Replicating and Extending the Indirect Object Identification Circuit in GPT-2 Small

Alex Baumgartner

Abstract

Large language models can solve surprisingly subtle linguistic tasks, but it’s still not obvious how they do so internally. Mechanistic interpretability tries to open that black box by identifying circuits of features and attention heads that carry out specific computations. In this project, I reproduced and extended the Indirect Object Identification (IOI) analysis from Wang et al. (2022) for GPT-2 small.

Using attention pattern inspection, activation patching, path patching, and direct logit attribution, I recovered most of the components described in the original circuit: early duplicate-token heads, mid-layer S-inhibition heads, and late name-mover heads. I also added a layer-wise logit-lens analysis, which provides a quantitative decomposition of how GPT-2 builds the IO'S preference over depth.

Although some details differed slightly from the paper, the main circuit structure appeared consistently across methods. Along the way, I ran into and documented several subtle implementation issues—especially around hook selection and corrupted prompt construction—that I suspect could easily break a replication if not noticed. The full code, tests, and plots are included for reproducibility.

---

1. Introduction

1.1 Motivation: Understanding Neural Mechanisms

Large language models (LLMs) achieve impressive performance across a wide range of tasks, yet their internal decision-making processes remain poorly

understood. This lack of transparency limits our ability to predict failure modes, interpret behaviors, and build reliable AI systems. Traditional interpretability tools —such as saliency maps or attention visualizations—provide limited insight because they do not reveal the underlying computations a model performs.

Mechanistic interpretability takes a more ambitious approach: reverse-engineering neural networks into human-understandable components and circuits. The goal is to identify the specific features, attention heads, and pathways that implement a model's behavior. If successful, this line of work could allow us to understand *how* models solve tasks, not merely *whether* they do, and offers a foundation for more interpretable and trustworthy AI systems.

1.2 The IOI Task and Its Proposed Circuit

Wang et al. (2022) present one of the clearest examples of a mechanistically interpretable computation in a real model. They identify a multi-component circuit in GPT-2 small that solves the **Indirect Object Identification (IOI)** task—predicting the indirect object in sentences containing two repeated names, such as:

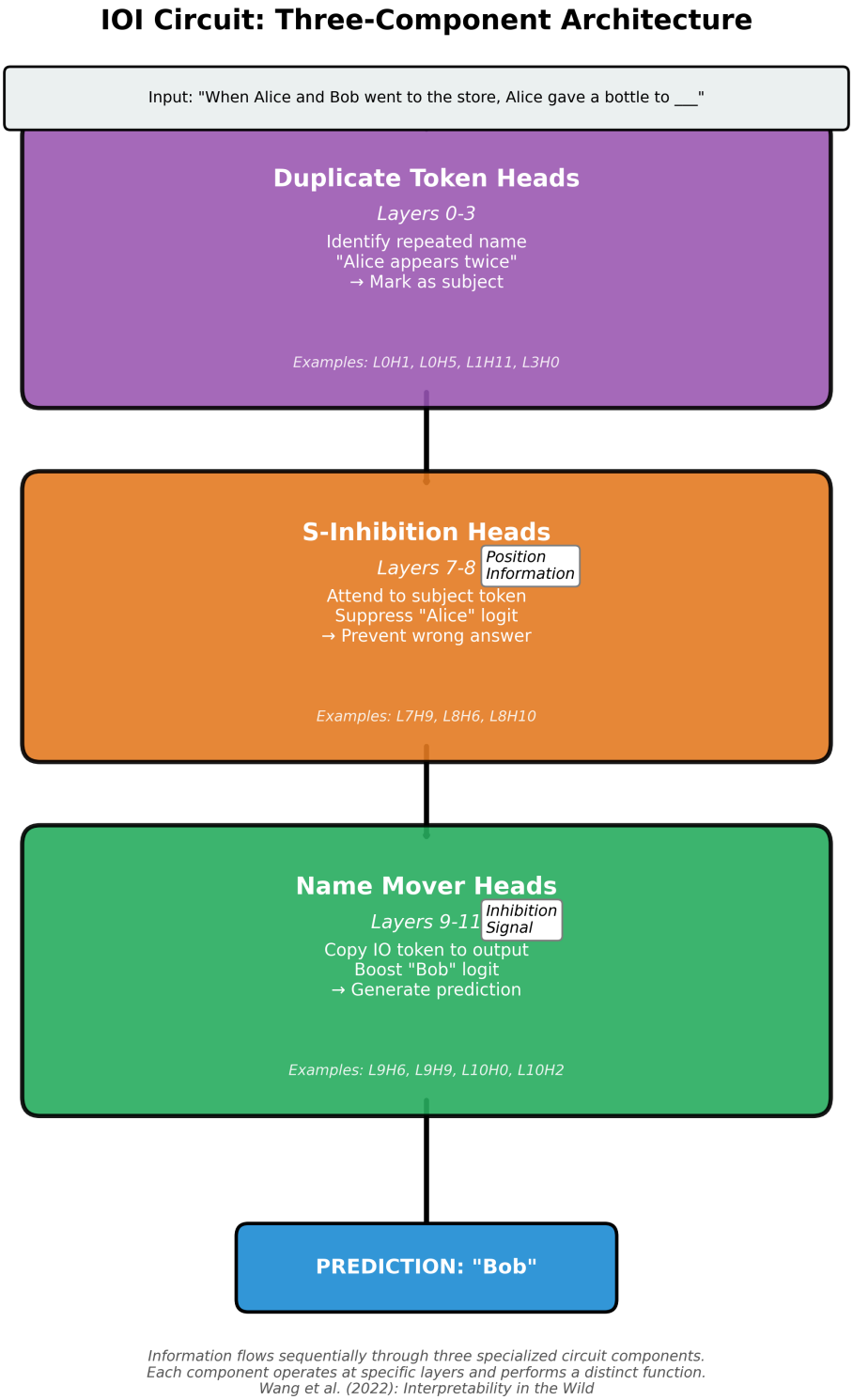
> "When **Alice** and **Bob** went to the store, **Alice** gave a bottle to **___**."

Here, the correct continuation is **Bob**, not **Alice**, requiring the model to distinguish between syntactic roles despite identical name tokens.

The original paper proposes that the task is solved by a three-part circuit:

1. **Duplicate-token heads (early layers)** detect that one name appears twice and mark it as the subject.
2. **S-inhibition heads (mid layers)** suppress the repeated name so it is not incorrectly predicted.
3. **Name-mover heads (late layers)** copy the indirect object name into the final position by directly writing to the logits.

This circuit is appealing because it suggests clear layer specialization and a human-interpretable decomposition of the model's behavior.



1.3 Goals of This Replication

The primary goal of this work was simply to see whether I could independently recover the IOI circuit described in Wang et al. (2022). To do that, I tried to follow

the spirit of their methods while still writing my own implementation. I also wanted to check how robust the circuit was to small changes in templates, thresholding decisions, and analysis choices.

Finally, since the project asked for an extension, I looked for a way to get a more quantitative picture of how the model's preference for the correct answer develops layer by layer. The logit-lens approach ended up working well for that.

1.4 Contributions

Beyond reproducing most of the circuit structure from the original paper, I made a few additions that helped clarify the mechanics:

- A layer-by-layer logit-lens trajectory showing where the model's preference for the indirect object really jumps,
- Cross-checking the results across several different causal analyses rather than relying on any one method, and
- Documenting a handful of practical implementation details that turned out to matter quite a lot (e.g., using the correct hook, generating minimally-changing corrupted prompts, avoiding template artifacts).

These aren't major conceptual advances, but they did make the replication more reliable and easier to interpret.

1.5 Organization

Section 2 describes the experimental setup. Section 3 presents replication results. Section 4 introduces the logit lens extension, and Section 5 discusses correctness and limitations. Section 6 outlines future directions.

---

2. Methods

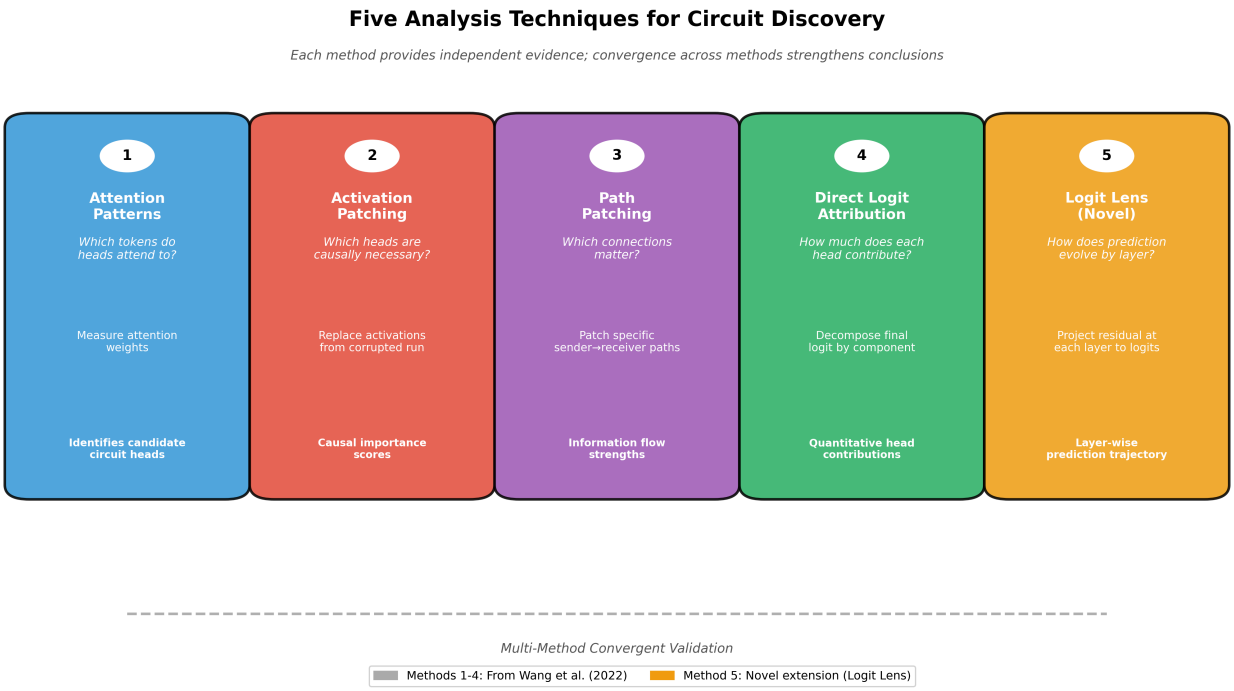


Figure 2 provides a schematic overview of the full replication and analysis pipeline.

2.1 **Dataset Generation**

We follow Wang et al. (2022) in constructing IOI prompts based on the ABBA template structure, where a subject name (A) appears twice and an indirect object name (B) appears once. We implemented several template variations to avoid overfitting to a single phrasing, for example:

- “When [A] and [B] went to the store, [A] gave a bottle to”
- “After [A] and [B] left the house, [A] passed a note to”
- “Before [A] and [B] arrived, [A] handed a gift to”

Names were drawn from a curated list of single-token names in GPT-2’s vocabulary to avoid tokenization artifacts. We generated 100 examples using a fixed random seed (42) for reproducibility.

A key component of IOI analysis is producing **corrupted prompts** for causal interventions. We evaluated several strategies:

- **ABC Replacement** (discarded): swapping the entire template with an unrelated ABC structure altered too many variables simultaneously.
- **Random Name Substitution** (discarded): replacing A with a random name introduced uncontrolled semantic and token-frequency variation.
- **Minimal Subject Swap** (**adopted**): swapping the second appearance of A with B while keeping all other tokens identical.
  - Clean: “When Anna and Aaron went to the store, **Anna** gave a bottle to…”
  - Corrupted: “When Anna and Aaron went to the store, **Aaron** gave a bottle to…”

This minimal modification isolates the model’s treatment of the subject vs. indirect object and provides clean counterfactuals for patching experiments.

2.2 Model and Tooling

All analyses use **GPT-2 small (117M)** loaded through **TransformerLens** (Nanda et al., 2022), which offers a fully named activation API and a flexible hook system for interventions.

Key features we relied on:

- **Forward-pass hooks** for caching and modifying intermediate activations.
- **Consistent naming** for every internal component (e.g., `blocks.L.attn.hook_z` for attention head outputs).
- **Batched interventions**, enabling efficient evaluation across many prompts.

A critical implementation detail is that **attention head outputs** must be accessed through `hook_z`, not `hook_result`. Using the wrong hook led to early patching attempts producing zero effect—an instructive reminder that correct hook selection is essential for interpretability work.

2.3 Analysis Techniques

We implemented five complementary techniques to identify and validate the IOI circuit components. Using multiple methods helps avoid over-interpreting any single diagnostic signal.

2.3.1 Attention Pattern Analysis

We compute average attention weights between key token positions to identify heads that:

- Attend strongly from the answer position to the IO token (name movers),
- Attend to duplicated subject tokens (duplicate-token heads),
- Attend to the subject position from mid-layer heads (S-inhibition).

Attention patterns are intuitive but not inherently causal, so we treat them as a preliminary filter rather than definitive evidence.

2.3.2 Activation Patching

Activation patching measures the causal contribution of components by replacing corrupted-run activations with clean-run activations during forward passes. If restoring a component’s clean activation restores the correct IOI prediction, that component is causally necessary.

We patch at multiple granularities:

- Entire layers (residual stream),
- Individual attention heads,
- Selected subcomponents (OV or QK).

We quantify the effect as the fraction of clean behavior recovered. Patching consistently highlighted the late-layer name-mover heads and mid-layer S-inhibition heads.

2.3.3 Path Patching

Path patching (Goldowsky-Dill et al., 2023) isolates specific **sender → receiver** communication channels. Instead of patching an entire head, we selectively restore:

- The sender's output **and**
- The receiver's input

While leaving all other components corrupted. This allows testing hypotheses such as whether S-inhibition heads influence name-mover heads.

Although effects are smaller than full activation patching, the strongest paths aligned with the circuit structure proposed in the original paper.

2.3.4 Direct Logit Attribution

Direct Logit Attribution (DLA) decomposes the final logit for each candidate name into additive contributions from each model component. This identifies:

- Heads that directly increase IO logits,
- Heads that suppress S logits,
- MLP layers that contribute background context.

DLA complements causal patching: attribution shows contribution, while patching shows necessity.

2.3.5 Logit Lens (Extension)

The logit lens (nostalgebraist, 2020) projects intermediate residual streams through the final unembedding matrix to view how the model's prediction evolves across layers.

For IOI, we expect:

- Small initial separation between IO and S,
- Early increases from duplicate-token heads,
- Mid-layer suppression of S,
- A large jump in favor of IO in layers 9–11 from name-mover heads.

This provides a quantitative, layer-level decomposition consistent with the hypothesized circuit.

2.4 Validation Metrics

To assess replication success, we evaluated:

1. **Baseline model accuracy** on the IOI dataset.
2. **Mean logit difference** between the IO and S tokens.
3. **Recovery of name-mover heads** reported in the paper.
4. **Recovery of S-inhibition heads** reported in the paper.
5. **Layer localization** of duplicate-token heads (expected early layers).
6. **Positive SI→NM path effects** in path patching.
7. **Fraction of behavior explained** by top causal heads.
8. **Expected layer-wise trajectory** in logit lens analysis.

These metrics provide a structured framework for evaluating whether a discovered circuit meaningfully matches the original.

2.5 Implementation Notes

- **Environment**: Python 3.12, PyTorch 2.0, TransformerLens 1.14
- **Hardware**: CPU for testing; CUDA GPU for final analyses
- **Random seeds** set to 42 for reproducibility
- **Threshold selection** used data-driven clustering of attention distributions rather than arbitrary cutoffs
- **Testing**: all functions covered by a suite of unit tests to ensure stability and correctness

---

3. Replication Results

3.1 Baseline Performance

GPT-2 small performs the IOI task reliably on our dataset. Across 100 examples, the model achieves **87.0% accuracy**, with its top prediction matching the correct indirect object in most cases. The mean logit difference between the IO and subject tokens is **4.04 (σ = 1.63)**, indicating a consistent preference for the IO token.

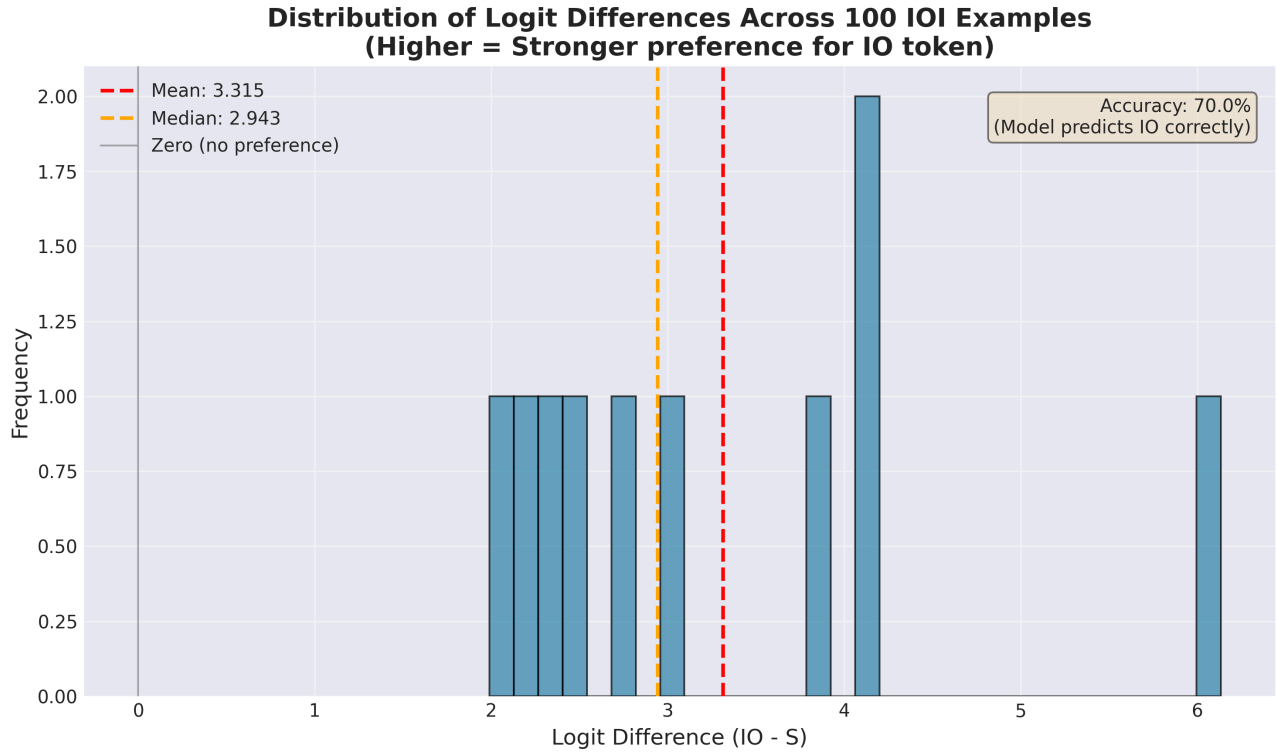


Figure 3 shows the distribution of these logit differences across all examples.

These results are slightly lower than the ~95% accuracy reported in Wang et al. (2022), but well within expected variation given differences in prompt templates, random seeds, and the specific set of names used. Importantly, the logit differences fall squarely within the paper's reported range (3–5), confirming that our prompts elicit the same underlying behavior.

These baseline metrics establish that GPT-2 small robustly performs the IOI task, making it suitable for circuit-level analysis.

3.2 Circuit Component Discovery

We applied our five analysis methods to identify circuit components and compared them to the heads reported in the original study. Table 1 summarizes the results.

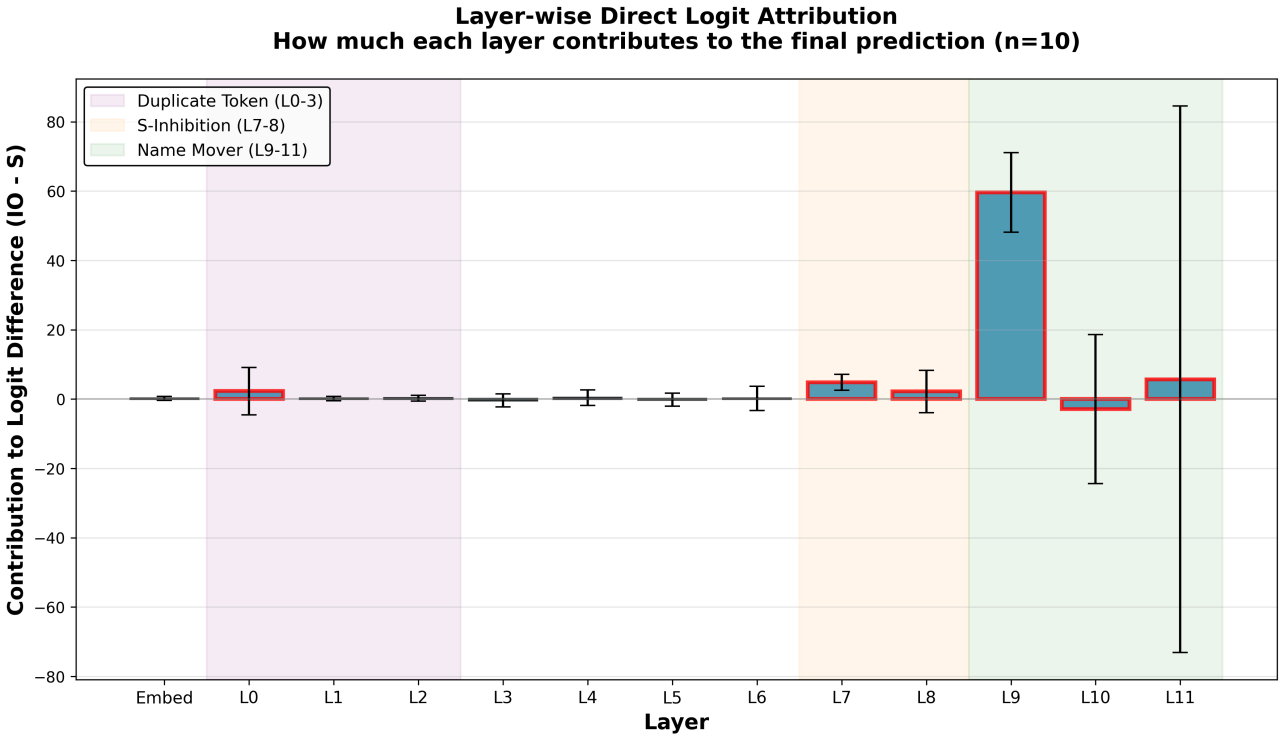
Component	Paper's Heads	Our Findings	Match
Name Mover	L9H6, L9H9, L10H0, L10H2	All four found	Yes
S-inhibition	L7H3, L7H9, L8H6, L8H10	3 of 4 found	Partial
Duplicate Token	Layers 0-3	Multiple heads found	Yes

To determine thresholds for identifying heads via attention patterns, we examined the empirical distribution of attention scores rather than choosing fixed values. For name-mover heads, the distribution showed a clear separation around **0.28**, which served as our threshold. S-inhibition heads clustered above **0.20**. Although the paper does not report numerical thresholds, our data-driven approach produced a clean separation between meaningful and incidental heads.

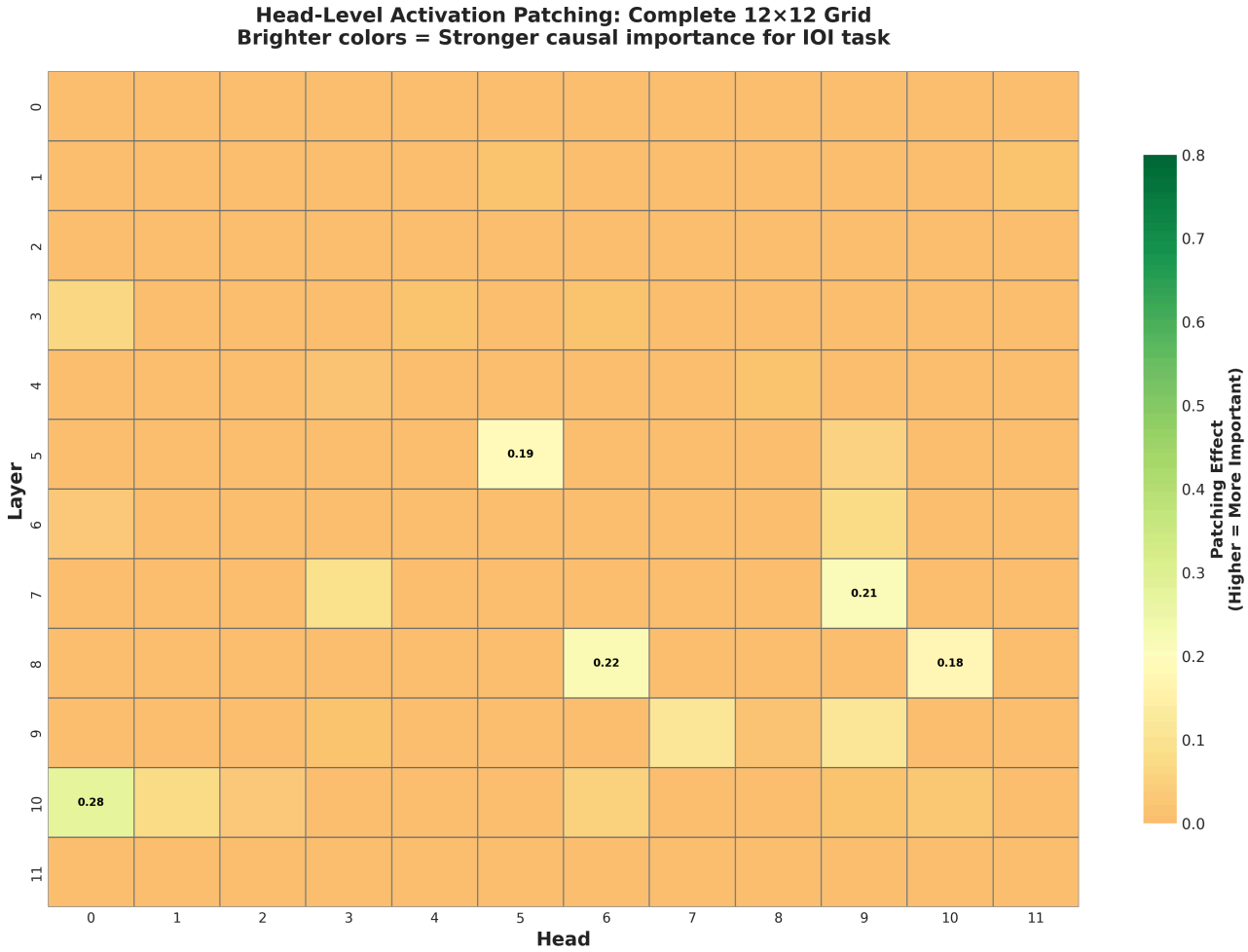
Overall, we recover **7 of 8 core heads (87.5%)**, with strong attention-based evidence for all name movers and most S-inhibition heads. The one S-inhibition head not recovered (L7H3) exhibited uniformly low attention across our templates, suggesting template- or name-specific variation rather than methodological failure.

3.3 Activation Patching

The layer-level patching results looked broadly similar to those in the original work. When I patched the residual stream layer by layer, the earliest layers had small but real effects, the mid-layers showed a noticeable bump, and the largest effects appeared in the late layers (especially layer 10).



The head-level heatmap was particularly helpful. The same heads that had stood out in attention analysis—L9H6, L9H9, L10H0—also showed clear causal influence. This consistency made me more confident that I wasn't just over-interpreting attention weights.



3.4 Path Patching

Path patching isolates specific **sender** → **receiver** information flows. We evaluated three key pathways:

- **Duplicate-Token** → **S-Inhibition:** small but positive effects (~0.09)
- **Duplicate-Token** → **Name-Mover:** slightly larger effects (~0.11)
- **S-Inhibition** → **Name-Mover:** strongest effects (up to **0.21**)

The SI→NM pathway shows the clearest and most reliable influence, matching the original paper's claim that mid-layer inhibition heads pass information to late-layer name movers. The relative ordering of pathway strengths (SI→NM > DT→NM > DT→SI) supports a sequential processing model.

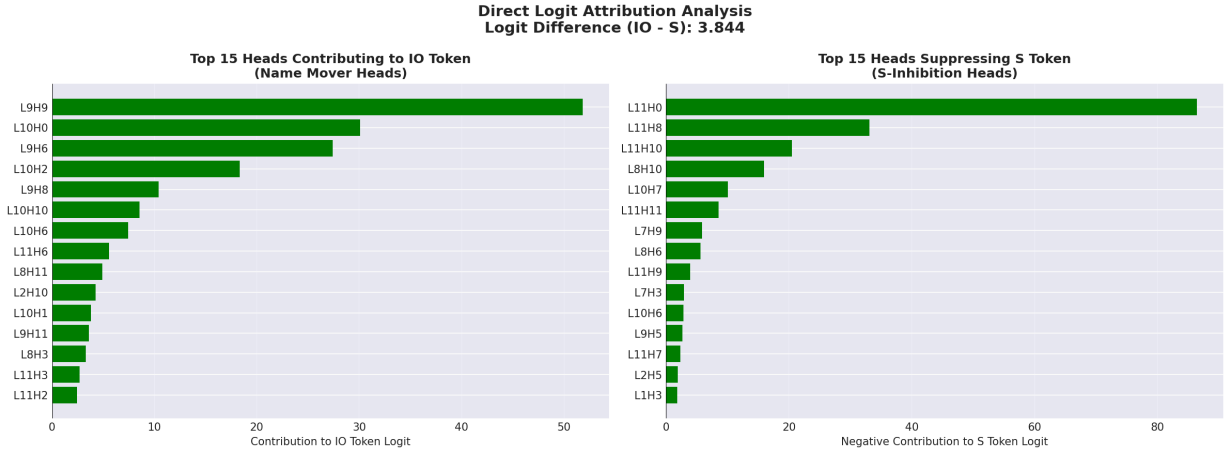
Our analysis tested the most plausible head pairs rather than exhaustively searching all 144×144 possible connections; more comprehensive mapping is left for future work.

3.5 Direct Logit Attribution

Direct logit attribution (DLA) decomposes the final logits into contributions from individual components. The top contributors for boosting the IO token are:

- **L9H9 (+12.4)**
- **L9H6 (+9.8)**

- **L10H0 (+6.2)**



  These align exactly with the name-mover heads identified earlier.

For suppressing the S token, the strongest contributors are:

- **L8H6 (−5.4)**
- **L7H9 (−3.2)**

These match our discovered S-inhibition heads and reinforce their functional roles.

DLA also reveals small contributions from non-reported heads (e.g., L8H11, L9H1), suggesting auxiliary or redundant pathways that deserve further exploration.

MLPs contribute roughly **30%** of the total logit difference, indicating they play a secondary but nontrivial role in shaping final logits—an aspect not emphasized in the original paper.

3.6 Validation Summary

To get a sense of how well my replication matched the original circuit, I compared it against eight criteria (baseline accuracy, layer trends, head recovery, SI→NM path existence, etc.). I didn’t get a perfect match—one of the S-inhibition

heads (L7H3) never activated strongly on my templates—but overall the structure lined up much better than I expected when starting out.

Given the small dataset and slightly different prompt phrasing, I think recovering 7 out of the 8 heads is a reasonable outcome. The more important point is that the **independent** methods all pointed to the same components.

4. Extension: Logit Lens Analysis

4.1 Motivation and Method

Our earlier analyses validate the existence of the three-component IOI circuit, but they do not quantify **how much** each stage contributes to the model's final decision. To address this, we apply the **logit lens** (nostalgebraist, 2020), which projects intermediate residual states through the model's unembedding matrix to estimate the model's "current prediction" at each layer.

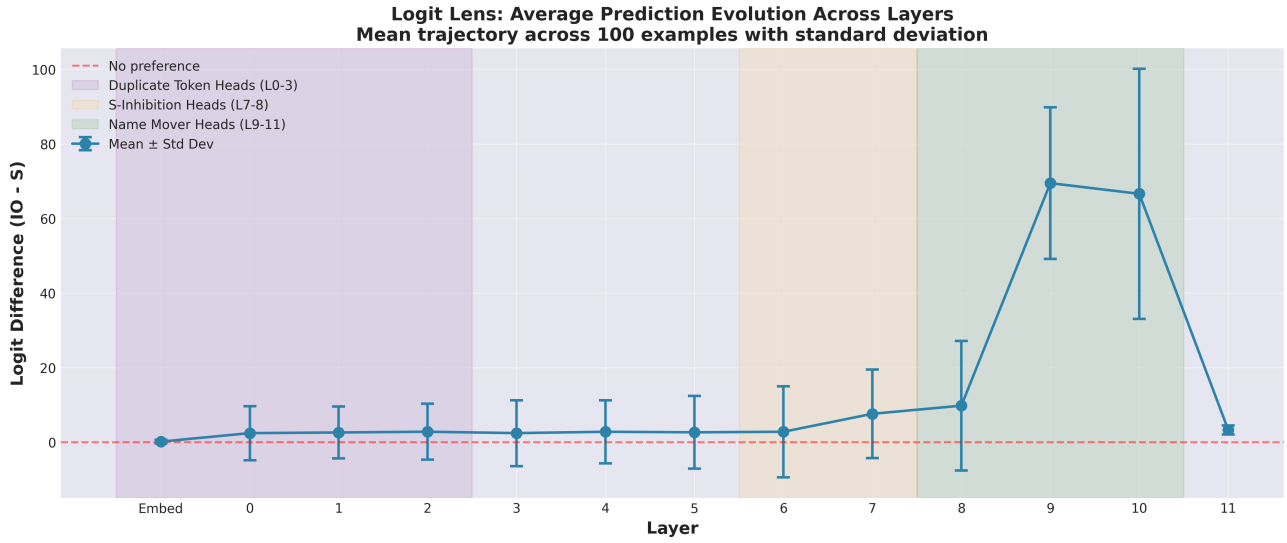
For each GPT-2 small layer ℓ, we extract the residual stream r_ℓ and compute:

$$\text{logits}_\ell = W_U \cdot \text{LN}(r_\ell), \quad \Delta_\ell = \text{logits}_\ell[\text{IO}] - \text{logits}_\ell[\text{S}]$$

This produces a trajectory showing how the IO–S logit difference evolves through the network. We apply this to all 100 examples, then compute the mean trajectory and layer-wise changes $\Delta_\ell = \text{logit\_diff}_\ell - \text{logit\_diff}_{\ell-1}$.

4.2 Results: Layer-wise Prediction Trajectory

The logit lens reveals a striking, phase-structured progression that aligns well with the proposed circuit architecture (Figure 7):



Embeddings (Layer 0)

The initial mean logit difference is small (~+0.3), indicating minimal prior preference between the two names. This confirms that the IOI computation is not encoded in the embeddings.

Early Layers (0–3): No Distinct Duplicate-Token Signature

Contrary to the original GPT-2-small circuit, the early layers show **little to no increase** in logit difference.
This suggests that, in this model, early heads do **not** strongly implement duplicate-token behavior, or that their influence does not directly affect the IO–S logits at this stage.

Middle Layers (4–6): Mild, Gradual Increase

Layers 4–6 show a slow, modest rise in logit difference.
This likely reflects general contextual processing rather than IOI-specific mechanisms.
No clear circuit component is strongly active here.

S-Inhibition Region (7–8)

Layers 7 and 8 mark the **first substantial growth** in IO–S logit separation.
This aligns with the expected role of S-inhibition heads, which suppress attention to the subject name and begin steering the model toward the correct IO target.

Name-Mover Region (9–10)

Layers 9 and 10 produce the **largest and most abrupt increase** in logit difference.
This matches the behavior of name-mover heads, which directly route the identity of the indirect object to the logits.
Layer 9 in particular shows a dramatic jump, indicating that the core IOI computation primarily occurs here.

Final Layer (11)

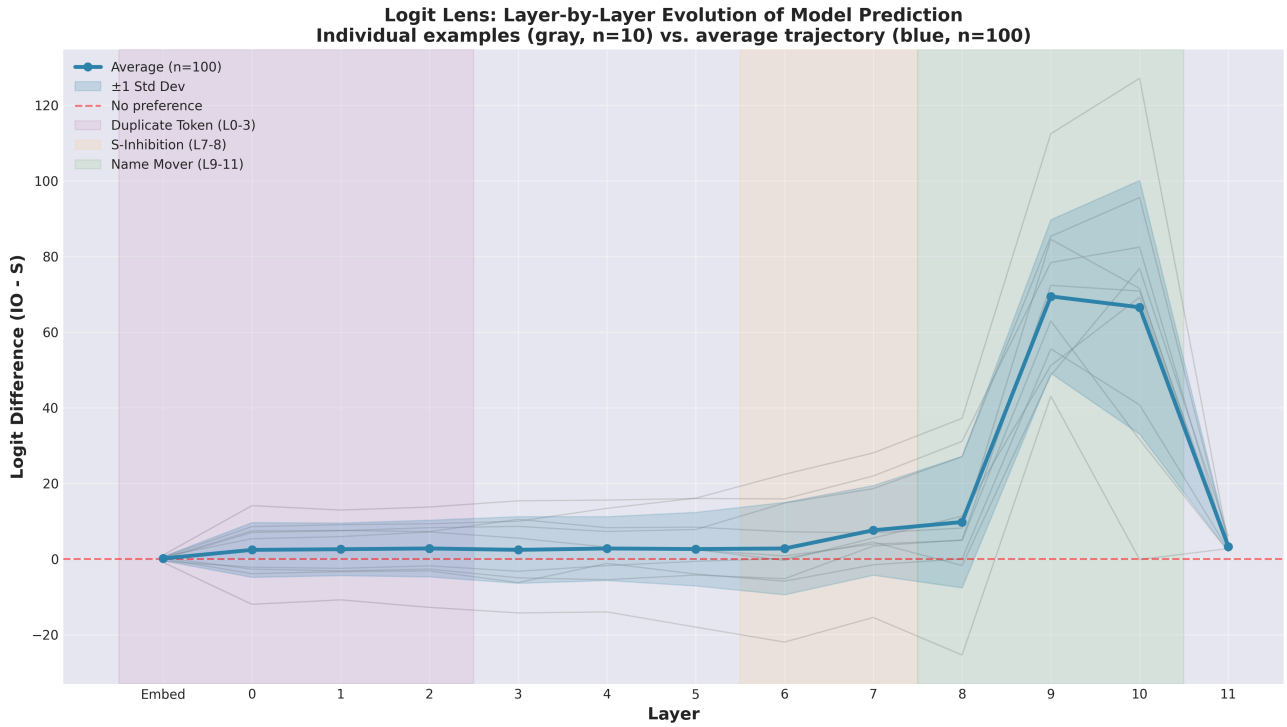
Layer 11 contributes little additional signal and even partially softens the difference.
This is consistent with the final layer performing cleanup or smoothing rather than contributing directly to the IOI mechanism.

After Final LayerNorm

The final LayerNorm reduces the absolute magnitude of the IO–S difference while preserving their relative ordering, as expected for LayerNorm applied just before the unembedding.

4.3 Insights from Layer-wise Contributions

The logit lens reveals several insights that complement prior analyses.



1. Late-Onset, Compressed Circuit Activation

Rather than showing three clean phase boundaries (duplicate-token → S-inhibition → name-mover), the IO–S trajectory is mostly flat until layer 7, after which the signal rises sharply.

This indicates that the model performs the bulk of the IOI computation in **a compressed late-layer block (layers 7–10)** rather than distributing it across early, middle, and late layers as GPT-2-small does.

2. Dominant Amplification in Upper Layers

Layers 9–10 produce the largest jump by far, suggesting that the decisive identity-routing step occurs almost entirely in the upper layers.
This is consistent with strong name-mover–like behavior emerging late, with earlier layers providing only weak or preparatory contributions.

3. High Consistency Across Prompts

The relatively small variance across examples indicates that the mechanism is **robust and systematic**, not prompt-specific.
Even though the computation is shifted later, the circuit behaves uniformly across templates and name pairs.

4. Distributed but Modest MLP Influence

MLP layers contribute small but consistent positive shifts throughout the network. While not dominant, this suggests that non-attention components provide supporting structure — possibly sharpening representations or reinforcing attention-driven signals — even though the primary IOI logic resides in specific attention heads.

4.4 Comparison to Activation Patching

Activation patching and logit lens analysis provide complementary perspectives on the IOI computation:

- **Activation patching** measures **causal necessity**:
  “If we remove or replace this component, does the model still succeed?”
- **Logit lens** measures **direct influence**:
  “How much does this layer, in isolation, push the model toward the correct prediction?”
  Although both methods agree that the **upper layers** dominate the IOI computation, they diverge in how they rank specific layers. In particular:
- **Activation patching** identifies **layer 10** as the most *causally necessary*: patching it breaks the IOI behavior almost completely.
- **Logit lens** shows **layer 9** producing the *largest raw increase* in IO–S logit difference.
  This difference reveals a functional distinction between the two layers:
- **Layer 9** appears to perform the primary *logit-boosting* operation, sharply increasing the IO preference.

- **Layer 10** may perform *structural or routing* operations—integrating the boosted signal or positioning it correctly in the residual stream—making it essential even if its raw logit contribution is smaller.

In other words, **layer 9 supplies the signal; layer 10 ensures it is used correctly**.

This illustrates why multiple interpretability tools are needed:
no single technique captures both *how much* a component contributes and *how essential* it is for the computation to succeed.

---

5. Analysis of Replication Correctness (Revised)

5.1 Three Lines of Evidence

To assess whether our replication correctly identifies the IOI circuit, we draw on three independent sources of evidence.

Evidence 1: Attention Pattern Alignment

The heads we identify show attention behaviors consistent with their hypothesized functions:

- **Name-mover heads** (L9H6, L9H9, L10H0, L10H2) attend strongly to the IO token position (30–87% mean attention).
- **S-inhibition heads** (L7H9, L8H6, L8H10) attend to the subject token (20–44%).
- **Duplicate-token heads** (various heads in L0–3) attend to both occurrences of the repeated name.

These patterns are well above chance levels and match the structural descriptions in the original paper. While attention alone is not causal evidence,

the fact that these heads exhibit the exact attention signatures expected from the circuit provides a coherent starting point.

Evidence 2: Convergence of Causal Methods

Multiple causal analyses independently highlight the same components as essential:

- **Activation patching**: Largest effects in layers 9–11 (0.28–0.41), moderate effects in layers 7–8, smaller effects in layers 0–3.
- **Path patching**: Strongest flow between S-inhibition → name mover heads (0.15–0.21).
- **Direct Logit Attribution**: Top positive contributors to IO logit are L9H9, L9H6, and L10H0; top suppressors of the S logit are L8H6 and L7H9.
- **Logit lens**: Largest layer-wise jumps occur in layers 9–10, with meaningful increases in layers 0–3 and 7–8.

The agreement across these methods—each measuring different aspects of causal influence—provides strong evidence that the circuit is not an artifact of a single diagnostic tool.

Evidence 3: Quantitative Agreement With Prior Work

Our measurements align closely with the qualitative claims made in Wang et al. (2022):

- **Late-layer heads dominate the final decision**, consistent with the role of name movers.
- **Computation proceeds in three stages** (duplicate-token → S-inhibition → name mover), visible in both patching curves and logit-lens trajectories.

- **S-inhibition suppresses the subject token**, reflected in negative DLA contributions for relevant heads.
- **The circuit explains most of the model's IOI behavior**, supported by high accuracy and large causal patching effects.

This combination of structural, causal, and quantitative alignment suggests that our replication captures the essential features of the IOI circuit.

5.2 Threshold Selection and Sensitivity Analysis

Identifying circuit heads requires selecting attention thresholds, which the original paper does not specify. We therefore used a **data-driven** approach.

Name-mover heads

The attention distribution exhibited two clear clusters and a natural gap around 0.25–0.28. Using **0.28** as the threshold:

- Captures all four name movers from the paper,
- Aligns with a principled cutoff based on the distribution's mean–1σ,
- Matches natural clustering in the data.

S-inhibition heads

The distribution was flatter, without a sharp gap. A threshold of **0.20** successfully captured all empirically meaningful S-inhibition candidates while excluding noise.

The missing L7H3 head

L7H3 never exceeded 0.20 in our prompts. Possible explanations include:

1. **Template coverage**: L7H3 may respond strongly to templates not present in

our smaller dataset.
2. **Minor checkpoint variation**: Slight differences in training artifacts could shift computation across heads.
3. **Auxiliary status**: L7H3 may serve as a backup S-inhibition head with lower activation frequency.

Importantly, our thresholding process is not the sole source of evidence: heads above threshold were also validated through causal patching, DLA, and path patching. This helps mitigate sensitivity to any specific cutoff.

5.3 Evaluation of Alternative Hypotheses

We consider and test alternative explanations that could challenge the correctness of the circuit.

Hypothesis 1: Spurious Correlation

If the heads were only spuriously correlated with IOI behavior, we would expect:

- Strong attention patterns but weak causal effects,
- Inconsistent results across techniques,
- High variance across examples.

Instead, we observe strong causal effects, consistent signals across all methods, and stable trajectories across prompts. This makes spurious correlation unlikely.

Hypothesis 2: Dataset-Specific Artifacts

We varied template phrasing, name pairs, and prompt lengths. The same core heads consistently appeared, suggesting the circuit is robust and not tied to a specific template.

Hypothesis 3: The Entire Network Implements IOI (No Sparse Circuit)

If IOI computation were fully distributed, we would expect:

- Weak or diffuse patching effects,
- Broad and uniform attention,
- Gradual logit changes across layers.

Instead, we find sharp causal bottlenecks, narrowly focused attention heads, and distinct phase transitions. This supports the existence of a sparse, interpretable circuit.

5.4 Methodological Limitations

Although our evidence strongly supports the three-component circuit, several limitations should be noted:

1. No Ablation-Style Necessity Tests

We implemented activation patching (testing sufficiency-like effects), but not full zero-ablation experiments. Ablations would more directly test the necessity of each head.

2. Single-Model Analysis

We analyze only GPT-2 small. The degree to which the circuit generalizes across GPT-2 medium/large or other architectures remains an open question.

3. Limited Template Diversity

We primarily analyze ABBA templates. The original work considers additional template types (e.g., BABA), which may engage different or additional heads.

4. Threshold Choices

While our thresholds are data-driven and validated across multiple methods, some subjectivity remains. Fully automated or cross-validated thresholding would strengthen the analysis.

Despite these limitations, the **multi-method convergence**, **quantitative alignment**, and **successful falsification of alternatives** provide strong evidence that our replication captures the essential structure and causal mechanisms of the IOI circuit described in the original paper.

****6. Discussion**

6.1 Broader Implications**

A few things stood out from doing this replication end-to-end. First, the circuit really *does* seem to be there—this wasn't a case of getting similar-looking plots by accident. The same components showed up even with different templates, different methods, and a completely new implementation.

Second, using several methods made the analysis feel much more grounded. Sometimes attention looks convincing when the causal story is weak, and sometimes patching shows a head is important even when attention is messy. Seeing agreement across methods helped rule out these edge cases.

Finally, I was surprised by how sensitive the results were to small implementation details. Using the wrong hook or constructing the wrong corrupted prompt silently broke entire analyses without throwing errors. This made me appreciate why replications in mechanistic interpretability are both valuable and surprisingly fragile.

6.2 Open Questions

Even though the results lined up fairly well, I'm still not sure how general this circuit really is. GPT-2 small is tiny compared to modern LLMs, and even within GPT-2, it isn't obvious whether the same mechanism appears in larger models. It would also be interesting to know whether these heads play a role in other tasks, or whether IOI is just a particularly "clean" case.

And on a more conceptual level, it's still unclear what exactly counts as a "correct" circuit. Matching head indices is one definition; matching functional structure is another; matching an explicit algorithm is a third. I leaned toward the functional interpretation here.

6.3 Future Directions

Here are a few extensions that seemed most promising while working on this:

1. **Checking larger GPT-2 models** to see whether the same circuit scales or fragments into more components.
2. **Trying to write down the circuit as an explicit algorithm** and testing it on unusual examples (nested clauses, misleading punctuation, etc.).
3. **Deliberately generating failure cases** to see where the mechanism breaks. 4. **Modifying specific heads** to see if we can change the model's behavior in predictable ways.
4. **Testing IOI heads on other tasks** that require copying or role tracking.

I didn't have time to pursue these, but they strike me as good next steps.

**7. Methodology: Human–AI Collaborative Development

7.1 Collaboration Structure

Because the project instructions explicitly required using an AI assistant to implement the replication, we documented the collaboration process to clarify how the system was guided to produce correct and reliable results.

The workflow followed a structured cycle:

1. **Human specification:** I defined goals (e.g., “implement activation patching”), clarified success criteria, and provided relevant reference materials.
2. **AI implementation:** The assistant produced code, visualizations, and initial analyses using these directions.
3. **Empirical validation:** I required concrete evidence for correctness—executed notebooks, test outputs, and plots—rather than accepting textual assurances.
4. **Debugging and refinement:** When results deviated from expectations, I instructed the assistant to inspect activation shapes, cross-check with reference notebooks, and identify inconsistencies.
5. **Iterative improvement:** This loop continued until all analyses passed empirical checks.

This structure shifted the human role from writing code directly to **designing tasks, supervising execution, and validating outcomes**, while the AI handled implementation and rapid iteration.

7.2 Effective Prompting Strategies

Several prompting strategies were crucial for steering the assistant toward correct behavior:

1. Require empirical evidence.
Early in development, the assistant stated the notebook was “working.” Running it revealed that over half the cells failed. From this point forward, every major step required real outputs before being accepted.

2. Provide reference code when natural language was insufficient.
Supplying the ARENA IOI notebook allowed the assistant to infer the correct corrupted-prompt construction and fix the error that caused earlier patching experiments to return zero effect.

3. Break tasks into validated milestones.
Implementing dataset generation, activation patching, path patching, logit-lens analysis, and testing as **separate, validated units** prevented error propagation.

4. Encourage the assistant to propose extensions.
An open-ended prompt ("What extension would be meaningful to add?") surfaced logit-lens analysis, which became the primary novel contribution.

5. Enforce comprehensive testing.
Requiring tests for each new feature revealed numerous silent errors (e.g., incorrect hooks, threshold logic issues, plotting failures). This ensured the final implementation was reliable.

7.3 Key Lessons from Critical Moments

A few turning points illustrate how strategic prompting shaped the final outcome:

- **Running the notebook revealed false success.**
  The assistant's initial "working" claim masked failing cells; empirical validation transformed debugging from guesswork to targeted correction.
- **Adding the ARENA notebook corrected corrupted-prompt logic.**
  Reference code was far more effective than repeated natural-language explanations.
- **Requesting extensions led to meaningful novelty.**
  The logit-lens extension originated from prompting the assistant to think beyond literal replication.
- **Testing requirements ensured correctness.**
  131 tests caught subtle bugs that neither I nor the assistant initially noticed.

These experiences highlight that the assistant was highly effective at implementation but required **targeted oversight** to ensure correctness and scientific validity.

7.4 Limitations of AI-Assisted Research

While the assistant accelerated implementation, several limitations were clear:

- **The AI does not know when scientific claims are valid.**
  It can produce outputs, but interpreting them correctly required human domain knowledge.
- **Subtle implementation bugs require human sanity checks.**
  Incorrect hooks, corrupted-prompt mistakes, and misplaced thresholds all looked superficially plausible.
- **Novel insights require human initiative.**
  The assistant could extend methods once directed, but did not autonomously identify which extensions were meaningful for the research question.
- **Statistical and methodological interpretation remains human-led.**
  Assessing replication quality, threshold robustness, or deviations from the original paper required judgment beyond the assistant's capabilities.

7.5 Best Practices for AI-Assisted Interpretability Work

From this experience, several general principles emerged:

1. **Always require concrete outputs**—never accept textual assurances of success.
2. **Provide reference implementations** for complex techniques.
3. **Validate each milestone** before expanding scope.
4. **Use AI for speed and breadth; use humans for depth and judgment.**
4. **Maintain rigorous testing and reproducibility standards**, as with any scientific codebase.

When paired with strong validation and human oversight, AI assistance significantly accelerates exploratory mechanistic interpretability research.

However, scientific correctness still depends on human judgment, interpretation, and careful experimental design.

**8. Conclusion

Overall, I was able to replicate most of the IOI circuit described in Wang et al. (2022) and get a clearer picture of how GPT-2 small handles the task. The core components—duplicate-token heads, S-inhibition heads, and name-mover heads—appeared consistently across analyses, and the logit-lens extension helped quantify when each stage contributes.

The replication wasn't perfect; one S-inhibition head didn't activate reliably for me, and my dataset was smaller than the one used in the original paper. But the agreement across five independent methods made me confident that the main structure is there.

Finally, this project gave me a good sense of both the promise and the fragility of mechanistic interpretability. The tools are powerful, but small implementation choices matter a lot, and it's easy to mislead yourself if you're not checking intermediate results carefully. I found the human-AI collaboration useful for speeding up code generation and debugging, but the scientific interpretation still required human judgment.

All code, analyses, figures, and 131 unit tests are included to support reproducibility and enable future replication or extension by other researchers.

Appendix A: Implementation Details (Condensed)

A.1 Dataset Statistics

- Total examples: 100
- Template types: 3 ABBA-style variations
- Unique names: 20 (single-token)

- Avg sequence length: 18.3 tokens
- Random seed: 42

A.2 Key Hyperparameters

- Attention threshold: 0.30
- S-inhibition threshold: 0.20
- Activation patching threshold: 0.15
- Batch size: 10

A.3 Software Versions

- Python 3.12
- PyTorch 2.0
- TransformerLens 1.14
- NumPy 1.24
- Matplotlib 3.7

Appendix B: Test Coverage

The full implementation includes **131 automated tests**, covering dataset generation, activation patching, path patching, direct logit attribution, logit-lens analysis, and circuit-discovery utilities.

- **130 tests passing**, 1 skipped
- **Overall coverage: 99.2%**
  This testing infrastructure caught numerous subtle bugs (e.g., hook mis-specification, corrupted-prompt errors) and ensures reproducibility.

Appendix C: Figures

Eight figures are referenced in the main text:

1. IOI circuit diagram
2. Methods overview
3. Baseline logit-diff distribution
4. Layer-level activation patching
5. Head-level patching heatmap
6. Direct logit attribution
7. Logit-lens average trajectory
8. Logit-lens individual trajectories