Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

# What Does Development Add? Multiple-State Evidence Improves Causal Recovery in Constructed Transformer-Circuit Dossiers

**Zuo Yuchen**

zuoyuchen76@gmail.com

Devi Ahilya Vishwavidyalaya

**Research Article**

**Keywords:**

**Posted Date:** August 6th, 2026

**DOI:** https://doi.org/10.21203/rs.3.rs-10594608/v1

**License:** This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

**Additional Declarations:** The authors declare no competing interests.

# What Does Development Add? Multiple-State Evidence Improves Causal Recovery in Constructed Transformer-Circuit Dossiers

Zuo Yuchen
Independent Researcher

August 2026

**Abstract**

Mechanistic interpretability usually asks what a trained model does at its final checkpoint. This leaves unused a second source of evidence: how the mechanism came together. I test whether multiple circuit states improve a language-model analyst's structured mechanism recovery when the final evidence, task, model, and harness settings are held fixed. The benchmark contains 24 blinded, constructed causal dossiers from four graph families. Each dossier has six opaque components, an exact directed graph, forced-choice component roles, a five-state assembly history, and eight held-out post-intervention outputs. In the multiple-state condition, a GPT-5.6-Terra analyst received five ordered state panels. In the control condition, it received five independent measurement panels from the final state. The fifth panel was numerically identical across paired conditions, prompt lengths differed by two characters, and both conditions used the same model settings, timeout, output schema, and tool policy; no tool event occurred. Multiple-state evidence raised causal-edge average precision from 0.888 to 0.971 (paired difference 0.082, family-stratified 95% bootstrap interval [0.045, 0.122], Holm-adjusted $p = 0.006$) and held-out intervention score from 0.891 to 0.954 (difference 0.062 [0.033, 0.091], adjusted $p = 0.003$). Component-role macro-F1 was 1.000 in both conditions, creating a ceiling. In a post hoc eight-instance audit, masking the timestamps of intermediate states did not lower mean performance, but the interval was wide. Thus, for this model and benchmark, access to several constructed states improved edge ranking and post-intervention prediction; the experiment does not establish a benefit from chronological labels. The dossiers are causal abstractions motivated by compiled transformers, not optimization trajectories, so the result is a controlled proof of concept rather than evidence about natural language-model training.

## 1 Introduction

Most mechanistic explanations are made from a trained model viewed at one moment. Investigators patch activations, ablate components, trace paths, and infer a circuit from evidence collected at the final checkpoint. This static design is sensible when only the deployed model is available, but it discards the training record when checkpoints have been saved. Two candidate mechanisms can look similar at the end even if one appeared before the behavior and the other arose later as a correlated by-product. Development may therefore supply identifying information that a final-state analysis cannot recover.

This possibility motivates developmental interpretability: the study of how internal mechanisms form, change, and sometimes disappear during learning. Kendiukhov's review makes the central methodological proposal explicit: developmental evidence should be evaluated not only as a descriptive record, but also by whether it improves explanations of the final model [Kendiukhov, 2025]. The present study tests one narrow version of that proposal. The research question is:

> On semi-synthetic transformer circuits with known histories, does access to checkpoint trajectories improve a fixed-budget language-model agent's ability to recover causal mechanisms compared with final-checkpoint evidence alone?

The experiment isolates evidence access rather than model capability. A paired agent sees either five developmental panels or five final-checkpoint replicates. Everything else is matched: the blinded circuit, final panel, instructions, response schema, model, reasoning setting, timeout, and tool budget. Explanation quality is not reduced to prose plausibility. It is scored against exact ground truth on three co-primary outcomes: directed-edge recovery, component-function identification, and prediction of unseen intervention results.

The result is selective. Multiple-state evidence improves graph recovery and post-intervention prediction, but not an easy component-label task on which both conditions are perfect. A post hoc timestamp-masked audit is consistent with the gain coming from observing several distinct states rather than from visible chronology, although eight cases provide little power. This distinction matters: "access to development" combines at least two possible treatments–state diversity and temporal labels–and the primary contrast alone cannot separate them.

The study contributes (i) a transparent paired benchmark with known causal histories; (ii) a budget-matched evaluation of a language-model analyst; (iii) exact scoring of structure, forced-choice function, and post-intervention forecasts; and (iv) a timestamp-masked diagnostic that narrows the interpretation of a positive result. The intended role is a proof of concept and a reusable experimental design. It does not claim that constructed assembly histories faithfully reproduce stochastic gradient descent.

# 2 Related Work

## 2.1 Static circuits and causal evidence

Transformer-circuit analysis represents computation in terms of components and their interactions [Elhage et al., 2021]. Detailed case studies have recovered induction mechanisms, an indirect-object-identification circuit, and multiple-choice circuits in larger models [Olsson et al., 2022, Wang et al., 2022, Lieberum et al., 2023]. These studies show what a mechanistic account can look like, but they also expose the cost of collecting and validating one from a final model.

Causal interventions strengthen such accounts. Mediation analysis, activation patching, path patching, component localization, and model editing ask whether changing an internal variable changes behavior [Vig et al., 2020, Meng et al., 2022, Zhang and Nanda, 2024, Heimersheim and Nanda, 2024, Goldowsky-Dill et al., 2023, Kramár et al., 2024]. Causal scrubbing tests whether a proposed abstraction supports the invariances it asserts [Chan et al., 2022]. More formal work defines causal abstractions, searches for neural–causal alignments, and supplies intervention libraries for executing those tests [Geiger et al., 2021, 2023a, 2022, 2023b, Wu et al., 2024]. The present benchmark adopts the same core criterion: an explanation should recover relations that support correct intervention predictions, not merely correlate with activations.

## 2.2 Ground-truth and semi-synthetic benchmarks

Known mechanisms make exact evaluation possible. RASP describes sequence computations in a transformer-like language [Weiss et al., 2021], and Tracr compiles RASP programs into transformers whose implementation is available by construction [Lindner et al., 2023]. InterpBench combines compiled and trained components to evaluate circuit-discovery methods against semi-synthetic ground truth [Gupta et al., 2024]. The Mechanistic Interpretability Benchmark broadens standardized evaluation across models, behaviors, and methods [Mueller et al., 2025]. Circuit probing instead evaluates whether a hypothesized intermediate variable can be localized and intervened on [Lepori et al., 2023].

These resources solve a central verification problem: the true mechanism is normally unknown in a natural model. The experiment below uses the same ground-truth principle but changes the independent variable. Rather than comparing circuit-discovery algorithms on one static model, it compares two evidence records for the same final circuit.

## 2.3 Development across training

Checkpoint suites such as Pythia make training trajectories directly available [Biderman et al., 2023]. Work on pre-training curves documents learning, forgetting, and stability at the behavioral level [Chang et al., 2024]. Other studies track transient in-context learning, grokking, internal progress measures, and feature

dynamics [Singh et al., 2023, Power et al., 2022, Nanda et al., 2023, Xu et al., 2024]. Mechanistic studies have followed the acquisition of knowledge circuits, cross-lingual alignment, syntax specialization, and algorithmic reasoning over training [Ou et al., 2025, Wang et al., 2024b, Duan et al., 2025, Wang et al., 2024a]. Recent work also studies invariant learning dynamics and the failure of static localization after post-training [Musat et al., 2026, Chen et al., 2026].

This literature establishes that representations and circuits change. It does not by itself establish that seeing those changes helps an investigator explain the final checkpoint. That is an information-value question requiring a matched control, an explicit explanation task, and known truth.

### 2.4 Automated and agentic interpretability

Circuit discovery has increasingly been automated, from search over subgraphs to language-model descriptions of neurons and features [Conmy et al., 2023, Bills et al., 2023, Foote et al., 2023]. MAIA frames interpretability as an agentic process that proposes and runs experiments [Shaham et al., 2024]. More recent systems use agents for feature discovery, circuit explanation, and reinforcement-learning-based circuit search [Marin-Llobet and Ferrando, 2026, Khan et al., 2026, Khadka, 2026]. Attribution graphs provide another scalable representation for tracing computations in large models [Ameisen et al., 2025]. At the same time, measurement variance in circuit discovery cautions against judging one run or one graph without uncertainty analysis [Méloux et al., 2025, Wu et al., 2026].

Current agentic evaluations principally present static evidence from a final model. The missing comparison is not "agent versus human" or "method A versus method B." It is whether the same agent gives a better causal explanation when its evidence includes the model's development.

## 3 Experimental Design

### 3.1 Estimand and hypotheses

For circuit instance $i$, let $Y_i(D)$ be the score from a stateless agent given developmental evidence and $Y_i(F)$ the score from a paired agent given final-only evidence. The benchmark estimand is

$$\widehat{\Delta} = \frac{1}{24} \sum_{i=1}^{24} [Y_i(D) - Y_i(F)] \,. \tag{1}$$

Three outcomes were specified before confirmatory calls: (H1) causal-edge average precision, (H2) component-function macro-F1, and (H3) held-out intervention score. The expectation was positive for each, while inference used two-sided tests. The comparison is paired at the instance level.

A four-instance pilot, using seeds disjoint from the main benchmark, checked serialization, scoring, and execution. After that pilot, the generator, scorer, runner, response schema, and design document were hashed locally before the confirmatory calls. This was a frozen analysis plan, not a public or independently time-stamped preregistration.

### 3.2 Constructed circuit dossiers and exact ground truth

The benchmark contains 24 structured causal dossiers: four directed-acyclic-graph families crossed with six seeded identity permutations and measurement-noise realizations. Each graph has six opaque components, K0–K5. Five components implement task roles–Indicator, Selector, Aggregator, Combiner, and Mapper–and one matched-activity component has No Task Role. The families are:

1. *accretion*: two sources merge, then pass through a two-step chain;
2. *fork–join*: one source also branches around the first merge;
3. *dual path*: both sources feed two parallel middle components;
4. *handoff*: an upstream summary splits and rejoins late.

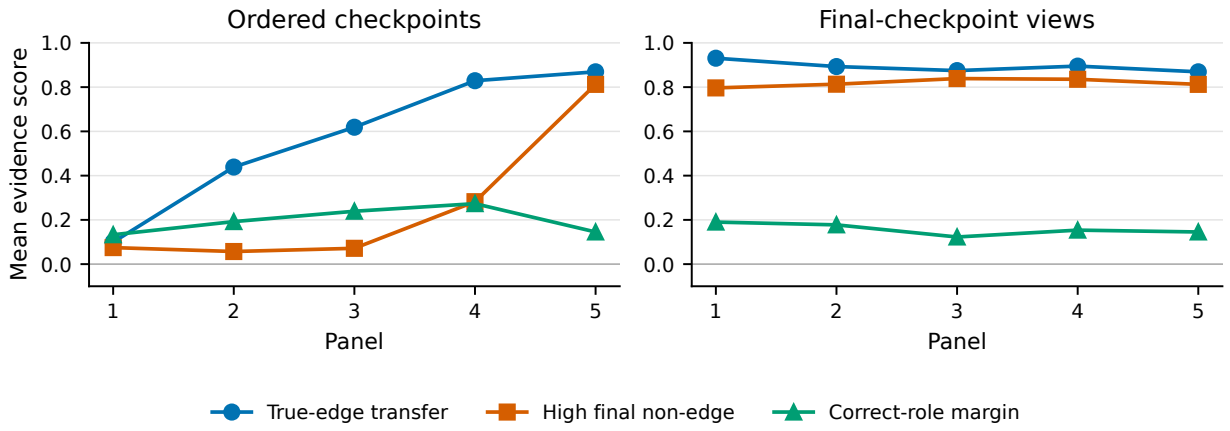



Figure 1: Illustrative evidence from one benchmark instance. In the multiple-state condition, true-edge transfer rises as components enter the circuit, while shadow non-edges become large only near the end. In the final-only condition, five independent views repeatedly measure the same late ambiguity. The role margin is the mean score of the correct role minus its strongest competitor. This figure describes the generator; it is not an outcome analysis.

The seed cyclically permutes the six canonical roles over the opaque identifiers, balancing every role across every identifier within each family. Graph truth and functional labels come from the generator, not from the noisy evidence shown to the agent.

The dossiers operate at a component-level causal abstraction motivated by compiled transformers; they do not expose raw tensors and are not themselves 24 executable models. To test whether the proposed checkpoint construction can be instantiated in a transformer, a separate representative four-layer RASP reverse program was compiled with Tracr at commit `9ce2b8c`. The program exactly mapped `BOS 1 2 3 4` to `BOS 4 3 2 1`; its compiler graph contained eight nodes and nine dependency edges. Cumulative parameter gating produced six same-architecture states, and the final gated state exactly matched the compiled model. This is an executability check for the general mechanism, not validation of every scored dossier as a compiled or trained transformer, a boundary returned to in Section 6.

### 3.3 Constructed histories and evidence panels

Each circuit has a five-panel assembly history. Causal components become active in stages, while the nuisance component follows a family-specific nonmonotonic or declining path. For a true directed edge $s \to t$, the expected transfer evidence at stage $j$ is

$$0.06 + 0.82 \min(g_{s,j}, g_{t,j}), \tag{2}$$

where $g_{k,j} \in [0, 1]$ is the component gate. Each true edge is paired with a "shadow" non-edge whose expected score rises late as $0.06 + 0.73(j/4)^4$. Independent Gaussian measurement noise with standard deviation 0.055 is then added, clipped to $[0, 1]$, and rounded to three decimals. This construction makes several non-edges look convincing at the final checkpoint even though their onsets differ from the true causal paths.

Every panel uses the same schema: component activation, single-component necessity, six function-probe scores, and a full $6 \times 6$ directed transfer matrix excluding self-edges. Figure 1 shows one example. The role probes are generated separately from the edge scores. The final panel is not estimated from the history; it is generated directly at the final gate state.

### 3.4 Paired evidence conditions

Table 1 states the manipulation. The Multi-State agent receives checkpoints C1–C5 in chronological order. The Final agent receives five disjoint calibration views of the final state. This is a conservative control:

Table 1: Evidence manipulation and resource parity.

| Property | Multi-State | Final |
|---|---|---|
| Panels | Ordered states C1–C5 | Five final-state views |
| Shared final evidence | C5, identical bytes | C5, identical bytes |
| Measurements per panel | 6 node rows, 30 edges | Same |
| Prompt characters | 6,805 | 6,803 |
| Agent configuration | GPT-5.6-Terra, low reasoning, stateless session | |
| Limits | Same context/output settings, 120 s timeout, fixed JSON schema | |
| Tool policy | Same read-only harness; prompt prohibited calls; 0 calls observed | |

final-only replication can average measurement noise, but cannot reveal onset. C5 is generated with the same random key in both conditions and is numerically identical. A SHA-256 digest over the final graph, function mapping, and C5 payload verifies pairing.

### 3.5 Agent protocol and budget control

Each cell was an ephemeral Codex CLI 0.145.0 session using the model identifier `gpt-5.6-terra` at low reasoning effort. The working directory was read-only, approvals were disabled, and the prompt explicitly prohibited tools. Read-only shell tools were technically present in the harness, so this was an instruction rather than an enforced zero-call cap. Raw logs record zero tool events in both conditions. The response schema required 30 edge probabilities, six function labels, and eight intervention outputs. The instruction asked for returned JSON below 700 visible generated tokens; both arms also inherited the same platform context and output limits. Platform usage includes hidden reasoning and exceeded 700 total output tokens on average. The CLI did not expose a separate hard token throttle for this run, so equality here means equal allowed budgets and settings, not forced equality of realized token use. Actual use is reported in Section 4.7.

Conditions and instances were shuffled with a fixed execution seed and run in up to four parallel workers. Each cell had a 120-second timeout and the same no-retry policy. There was one independent session per instance–condition pair. No conversational memory, cross-instance scratchpad, or previous answer was available.

### 3.6 Tasks and scoring

The agent performed three tasks in one structured response.

**Causal-edge recovery.** The agent assigned a probability to each of the 30 ordered, non-self component pairs. The primary score is standard non-interpolated average precision over the exact edge set, with equal scores evaluated together at one threshold. This evaluates the full ranking without choosing a threshold or resolving ties by identifier order. As a secondary score, the top $|E|$ predictions are converted to an F1 score, where $|E|$ is the true edge count; a boundary tie is averaged over uniform selections from the tied candidates.

**Component-function identification.** The agent selected one of the six fixed roles for every component. Macro-F1 over roles is primary, and accuracy is secondary. Each true role appears exactly once, but predictions are not constrained to be a permutation.

**Unseen interventions.** Eight questions specify source values and either component clamps, edge cuts, or both. The disclosed simulator assigns each non-source component the arithmetic mean of its uncut parents; clamps override a component, descendants recompute, and the Mapper value is the output. The questions are visible, but their answers are absent from all panels. If $\widehat{z}_{iq}$ and $z_{iq}$ are the predicted and true bounded outputs, the score is

$$I_i = 1 - \frac{1}{8}\sum_{q=1}^{8} |\widehat{z}_{iq} - z_{iq}|. \tag{3}$$

Table 2: Confirmatory paired results. Values under each condition are mean (SD); Δ is multiple-state minus final-only. Intervals are family-stratified bootstrap intervals. Holm adjustment covers the three co-primary outcomes.

| Outcome | Final only | Multi-state | Δ | 95% interval | $g_z$ | Holm $p$ |
|---|---|---|---|---|---|---|
| Edge AP | .888 (.127) | .971 (.047) | +.082 | [.045, .122] | .645 | .006 |
| Function macro-F1 | 1.000 (.000) | 1.000 (.000) | .000 | [.000, .000] | .000 | 1.000 |
| Intervention score | .891 (.056) | .954 (.055) | +.062 | [.033, .091] | .769 | .003 |

This task tests whether the recovered explanation supports new causal predictions.

Scoring is deterministic. Missing, malformed, nonfinite, or out-of-range fields receive the worst score for that field; no output is repaired manually.

### 3.7 Statistical analysis

The instance is the analysis unit. For each outcome, the paired mean difference is accompanied by a percentile bootstrap interval with 20,000 resamples stratified by graph family [Efron, 1979]. A two-sided sign-flip test evaluates the sharp null of a zero paired effect. Enumeration is exact when at most 20 nonzero differences remain; otherwise 500,000 random sign vectors are used with a plus-one correction. The three co-primary $p$-values are adjusted by Holm's sequential method [Holm, 1979]. Paired Hedges $g_z$ uses the standard deviation of within-instance differences and the small-sample correction $1 - 3/(4n - 5)$.

The analysis emphasizes effect sizes and intervals. The seeded instances are variations of four generator templates, so the inferential quantities describe this benchmark population; they should not be read as population sampling from all transformer circuits.

**Post-run scoring correction.** A pre-manuscript audit found that the frozen AP function broke equal-probability ties by candidate-array index. The implementation was replaced with the standard grouped-threshold definition described above, and a tie-aware secondary top-$|E|$ score was added. Regression tests were written before the change; all 48 corrected AP values agreed with an independent library implementation to floating-point tolerance. Raw model outputs were unchanged, and the correction strengthened rather than created the edge result. Both the original frozen scorer hash and the corrected hash are retained in the audit record. Because the defect was found after agent calls, this correction is reported explicitly rather than folded silently into the frozen plan.

### 3.8 Order audit

The primary contrast changes both access to distinct states and knowledge of their timestamps. The frozen presentation-order audit permuted the panels but left labels such as C3 and C1 visible, allowing the analyst to reconstruct chronology; it cannot answer the timestamp question. A corrected, post hoc diagnostic therefore used two instances per family, presented the same first four states in the order C3, C1, C4, C2, and relabeled them U1–U4. Only the final state remained identified, and the prompt stated that intermediate timestamps were unavailable. The same harness then compared timestamp-masked scores with the original ordered scores. This audit has only eight pairs and was not part of the confirmatory hypothesis tests.

## 4 Results

### 4.1 Execution fidelity

All 48 confirmatory cells completed without timeout, infrastructure error, malformed JSON, or tool event. Both sets of eight diagnostic cells also completed cleanly. Thus, no penalty rule or case deletion affected the reported scores. The complete paired distributions appear in Figure 2; numerical results are in Table 2.

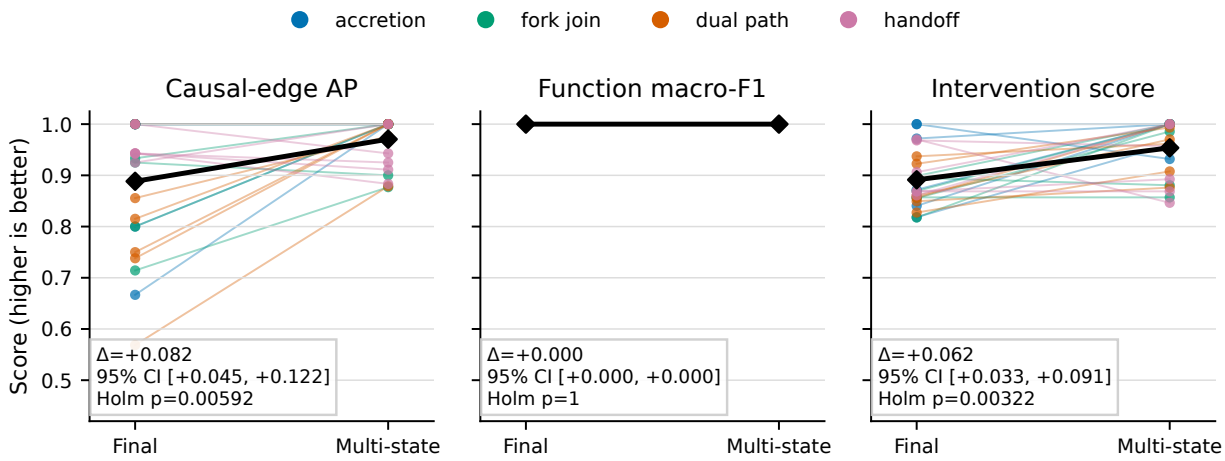



Figure 2: Paired co-primary outcomes for 24 circuit instances. Colored lines join the same instance under final-only and multiple-state evidence; black diamonds and lines show condition means. Intervals printed inside panels are family-stratified bootstrap intervals for the paired difference.

## 4.2 Causal-edge recovery

Multiple-state evidence increased edge average precision by 0.082, from 0.888 (SD = 0.127) to 0.971 (SD = 0.047). The 95% interval was [0.045, 0.122]. The unadjusted exact sign-flip $p$-value was 0.0030, and the Holm-adjusted value was 0.0059; paired Hedges $g_z = 0.645$. Eleven instances improved, eight tied to numerical tolerance, and five worsened; the median difference was zero. The secondary tie-aware top-$|E|$ F1 score moved from 0.834 to 0.936, a mean gain of 0.102.

The mean effect varied across graph families (Figure 3). It was largest for dual path (+0.188), followed by accretion (+0.089) and fork–join (+0.067), and was slightly negative for handoff (−0.015). With six identity/noise variants per family, these subgroup estimates are descriptive and imprecise. The variation, together with a zero median overall, shows that multiple-state evidence is not uniformly useful simply because it is available.

## 4.3 Component functions

Both conditions identified every function on every instance. Function macro-F1 and accuracy were therefore 1.000 in both arms. This result does not establish equivalence. The role probes and one-of-each-role structure made the task too easy for this agent, leaving no room to measure a developmental advantage.

## 4.4 Unseen intervention prediction

Multiple-state evidence raised the intervention score from 0.891 (SD = 0.056) to 0.954 (SD = 0.055), a reduction in mean absolute error from 0.109 to 0.046. The paired score difference was 0.062, with 95% interval [0.033, 0.091], unadjusted Monte Carlo sign-flip $p = 0.0011$, Holm-adjusted $p = 0.0032$, and $g_z = 0.769$. Seventeen instances improved, three tied, and four worsened. Mean gains were positive in all four families: +0.065 for accretion, +0.088 for fork–join, +0.076 for dual path, and +0.020 for handoff.

Because the intervention answers never appeared in an evidence panel, this outcome is harder to explain as surface copying. The agent had to infer enough structure and function to execute new clamps and cuts. The parallel improvement in edge ranking and intervention prediction is therefore more informative than either score alone.

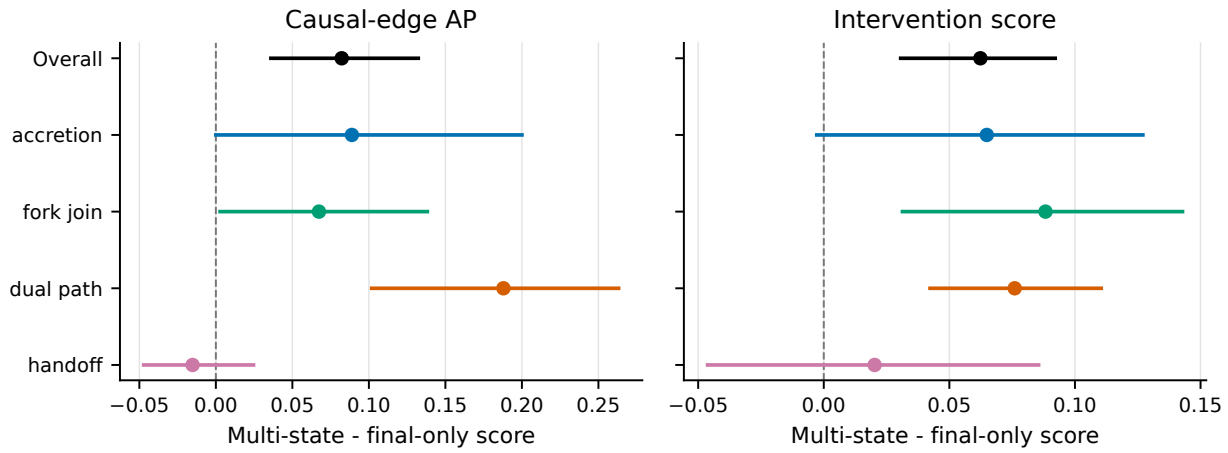



Figure 3: Overall and family-specific paired effects for the two outcomes that improved. Points are mean multiple-state-minus-final differences. Lines are unstratified percentile bootstrap intervals within the displayed group; family intervals contain only six instances and are descriptive.

Table 3: Post hoc timestamp-masked audit ($n = 8$). The interval and $p$-value compare ordered labels with an unordered set of intermediate states.

| Outcome | Final only | Ordered | Unordered | Ord.–unord. | 95% interval | $p$ |
|---|---|---|---|---|---|---|
| Edge AP | .912 | .978 | .987 | −.009 | [−.046, .025] | .750 |
| Function macro-F1 | 1.000 | 1.000 | 1.000 | .000 | [.000, .000] | 1.000 |
| Intervention score | .870 | .959 | .966 | −.007 | [−.056, .043] | .781 |

## 4.5 Post hoc timestamp-masked audit

Table 3 and Figure 4 report the eight-instance timestamp-masked audit. Removing intermediate timestamps did not lower either mean substantive score. Edge AP was 0.978 with ordered labels and 0.987 with timestamps masked (ordered minus unordered −0.009, 95% interval [−0.046, 0.025], $p = 0.750$). Intervention score was 0.959 versus 0.966 (difference −0.007, interval [−0.056, 0.043], $p = 0.781$). Function score remained at ceiling.

This small, post hoc audit cannot establish equivalence or prove that timestamps never matter. Its intervals permit moderate effects in either direction. It provides no evidence that visible intermediate chronology caused the primary gain, so the defensible treatment label is *access to multiple constructed states*. A benefit from temporal order remains unconfirmed.

## 4.6 Post hoc mean-transfer heuristic

To check whether the final-only package was intrinsically uninformative, a simple non-language-model heuristic averaged the five transfer scores for each candidate edge and ranked candidates by that mean. It achieved mean AP 0.995 on final-only dossiers and 1.000 on multiple-state dossiers. Thus, both evidence packages contain enough signal for a purpose-built estimator to recover nearly all edges. The larger 0.082 analyst effect measures how this fixed language model used the two presentations; it is not an information-theoretic lower bound on the value of developmental data. This diagnostic was devised after the confirmatory result and is not included in hypothesis testing.

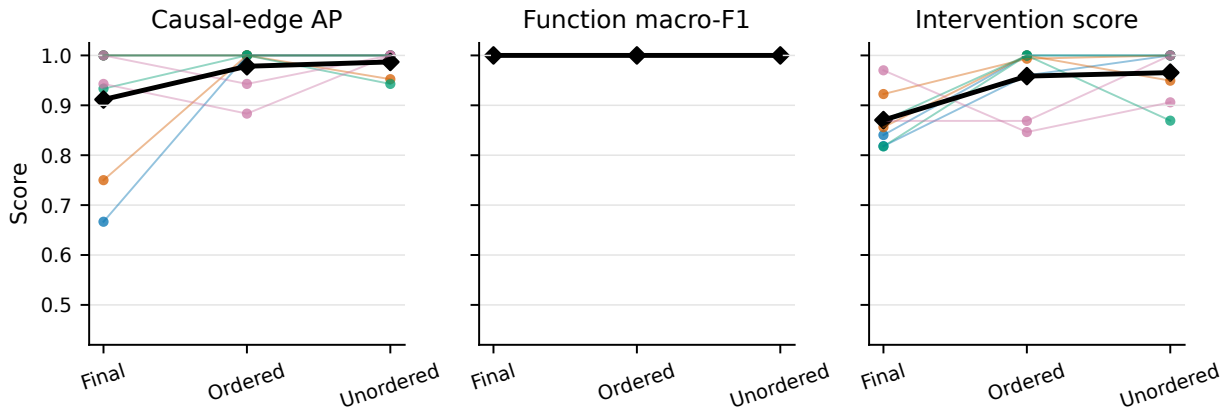



Figure 4: Timestamp-masked audit outcomes for two instances per family. Ordered and unordered conditions contain the same five states; the latter masks intermediate timestamps while identifying the final state. Black diamonds show means.

Table 4: Platform-reported resource use per confirmatory cell, mean (SD) unless noted. Reasoning tokens are a subset of reported output accounting.

| Quantity | Final only | Multi-state |
|---|---|---|
| Input tokens | 17,963 (525) | 17,841 (382) |
| Output tokens | 1,271 (408) | 1,017 (369) |
| Reasoning output tokens | 1,074 (404) | 819 (366) |
| Elapsed seconds | 26.23 | 21.76 |
| Tool events, total | 0 | 0 |

### 4.7 Resource use and manipulation checks

The serialized evidence was nearly exactly matched in size: 6,805 characters for every multiple-state prompt and 6,803 for every final-only prompt. Platform-reported realized usage was also similar in input but not artificially equalized after generation (Table 4). If anything, the final-only sessions consumed more output and reasoning tokens and took longer. The multiple-state advantage is therefore not explained by greater realized generation in that condition. Cached-input counts differ because they are execution-system accounting, not evidence supplied to the analyst. The read-only working directory contained benchmark artifacts, including ground truth; this would have allowed leakage through a tool call, but the raw event streams record none. Future runs should place hidden truth outside the analyst's filesystem even when tools are prohibited.

The pilot and main run took 57.5 and 295.4 wall-clock seconds, respectively, with four workers. The original presentation-order audit took 46.7 seconds, and the corrected timestamp-masked audit took 70.2 seconds. The Tracr feasibility run took under two minutes. Total empirical wall time was therefore under ten minutes, well below the two-hour compute limit.

## 5 Discussion

The experiment gives a qualified "yes" to the operational research question. When this fixed analyst received several states from a known constructed history, it recovered causal edges more accurately and made better predictions for unseen interventions than when it received repeated final-state evidence. The paired final panel rules out better final evidence, and the matched prompt shape, model, and allowed resources rule out simple budget differences. The post hoc heuristic shows, however, that this is a conditional result about how the language model used the packages, not proof that final-state evidence was insufficient in principle.

Why did the extra states help the analyst? The generator creates late-rising shadow non-edges, and the

prompt explains that late co-activation can be misleading. At the final state, these alternatives have high and precisely repeated transfer scores. Earlier states reveal which relations remain strong across different partial circuits. This is an engineered test of whether the analyst can use cross-state variation. The timestamp-masked audit produced no detectable loss, but its small sample does not establish that order is irrelevant. The present evidence supports state diversity more directly than temporal sequencing.

This distinction sharpens the developmental-interpretability agenda. A checkpoint archive can help for at least two reasons. First, it supplies varied mechanistic contexts: components and paths differ across checkpoints. Second, timestamps can identify precedence and transitions. The present experiment isolates neither factor confirmatorily, but its post hoc audit is consistent with the first. Future benchmarks should manipulate state diversity and timestamps orthogonally instead of treating a trajectory as one indivisible input.

The perfect function score is also informative for benchmark design. Forced-choice labels with strong local probes can be solved statically even when the graph remains ambiguous. Harder tests should include repeated roles, compositional functions, abstention, and labels that require behavioral experiments rather than direct probe comparison. Otherwise a ceiling can make a meaningful source of evidence look irrelevant.

More broadly, developmental evidence should be evaluated by predictive consequences. A convincing account should forecast what happens under a new ablation, clamp, path patch, or distribution shift. The intervention result here moves in step with graph recovery and offers such a check. This does not guarantee that the agent's internal rationale was faithful, but it demonstrates that its structured explanation carried useful causal information.

# 6 Limitations

**Constructed rather than learned histories.** The main limitation is ontological. The checkpoint sequences are designed circuit-assembly histories with a transparent measurement model, not checkpoints produced by gradient descent. A representative Tracr model establishes executability, but the 24 scored dossiers are component-level abstractions rather than 24 independently compiled and trained transformers. Natural training may add path dependence, representational rotation, superposition, optimizer state, and temporary mechanisms absent here.

**Four graph templates.** The 24 instances are six blinded permutations and noise realizations of only four families. Treating instances as the analysis unit captures reliability under those variations, not structural diversity across the space of transformer circuits. The bootstrap and sign-flip tests do not remove this template dependence. Replication should increase the number of independently generated programs and treat program family as a higher-level sampling unit.

**One agent and one run per cell.** Results may depend on the selected proprietary model, its unreported training data, and the platform's default sampling behavior. One response per cell does not estimate within-model stochastic variance. Model identifiers and raw event logs improve auditability but cannot make a changing hosted model exactly reproducible. A stronger study would cross agents, decoding seeds, and explanation scaffolds.

**Evidence is precomputed.** The analyst received summary statistics rather than weights, activations, and an experimental interface. Identical tool policies and zero realized calls simplify the comparison, but they omit adaptive investigation–one of the main attractions of agentic interpretability. The truth files were present in the read-only working directory; zero logged calls rule out observed access, but isolation would be safer. A future fixed-cost design should give both conditions the same intervention API and ask whether multiple-state analysts choose better experiments.

**Order audit power.** The timestamp-masked audit is post hoc and contains eight pairs. Its intervals rule out neither modest benefits nor modest harms from timestamps, and the final checkpoint remained labeled. The audit supports a cautious interpretation; it is not an equivalence test.

**Engineered final-state ambiguity.** Late shadow non-edges are deliberately built into the measurement model, and the prompt tells the analyst that late co-activation can produce high final-only scores. This directly cues the intended inference. Moreover, a post hoc mean-transfer heuristic nearly solves both arms, so the final package is not information-theoretically ambiguous. The experiment tests one analyst's use of an engineered multi-state cue; it does not show that real histories are inherently more identifying. Benchmarks should sweep ambiguity, remove leading descriptions, and compare purpose-built baselines.

**Coupled outcomes.** The intervention set includes cuts of sampled true edges and a known non-edge, so graph truth influences both edge scoring and the questions used for post-intervention prediction. The intervention endpoint is a useful consistency check, but it is not statistically or conceptually independent of edge recovery.

# 7 Future Work

The next experiment should compile a larger set of RASP programs, train or fine-tune matched semi-synthetic transformers, and derive evidence from actual activation and intervention measurements at saved checkpoints. A $2 \times 2$ design could cross access to multiple states with access to their timestamps. Another factor could compare uniformly spaced checkpoints with transition-focused selection. Hierarchical analysis across programs, seeds, and agent repetitions would then separate program diversity from response stochasticity.

For natural models, the closest analogue is to choose behaviors with documented emergence windows in dense checkpoint suites. An explainer could receive the same final-checkpoint tools in both arms, while the developmental arm may query earlier checkpoints under a matched total intervention budget. Precommitted hidden tests should then score final-circuit localization and predictions of held-out edits. Such a study would connect developmental descriptions to the standards already used for causal intervention and circuit evaluation.

# 8 Conclusion

For one language model and prompt on 24 constructed variants of four causal-graph templates, multiple-state evidence improved mean causal-edge ranking and post-intervention output prediction relative to replicated final-state evidence under the same harness settings. Component-role performance was saturated. A small post hoc audit found no loss after intermediate timestamps were masked, but it did not establish equivalence or isolate chronology. The result supports developmental evidence as a promising source of structured variation for a language-model analyst; validation on many independently learned checkpoint trajectories remains the decisive next step.

# Data and Code Availability

The codebase will be released upon acceptance. It will include the frozen generator and hashes, visible prompts, response schema, raw agent event logs, exact scoring code, per-instance results, statistical analysis, and figure scripts.

# Ethical Considerations

The study used no human participants, personal data, or user-generated content. Its main risk is epistemic: a positive result on transparent synthetic evidence could be overgeneralized to natural models. The manuscript therefore reports the constructed nature of the histories, the single-agent design, and the limited circuit population directly.

# A Exact Benchmark Structure

The canonical edge sets before identifier permutation were:

$$\begin{aligned}
\text{accretion: } & 0\to2,\ 1\to2,\ 2\to3,\ 3\to4,\\
\text{fork–join: } & 0\to2,\ 1\to2,\ 0\to3,\ 2\to3,\ 3\to4,\\
\text{dual path: } & 0\to2,\ 1\to2,\ 0\to3,\ 1\to3,\ 2\to4,\ 3\to4,\\
\text{handoff: } & 0\to2,\ 1\to2,\ 2\to3,\ 2\to4,\ 3\to4.
\end{aligned}$$

Canonical nodes 0–5 correspond to Indicator, Selector, Aggregator, Combiner, Mapper, and No Task Role. For seed $r \in \{0,\dots,5\}$, canonical node $k$ is serialized as $\mathrm{K}((k+r) \bmod 6)$. This Latin-cycle assignment prevents any opaque identifier from carrying a stable role cue.

Source values for each intervention are drawn deterministically from a seeded uniform distribution on $[0.15, 0.95]$ and rounded to two decimals. The eight question forms comprise three component clamps, three single-edge cuts (including one known non-edge), one two-component clamp, and one combined clamp and true-edge cut. Answers are computed from the hidden graph before evidence noise is generated.

# B Prompt and Output Contract

Every prompt opens with the same task definition and identifies its evidence mode. It defines a true edge as a designated residual contribution consumed by a target, explains activation and necessity fields, lists the six role meanings, and states whether panels are ordered checkpoints or final-state calibration views. It then prints five panels, a fixed candidate-edge index E00–E29, the intervention simulator, and questions Q0–Q7. The returned JSON has exactly three arrays:

```
{
  "edge_probabilities": [30 numbers in [0,1]],
  "function_labels": [6 labels aligned to K0..K5],
  "intervention_outputs": [8 numbers in [0,1]]
}
```

The prompt explicitly states that C5 is final and identical across paired conditions. No answer key, hidden graph, canonical-to-opaque mapping, or intervention output is serialized into the visible dossier.

# C Additional Reproducibility Details

The pilot used one held-out instance from each family with seeds 90–93. The confirmatory set used seeds 0–5. Before confirmatory execution, SHA-256 digests were recorded for the benchmark generator, scorer, agent runner, JSON schema, and design document. Main and audit cells were shuffled using seed 20260804. After the disclosed tie correction, the main exact edge test enumerated 65,536 sign assignments over 16 nonzero differences. The intervention test used 500,000 Monte Carlo assignments. The family-stratified bootstrap used independent fixed seeds for each outcome.

The Tracr check used Python 3.12.9, JAX 0.11.0, Haiku 0.0.17, and Tracr commit `9ce2b8c`. Ground truth was extracted through Tracr's RASP graph, basis inference, Craft-component annotation, and layer-allocation passes. Whole attention/MLP subtrees were cumulatively gated while embeddings stayed fixed. This gives exact half-layer checkpoints; finer component gates would require head-slice and MLP-subspace masks. The complete commit identifier is retained in the experiment record.

# D Reference Verification

Bibliographic records were checked against primary publisher pages, conference proceedings, official project pages, or arXiv records on August 4, 2026. The local audit records the checked URL and metadata for each cited source. Recent 2026 works are identified as arXiv preprints rather than presented as peer-reviewed publications.

# References


Emmanuel Ameisen, Jack Lindsey, Adam Pearce, Wes Gurnee, Nicholas L. Turner, Brian Chen, Craig Citro, David Abrahams, Shan Carter, Basil Hosmer, Jonathan Marcus, Michael Sklar, Adly Templeton, Trenton Bricken, Callum McDougall, Hoagy Cunningham, Thomas Henighan, Adam Jermyn, Andy Jones, Andrew Persic, Zhenyi Qi, T. Ben Thompson, Sam Zimmerman, Kelley Rivoire, Thomas Conerly, Chris Olah, and Joshua Batson. Circuit tracing: Revealing computational graphs in language models. *Transformer Circuits Thread*, 2025. URL https://transformer-circuits.pub/2025/attribution-graphs/methods.html.

Stella Biderman, Hailey Schoelkopf, Quentin Gregory Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, Usvsn Sai Prashanth, Edward Raff, Aviya Skowron, Lintang Sutawika, and Oskar Van Der Wal. Pythia: A suite for analyzing large language models across training and scaling. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 2397–2430, 2023. URL https://proceedings.mlr.press/v202/biderman23a.html.

Steven Bills, Nick Cammarata, Dan Mossing, Henk Tillman, Leo Gao, Gabriel Goh, Ilya Sutskever, Jan Leike, Jeff Wu, and William Saunders. Language models can explain neurons in language models. OpenAI research publication, 2023. URL https://openaipublic.blob.core.windows.net/neuron-explainer/paper/index.html.

Lawrence Chan, Adrià Garriga-Alonso, Nicholas Goldowsky-Dill, Ryan Greenblatt, Jenny Nitishinskaya, Ansh Radhakrishnan, Buck Shlegeris, and Nate Thomas. Causal scrubbing: a method for rigorously testing interpretability hypotheses. *AI Alignment Forum*, 2022. URL https://www.alignmentforum.org/posts/JvZhhzycHu2Yd57RN/causal-scrubbing.

Tyler A. Chang, Zhuowen Tu, and Benjamin K. Bergen. Characterizing learning curves during language model pre-training: Learning, forgetting, and stability. *Transactions of the Association for Computational Linguistics*, 12:1346–1362, 2024. URL https://arxiv.org/abs/2308.15419.

Hang Chen, Jiaying Zhu, Hongyang Chen, Hongxu Liu, Xinyu Yang, and Wenya Wang. Navigating by old maps: The pitfalls of static mechanistic localization in LLM post-training. *arXiv preprint arXiv:2605.06076*, 2026. URL https://arxiv.org/abs/2605.06076.

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

Xufeng Duan, Zhaoqian Yao, Yunhao Zhang, Shaonan Wang, and Zhenguang G. Cai. How syntax specialization emerges in language models. *arXiv preprint arXiv:2505.19548*, 2025. URL https://arxiv.org/abs/2505.19548.

Bradley Efron. Bootstrap methods: Another look at the jackknife. *The Annals of Statistics*, 7(1):1–26, 1979. doi: 10.1214/aos/1176344552. URL https://doi.org/10.1214/aos/1176344552.

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

Alex Foote, Neel Nanda, Esben Kran, Ioannis Konstas, Shay Cohen, and Fazl Barez. Neuron to graph: Interpreting language model neurons at scale. *arXiv preprint arXiv:2305.19911*, 2023. URL https://arxiv.org/abs/2305.19911.

Atticus Geiger, Hanson Lu, Thomas Icard, and Christopher Potts. Causal abstractions of neural networks. In *Advances in Neural Information Processing Systems 34*, 2021. URL https://arxiv.org/abs/2106.02997.

Atticus Geiger, Zhengxuan Wu, Hanson Lu, Josh Rozner, Elisa Kreiss, Thomas Icard, Noah Goodman, and Christopher Potts. Inducing causal structure for interpretable neural networks. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 7324–7338, 2022. URL https://proceedings.mlr.press/v162/geiger22a.html.

Atticus Geiger, Duligur Ibeling, Amir Zur, Maheep Chaudhary, Sonakshi Chauhan, Jing Huang, Aryaman Arora, Zhengxuan Wu, Noah Goodman, Christopher Potts, and Thomas Icard. Causal abstraction: A theoretical foundation for mechanistic interpretability. *arXiv preprint arXiv:2301.04709*, 2023a. URL https://arxiv.org/abs/2301.04709.

Atticus Geiger, Zhengxuan Wu, Christopher Potts, Thomas Icard, and Noah D. Goodman. Finding alignments between interpretable causal variables and distributed neural representations. *arXiv preprint arXiv:2303.02536*, 2023b. URL https://arxiv.org/abs/2303.02536.

Nicholas Goldowsky-Dill, Chris MacLeod, Lucas Sato, and Aryaman Arora. Localizing model behavior with path patching. *arXiv preprint arXiv:2304.05969*, 2023. URL https://arxiv.org/abs/2304.05969.

Rohan Gupta, Iván Arcuschin, Thomas Kwa, and Adrià Garriga-Alonso. Interpbench: Semi-synthetic transformers for evaluating mechanistic interpretability techniques. In *Advances in Neural Information Processing Systems 37, Datasets and Benchmarks Track*, 2024. URL https://arxiv.org/abs/2407.14494.

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

Sture Holm. A simple sequentially rejective multiple test procedure. *Scandinavian Journal of Statistics*, 6(2): 65–70, 1979. URL https://www.jstor.org/stable/4615733.

Ihor Kendiukhov. A review of developmental interpretability in large language models. *arXiv preprint arXiv:2508.15841*, 2025. URL https://arxiv.org/abs/2508.15841.

Barsat Khadka. MechRL: Reinforcement learning agents perform circuit discovery for mechanistic interpretability. *arXiv preprint arXiv:2605.26343*, 2026. URL https://arxiv.org/abs/2605.26343.

Ayan Antik Khan, Harsh Kohli, Yuekun Yao, Huan Sun, and Ziyu Yao. Can language model agents be helpful circuit explainers in mechanistic interpretability? *arXiv preprint arXiv:2606.24026*, 2026. URL https://arxiv.org/abs/2606.24026.

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

Michael A. Lepori, Thomas Serre, and Ellie Pavlick. Uncovering intermediate variables in transformers using circuit probing. *arXiv preprint arXiv:2311.04354*, 2023. URL https://arxiv.org/abs/2311.04354.

Tom Lieberum, Matthew Rahtz, János Kramár, Neel Nanda, Geoffrey Irving, Rohin Shah, and Vladimir Mikulik. Does circuit analysis interpretability scale? evidence from multiple choice capabilities in chinchilla. *arXiv preprint arXiv:2307.09458*, 2023. URL https://arxiv.org/abs/2307.09458.

David Lindner, János Kramár, Sebastian Farquhar, Matthew Rahtz, Thomas McGrath, and Vladimir Mikulik. Tracr: Compiled transformers as a laboratory for interpretability. In *Advances in Neural Information Processing Systems 36*, 2023. URL https://arxiv.org/abs/2301.05062.

Arnau Marin-Llobet and Javier Ferrando. Automated interpretability and feature discovery in language models with agents. *arXiv preprint arXiv:2605.01555*, 2026. URL https://arxiv.org/abs/2605.01555.

Maxime Méloux, François Portet, and Maxime Peyrard. Mechanistic interpretability as statistical estimation: A variance analysis. *arXiv preprint arXiv:2510.00845*, 2025. URL https://arxiv.org/abs/2510.00845.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in GPT. In *Advances in Neural Information Processing Systems 35*, 2022. URL https://arxiv.org/abs/2202.05262.

Aaron Mueller, Atticus Geiger, Sarah Wiegreffe, Dana Arad, Iván Arcuschin, Adam Belfki, Yik Siu Chan, Jaden Fiotto-Kaufman, Tal Haklay, Michael Hanna, Jing Huang, Rohan Gupta, Yaniv Nikankin, Hadas Orgad, Nikhil Prakash, Anja Reusch, Aruna Sankaranarayanan, Shun Shao, Alessandro Stolfo, Martin Tutek, Amir Zur, David Bau, and Yonatan Belinkov. MIB: A mechanistic interpretability benchmark. In *Proceedings of the 42nd International Conference on Machine Learning*, 2025. URL https://arxiv.org/abs/2504.13151.

Tiberiu Musat, Tiago Pimentel, Nicolas Zucchet, and Thomas Hofmann. Invariant learning dynamics of transformers in inductive reasoning tasks. *arXiv preprint arXiv:2607.11875*, 2026. URL https://arxiv.org/abs/2607.11875.

Neel Nanda, Lawrence Chan, Tom Lieberum, Jess Smith, and Jacob Steinhardt. Progress measures for grokking via mechanistic interpretability. *arXiv preprint arXiv:2301.05217*, 2023. URL https://arxiv.org/abs/2301.05217.

Catherine Olsson, Nelson Elhage, Neel Nanda, Nicholas Joseph, Nova DasSarma, Tom Henighan, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Scott Johnston, Andy Jones, Jackson Kernion, Liane Lovitt, Kamal Ndousse, Dario Amodei, Tom Brown, Jack Clark, Jared Kaplan, Sam McCandlish, and Chris Olah. In-context learning and induction heads. *arXiv preprint arXiv:2209.11895*, 2022. URL https://arxiv.org/abs/2209.11895.

Yixin Ou, Yunzhi Yao, Ningyu Zhang, Hui Jin, Jiacheng Sun, Shumin Deng, Zhenguo Li, and Huajun Chen. How do LLMs acquire new knowledge? a knowledge circuits perspective on continual pre-training. In *Findings of the Association for Computational Linguistics: ACL 2025*, 2025. URL https://arxiv.org/abs/2502.11196.

Alethea Power, Yuri Burda, Harri Edwards, Igor Babuschkin, and Vedant Misra. Grokking: Generalization beyond overfitting on small algorithmic datasets. *arXiv preprint arXiv:2201.02177*, 2022. URL https://arxiv.org/abs/2201.02177.

Tamar Rott Shaham, Sarah Schwettmann, Franklin Wang, Achyuta Rajaram, Evan Hernandez, Jacob Andreas, and Antonio Torralba. A multimodal automated interpretability agent. *arXiv preprint arXiv:2404.14394*, 2024. URL https://arxiv.org/abs/2404.14394.

Aaditya K. Singh, Stephanie C. Y. Chan, Ted Moskovitz, Erin Grant, Andrew M. Saxe, and Felix Hill. The transient nature of emergent in-context learning in transformers. In *Advances in Neural Information Processing Systems 36*, 2023. URL https://arxiv.org/abs/2311.08360.

Jesse Vig, Sebastian Gehrmann, Yonatan Belinkov, Sharon Qian, Daniel Nevo, Simas Sakenis, Jason Huang, Yaron Singer, and Stuart Shieber. Causal mediation analysis for interpreting neural NLP: The case of gender bias. *arXiv preprint arXiv:2004.12265*, 2020. URL https://arxiv.org/abs/2004.12265.

Boshi Wang, Xiang Yue, Yu Su, and Huan Sun. Grokked transformers are implicit reasoners: A mechanistic journey to the edge of generalization. In *Advances in Neural Information Processing Systems 37*, 2024a. URL https://arxiv.org/abs/2405.15071.

Hetong Wang, Pasquale Minervini, and Edoardo M. Ponti. Probing the emergence of cross-lingual alignment during LLM training. In *Findings of the Association for Computational Linguistics: ACL 2024*, 2024b. URL https://arxiv.org/abs/2406.13229.

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

Gail Weiss, Yoav Goldberg, and Eran Yahav. Thinking like transformers. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 11080–11090, 2021. URL https://proceedings.mlr.press/v139/weiss21a.html.

Frank Zhengqing Wu, Francesco Tonin, and Volkan Cevher. Demystifying variance in circuit discovery of LLMs. *arXiv preprint arXiv:2606.16920*, 2026. URL https://arxiv.org/abs/2606.16920.

Zhengxuan Wu, Atticus Geiger, Aryaman Arora, Jing Huang, Zheng Wang, Noah D. Goodman, Christopher D. Manning, and Christopher Potts. pyvene: A library for understanding and improving PyTorch models via interventions. *arXiv preprint arXiv:2403.07809*, 2024. URL https://arxiv.org/abs/2403.07809.

Yang Xu, Yi Wang, Hengguan Huang, and Hao Wang. Tracking the feature dynamics in LLM training: A mechanistic study. *arXiv preprint arXiv:2412.17626*, 2024. URL https://arxiv.org/abs/2412.17626.

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