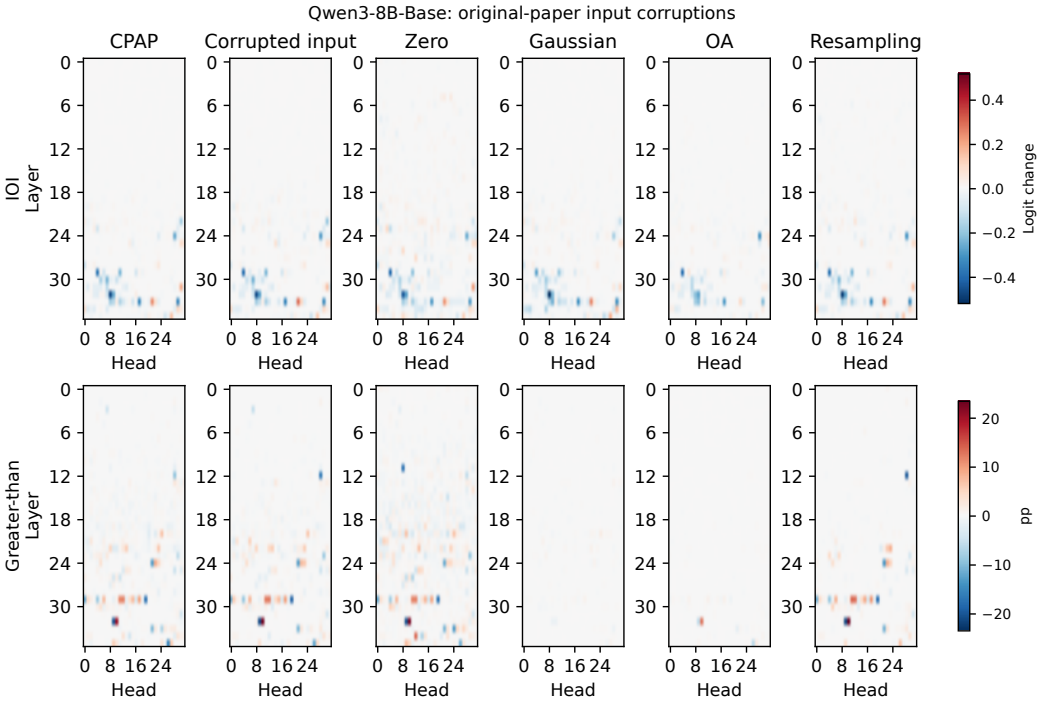


Figure 10: Qwen3-8B-Base IOI and Greater-than effects over all 36 layers. Greater-than predicts the units digit with the tens digit supplied; effects are in percentage points. Maps use each method's available observations and a shared, unclipped scale within each row.