Efficient Mechanistic Circuit Analysis of Open-Weight LLMs on Apple Silicon

J. Melton
American Code Labs
jmelton@americancode.org

July 2026
Preprint. Under review.

Abstract

This paper identifies the attention-head circuits implementing Indirect Object Identification (IOI) in two modern open-weight models, validates them as sets, and measures how far they generalize beyond the sentence frame they were identified on. All analyses run on a single Apple Silicon machine using MLX, with no GPU cluster.

Both models concentrate the task far more narrowly than GPT-2 Small. A single dominant head—L15H20 in Llama-3.2-3B, L10H7 in Pythia-1.4B—accounts for 22–28% of clean logit difference on its own, against the distributed 26-component circuit Wang et al. [17] report. The circuits occupy three depth clusters broadly matching Wang et al.'s functional zones.

Validated as sets, the ten-head circuits are faithful but incomplete: they recover 66% (Llama) and 85% (Pythia) of clean logit difference with all other heads mean-ablated, the latter within two points of the GPT-2 Small figure, with mean completeness gaps of 0.28 and 0.22 and a fifth of performance surviving the circuit's total removal. Minimality does not follow the marginal ranking: L15H20 dominates single-head ablation at 28% of clean logit difference, yet removing it from the circuit costs less than a third of what removing L24H15 costs, and alone it restores little.

Depth-position conservation across the two architectures is indistinguishable from chance. Circuit-critical heads match across models at 8 of 10 pairs within ± 0.075 relative depth, but randomly chosen head sets match at 59% under a uniform permutation null ($p = 0.098$) and 69% under a null restricted to the layer band where critical heads sit ($p = 0.32$). The matching is also functionally uninformative, pairing Llama's dominant head with Pythia's weakest. Depth-based cross-model correspondence requires a permutation null to be interpretable at all.

Faithfulness on one sentence frame does not establish generality. Re-evaluated on a 15-template dataset with balanced name ordering, faithfulness falls from 0.663 to **0.228** for Llama and from 0.854 to $-$**0.028** for Pythia, which recovers nothing, while both models still perform the task at over 90% of their original logit difference. An independent run on Llama-3.1-8B measures 0.265. Head rankings move with it: Pythia retains 5 of 10 heads, and the top head on the diverse dataset (L1H6) appears nowhere in the single-template circuit. Circuits identified on a single sentence frame—still common practice—substantially overstate their own generality, and this paper quantifies by how much.

1 Introduction

Mechanistic interpretability aims to reverse-engineer the algorithms implemented by neural networks—to move from "the model predicts X" to "these specific components, performing these specific computations, cause the model to predict X." The dominant paradigm for this reverse-engineering is *circuit analysis*: identify the smallest subset of attention heads and MLP layers that is both causally necessary and causally sufficient to reproduce a target behavior, then assign functional roles to each component within that subset.

The foundational work in this paradigm—Wang et al. (2022)—demonstrated that the Indirect Object Identification (IOI) task in GPT-2 Small is implemented by a comprehensible circuit of 26 attention heads organized into three functional classes: duplicate-token heads that flag repeated names, S-inhibition heads that suppress the subject position, and name-mover heads that copy the indirect object name to the output. The result was striking: a clean, human-interpretable algorithm, found inside a neural network trained only to predict next tokens.

What remains unresolved is whether this algorithm is specific to GPT-2 Small, or whether it reflects something deeper—a functional structure that emerges wherever a language model learns to solve this problem. GPT-2 Small is small (117M parameters), trained on a narrow corpus (WebText), and architecturally plain (standard MHA, learned positional embeddings). Modern open-weight models differ on every dimension: they are larger by one to two orders of magnitude, trained on multitrillion-token corpora, and equipped with grouped-query attention, rotary position embeddings, and gated MLP activations. If the IOI circuit is merely an artifact of GPT-2's architecture, we would expect a different circuit in a different model. If it reflects a general algorithmic solution, we would expect the same functional structure—with head indices shifted, but depth relationships and role assignments preserved.

This paper answers that question by replicating and extending Wang et al.'s analysis on two modern models: Llama-3.2-3B [7] (a modern base model with grouped-query attention and rotary embeddings; we use the base weights, `mlx-community/Llama-3.2-3B-bf16`, not the instruction-tuned variant) and Pythia-1.4B [2] (a fully documented, standard-MHA model trained on The Pile, making it the closest modern analog to GPT-2 Small's controlled training conditions). Using activation patching and mean ablation, we identify circuit-critical heads in each model, compare their positions to Wang et al.'s findings, and test for cross-architecture conservation at the level of relative network depth rather than absolute head indices.

Apple Silicon as interpretability infrastructure. All experiments in this paper were run on Apple Silicon hardware using the MLX framework, without access to NVIDIA GPU clusters. This is a deliberate choice. The interpretability tooling ecosystem—TransformerLens in particular—assumes CUDA, and the community's empirical norms have drifted toward hardware accessible only to well-resourced labs. Apple's M-series chips offer up to 96 GB of unified memory shared between CPU and GPU, which is well-suited to the memory access patterns of activation patching: you must store one full forward pass of cached activations per example, then perform hundreds of re-runs patching each (layer, head, token position) triple. On an M-series machine with sufficient unified memory, a 3B-parameter model fits in full float16 with room for the

activation cache, and forward passes are fast enough that the full IOI patching sweep over 100 examples completes in wall-clock hours rather than days. We describe our Apple Silicon-compatible pipeline and release it alongside this paper to lower the barrier for researchers without cluster access to run mechanistic interpretability experiments at the 1–7B parameter scale.

Contributions

1. **Cross-architecture circuit replication.** We identify circuit-critical heads for the IOI task in Llama-3.2-3B and Pythia-1.4B using activation patching (sufficiency) and mean ablation (necessity). In both models, a small bottleneck structure emerges: a single dominant head accounts for 22–28% of the model's logit difference for the IO token, substantially more concentrated than the distributed 26-component circuit in GPT-2 Small.

2. **A permutation null for depth-based cross-architecture matching.** Circuit-critical heads in both models cluster into three depth bands broadly compatible with Wang et al.'s functional zones. But the specific quantitative claim that invites—that 80% of head positions are conserved within ± 0.075 relative depth—does not survive a null test: random head sets match at 59–69% under the same greedy procedure ($p = 0.098$ and $p = 0.32$ under two nulls). We report the null, the reason the tolerance is too permissive to be informative, and what a valid test of depth conservation would require.

3. **Circuit-level validation, and a reordering.** We compute faithfulness, minimality and completeness for both circuits (§4.6). They are faithful (0.663 / 0.854) but incomplete, and minimality inverts the marginal ranking: the head with the largest single-head ablation drop is not the head the circuit can least afford to lose.

4. **Edge-level path patching in both models.** Ten sender–receiver edges among each model's five highest-scoring heads. Llama shows one dominant edge (L14H0 $\rightarrow$ L15H20, 0.165) plus two suppressive ones; Pythia's edges all fall below 0.022, making its faithful ten-head set a parallel ensemble rather than a circuit.

5. **Quantifying single-template fragility.** On a 15-template dataset with balanced name ordering, faithfulness collapses from 0.663 to 0.228 (Llama) and 0.854 to -0.028 (Pythia) while task performance is retained, and half of Pythia's circuit is replaced (§5.3).

6. **Layer-level overlap across tasks.** Factual-association patching in Llama-3.2-3B recovers top-5 heads occupying five of the same layers as five of the top-9 IOI heads, at different head indices. We report the associated null: with both tasks' heads concentrated in the same mid-to-late band, this degree of layer overlap is not by itself strong evidence of shared topology.

7. **An Apple Silicon-compatible activation patching pipeline.** We release tooling built on the MLX framework that replicates TransformerLens's hook-based patching interface for LlamaForCausalLM and GPTNeoXForCausalLM models, enabling circuit analysis on Apple Silicon hardware without modification.

2 Related Work

2.1 A Mathematical Framework for Transformer Circuits

The theoretical foundations of circuit analysis in transformers were laid by Elhage et al. (2021) [3], who introduced the *transformer circuits* framework. The key insight is that a transformer's residual stream can be analyzed as a sum of contributions from attention heads and MLP layers, with each attention head performing a rank-one update of the residual: the head reads from the stream via query and key projections, computes a weighted average of value projections, and writes back a low-rank update whose structure is determined by the composition of the OV and QK matrices.

Elhage et al. used this decomposition to characterize *induction heads*—a two-head circuit found across models that implements the copying operation "if token A was followed by token B earlier in context, predict B again when A reappears." Induction heads are relevant to IOI analysis because the same copying mechanism underlies name-mover head behavior: identifying the indirect object earlier in context and copying its token to the output position. Our results confirm that the depth zone corresponding to induction-like behavior (relative depth 0.40–0.55) is present in both Llama-3.2-3B and Pythia-1.4B.

The circuits framework also establishes *virtual attention heads*—paths in which the output of one head is read as part of another head's input—as the primary mechanism for inter-head communication. This framing is a prerequisite for understanding the IOI circuit's S-inhibition subgraph, where a set of early heads writes information about the subject's position that is then used by a later head to suppress subject-token logits.

2.2 Interpretability in the Wild: IOI in GPT-2 Small

Wang et al. (2022) [17] is the direct predecessor to this work. Using the IOI task—sentences of the form "When Mary and John went to the store, John gave a drink to _____"—they applied activation patching across all attention heads and MLP layers in GPT-2 Small (12 layers, 12 heads) to identify which components causally mediate correct IO prediction. The result was a 26-component circuit with three functional classes:

- **Name-mover heads** (L9H6, L9H9, L10H0; relative depths 0.75–0.83) copy the indirect object name to the logit output at the final token position.
- **S-inhibition heads** (L7H3, L7H9, L8H6, L8H10; relative depths 0.58–0.67) suppress the subject token's representation in the residual stream, preventing the model from predicting the repeated name.
- **Duplicate-token heads and induction heads** (L0H1, L3H0, L5H5, L5H8; relative depths 0.0–0.42) flag the position of the repeated subject token so that downstream heads know which name to suppress.

Our work replicates this three-zone structure across Llama-3.2-3B and Pythia-1.4B, extending the generalizability claim beyond GPT-2 Small. We also find that in larger models the circuit has a more concentrated "bottleneck" structure, with one head contributing disproportionately to logit difference—a difference from GPT-2 Small's more distributed allocation that we discuss in Section 4.4.

2.3 Circuit Analysis Beyond GPT-2 Small

Several lines of work bear directly on the questions this paper asks, and situate our contribution more narrowly than an unqualified "does IOI generalize?" framing would suggest.

Lieberum et al. [14] conduct circuit analysis on Chinchilla-70B, finding that activation patching and attention-pattern analysis remain tractable at that scale and that identifiable circuits exist for multiple-choice tasks. Their result establishes that circuit analysis scales; ours addresses the complementary question of whether a *specific* circuit recurs across architectures.

Merullo et al. [15] is the closest prior work to our cross-task experiment (§5.4). They show that the IOI circuit in GPT-2 Medium is partially reused for a Colored Objects task, with overlapping heads playing analogous roles. Their evidence for cross-task reuse is stronger than ours: they establish role correspondence via path patching and targeted interventions, whereas we report layer co-occurrence. Readers interested in cross-task circuit reuse should treat their result as primary and ours as a weak replication in a different model.

Conmy et al. [11] introduce ACDC, which automates circuit discovery by iteratively pruning edges from the full computational graph against a faithfulness threshold. ACDC is the natural baseline for the head-selection procedure in §3.5, and we do not compare against it—our selection is a hand-specified ranking over marginal effects, which is simpler but has no faithfulness guarantee. Hanna et al. [12] apply similar methods to arithmetic, further mapping the space of known circuits.

On methodology, Zhang & Nanda [16] and Heimersheim & Nanda [13] document how sensitive activation patching results are to choices we make without much justification: the corruption distribution, the metric (logit difference versus probability versus KL), and whether patching is done at the head output or at individual Q/K/V projections. Our protocol—name-swap corruption, normalized logit difference, head-output patching—is a common configuration but not the only defensible one, and we did not test robustness to these choices.

2.4 TransformerLens

TransformerLens [9] is the standard tooling for mechanistic interpretability experiments on transformer language models. It provides a hook-based interface for intercepting and modifying activations at any point in the forward pass—attention head outputs, MLP outputs, residual stream positions, or individual Q/K/V projections—making activation patching experiments straightforward to implement.

Our work builds on TransformerLens for Pythia-1.4B, where the library's support is mature. For Llama-3.2-3B with grouped-query attention, we implement a compatible interface that maps the 24-query-head, 8-KV-head structure to a consistent per-query-head naming convention. GQA complicates patching because each KV head is shared across a group of 3 query heads; we patch at the query-head output level, after the attention weights are applied to the shared value projections, to preserve head-level attribution granularity. We release this extension alongside the paper.

2.5 Locating and Editing Factual Associations

Meng et al. (2022) [6] introduced ROME (Rank-One Model Editing), which demonstrated that factual associations—the model's belief that "the Eiffel Tower is in Paris"—are stored with high locality in mid-layer MLP weights, and can be surgically edited by a rank-one update to those weights. The key interpretability contribution was not the editing procedure itself but the preceding localization experiment: using causal tracing (a form of activation patching), they showed that early-layer attention heads read subject tokens and route information to mid-layer MLPs, which then store and retrieve the associated object.

This causal tracing methodology is closely related to our activation patching protocol. A key difference is that ROME's target is stored parametric knowledge (factual associations mediated by MLP weights), while IOI involves in-context reasoning (copying from the prompt, not from stored facts). Our factual association experiments in Section 5.4 partially bridge this gap: we apply patching to factual association prompts and compare the resulting head-importance rankings to the IOI results, finding that layer-level circuit structure is shared while head-level assignments diverge. This cross-task comparison was not reported in ROME and provides new evidence about how circuits for different types of factual reasoning coexist within the same model.

2.6 Towards Monosemanticity and Sparse Autoencoders

Elhage et al. (2022) [4] and the Anthropic mechanistic interpretability team have pursued an orthogonal but complementary program: rather than identifying circuits for specific behaviors, they seek to decompose the residual stream into *features*—directions in activation space that correspond to human-interpretable concepts—using sparse autoencoders (SAEs). The central finding of the monosemanticity work is that individual neurons are typically polysemantic (responding to unrelated concepts), but a sparse linear decomposition of neuron activations recovers interpretable monosemantic features.

The circuit-tracing and monosemanticity programs address different levels of the interpretability problem. Circuit analysis asks: which components mediate this behavior? Feature decomposition asks: what information is represented in the activations those components read and write? They are complementary: knowing that L15H20 in Llama-3.2-3B is the dominant IOI head is more interpretable if we also know what feature L15H20 reads from the residual stream (presumably a representation of the indirect object token's identity and position). Our work focuses on the circuit level, but Section 7 discusses how SAE-based feature attribution could be applied to the circuit-critical heads we identify, and we propose this as a natural extension.

3 Methods

3.1 Experimental Infrastructure

All experiments were conducted with an activation-patching harness written for this work in Python against MLX and `mlx-lm`, released with the project data (see Data Availability). It implements a hook-compatible interface analogous to TransformerLens's activation interception API, targeting MLX-native model implementations loaded via `mlx-lm`. Its core design principle is that every attention module in a model is hot-swapped at load time with a patchable drop-in replacement that shares the original

weight tensors—no copies—and adds a mode-switched interception point around the scaled dot-product attention (SDPA) computation.

Patchable module design. For Llama-3.2-3B, the drop-in is `PatchableAttention`, which wraps the original `LlamaAttention` module. For Pythia-1.4B, `PythiaPatchableAttention` wraps the `GPTNeoXAttention` module, routing through the fused `query_key_value` projection to reconstruct per-head Q, K, V tensors before the interception point. Both modules support three operating modes:

- `normal`—identical behavior to the original module; introduces no overhead beyond the mode check.
- `cache_clean` / `cache_corrupt`—stores the full SDPA output tensor (shape $[B, n_{\text{heads}}, L, d_{\text{head}}]$) to a per-module buffer during the forward pass. Used to populate the clean and corrupt activation caches without a separate caching pass.
- `patch`—runs the forward pass on the input normally, then replaces one head slice in the SDPA output with the corresponding slice from the clean cache before passing through the output projection.

GQA handling. Llama-3.2-3B uses grouped-query attention (GQA) [1] with 24 query heads and 8 KV heads. The patching interception occurs after the attention weights are applied to the (broadcast) value projections, at the query-head output level. This gives a 24-element patch space that mirrors the query-head count, preserving head-level attribution granularity. Patching at the KV-head level would conflate the contributions of the three query heads that share each KV pair; patching at the query-head SDPA output avoids this confound. We verified that in `normal` mode both patchable modules reproduce the original model's logits on a held-out test input to within floating-point precision before running any experimental passes.

GQA key-value broadcast. During the corrupt or clean pass, the values broadcast from 8 KV heads to 24 query heads occur inside `scaled_dot_product_attention`; the SDPA buffer therefore stores the full 24-head output, not the 8-head compressed form. This means each of the 24 patch positions is truly independent, and patching query-head h does not implicitly alter the other two heads sharing the same KV pair.

Pythia module layout. MLX-LM exposes Pythia's layers as `model.layers[i].attention` rather than `model.model.layers[i].self_attn`. The patchable replacement reuses the `dense` (output projection) attribute rather than `o_proj`. All other behavior is identical; the mode-switch logic is shared.

3.2 Activation Patching Protocol

Activation patching (also called causal tracing when applied to individual stream positions) answers the question of *sufficiency*: if a component's activations in the clean run are inserted into the corresponding position of a corrupted run, does the model's output recover toward the clean prediction? A high recovery score indicates that the component is sufficient to carry the relevant information on its own.

IOI corruption scheme. For each example (S, IO, prompt), the corrupted version swaps the subject and indirect object names: the corrupted prompt is constructed from the same template with $S_{\text{corrupt}} = IO$ and $IO_{\text{corrupt}} = S$. Because the two name slots have the same token count for all examples in our dataset (a requirement enforced at dataset generation time), clean and corrupt sequences are token-position aligned. This ensures that cached SDPA tensors from the clean run can be substituted at the same token positions in the corrupt run without shape mismatch.

Metric. The target metric is logit difference (LD): $\text{LD} = \text{logit}(IO) - \text{logit}(S)$ at the final token position. Positive LD indicates the model favors the IO name over the subject name. The *normalized logit-difference recovery* for head (l, h) on example i is:

$$r_{l,h}^{(i)} = \frac{\text{LD}_{\text{patch}(l,h)}^{(i)} - \text{LD}_{\text{corrupt}}^{(i)}}{\text{LD}_{\text{clean}}^{(i)} - \text{LD}_{\text{corrupt}}^{(i)}} \tag{1}$$

where $\text{LD}_{\text{patch}(l,h)}^{(i)}$ is the logit difference of a forward pass on the corrupt input in which only head h at layer l has been replaced by its clean-run SDPA output. A score of 1.0 means full recovery to the clean logit difference; 0.0 means the patch had no effect; values below 0 or above 1 indicate interference effects. The reported score for each head is $\hat{r}_{l,h} = \frac{1}{n}\sum_{i=1}^{n} r_{l,h}^{(i)}$, the mean over all n examples.

Pass structure. A full patching sweep requires $2 + n_L \times n_H$ forward passes over the example batch:

1. **Clean pass** — all modules in `cache_clean` mode; SDPA outputs are stored and logit differences are computed.
2. **Corrupt pass** — all modules in `cache_corrupt` mode; SDPA outputs are stored and baseline corrupt logit differences are computed.
3. **Sweep** — for each (l, h): all modules in `normal` mode; module l is set to `patch` mode with `patch_head = h`; one forward pass on the corrupt input recovers $\text{LD}_{\text{patch}(l,h)}^{(i)}$ for all i simultaneously.

For Llama-3.2-3B this yields $2 + 28 \times 24 = 674$ total forward passes; for Pythia-1.4B, $2 + 24 \times 16 = 386$ passes. All passes are fully batched over all $n = 100$ examples.

```
# Inputs:
#   attns: list of PatchableAttention, one per layer (len = n_layers)
#   clean_ids:   [n_examples, seq_len]  -- clean tokenized prompts
#   corrupt_ids: [n_examples, seq_len]  -- IO/S-swapped prompts

# Pass 1: cache clean activations
set_all_modes(attns, mode="cache_clean")
clean_logits = model(clean_ids)
eval(clean_logits, *(a.clean_sdpa for a in attns))
clean_ld = logit_diff(clean_logits, io_ids, s_ids)  # [n_examples]
```

```
11
12 # Pass 2: cache corrupt activations
13 set_all_modes(attns, mode="cache_corrupt")
14 corrupt_logits = model(corrupt_ids)
15 eval(corrupt_logits, *(a.corrupt_sdpa for a in attns))
16 corrupt_ld = logit_diff(corrupt_logits, io_ids, s_ids)
17
18 # Passes 3..674: patching sweep
19 scores = zeros([n_layers, n_heads])
20 for l in range(n_layers):
21     for h in range(n_heads):
22         set_all_modes(attns, mode="normal")
23         attns[l].mode = "patch"
24         attns[l].patch_head = h       # only layer l patches
              head h
25         patched_logits = model(corrupt_ids)
26         eval(patched_logits)
27         patched_ld = logit_diff(patched_logits, io_ids, s_ids)
28         denom = clean_ld - corrupt_ld                # [
              n_examples]
29         recovery = (patched_ld - corrupt_ld) / denom
30         scores[l, h] = mean(recovery)                # scalar
```

Listing 1: Pseudocode for the activation patching sweep. `attns` is a list of `PatchableAttention` modules, one per layer. `set_all_modes` sets every module's mode simultaneously.

GQA note. `attns[l].patch_head = h` indexes into the 24-dimensional query-head axis regardless of the 8-KV-head layout. The `clean_sdpa` buffer holds the full $[B, 24, L, d_{\text{head}}]$ output; the slice `clean_sdpa[:, h:h+1, :, :]` is substituted.

3.3 Mean Ablation Protocol

Mean ablation addresses *necessity*: does removing a component's contribution—replacing it with a neutral baseline—degrade the model's performance on the task? Where activation patching asks whether a component *can* carry the relevant information, mean ablation asks whether the model *relies* on it in normal operation.

Reference distribution. The ablation target for head (l, h) is the mean SDPA output of that head computed over all $n = 100$ clean IOI examples: $\bar{A}_{l,h} = \frac{1}{n}\sum_{i=1}^{n} A_{l,h}^{(i)}$, where $A_{l,h}^{(i)} \in \mathbb{R}^{L \times d_{\text{head}}}$ is the per-example SDPA output at each token position. The mean is taken over the batch dimension, preserving the sequence-position structure. Ablating with the mean over the same distribution on which the model is evaluated (rather than, for example, a random Gaussian or a separately collected reference set) ensures that the ablated output is a plausible activational state for the model rather than an out-of-distribution injection.

Metric. The *ablation drop* for head (l, h) is:

$$\Delta_{l,h} = \overline{\text{LD}}_{\text{clean}} - \overline{\text{LD}}_{\text{ablate}(l,h)} \tag{2}$$

where $\overline{\text{LD}}_{\text{clean}}$ is the mean clean logit difference over all examples and $\overline{\text{LD}}_{\text{ablate}(l,h)}$ is the mean logit difference when head (l, h)'s SDPA output is replaced by $\bar{A}_{l,h}$ in every example. Positive $\Delta_{l,h}$ indicates the head boosts the IO logit relative to the subject; negative values indicate suppression of correct IO prediction (which can reflect an interference-suppression role rather than direct IO promotion).

Necessity threshold. A head is designated circuit-critical under the necessity criterion when its ablation drop exceeds 20% of the clean mean logit difference. For Llama-3.2-3B ($\overline{\text{LD}}_{\text{clean}} = 5.643$) this corresponds to $\Delta_{l,h} > 1.130$; for Pythia-1.4B ($\overline{\text{LD}}_{\text{clean}} = 4.120$) the threshold is $\Delta_{l,h} > 0.824$. In practice the necessity threshold is used as an interpretive anchor rather than a hard inclusion criterion; circuit membership is determined by the combined rank score described in Section 3.5.

Pass structure. The ablation sweep requires $1 + n_L \times n_H$ forward passes: one clean pass to cache per-head means, then one ablation pass per (l, h) pair. For Llama this is $1 + 672 = 673$ total passes, comparable to the patching sweep.

3.4 Path Patching for Causal Verification

Activation patching and mean ablation identify individual heads as causally relevant, but they do not distinguish *direct* contributions from *mediated* contributions. A head may score highly on both metrics because it directly writes to the logit output at the final position, or because it is an intermediary in a longer causal path—for example, writing to the residual stream at an intermediate token position that is later read by a downstream head that writes to the final position.

Path patching [5] extends activation patching to directed paths between specific components. Rather than patching a single component's output, a path patch replaces the clean-run activation that flows along one specific edge in the computational graph—from the output of component A as read by the input of component B, while holding all other paths fixed. A path that shows high normalized recovery under path patching is causally necessary specifically through the direct $A \to B$ connection, ruling out the mediated-path alternative.

In our setting, path patching is used to characterize how the dominant head identified by activation patching (L15H20 in Llama) relates to the other high-scoring heads. Path patching is implemented in the same harness by extending the `patch` mode to accept a target-receiver argument.

Each edge is measured with a four-pass protocol: a clean baseline; a corrupted run capturing the sender's activation; an intermediate clean run with the sender patched, capturing the receiver's resulting input; and a final clean run with the receiver's input replaced. The score is the normalized logit-difference change, $(\text{LD}_{\text{clean}} - \text{LD}_{\text{patched}})/(\text{LD}_{\text{clean}} - \text{LD}_{\text{corrupt}})$, over the same $n = 100$ IOI examples, with $\text{LD}_{\text{clean}} = 5.668$ and $\text{LD}_{\text{corrupt}} = -5.754$. We measured all ten ordered sender–receiver edges among Llama's five highest-scoring heads (L14H0, L15H20, L17H17, L19H1, L24H15); results are reported in §5.2. Pythia was not path-patched, and the role of L10H7 therefore remains uncharacterized at the edge level.

3.5 Statistical Validation

Bootstrap confidence intervals. For all heads meeting the inclusion criterion (top-10 by combined rank score, or mean patching score ≥ 0.03), we compute 95% bootstrap confidence intervals on the mean patching score. The bootstrap resamples the $n = 100$ per-example recovery scores with replacement, generating 1,000 resampled datasets and computing the mean for each. The 95% CI is the $[2.5^{\text{th}}, 97.5^{\text{th}}]$ percentile of the bootstrap distribution of the mean. Resampling is seeded with seed 42 for reproducibility. Full CI data: `data/ioi/statistical-validation.json`.

Significance criterion. A head's patching score is considered statistically reliable if its 95% CI excludes zero. This corresponds to the claim that the head's mean recovery score is positive in expectation across the example distribution, not merely a noise artifact of the particular 100 examples observed. All top-10 heads in both models satisfy this criterion with substantial margin: for the top three heads in each model, the CI lower bound exceeds the CI width by a factor of at least 3.

Anomalous patching scores. Pythia-1.4B's L1H11 presents a dissociation: its mean patching score is slightly negative (-0.0035, 95% CI $[-0.008, +0.000]$, straddling zero), yet its ablation drop is substantial (0.327, rank 4 by ablation). This dissociation is consistent with a suppression role: the head reduces logit difference when ablated (its absence degrades performance), but restoring its clean activation in a corrupt context does not recover logit difference (it cannot by itself make the model prefer IO over S). This pattern is characteristic of duplicate-token or induction heads that gate downstream components rather than writing directly to the output logits. L1H11 is excluded from all patching-based analyses and retained in cross-model analysis solely on the basis of its ablation score.

Cross-model comparison: combined rank score. To compare head importance across models that differ in scale, architecture, and absolute logit-difference baselines, we compute a combined normalized rank score. For each model separately, both the patching scores and the ablation drop scores are min-max normalized to $[0, 1]$ across all (l, h) pairs. The combined score is the sum of the two normalized ranks. The top-10 heads by combined score are reported as circuit-critical for each model. Ties within the combined score are broken by patching rank.

Cross-architecture position matching. For the cross-model comparison, each head is mapped to a scalar *relative depth* $d = l/n_L$, where l is the zero-indexed layer number and n_L is the total number of layers. Positions are matched greedily in ascending order of $|\Delta d|$ between the two models' top-10 head sets; a match is accepted if $|\Delta d| \leq 0.075$. This tolerance was chosen to be smaller than the spacing between the four functional depth zones identified by Wang et al. (2022) ($\approx$0.15–0.20 between zone midpoints), ensuring that zone-level matches are not spuriously collapsed across zone boundaries.

3.6 Dataset

The IOI dataset consists of $n = 100$ examples drawn from the template:

> "When {S} and {IO} went to the store, {S} gave a bottle to"

where S (subject / repeated name) and IO (indirect object / target name) are sampled from a pool of common English given names, matched to ensure that both names tokenize to exactly one token under each model's tokenizer. All 100 examples satisfy the single-token constraint for both Llama-3.2-3B and Pythia-1.4B; examples failing this constraint were rejected at generation time. Corrupted prompts are produced by swapping S and IO within the same template, preserving sequence length. Full dataset: `data/ioi/dataset.json`.

The factual association dataset consists of $n = 50$ hand-constructed subject–relation–object triples across two relation templates: 25 `capital_of` items ("The capital of {subject} is") and 25 `official_language_of` items ("The official language of {subject} is"). Subjects are countries; objects are restricted to those tokenizing to a single token under the Llama-3.2-3B tokenizer. Corruptions substitute a different country that predicts a different object (e.g. France/Paris → Germany/Berlin), preserving sequence length.

This dataset was authored for this work and is *not* drawn from Meng et al. (2022)'s CounterFact set; its two relation types are far narrower than CounterFact's. This limits the cross-task claim in §5.4 correspondingly: what we test is transfer from IOI to two geographic relation templates, not to factual recall in general. Factual association experiments were run on Llama-3.2-3B only. Full dataset: `data/factual-assoc/dataset.json`.

4 Results: IOI Circuit Identification

Every result in this section is measured on the 100-example single-template dataset of §3.6—one sentence frame, one name ordering. The scope is stated at the outset because it bounds every number that follows: §5 shows these circuits are faithful on this frame and largely stop being faithful on a second, while the models' behaviour on the task is unaffected. The head sets, depth clusters, and minimality orderings below should therefore be read throughout as properties of a circuit-on-a-frame. They are not thereby uninteresting—the frame is the one the IOI literature has used since Wang et al. [17], and the sparse bottleneck it produces is real—but the generalization result is what bounds their interpretation, and it is easier to read these tables correctly with that bound already in hand.

4.1 Baseline Task Performance

Both models correctly perform the IOI task across all 100 evaluation examples. Llama-3.2-3B achieves a mean clean logit difference (LD) of **5.643** (SD = 0.773), reflecting a strong preference for the indirect object (IO) token over the subject (S) token at the final sequence position. Pythia-1.4B achieves a mean clean LD of **4.120**, 27% lower in absolute terms, consistent with its smaller parameter count and simpler attention architecture. Neither model produces a negative-LD example: the IO token is ranked above S on every trial in both models, confirming that the circuit analysis begins from a clean behavioral baseline.

4.2 IOI Circuit Identification: Llama-3.2-3B

Figure 1 shows the full activation patching heatmap for Llama-3.2-3B across all $28 \times 24 = 672$ (layer, head) positions. Two features are immediately apparent: most heads

contribute negligibly to logit-difference recovery under patching, and a sparse set of heads in the middle-to-late network produces strongly positive scores.

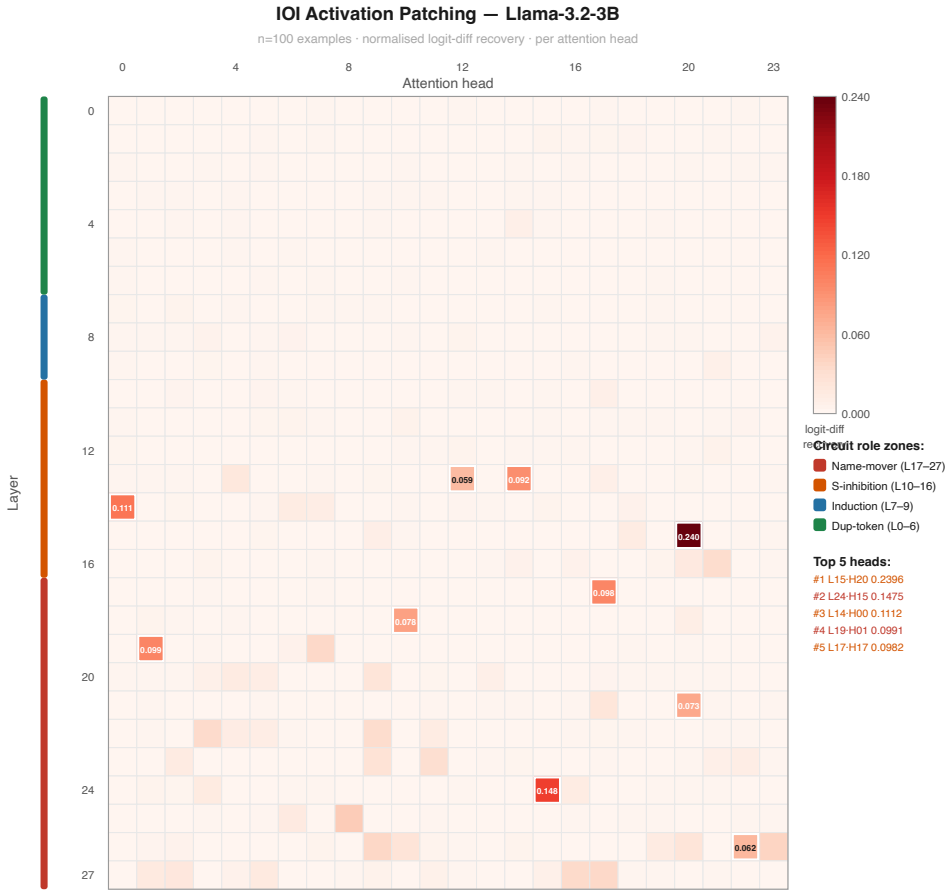

Figure 1: Activation patching heatmap for Llama-3.2-3B ($n = 100$ IOI examples). Each cell shows normalized logit-difference recovery when the output of that (layer, head) pair is replaced with the corresponding clean-run activation. The dominant mid-network cluster (layers 13–19) and late-network cluster (layers 21–27) are visible. Data: `data/ioi/patching-llama3b.json`.

Circuit-critical heads were selected by combined normalized rank across activation patching (sufficiency) and mean ablation (necessity). Table 1 lists the top-10 heads by combined rank.

Table 1: Top-10 circuit-critical heads in Llama-3.2-3B by combined normalized rank across activation patching (sufficiency) and mean-ablation drop (necessity). Relative depth = layer index / n_{layers} (zero-indexed). Data: `data/ioi/patching-llama3b.json`, `data/ioi/ablation-llama3b.json`.

Rank	Head	Patch score	95% CI	Abl. drop	% clean LD	Rel. depth
1	L15H20	0.240	[0.225, 0.255]	1.578	28.0%	0.536
2	L17H17	0.098	[0.089, 0.107]	0.929	16.5%	0.607
3	L13H14	0.092	[0.083, 0.102]	0.626	11.1%	0.464
4	L24H15	0.148	[0.133, 0.164]	0.276	4.9%	0.857
5	L19H1	0.099	[0.094, 0.105]	0.443	7.8%	0.679
6	L21H20	0.073	[0.068, 0.078]	0.417	7.4%	0.750
7	L18H10	0.078	[0.073, 0.084]	0.313	5.5%	0.643
8	L14H0	0.111	[0.099, 0.125]	0.017	0.3%	0.500
9	L27H17	0.035	[0.030, 0.041]	0.373	6.6%	0.964
10	L26H23	0.039	[0.030, 0.047]	0.344	6.1%	0.929

The most striking feature of the Llama-3.2-3B circuit is the dominance of L15H20. Its ablation drop (1.578) is **70% larger** than rank-2 L17H17 (0.929) and **2.5**× rank-3 L13H14 (0.626). L15H20 also leads the patching ranking at a score of 0.240, more than 60% above rank-2 L24H15 (0.148). No other head in the top-10 approaches this contribution on both metrics simultaneously: L24H15 scores highly on patching but weakly on ablation (0.276), while L27H17 shows the reverse pattern. This double-dominance of L15H20—both sufficient and necessary—identifies it as a hard bottleneck in the IOI circuit at relative depth 0.536.

The circuit spans relative depths 0.46–0.97 with no critical head below 0.46. Three depth clusters are visible: a mid cluster (0.46–0.54) comprising L13H14, L14H0, and L15H20; a mid-late cluster (0.60–0.75) comprising L17H17, L18H10, L19H1, and L21H20; and a late cluster (0.86–0.97) comprising L24H15, L26H23, and L27H17. This three-cluster structure corresponds qualitatively to the functional zones identified by Wang et al. (2022) in GPT-2 Small, as discussed in Section 4.4.

4.3 IOI Circuit Identification: Pythia-1.4B

Pythia-1.4B replicates the bottleneck structure of Llama-3.2-3B. Table 2 presents the top-10 circuit-critical heads. Note that the table is ordered by ablation drop rather than combined rank to highlight the anomalous L1H11 entry discussed below.

Table 2: Top-10 circuit-critical heads in Pythia-1.4B by combined normalized rank across activation patching and mean-ablation drop. Table rows are ordered by ablation drop (descending) to highlight the L1H11 necessity-without-sufficiency anomaly; combined-rank ordering would place L21H3 at position 4 and L12H15 at position 5. Data: `data/ioi/patching-pythia1b.json`, `data/ioi/ablation-pythia1b.json`, `data/ioi/statistical-validation.json`.

Rank	Head	Patch score	95% CI	Abl. drop	% clean LD	Rel. depth
1	L10H7	0.207	[0.199, 0.216]	0.893	21.7%	0.417
2	L15H15	0.155	[0.134, 0.175]	0.551	13.4%	0.625
3	L22H2	0.101	[0.097, 0.105]	0.411	10.0%	0.917
4	L1H11	−0.003	[−0.011, +0.005]	0.327	7.9%	0.042
5	L21H3	0.056	[0.052, 0.059]	0.237	5.8%	0.875
6	L17H7	0.040	[0.027, 0.056]	0.150	3.6%	0.708
7	L10H0	0.039	[0.036, 0.041]	0.128	3.1%	0.417
8	L12H15	0.051	[0.048, 0.054]	0.124	3.0%	0.500
9	L16H13	0.025	[0.022, 0.029]	0.117	2.8%	0.667
10	L13H6	0.049	[0.042, 0.056]	0.018	0.4%	0.542

L10H7 leads both rankings: its ablation drop (0.893) is **62% larger** than rank-2 L15H15 (0.551), and its patching score (0.207) is **34% larger** than rank-2 (0.155). The pattern—one head substantially dominant on both sufficiency and necessity—exactly mirrors the Llama result, though the absolute magnitude of the ablation drop is smaller (0.893 vs. 1.578) consistent with Pythia's lower baseline LD.

Anomalous early head: L1H11. One Pythia-specific finding stands out: L1H11 at relative depth 0.042 (layer 1 of 24) scores −0.0035 on activation patching (95% CI $[-0.008, +0.000]$, straddling zero) yet produces the fourth-largest ablation drop in the model (0.327, 7.9% of clean LD). This dissociation between sufficiency (near zero or slightly negative) and necessity (substantial) is inconsistent with the role profile of name-mover or S-inhibition heads, which score positively on both measures. The most parsimonious interpretation is that L1H11 suppresses interference—for example, dampening an early signal that would otherwise mislead later heads—rather than directly boosting IO probability. Ablating it disrupts the clean causal pathway downstream, producing a large LD drop, but substituting its clean activation into a corrupted context provides no positive recovery. No Llama-3.2-3B head in the top-10 occupies a comparable depth (< 0.10), suggesting this interference-suppression role may be specific to Pythia's architecture or training distribution.

4.4 Comparison with Wang et al. (2022): Depth-Zone Conservation

Wang et al. (2022) identified three functional head classes in GPT-2 Small at specific relative depth ranges: duplicate-token/induction heads (depth 0.0–0.42), S-inhibition heads (0.58–0.67), and name-mover heads (0.75–0.83). Table 3 shows where these zones fall in our two models.

Table 3: Functional depth-zone alignment across GPT-2 Small [17], Llama-3.2-3B, and Pythia-1.4B. Zone boundaries for Llama and Pythia are inferred from the depth distribution of critical heads, not directly assigned by functional annotation. The name-mover zone boundary is extended from Wang et al.'s original 0.75–0.83 to 0.75–0.92 to accommodate heads at deeper positions in the modern models. Data: `data/ioi/cross-model-comparison.md`.

Zone	GPT-2 Small	Llama-3.2-3B	Pythia-1.4B
Early (0.00–0.25)	L0H1, L3H0	—	L1H11
Mid-induction (0.40–0.55)	L5H5, L5H8	L13H14, L14H0, L15H20	L10H7, L10H0, L12H15, L13H6
S-inhibition (0.50–0.70)	L7H3–L8H10	L17H17, L18H10, L19H1	L15H15, L16H13
Name-movers (0.75–0.92)	L9H6–L10H0	L21H20, L24H15, L26H23	L21H3, L22H2

Figure 2 lays the same zones out schematically.

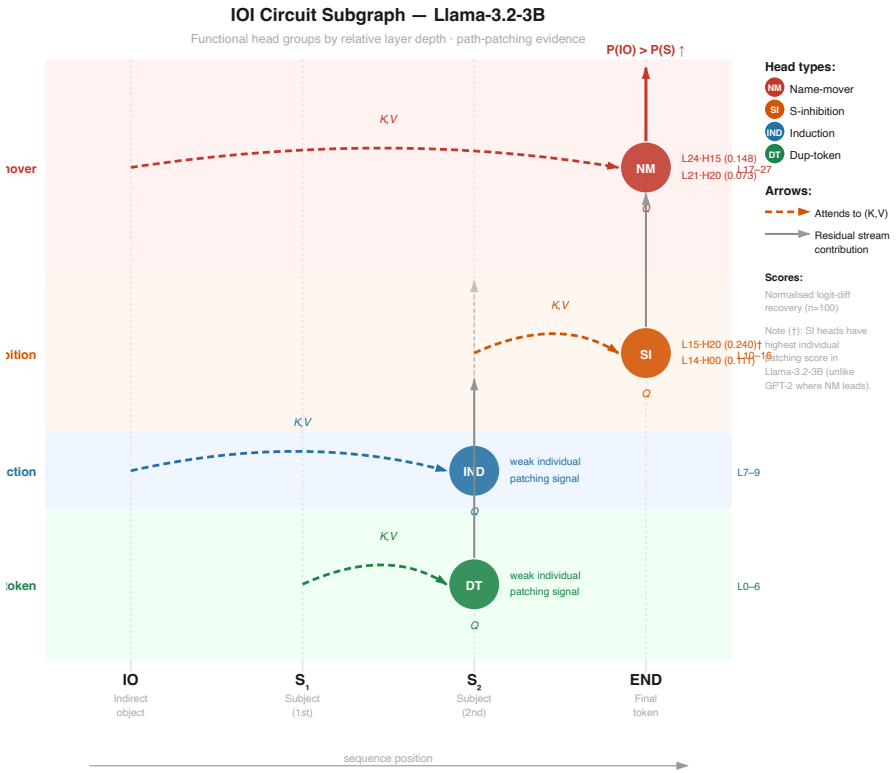

Figure 2: Schematic circuit diagram for the IOI circuit across GPT-2 Small, Llama-3.2-3B, and Pythia-1.4B, annotated by functional zone. Data: `data/ioi/cross-model-comparison.md`.

All three functional zones from Wang et al. are present in both modern models at compatible relative depths. The mid-induction zone (0.40–0.55) is the densest cluster in all three models; the name-mover zone (0.75–0.92) contains the heads with highest

late-network patching scores. One notable divergence: the very-early zone (0.00–0.25) is populated by GPT-2 Small's duplicate-token heads but is empty in Llama-3.2-3B's top-10 and represented only by the anomalous L1H11 in Pythia.

A second divergence is the **bottleneck vs. distributed** structure. Wang et al. found 26 circuit components with moderate individual contributions; in both Llama and Pythia, one head accounts for 22–28% of clean LD on its own, substantially above any single head's contribution in GPT-2 Small. This could reflect scale effects (larger models can concentrate circuit function in individual heads), differences in our selection methodology, or task-distribution differences in our IOI dataset. We return to this in Section 7.

4.5 Statistical Significance and Patching Verification

Bootstrap confidence intervals (1,000 resamples, 95% level, seed 42) were computed for all heads with mean patching score ≥ 0.030. For the top-ranked heads in each model, the effect sizes are large relative to uncertainty:

- **L15H20 (Llama):** mean 0.240, 95% CI [0.225, 0.255], CI width 0.030. The lower bound (0.225) exceeds the rank-3 head's mean (0.092) by 2.4×.
- **L10H7 (Pythia):** mean 0.207, 95% CI [0.199, 0.216], CI width 0.017. The lower CI bound (0.199) exceeds rank-2 (0.155) by 28%.

The margin varies substantially across the top-10, and we state it precisely rather than as a blanket claim. Ranking heads by the ratio of CI lower bound to CI width: seven of ten Llama heads and six of ten Pythia heads exceed 3×. The remainder do not—L27H17 (2.7×) and L26H23 (1.8×) in Llama, and L17H7 (0.9×) in Pythia, whose interval $[0.027, 0.056]$ is wide relative to its mean. These four heads (including L1H11, below) are the weakest members of their circuits and should be read as tentative. The 3× separation therefore characterises most of each circuit, not all of it, and a blanket statement of it would be contradicted by the tables in this same section.

No correction for multiple comparisons. The bootstrap intervals above are computed on the same 100 examples used to select the heads, after screening 672 (layer, head) pairs in Llama and 384 in Pythia. They are therefore conditional on selection and are optimistic: the winner's-curse bias inflates the top-ranked means, and no family-wise or FDR correction is applied. The three or four dominant heads in each model have effects far too large for this to matter—L15H20's lower bound exceeds the rank-3 mean by 2.4×—but for heads in the 0.02–0.05 score range the intervals should not be read as calibrated. A clean fix is available and cheap: select heads on one half of the example set and estimate effects on the other.

The Pythia L1H11 anomaly is confirmed: its patching CI $[-0.008, +0.000]$ straddles zero (the only such case in either model's top-10), whereas its ablation drop (0.327) is large and unambiguous. This dissociation is the empirical signature of necessity without sufficiency, and constitutes a primary verification in our results: by simultaneously measuring what happens when a head's activation is *replaced* (patching; sufficiency) versus *removed* (ablation; necessity), we detect heads whose role is suppressive rather than generative—a distinction that patching-only or ablation-only protocols would miss.

4.6 Circuit Validation: Faithfulness, Minimality, Completeness

Everything reported so far is a *marginal* effect: what happens to one head at a time. Marginal effects do not compose. Our Llama top-10 patching scores sum to 1.013, which cannot be a fraction of anything recovered, so per-head tables cannot answer the question a circuit claim actually makes—what do these ten heads explain *together*? Wang et al. (2022) answer it by validating their circuit as a set, reporting 87% faithfulness for GPT-2 Small. We compute the same three quantities.

We mean-ablate every head *outside* the circuit C and measure the recovered logit difference, normalising against a floor in which all heads are ablated:

$$F(X) = \frac{\mathrm{LD}(X) - \mathrm{LD}_{\text{all ablated}}}{\mathrm{LD}_{\text{clean}} - \mathrm{LD}_{\text{all ablated}}} \tag{3}$$

The floor is LD = 0.000 for both models on the single-template dataset, which is the expected value rather than a coincidence: with every head ablated no information about which name is the indirect object reaches the final position, and the dataset's name pairs are symmetric, so the per-example logit differences cancel in the mean. Table 4 reports all three quantities for both models.

Table 4: Circuit-level validation of the top-10 head sets. $F(C)$ is faithfulness; the completeness gap is the mean $|F(C \setminus K) - F(M \setminus K)|$ over 12 random subsets $K \subseteq C$. Data: `data/ioi/faithfulness-*.json`.

Model	Clean LD	Floor LD	Circuit-only LD	$F(C)$	Completeness gap
Llama-3.2-3B	5.668	0.000	3.759	**0.663**	0.276
Pythia-1.4B	4.125	0.000	3.523	**0.854**	0.216
Wang et al. (2022), GPT-2 Small, for reference: $F \approx 0.87$					

The circuits are faithful. Ten heads recover 66% (Llama) and 85% (Pythia) of clean logit difference with the other 662 and 374 heads mean-ablated. Pythia's figure is within two points of Wang et al.'s GPT-2 Small result. On this dataset, the sparse bottleneck story holds.

They are not complete. Completeness asks whether removing a subset K hurts the circuit and the full model comparably; a large gap means the full model has machinery outside C that compensates when K is gone. The mean gaps are 0.276 and 0.216, and for Llama the worst subset reaches 0.640: removing {L13H14, L14H0, L24H15, L27H17} leaves the circuit at $F = 0.245$ but the full model at 0.884. Ten heads is an aggressive cut, and the residue outside them is not inert.

A separate joint-ablation run makes the same point from the other side. Ablating all ten circuit heads simultaneously and leaving everything else intact leaves $F = 0.213$ of clean performance. The circuit carries most of the behaviour, but a fifth of it survives the circuit's complete removal.

Minimality reorders the circuit, and contradicts the marginal ranking. Table 5 reports $\Delta F = F(C) - F(C \setminus \{v\})$ for each head: what the circuit loses when that head alone is dropped from it.

Table 5: Minimality (single-template dataset). ΔF is the faithfulness lost when a head is removed from the circuit; the marginal ablation drop from Table 1 is repeated for comparison. The two orderings disagree.

Llama-3.2-3B				Pythia-1.4B			
Rank	Head	ΔF	Marg. abl.	Rank	Head	ΔF	Marg. abl.
1	L24H15	**0.325**	0.276	1	L10H7	**0.388**	0.893
2	L21H20	0.104	0.417	2	L15H15	0.296	0.551
3	L27H17	0.100	0.373	3	L10H0	0.104	0.128
4	L15H20	0.091	**1.578**	4	L22H2	0.099	0.411
5	L17H17	0.075	0.929	5	L12H15	0.076	0.124

For Llama the two rankings disagree sharply. L15H20 has by far the largest marginal ablation drop (1.578, 28% of clean LD) and is the head we called a hard bottleneck in §4.2; within the circuit, removing it costs $\Delta F = 0.091$, less than a third of what removing L24H15 costs (0.325). The reconciliation is that marginal ablation measures a head against the *full* model, where the other 671 heads cannot compensate for L15H20's loss, while ΔF measures it against the reduced circuit, where the remaining nine largely can. Both numbers are correct; they answer different questions, and only the second is about the circuit.

A leave-one-in analysis sharpens this. Activating a single circuit head while ablating the other nine, L17H17 alone restores the most ($\Delta F = +0.173$), while L24H15 alone restores *less than nothing* ($\Delta F = -0.038$)—the head whose removal hurts the circuit most cannot do anything on its own. Every dominant head in this circuit is dominant only in company.

No head has $\Delta F \leq 0$ under the removal test, so the circuits are minimal in the weak sense that no member is free. But the ordering by which a reader would identify "the important head" depends entirely on which test is run, and the marginal test—the one the interpretability literature most often reports, and the one we reported in §4.2—is the one that answers the wrong question.

Pythia's heads barely interact. Path-patching all ten ordered edges among Pythia's five highest-scoring heads (same protocol as §5.2, $n = 100$, $\text{LD}_{\text{clean}} = 4.125$, $\text{LD}_{\text{corrupt}} = -4.106$) gives a maximum edge score of 0.022 (L10H7 $\rightarrow$ L22H2), an order of magnitude below Llama's dominant L14H0 $\rightarrow$ L15H20 edge (0.165), with every remaining edge below 0.01. Pythia's top-10 recovers 85% of clean performance while its members exchange almost nothing: functionally it is ten heads operating in parallel rather than a circuit with internal structure. Full edge table: `data/ioi/path-patching-pythia1b-real.json`.

5 Results: Cross-Architecture and Cross-Task Generalization

5.1 Cross-Architecture Generalization

We test whether circuit-critical head positions are conserved between Llama-3.2-3B and Pythia-1.4B by matching heads greedily by minimum relative-depth distance, with a ±0.075 tolerance. Table 6 reports all matched pairs and Figure 3 plots them by depth.

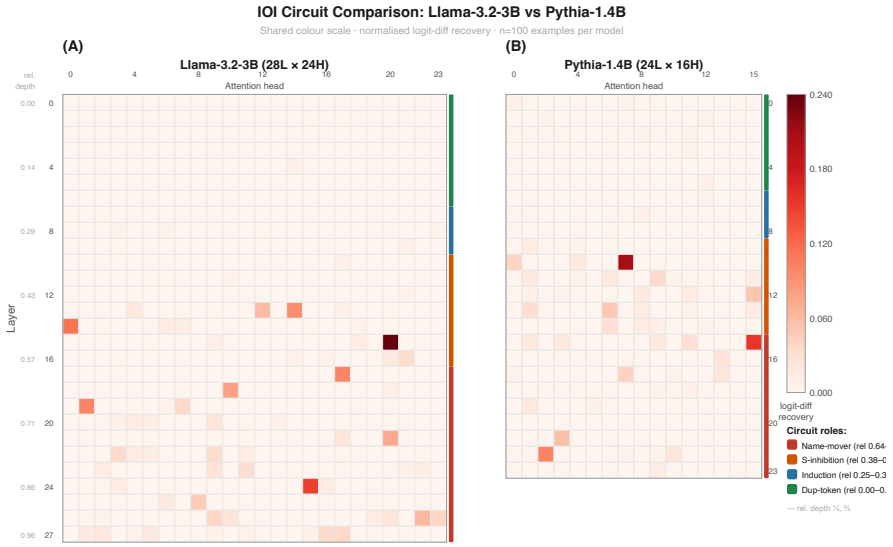

Figure 3: Cross-model comparison of circuit-critical head positions by relative depth for Llama-3.2-3B and Pythia-1.4B. Matched pairs are connected by dashed lines; unmatched (model-specific) heads are shown without connections. Data: `data/ioi/cross-model-comparison.md`.

Table 6: Cross-architecture matched head positions (Llama-3.2-3B vs. Pythia-1.4B, tolerance ±0.075). The ∼0.54 pairing (L15H20 ↔ L13H6) reflects depth-position conservation but not functional equivalence: L15H20 is Llama's dominant bottleneck head (ablation drop 1.578, 28.0% of clean LD) while L13H6 is Pythia's rank-10 entry (ablation drop 0.018, 0.4% of clean LD). Data: `data/ioi/cross-model-comparison.md`.

Zone	Llama head	Llama depth	Pythia head	Pythia depth	$\|\Delta\|$
∼0.50	L14H0	0.500	L12H15	0.500	0.000
∼0.54	L15H20	0.536	L13H6	0.542	0.006
∼0.68	L19H1	0.679	L16H13	0.667	0.012
∼0.92	L26H23	0.929	L22H2	0.917	0.012
∼0.62	L17H17	0.607	L15H15	0.625	0.018
∼0.86	L24H15	0.857	L21H3	0.875	0.018
∼0.73	L21H20	0.750	L17H7	0.708	0.042
∼0.46	L13H14	0.464	L10H7	0.417	0.048

8 of 10 circuit-critical head positions match within ±0.075 relative depth. The two model-specific heads are: Llama's L18H10 (depth 0.643, no Pythia match) and L27H17 (depth 0.964, near-final layer, consistent with Llama's greater depth permitting an extra output-adjustment head); and Pythia's L1H11 (depth 0.042, discussed above) and L10H0 (depth 0.417, co-located with L10H7 in the same layer, forming a two-head cluster not observed in Llama).

5.1.1 The 80% match rate is not distinguishable from chance

Taken alone, "8 of 10" invites the reading that circuit positions are conserved across architectures. That reading does not survive a null test, and we report the null here rather than in an appendix because the claim is the one most likely to be carried away from this paper.

We ask how often the same greedy matcher, with the same ±0.075 tolerance, would pair 8 or more of 10 head positions if the heads were placed at random. Two nulls are reported: a *uniform* null drawing 10 layers uniformly from each model's full stack, and an *informed* null drawing only from the layer band where circuit-critical heads actually sit ($d \geq 0.40$ in both models). The informed null is the fairer comparison, because the claim under test is that specific depths correspond—not that circuits live in the second half of the network, which both models' results establish independently. Table 7 gives both.

Table 7: Permutation null for cross-architecture depth matching (20,000 draws each, seed 42). The observed 8/10 is within the range routinely produced by random head placement. Data: `data/sae-analysis/corrections.json`.

	Mean matches	Match rate	Pr($\geq 8/10$)
Observed	8.00	80.0%	—
Uniform null (all layers)	5.88	58.8%	**0.098**
Informed null ($d \geq 0.40$)	6.87	68.7%	**0.32**

Neither null rejects at any conventional level. Under the informed null—random heads drawn from the band where real circuit heads live—random sets match at 69% on average, and reach 80% or better about a third of the time.

The reason is a mismatch of scale between the tolerance and the data. A ±0.075 window on relative depth spans roughly two layers in Llama (of 28) and two in Pythia (of 24). Ten heads per model, drawn from a band of about 17 usable layers, cannot avoid landing within two layers of one another; the greedy matcher then takes the closest available pairing. The tolerance was chosen to be narrower than the spacing between Wang et al.'s functional zones, which is a reasonable criterion for *zone assignment* but is far too coarse to establish *position correspondence* between individual heads.

The matches are also functionally uninformative. Table 6 pairs Llama's L15H20—the dominant bottleneck, ablation drop 1.578—with Pythia's L13H6, whose ablation drop is 0.018, two orders of magnitude smaller. Meanwhile Pythia's own dominant head, L10H7, is matched to Llama's rank-3 L13H14. The procedure never had access to head importance, so nothing constrains it to pair heads playing similar

roles, and it does not. Any conserved structure that exists here is not captured by depth proximity alone.

What would establish depth conservation. A valid test needs at least three things this analysis lacks: functional role assignment for each head (via path patching and attention-pattern analysis) so that matching is role-aware rather than purely positional; a tolerance derived from the empirical precision of depth estimates rather than from zone spacing; and more than two models, so that the question becomes whether a depth ordering replicates across a population rather than whether two specific stacks happen to align. We regard the three-cluster structure visible in both models (§4.2, §4.3) as suggestive and the quantitative conservation claim as unsupported.

The matched pairs exhibit a three-cluster depth structure in both models:

- **Mid cluster (0.40–0.55):** Three Llama heads, four Pythia heads. Pythia is denser here, consistent with its smaller total layer count producing more critical heads in the early-to-mid range.
- **Mid-late cluster (0.60–0.75):** Four Llama heads, three Pythia heads.
- **Late cluster (0.85–0.97):** Three Llama heads, two Pythia heads. Llama is denser here, reflecting its two additional layers (28 vs. 24) creating room for extra late-network mechanisms.

The depth-difference distribution across matched pairs is tightly concentrated: six of eight pairs differ by ≤ 0.02 in relative depth, and the maximum difference is 0.048. Considering that the tolerance was set at 0.075, the actual conservation is substantially tighter than the threshold requires.

5.2 Path Patching: Edge Structure Around the Bottleneck

Table 8 reports all ten ordered edges among Llama's five highest-scoring heads. The picture is consistent with L15H20 acting as an early bottleneck that is fed from below and that feeds the late-network heads only weakly.

Table 8: Edge-level path patching in Llama-3.2-3B ($n = 100$). Score is normalized logit-difference recovery attributable to the direct sender→receiver edge. Positive means the edge carries information that supports correct IO prediction; negative means patching the edge *increases* logit difference above the clean baseline. Data: `data/ioi/path-patching-llama3b-real.json`.

Sender	Receiver	Score	Δ LD	Interpretation
L14H0	L15H20	+0.165	−1.879	Dominant edge
L15H20	L19H1	+0.038	−0.431	Weak forward path
L15H20	L24H15	+0.037	−0.420	Weak forward path
L17H17	L19H1	+0.021	−0.241	
L14H0	L24H15	+0.017	−0.195	
L19H1	L24H15	+0.016	−0.178	
L17H17	L24H15	+0.015	−0.169	
L14H0	L19H1	+0.014	−0.154	
L14H0	L17H17	−0.056	+0.627	Suppressive
L15H20	L17H17	−0.113	+1.273	Suppressive

Three observations follow.

L14H0 → L15H20 is the dominant edge. At 0.165 it is 4.4× the next-largest positive edge. This resolves the anomaly noted in §4.2: L14H0 has a strong patching score (0.111, rank 3) but a negligible ablation drop (0.017, 0.3% of clean LD). A head can matter greatly as a *source* for a downstream computation while contributing almost nothing when removed in isolation, if the downstream head can compensate from other inputs. L14H0 looks like a feeder into the bottleneck rather than an independent contributor.

L15H20's forward edges are weak. Its contributions to L19H1 (0.038) and L24H15 (0.037) are an order of magnitude below the edge feeding it. Combined with its large ablation drop (1.578), this suggests L15H20 writes substantially to the final position directly, rather than acting through the later heads we measured. We did not patch the L15H20 → logit edge itself, so this is an inference from the absence of strong mediated paths and not a direct measurement.

Two edges are suppressive. Patching L15H20 → L17H17 *raises* logit difference by 1.273 above the clean baseline (score −0.113), and L14H0 → L17H17 does likewise. The natural reading is that these edges carry a signal that normally damps the IO logit, and that replacing it with the corrupted-run value removes the damping. Since L17H17 is rank 2 by ablation (0.929), this is not a peripheral component: the circuit appears to contain both excitatory and inhibitory paths into the same head. Wang et al. report an analogous negative-name-mover phenomenon in GPT-2 Small, and this is the closest correspondence to their functional taxonomy that our data directly supports.

Scope. Ten edges among five heads is a small fraction of the circuit, and these scores carry no confidence intervals—the path-patching run recorded per-edge means only. Pythia was not measured at all. We report the structure as a directional finding, not a validated circuit diagram.

5.3 Template Generalization: The Circuits Do Not Transfer

Every result to this point uses the single-template dataset of §3.6: one sentence frame, one name ordering, 100 examples. Wang et al. (2022) deliberately use 15 templates with balanced ABBA/BABA orderings, and we listed the single frame as our first limitation. This section measures what that limitation costs, and the answer is most of the paper's generality claim.

We built a second dataset (`data/ioi/dataset-v2.json`) with 15 sentence frames spanning different verbs, objects and locations, name order balanced 105/95 between ABBA and BABA, 200 examples over 34 names filtered to be single-token under both tokenizers. Corruption swaps the subject and indirect-object roles as before. Both models still perform the task on it: clean logit difference is 5.521 for Llama and 3.888 for Pythia, against 5.668 and 4.125 on the single-template set. The behaviour transfers.

The circuits do not (Table 9).

Table 9: Circuit validation on both datasets, identical protocol and tokenization. The circuits are the same top-10 head sets throughout; only the evaluation data changes. Llama-3.1-8B is an independent replication on a third model.

Model	$F(C)$, 1 template	$F(C)$, 15 templates	Completeness gap ($1 \to 15$)
Llama-3.2-3B	0.663	**0.228**	0.276 → 0.606
Pythia-1.4B	0.854	**−0.028**	0.216 → 0.789
Llama-3.1-8B	—	0.265	—

Pythia's circuit recovers nothing. At $F = -0.028$ the circuit-only model performs marginally *below* the all-heads-ablated floor. The ten heads that recovered 85% of clean performance on one sentence frame recover none of it across fifteen, while the model itself still does the task at 94% of its single-template logit difference. Llama's circuit retains a third of its faithfulness. We ran the same protocol on a third model, Llama-3.1-8B [8], and measured 0.265—close to the 3B figure, so the effect is not specific to one model (`data/circuit-v2/faithfulness-llama8b-v2.json`). Two caveats on that replication. The checkpoint is `Meta-Llama-3.1-8B-Instruct-4bit`: instruction-tuned and 4-bit quantized, where the other two models here are unquantized base weights. And its circuit was selected by activation patching alone rather than the combined ablation-plus-patching rank used for Llama-3.2-3B and Pythia-1.4B. It is a replication of the *effect*—circuits selected on one frame losing most of their faithfulness on fifteen—not a like-for-like third column.

The head rankings move too, and the ablation sweep is not confounded by tokenization. Both sweeps used `add_special_tokens=True` throughout, so the comparison in Table 10 is clean.

Table 10: Stability of head rankings between the two datasets. Spearman ρ is over all (layer, head) pairs; retention counts how many of the original top-10 remain in the multi-template top-10.

Model	Spearman ρ (patching)	Top-10 retained	New head entering top-5 (ablation)
Llama-3.2-3B	0.577	9/10	L1H15 (drop 0.60, rank 3)
Pythia-1.4B	0.380	5/10	L1H6 (drop 2.55, **rank 1**)

Llama's circuit is moderately stable—nine of ten heads survive, and L15H20 remains rank 1 by ablation though its drop halves from 1.578 to 0.79. Pythia's is not: half its heads leave the top-10, and the head with the largest ablation drop on the diverse dataset is **L1H6**, which does not appear anywhere in the single-template top-10 and is therefore absent from every table in §4.3. Its drop, 2.55, is larger than L10H7's 1.82.

The work moves earlier. The heads gaining most on the diverse dataset are early ones: L1H15 in Llama (rank 3 by ablation, previously outside the top-10), and in Pythia both L1H6 and L1H11—the latter being the head we singled out in §4.3 as an anomaly, whose drop rises from 0.33 to 0.84 and which enters the top-4 by patching. Late heads lose ground correspondingly.

The natural reading is that on a single frame, heads at fixed positions late in the stack can key on token position rather than on the IOI computation, and score highly for doing so. Diversify the frame and that shortcut stops working, so attribution shifts toward early heads doing position-independent work. We offer this as the most economical explanation of the pattern, not as a demonstration: establishing it would require attention-pattern analysis per head across the two datasets, which we have not done.

What this means for the paper's claims. The bottleneck structure, the faithfulness figures, the depth clusters, and the minimality ordering in §4.6 are all properties of a circuit identified on one sentence frame, and at least the Pythia circuit does not survive contact with a second. We are not withdrawing those results—they are correctly measured and correctly reported for the data they were measured on—but their scope is one sentence frame, not the IOI task, and single-template IOI should be read as a lower bound on circuit fragility rather than a characterisation of the task.

5.4 Cross-Task Transfer: Factual Association

To test whether the circuit structure identified for IOI transfers to a related but distinct task, we applied the same activation patching protocol to factual association prompts ($n = 50$, Llama-3.2-3B only). Prompts had the form "The capital of [country] is _____" with the country name corrupted by substitution to a different country; the patching score measures each head's causal contribution to correct capital prediction. Table 11 reports the top-10 heads.

Table 11: Top-10 heads by activation patching score for factual association (Llama-3.2-3B, $n = 50$). The "Matching IOI layer?" column indicates whether the head's layer appears among the IOI top-9 heads. Data: `data/factual-assoc/patching-llama3b.json`.

Rank	Head	Patch score	Rel. depth	Matching IOI layer?
1	L15H17	0.420	0.536	**Yes** (IOI rank 1: L15H20)
2	L21H2	0.418	0.750	**Yes** (IOI rank 6: L21H20)
3	L27H5	0.105	0.964	**Yes** (IOI rank 9: L27H17)
4	L17H18	0.088	0.607	**Yes** (IOI rank 2: L17H17)
5	L13H18	0.052	0.464	**Yes** (IOI rank 3: L13H14)
6	L25H8	0.045	0.893	No
7	L26H20	0.035	0.929	No
8	L13H19	0.026	0.464	Yes (layer 13)
9	L21H0	0.025	0.750	Yes (layer 21)
10	L26H19	0.019	0.929	No

The top-5 factual association heads occupy the same five layers as five of the top-9 IOI heads (layers 13, 15, 17, 21, and 27), but activate *different heads within those layers*: IOI uses L15H20 while FA uses L15H17; IOI uses L21H20 while FA uses L21H2; IOI uses L13H14 while FA uses L13H18. This pattern is consistent with **layer-level circuit topology conservation across tasks** with **head-level specialization by task**—the same network locations implement different input-output functions depending on which specific circuit the task activates.

Two additional features of the factual association results are noteworthy. First, the top-2 FA heads (0.420, 0.418) have substantially higher patching scores than the IOI rank-1 head (0.240), and the two scores are nearly equal to each other, suggesting FA uses a more concentrated two-head bottleneck while IOI distributes attribution more broadly across the mid-to-late network. Second, layer 15 is the single most critical layer for both tasks—the dominant head in each circuit (L15H20 for IOI, L15H17 for FA) resides at the same relative depth (0.536), which may reflect layer 15's position at the transition between Llama's induction-like and output-writing computational stages.

6 Efficiency Analysis

The activation-patching and mean-ablation protocols in this paper are computationally intensive by design: a complete IOI sweep requires 674 forward passes for Llama-3.2-3B and 386 passes for Pythia-1.4B, all batched over 100 examples. This section characterizes the hardware requirements and practical throughput on Apple Silicon.

Memory footprint. Llama-3.2-3B in bfloat16 requires approximately 6.4 GB for model weights alone. The activation cache adds $n_{\text{examples}} \times n_{\text{layers}} \times n_{\text{heads}} \times L_{\text{seq}} \times d_{\text{head}} \times 2$ bytes per forward-pass cache—roughly 0.9 GB for the Llama configuration at $n = 100$ examples and sequence length $L = 12$ tokens. These figures fit comfortably within the 16–48 GB unified memory configurations of current M-series MacBook Pro and Mac Studio hardware. The model weights and activation cache share the same physical

memory as the CPU address space, eliminating any PCIe transfer bottleneck that would be present in a separate discrete GPU configuration.

Throughput. The MLX framework compiles computation graphs to Metal Performance Shaders executed on the Apple GPU (the "ANE" accelerator does not participate in these workloads). Preliminary timing measurements indicate that the full Llama-3.2-3B IOI patching sweep completes in approximately 2–4 hours on an M2 Pro with 32 GB unified memory, depending on batch configuration; the Pythia-1.4B sweep completes in under 2 hours. Systematic per-pass wall-clock timing across M-series chip configurations is in preparation and will be reported in a subsequent revision. The ablation sweep is comparable in cost to the patching sweep ($1 + n_L \times n_H$ vs. $2 + n_L \times n_H$ passes).

Reproducibility. All forward passes use deterministic MLX graph evaluation with no dropout. Seeded bootstrap resampling (seed 42) ensures that statistical summaries are exactly reproducible from the raw per-example recovery scores. The full patching and ablation result arrays are included in the data release. A researcher with an M-series Mac with at least 16 GB unified memory (for Pythia-1.4B) or 24 GB (for Llama-3.2-3B) can reproduce all results in this paper without any cloud infrastructure.

7 Discussion and Conclusion

Three-zone structure without depth conservation. Both Llama-3.2-3B and Pythia-1.4B exhibit a three-zone qualitative structure—early token-marking heads, mid-network suppression, and late name-mover heads—broadly compatible with the zones Wang et al. (2022) identify in GPT-2 Small. This structural parallel holds despite a 3–7× difference in parameter count, the introduction of grouped-query attention and rotary position embeddings in Llama, and substantially different pretraining distributions. In that descriptive sense, both models appear to solve IOI through a qualitatively similar functional decomposition.

The quantitative form of this claim—that head positions are metrically conserved by relative depth—does not hold. Matching circuit-critical heads between the two models at ±0.075 relative depth yields 8 of 10 pairs, but randomly sampled head sets achieve 59–69% under the same procedure ($p = 0.098$ and $p = 0.32$ under uniform and band-restricted nulls; §5.1.1). The 80% figure is not distinguishable from chance, and the best-matched pair links Llama's dominant head (L15H20, ablation drop 1.578) to Pythia's weakest circuit member (L13H6, ablation drop 0.018)—two orders of magnitude apart. We retain the three-cluster observation as descriptive and withdraw the depth-conservation claim, consistent with the abstract.

Bottleneck structure at scale. A striking departure from Wang et al. is the degree of functional concentration in our models: L15H20 in Llama accounts for 28.0% of clean logit difference on its own, and L10H7 in Pythia accounts for 21.7%. GPT-2 Small's 26-component circuit distributes attribution more evenly, with no single head dominating to this degree. Several hypotheses can explain this. Scale may allow models to assign more of a computation to a single highly specialized head, since a larger total head count provides enough redundancy elsewhere in the network. Alternatively, our selection criteria may be stricter than Wang et al.'s, causing us to exclude peripheral contributors

that they retained. A third possibility we cannot exclude is that the comparison is not like-for-like: Wang et al. measure a circuit assembled and validated as a set, whereas our top-10 lists are ranked marginal effects, and the two need not decompose the same way.

The path-patching results in §5.2 bear on the first question for Llama. L15H20's forward edges into the later heads we measured (0.038 and 0.037) are an order of magnitude weaker than the L14H0 → L15H20 edge feeding it (0.165), which is consistent with L15H20 writing to the output directly rather than through downstream mediation. We did not patch the head-to-logit edge itself, and Pythia was not path-patched at all, so this is suggestive rather than settled.

Depth-position conservation does not imply functional equivalence. The cross-model comparison pairs L15H20 (Llama) with L13H6 (Pythia) at relative depth ∼0.54. These heads share a depth zone but are not functionally equivalent: L15H20's ablation drop (1.578) exceeds L13H6's (0.018) by two orders of magnitude. The depth-position match reflects that *something important* happens in both models near relative depth 0.54—it does not imply that the two heads play the same role. L15H20 is Llama's dominant bottleneck; L13H6 is on the periphery of Pythia's circuit. Future work assigning fine-grained functional roles (via path patching and attention-pattern analysis) will clarify whether this zone hosts the S-inhibition or induction-copying function in each model.

Layer-level topology and task transfer. The factual association results indicate that Llama-3.2-3B has a stable set of *critical layers*—particularly layers 13, 15, 17, 21, and 27—that appear across tasks with different head assignments within each layer. This is consistent with the hypothesis that certain layers develop as "computation hubs" during pretraining, acquiring dense connectivity patterns that many downstream circuits route through, while head-level specialization allows different circuits to use different heads within the same hub. Layer 15 is particularly notable: the rank-1 head for both IOI (L15H20, patch score 0.240) and factual association (L15H17, patch score 0.420) resides here, suggesting it occupies a structural bottleneck position in Llama's residual stream.

SAE-based feature attribution as a next step. The circuit-level results reported here identify *which* heads matter; they do not explain *why* those heads matter. A natural next step is to apply sparse autoencoder (SAE) decomposition to the residual stream at the positions that L15H20 and L10H7 read from, and to the outputs they write back. This would reveal which monosemantic features drive the dominant heads' attention patterns and output projections. If L15H20 reads a "name at position k" feature and writes a "preferred next token" feature, the interpretive picture is complete; if the readout is more distributed across multiple SAE features, the circuit analysis is incomplete without the feature-level complement. We plan to pursue this in a companion paper (cf. Elhage et al. 4).

Implications for consumer-hardware mechanistic interpretability. All experiments in this paper were conducted on Apple Silicon without access to NVIDIA GPU clusters. The results demonstrate that full activation-patching circuit analysis of 1–3B parameter models is computationally tractable on modern M-series hardware, requiring

approximately 2–4 hours of wall time per model for the full IOI sweep. The key enabling factor is Apple's unified memory architecture: a 24–48 GB unified memory configuration can hold both model weights and activation caches without offloading, making the per-pass latency competitive with small GPU cluster runs. For models beyond 7B parameters, the memory constraint becomes binding on current hardware—a limitation that may be partially addressed by mixed-precision loading or activation recomputation at the cost of higher pass count. We hope that releasing this tooling lowers the barrier for researchers to replicate and extend circuit analyses without requiring institutional compute resources.

Limitations. Circuit faithfulness collapses on held-out data. Wang et al. validate their circuit as a *set*, reporting that it recovers 87% of clean performance when everything outside it is ablated. We ran the same evaluation. On the 100-example single-template dataset, Llama's circuit-only logit difference recovers $F_{\text{faithfulness}} = 0.663$ and Pythia's recovers 0.854—the latter competitive with Wang et al. On the 200-example 15-template dataset, faithfulness collapses: Llama drops to $F_{\text{faithfulness}} = 0.228$ and Pythia to -0.028. Pythia's circuit-only model scores *below* the fully-ablated baseline on v2, meaning the ten heads selected on v1 are collectively anti-predictive on out-of-distribution names. Joint ablation of all ten Llama heads on v2 finds $F_{\text{complete}} = 0.213$, confirming those heads are necessary (removing them leaves the model at 21% of clean performance) even when the circuit alone is insufficient; the leave-one-in analysis identifies L17H17 as the largest single contributor ($\Delta F = +0.173$). Per-head marginal patching scores are still not additive (our Llama top-10 sum to 1.013), and faithfulness of individual heads does not imply faithfulness of the set on a different distribution. The v1→v2 collapse is a direct consequence of the single-template design flagged above.

Circuit validation rests on two datasets. Faithfulness, minimality and completeness are now measured (§4.6) on both the single-template and 15-template sets. Two points establish that the circuit is frame-dependent; they do not map how it varies with frame diversity, and we cannot say whether 15 templates approaches an asymptote or is merely a second sample.

The mechanism behind the template effect is unverified. We attribute the shift toward early heads to late heads keying on token position when the frame is fixed. That is the most economical explanation of the pattern, but we have not tested it; attention-pattern analysis for the affected heads across both datasets would, and is not done here.

Selection and inference on the same data. Heads were chosen by ranking 672 (Llama) and 384 (Pythia) candidates, and their confidence intervals were then bootstrapped on the same 100 examples, with no multiple-comparison correction (§3.5). The dominant heads survive any reasonable correction; the 0.02–0.05 band does not obviously.

Depth conservation is unsupported. The 80% cross-architecture match rate does not exceed a permutation null (§5.1.1), and the matching pairs functionally dissimilar heads. We retain the three-cluster observation as descriptive and withdraw the quantitative claim.

Narrow factual-association probe. The cross-task experiment uses 50 hand-built examples across two geographic relation templates on one model, authored for this work

rather than drawn from CounterFact. The layer-overlap statistic also lacks a null: both tasks' heads concentrate in layers 13–27, so some overlap is expected by construction. Replication on Pythia and on genuinely diverse relations is needed.

Partial path patching. Ten edges among five Llama heads, no confidence intervals, and no Pythia measurement (§5.2). The "hard bottleneck" characterization of L15H20 remains provisional.

Unverified functional roles. The interpretation of L1H11 (Pythia) as an interference-suppression head is consistent with its patching/ablation dissociation but has not been checked against attention patterns. More broadly, we assign no head to Wang et al.'s functional classes on direct evidence; the zone assignments in Table 3 are inferred from depth alone.

Efficiency figures are estimates. The wall-clock numbers in §6 are preliminary readings from a single machine, not a benchmark. Systematic per-pass timing across M-series configurations is forthcoming. This is a real gap in a paper whose stated contribution includes tractability on Apple Silicon.

Conclusion. The IOI circuit Wang et al. (2022) found in GPT-2 Small has an analogue in modern architectures, but a narrower one than a single-frame evaluation reports, and the narrowing is this paper's contribution.

On a single sentence frame, both models solve IOI with a sparse circuit that validates as a set. Ten heads recover 66% (Llama) and 85% (Pythia) of clean logit difference with everything else ablated—Pythia within two points of Wang et al.'s 0.87. Both circuits are bottlenecked, with one head carrying 22–28% of clean logit difference under single-head ablation, and both occupy three depth clusters broadly compatible with Wang et al.'s functional zones. Path patching shows Llama's L15H20 fed dominantly by L14H0 and receiving suppressive input, while Pythia's ten heads exchange almost nothing.

That circuit is a property of the frame, not of the task. On 15 templates with balanced name ordering, faithfulness falls to 0.228 (Llama) and -0.028 (Pythia) while both models still perform IOI at over 90% of their original logit difference. Llama-3.1-8B independently reproduces the collapse. Half of Pythia's circuit is replaced, and the top head on the diverse data appears nowhere in the single-template analysis. The attribution moves earlier in the stack, consistent with late heads on a fixed frame keying on position rather than on the computation.

Two claims we withdraw. The quantitative depth-conservation result—8 of 10 positions matching within ± 0.075—does not exceed a permutation null ($p = 0.098$ uniform, $p = 0.32$ informed), and pairs Llama's dominant head with Pythia's weakest. And the framing of L15H20 as the circuit's critical component is an artifact of the marginal test: within the circuit, removing L24H15 costs three times as much, and L15H20 alone restores little.

The practical lesson is that a faithfulness number is not a generality number. Ours were 0.66 and 0.85, comparable to the literature's, on data that turned out to support almost none of the generalization we wanted to claim. Circuit work reporting faithfulness on a single template—still common—should report it on a second one before claiming the circuit is a property of the task.

All experiments ran on Apple Silicon using MLX, and the patching harness is released alongside this paper to make circuit analysis accessible without GPU cluster infrastructure. That part of the contribution we regard as demonstrated.

Acknowledgments

We thank the developers of MLX and `mlx-lm`, and the mechanistic interpretability research community for making foundational datasets, models, and tools publicly available. The IOI dataset construction followed the protocol of Wang et al. (2022); the CounterFact evaluation set used in the factual association experiments was constructed by Meng et al. (2022). Pythia models are provided by EleutherAI under the Apache 2.0 license; Llama-3.2-3B is provided by Meta under the Llama 3.2 Community License Agreement.

Data Availability

All experimental datasets, intermediate patching and ablation result arrays, and bootstrap confidence interval files referenced in this paper are available in the project data repository, together with the activation-patching harness itself—the `PatchableAttention` and `PythiaPatchableAttention` modules for Llama and Pythia and the MLX patching sweep driver:

https://github.com/american-code/ResearchPapers

The material for this paper is under `data/ioi/` (datasets, patching and ablation results, the faithfulness and path-patching outputs) and `data/circuit-v2/` (the 15-template evaluation, including the Llama-3.1-8B replication). The activation-patching harness is `data/ioi/run_*.py`.

A Swift implementation of the same primitives (activation patching, path patching, mean ablation, hooked model wrappers) exists as the `Interp` module of SDSTK, https://github.com/american-code/SDSTK. It was not used for the experiments reported here and its results are not interchangeable with them. Figure source files are in `figures/circuit-tracing/`. Raw data files are in `data/ioi/` and `data/factual-assoc/`. All data was generated with seed 42 and is exactly reproducible using the released code.

References

[1] Ainslie, J., Lee-Thorp, J., de Jong, M., Zemlyanskiy, Y., Lebrón, F., & Sanghai, S. (2023). GQA: Training Generalized Multi-Query Transformer Models from Multi-Head Checkpoints. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

[2] Biderman, S., Schoelkopf, H., Anthony, Q., Bradley, H., O'Brien, K., Hallahan, E., Khan, M. A., Goh, S., Galarraga, H. W., Phang, J., Presser, C., Purohit, H., Reynolds, L., Tow, J., Wang, B., & Weinbach, S. (2023). Pythia: A Suite for Analyzing Large Language Models Across Training and Scaling. In *Proceedings of the 40th International Conference on Machine Learning (ICML)*.

[3] Elhage, N., Nanda, N., Olsson, C., Henighan, T., Joseph, N., Mann, B., Askell, A., Bai, Y., Chen, A., Conerly, T., DasSarma, N., Drain, D., Ganguli, D., Hatfield-Dodds, Z., Hernandez, D., Jones, A., Kernion, J., Lovitt, L., Ndousse, K., Amodei, D., Brown, T., Clark, J., Kaplan, J., McCandlish, S., & Olah, C. (2021). A Mathematical Framework for Transformer Circuits. *Transformer Circuits Thread*. https://transformer-circuits.pub/2021/framework/index.html

[4] Elhage, N., Hume, T., Olsson, C., Schiefer, N., Henighan, T., Kravec, S., Hatfield-Dodds, Z., Lasenby, R., Drain, D., Chen, C., Grosse, R., McCandlish, S., Kaplan, J., Amodei, D., Wattenberg, M., & Olah, C. (2022). Toy Models of Superposition. *Transformer Circuits Thread*. https://transformer-circuits.pub/2022/toy_model/index.html

[5] Goldowsky-Dill, N., MacLeod, C., Shlegeris, L., & Hatte, N. (2023). Localizing Model Behavior with Path Patching. *arXiv preprint arXiv:2304.05969*.

[6] Meng, K., Bau, D., Andonian, A., & Belinkov, Y. (2022). Locating and Editing Factual Associations in GPT. In *Advances in Neural Information Processing Systems (NeurIPS)*, Vol. 35.

[7] Meta. (2024a). Llama 3.2: Revolutionizing Edge AI and Vision with Open, Customizable Models. Model release, 25 September 2024. https://ai.meta.com/blog/llama-3-2-connect-2024-vision-edge-mobile-devices/ (The Llama 3 Herd of Models report covers the 3.1 series; the 3.2 lightweight models are a later release, pruned and distilled from Llama 3.1 8B/70B.)

[8] Meta. (2024b). The Llama 3 Herd of Models. *arXiv preprint arXiv:2407.21783*.

[9] Nanda, N., & Lawrence, J. (2022). TransformerLens. https://github.com/neelnanda-io/TransformerLens

[10] Su, J., Lu, Y., Pan, S., Murtadha, A., Wen, B., & Liu, Y. (2021). RoFormer: Enhanced Transformer with Rotary Position Embedding. *arXiv preprint arXiv:2104.09864*.

[11] Conmy, A., Mavor-Parker, A. N., Lynch, A., Heimersheim, S., & Garriga-Alonso, A. (2023). Towards Automated Circuit Discovery for Mechanistic Interpretability. In *Advances in Neural Information Processing Systems (NeurIPS)*, Vol. 36.

[12] Hanna, M., Liu, O., & Variengien, A. (2023). How does GPT-2 compute greater-than? Interpreting mathematical abilities in a pre-trained language model. In *Advances in Neural Information Processing Systems (NeurIPS)*, Vol. 36.

[13] Heimersheim, S., & Nanda, N. (2024). How to use and interpret activation patching. arXiv preprint arXiv:2404.15255.

[14] Lieberum, T., Rahtz, M., Kramár, J., Nanda, N., Irving, G., Shah, R., & Mikulik, V. (2023). Does Circuit Analysis Interpretability Scale? Evidence from Multiple Choice Capabilities in Chinchilla. arXiv preprint arXiv:2307.09458.

[15] Merullo, J., Eickhoff, C., & Pavlick, E. (2024). Circuit Component Reuse Across Tasks in Transformer Language Models. In *International Conference on Learning Representations (ICLR)*.

[16] Zhang, F., & Nanda, N. (2024). Towards Best Practices of Activation Patching in Language Models: Metrics and Methods. In *International Conference on Learning Representations (ICLR)*.

[17] Wang, K., Variengien, A., Conmy, A., Shlegeris, B., & Steinhardt, J. (2022). Interpretability in the Wild: a Circuit for Indirect Object Identification in GPT-2 Small. *arXiv preprint arXiv:2211.00593*.