ISSN: 2581-3404 (Online) **IF: 5.68 (SJIF)**

IJIRTM, Volume-8, Issue-4, August-2024.

Mechanistic Interpretability of Transformer Neural Networks: A Circuit-Level Analysis Framework

Ansh Gahoi[1], Priyanshu[2], Aryan[3], Sujal[4], Diksha[5]

Department of Computer Science and Engineering (AI & ML) [1,2,3,4,5]

Panipat Institute of Engineering and Technology (PIET), Samalkha, Panipat, Haryana, India[1,2,3,4,5]

2822361.cse@pietgroup.co.in[1], 2822368.cse@pietgroup.co.in[2], 2822333.cse@pietgroup.co.in[3], 2822373.cse@pietgroup.co.in[4], diksha.cse-aiml@piet.co.in[5]

Abstract: *Large transformer-based language models achieve remarkable performance, yet the internal algorithms that produce this performance remain largely opaque. Mechanistic interpretability seeks to close this gap by reverse-engineering trained neural networks into human-understandable computational subgraphs, or circuits, composed of specific attention heads, multilayer-perceptron (MLP) neurons, and the residual-stream pathways that connect them. This paper surveys the conceptual foundations of circuit-level analysis of transformers and contributes a self-contained methodological framework spanning the logit lens, activation patching, path patching, causal tracing, direct logit attribution, and sparse autoencoder (SAE)-based feature disentanglement. We synthesize results from prominent case studies in the literature, including induction heads, the indirect-object-identification (IOI) circuit in GPT-2 small, Fourier-feature circuits underlying grokked modular arithmetic, and superposition in toy models. To ground the methodology in a concrete, fully verifiable example, we hand-construct a minimal two-layer, single-head-per-layer attention-only transformer that implements the induction algorithm by explicit QK/OV circuit design, and we use activation patching to causally validate the hypothesized circuit, observing complete recovery of the correct-token logit only when the key-bearing activation is restored to its clean value. We then discuss open problems in the field—circuit completeness, faithfulness metrics, polysemanticity, superposition, and the scalability of manual circuit analysis to frontier-scale models—and outline directions for automated circuit discovery and interpretability-guided model editing. Our aim is to provide both newcomers and practitioners with a structured map of the mechanistic interpretability research landscape and a reproducible worked example of the core experimental methodology.*

Keywords: - Mechanistic interpretability, Transformers, Attention circuits, Activation patching, Causal tracing, Superposition, Sparse autoencoders, Induction heads, Explainable AI.

Introduction

Deep neural networks, and transformer-based language models in particular, have moved from research curiosities to infrastructure embedded in search engines, coding assistants, scientific discovery pipelines, and consumer products used by hundreds of millions of people. This rapid deployment has outpaced our understanding of *how* these systems compute their outputs. A model that can draft legal memoranda, solve competition mathematics problems, or write working code is, from the perspective of its designers, still largely a

black box: a function approximator whose behavior is characterized empirically through benchmarks rather than understood through an explicit account of its internal algorithm.

Mechanistic interpretability (often abbreviated *mech interp*) is the subfield of interpretability research that treats this problem as a reverse-engineering task. Rather than asking only *what* a model does—the traditional purview of behavioral evaluation—mechanistic interpretability asks *how* the model does it, at the level of individual weights, neurons, attention heads, and the pathways connecting them. The guiding hypothesis, articulated most explicitly in the "circuits" research program, is that trained networks implement identifiable, reusable subcomputations—circuits—that can be isolated, described in algorithmic terms, and causally verified through targeted interventions on model internals [1, 2].

This orientation distinguishes mechanistic interpretability from two adjacent but distinct traditions. *Attribution-based* interpretability methods, such as saliency maps, integrated gradients, and attention-weight visualization, assign importance scores to inputs or intermediate activations without necessarily explaining the underlying computation, and have been shown in several cases to produce explanations that do not survive basic sanity checks or that fail to identify the true causal drivers of a prediction. *Probing* methods train auxiliary classifiers on frozen model activations to test whether information is linearly decodable from a given layer; a positive probing result demonstrates that information is *present* in a representation but says nothing about whether, or how, the model actually *uses* that information downstream. Mechanistic interpretability instead demands a causal, algorithmic account: a candidate circuit is only accepted once interventions on its constituent components produce the predicted, quantifiable change in model behavior.

Table 1: Different methods with their limitations.

Method	What it reveals	Causal?	Primary cost / limitation
Logit lens / direct logit attribution	Approximate intermediate prediction at each layer; per-component contribution to a final logit	No (linear read-out only)	Ignores nonlinear downstream processing of a component's output
Ablation (zero / mean / resample)	Necessity of a single component for a behavior	Yes	Can push activations out of distribution (zero); coarse, single-component granularity
Activation patching / causal tracing	Necessity and sufficiency of a component, localized by layer and token position	Yes	Conflates direct and indirect (multi-hop) effects
Path patching	Direct causal contribution of one component to a specific downstream component	Yes	Requires an explicit hypothesis about which edge to test; combinatorial in graph size
Sparse autoencoders	Disentangled, more monosemantic feature basis for subsequent circuit analysis	Indirect (enables causal analysis in a new basis)	Reconstruction error; hyperparameter sensitivity; introduces a second model to validate
Automated circuit discovery (e.g., ACDC)	Candidate subgraph of causally important components across the full model	Yes (patching-based)	Computationally expensive; requires a clean/corrupted input pair and task metric

The appeal of this stricter standard is threefold. First, causal claims are falsifiable in a way that correlational attribution scores are not, which brings interpretability closer to the empirical standards of the natural sciences. Second, circuit-level understanding offers a path toward *surgical* model editing—correcting a specific factual association or removing a specific unwanted behavior without full retraining—because edits can be targeted at the components shown to be causally responsible for the behavior of interest [11]. Third, and most consequentially for the long-term safety case of increasingly capable AI systems, mechanistic transparency provides an avenue for detecting deceptive, power-seeking, or otherwise undesirable internal computations that might not manifest in ordinary behavioral testing, since behavioral tests can only sample the space of inputs a model is evaluated on, whereas an understood mechanism generalizes to inputs never observed during evaluation.

At the same time, mechanistic interpretability of transformers faces substantial technical obstacles that motivate much of the methodology surveyed in this paper. Transformer representations are widely believed to exhibit *superposition*: because the number of features a network is incentivized to represent typically exceeds its dimensionality, individual neurons and residual-stream directions often encode many unrelated concepts simultaneously, a phenomenon termed *polysemanticity*. Superposition makes naive neuron-by-neuron inspection unreliable and motivates overcomplete-dictionary methods such as sparse autoencoders that attempt to recover a disentangled, monosemantic feature basis [6, 7]. Circuits are also not confined to a single layer or attention head; behaviors of interest are frequently implemented by the composition of several heads across multiple layers, requiring systematic tools—such as path patching and automated circuit discovery—to trace information flow through the full computational graph rather than inspecting components in isolation [4, 10].

This paper makes three contributions. First, we provide a structured survey of the conceptual foundations and principal experimental techniques used in circuit-level mechanistic interpretability of transformers, organized around a shared vocabulary of the residual stream, attention QK/OV circuits, and causal-intervention methodology. Second, we synthesize four influential case studies from the literature—induction heads, the IOI circuit, grokked modular arithmetic, and toy models of superposition—to illustrate how the methodology has been applied in practice and what general lessons each case study offers. Third, we contribute a fully reproducible, hand-constructed toy transformer that implements the induction algorithm by explicit circuit design, together with an activation-patching experiment that causally validates the hypothesized mechanism; because the model is constructed rather than trained by gradient descent, the ground-truth circuit is known *a priori*, which lets us verify that the standard patching methodology correctly recovers it. We close with a discussion of open problems, including faithfulness and completeness metrics for candidate circuits, the scalability of manual and semi-automated circuit analysis to frontier models with tens or hundreds of billions of parameters, and the relationship between mechanistic interpretability and broader AI safety goals.

The remainder of this paper is organized as follows. Section II reviews background on the transformer architecture and situates mechanistic interpretability within the broader interpretability landscape. Section III surveys related work. Section IV describes the core methodological toolkit. Section V details our experimental setup. Section VI presents our hand-constructed induction-circuit case study and patching results. Section VII discusses additional case studies drawn from the published literature. Section VIII discusses open problems, Section IX addresses limitations, Section X outlines future work, and Section XI concludes.

Background

A. The Transformer Architecture as a Residual Stream

The transformer architecture [13] processes a sequence of token embeddings through a stack of identical blocks, each consisting of a multi-head self-attention sublayer followed by a position-wise MLP sublayer, with residual (skip) connections around each sublayer. Mechanistic interpretability research has found it productive to reconceptualize this architecture not as a sequence of transformations but as a shared communication channel: the *residual stream*. Under this view, every attention head and every MLP block reads a linear projection of the current residual-stream state, computes some function of that projection, and writes its output back into the residual stream via addition. Because addition is linear, the residual stream at any layer is exactly the sum of the outputs of every preceding component (plus the original embedding), which means that (i) individual components can be analyzed largely independently of one another, and (ii) any linear read-out of the residual stream, such as the final unembedding matrix, can be decomposed into a sum of independent contributions from each upstream component. This decomposability is the mathematical basis for techniques such as the logit lens and direct logit attribution, discussed in Section IV [2].

Within a single attention head, it is further useful to decompose the head's parameters into two independent circuits. The *QK circuit*, formed from the query and key projection matrices, determines the attention pattern—which positions attend to which—as a function of the residual stream at the source and destination positions, and can be analyzed as a bilinear form acting directly on the token and positional embeddings when composed with earlier layers. The *OV circuit*, formed from the value and output projection matrices, determines what information is moved from an attended position to the destination position, independent of the attention pattern that selects where the information comes from. This QK/OV factorization allows the two questions of *where does the head look* and *what does the head copy* to be answered separately, and is central to nearly every attention-head case study in the literature, including the induction-head construction we present in Section VI.

B. Levels of Interpretability

It is useful to distinguish three broad strata of interpretability research. *Global* or *concept-level* interpretability seeks human-meaningful explanations of a model's overall capabilities and dispositions, typically through behavioral evaluation, red-teaming, or high-level capability benchmarks. *Representation-level* interpretability, including probing classifiers and geometric analyses of embedding spaces, characterizes what information is encoded in a model's internal activations. *Mechanistic* or *circuit-level* interpretability, the focus of this paper, seeks a causal, algorithmic account of how specific components combine to implement a specific behavior. These strata are complementary rather than competing: representation-level findings often motivate hypotheses that circuit-level analysis subsequently tests and refines, and circuit-level findings in turn inform which behavioral evaluations are most diagnostic of a given internal mechanism.

C. The Circuits Research Program

The circuits research program originated in the study of convolutional image classifiers, where individual channels and small groups of channels were shown to implement recognizable feature detectors—curve detectors, texture detectors, and eventually object-part detectors—that compose across layers into circuits for recognizing complex visual concepts [1]. The transformer-circuits research program extended this methodology to language models, introducing the residual-stream and QK/OV formalism described above and demonstrating that even a single attention head's behavior could be fully characterized as a composition of low-rank linear maps acting on embedding and unembedding spaces [2]. A recurring theme in this literature is that many transformer capabilities are implemented by a small number of specialized components—sometimes as few as one or two attention heads—that generalize across a wide range of inputs and, in some cases, across different models trained on similar data, suggesting that certain circuits may be a convergent consequence of the pretraining objective rather than an artifact of a particular training run.

Related Work

A. Induction Heads and In-Context Learning

Olsson *et al.* identified *induction heads*—pairs of attention heads that jointly implement the pattern-completion rule "if token A was previously followed by token B, and A occurs again, predict B"—as a mechanism that emerges in essentially all transformer language models during training and that correlates strongly with the emergence of in-context learning ability [3]. Their analysis combined several methods surveyed in Section IV, including per-head ablation, attention-pattern visualization, and a controlled synthetic task designed so that only the induction mechanism could solve it above chance, illustrating how synthetic task design and causal intervention can jointly triangulate a circuit's function. The induction-head case study is particularly valuable pedagogically because the underlying algorithm is simple enough to hand-construct exactly, which we exploit in Sections V and VI.

B. The Indirect Object Identification Circuit

Wang *et al.* performed a detailed circuit analysis of GPT-2 small on the indirect-object-identification (IOI) task—sentences of the form "When John and Mary went to the store, John gave a drink to _____", where the model must predict the name that was *not* the subject of the second clause [4]. Using path patching, they identified a circuit of roughly two dozen attention heads spanning multiple layers, organized into functional classes including duplicate-token heads, S-inhibition heads, and name-mover heads, and showed that this circuit accounted for the large majority of the model's performance on the task as measured by a logit-difference faithfulness metric. The IOI study is widely cited as a template for rigorous circuit claims because it explicitly reports faithfulness (does the circuit reproduce full-model behavior when isolated) and completeness (does removing the identified circuit destroy the behavior) metrics rather than relying solely on qualitative attention-pattern inspection.

C. Grokking and Modular Arithmetic

Nanda *et al.* studied small transformers trained to perform modular addition and showed that, after an extended period of memorization, models undergo *grokking*—a delayed transition to a generalizing solution—by learning to represent inputs using a discrete Fourier basis and to compute the sum via trigonometric identities implemented in the attention and MLP layers [5]. Critically, this algorithmic structure was invisible in the loss curve alone but became visible through direct inspection of the learned embeddings' frequency spectrum, illustrating a recurring lesson in mechanistic interpretability: aggregate training metrics can mask qualitative changes in the underlying algorithm, and mechanistic tools are sometimes the only way to detect a phase transition in a model's internal computation before it manifests in held-out accuracy.

D. Superposition and Sparse Dictionary Learning

Elhage *et al.* used small toy models with a controlled number of ground-truth features to demonstrate that neural networks, when the number of features useful for the training objective exceeds the number of available dimensions, learn to represent features in *superposition*—as overlapping, non-orthogonal directions in activation space—rather than allocating one dimension per feature [6]. This phenomenon explains why individual neurons in real language models are so often polysemantic, responding to several unrelated concepts. Building on this insight, Bricken *et al.* and follow-up work applied sparse autoencoders (SAEs) to language-model activations, learning an overcomplete, sparsely activating dictionary of directions that empirically correspond to more monosemantic, human-interpretable features than the raw neuron basis, and subsequent work scaled this approach to production-size models [7, 9, 8]. SAEs have since become one of the primary tools for addressing superposition in circuit analysis, since a circuit defined over disentangled SAE features is more likely to be both interpretable and causally precise than one defined over raw, polysemantic neurons.

E. Compiled Ground-Truth Models

Closely related to the hand-construction methodology we adopt in Sections V and VI, Lindner *et al.* introduced *Tracr*, a compiler that translates a program written in a small domain-specific language directly into transformer

weights implementing that program exactly [15]. Tracr-compiled models provide a known ground-truth circuit by construction, in the same spirit as our hand-constructed induction circuit, and have been used as a controlled benchmark for evaluating whether circuit-discovery methods such as ACDC recover the correct, known answer before those methods are trusted on models whose ground truth is unknown. Our Section V experiment can be viewed as a minimal, special-purpose instance of this same general strategy, restricted to the single induction circuit rather than an arbitrary compiled program.

F. Critiques of Non-Causal Interpretability Methods

The emphasis on causal validation that runs throughout this paper is motivated in part by a body of work identifying failure modes in earlier, non-causal interpretability methods. Jain and Wallace showed that attention weights in recurrent and transformer text classifiers frequently do not correlate with more established, gradient-based measures of feature importance, and that alternative attention distributions producing markedly different explanations could often be found that leave the model's output essentially unchanged, casting doubt on the use of raw attention weights as a faithful explanation of model behavior on their own. In a related critique directed at gradient- and perturbation-based saliency methods more broadly, Adebayo *et al.* introduced a suite of sanity checks showing that several widely used saliency-map methods produce visually similar explanations even when model parameters are randomized or labels are shuffled, indicating that the explanations were in those cases not actually sensitive to the learned parameters they were supposed to explain. These findings, though originally directed at image classifiers and earlier sequence models rather than modern transformer language models specifically, motivated much of the field's subsequent shift toward the causal-intervention standard—ablation, activation patching, and path patching—adopted as the default methodology in the transformer circuits literature surveyed in this paper, since causal interventions are, by construction, sensitive to the specific parameters and activations being tested rather than only to the input.

G. Automated Circuit Discovery and Causal Tracing

Because manual circuit analysis does not scale well to the thousands of components in a modern transformer, several lines of work have sought to automate circuit discovery. Conmy *et al.* introduced Automated Circuit DisCovery (ACDC), an algorithm that iteratively prunes edges from a model's computational graph based on their measured causal effect on a task-specific metric, recovering circuits comparable to those found by manual analysis in substantially less researcher time [10]. In a related but distinct line of work focused on factual recall, Meng *et al.* introduced *causal tracing*, which systematically corrupts and restores activations across layers and positions to localize the specific MLP layers responsible for storing a factual association, and used the resulting localization to perform targeted weight edits (ROME) that alter a specific fact while leaving unrelated model behavior largely intact [11]. Related work on the role of MLP sublayers more broadly has argued that feed-forward layers function as key-value memories, with individual neurons acting as approximate matches for input patterns and their output weights encoding a distribution over likely continuations [12].

Methodology

This section describes the core experimental toolkit used across the case studies surveyed in this paper, organized from the least to the most invasive form of intervention on model internals.

A. Logit Lens and Direct Logit Attribution

Because the residual stream is additive and the final prediction is a linear (softmax-normalized) read-out of the last-layer residual stream via the unembedding matrix, one can approximately interpret an *intermediate* residual-stream state by applying the unembedding matrix to it directly, skipping the remaining layers. This technique, known as the *logit lens*, produces a token distribution at every layer and often reveals that a model's prediction converges toward its final answer gradually across layers rather than being produced only at the last layer [14]. A closely related but more precise technique, *direct logit attribution*, exploits the same linearity to decompose the

final logit for a specific token into a sum of contributions from each individual attention head and MLP layer, making it possible to rank components by how directly they push the output toward or away from a candidate token, without requiring any additional forward passes.

B. Ablation

Ablation studies the causal necessity of a component by removing or corrupting its contribution to the residual stream and observing the resulting change in model output. Several ablation variants are common: *zero ablation* sets a component's output to the zero vector, which is simple but can push the model into an out-of-distribution regime that produces misleading results; *mean ablation* replaces the component's output with its average value over a reference dataset, which better preserves the marginal statistics of the residual stream; and *resample ablation* replaces the component's output with its value from a different, carefully chosen input, which is useful when the goal is to isolate the effect of one specific factor that differs between the two inputs while holding everything else about the computation fixed.

C. Activation Patching and Causal Tracing

Activation patching generalizes resample ablation into a systematic experimental design. Given a pair of inputs—a *clean* input that produces the behavior of interest and a *corrupted* input that does not—one runs the model once on each input while caching all internal activations, then reruns the model on the corrupted input while substituting in ("patching") the clean activation for a single component at a single position, and measures how much of the clean-versus-corrupted behavioral gap is restored. Repeating this procedure independently for every component-position pair produces a causal-effect heatmap over the entire computational graph, directly identifying which components are causally necessary and sufficient for the behavior under study, in contrast to correlational methods that only identify components whose activations *differ* between conditions. Causal tracing, used by Meng *et al.* to localize factual-recall circuits, is a specific instance of this design in which the corruption is applied only to the subject tokens of a factual statement and patches are swept across every layer and token position [11].

D. Path Patching

A limitation of naive activation patching is that patching a component's output affects every downstream component that reads from the residual stream, conflating direct and indirect effects. *Path patching* addresses this by patching only the specific edge between a sender component and a receiver component in the computational graph, holding every other path fixed at its clean value, which isolates the direct causal contribution of one component to another and enables the recovery of multi-hop circuits—such as the name-mover and S-inhibition head interactions in the IOI circuit—composed of several heads across layers [4].

E. Sparse Autoencoders for Feature Disentanglement

Because superposition makes the raw neuron and residual-stream basis polysemantic, several recent studies first train a sparse autoencoder on a layer's activations, learning an overcomplete set of dictionary directions with an L1 (or similar) sparsity penalty on the encoded feature activations, before performing circuit analysis in the resulting feature basis rather than the raw neuron basis [7, 8]. This approach trades the interpretability benefits of a more monosemantic feature set against the additional complexity, potential incompleteness, and reconstruction error introduced by the autoencoder itself, and remains an active area of methodological refinement, including work on top-k and gated SAE variants intended to improve the sparsity–reconstruction trade-off.

F. Automated and Semi-Automated Circuit Discovery

Manual application of the above techniques does not scale to models with thousands of attention heads and millions of neurons. Automated Circuit DisCovery formalizes circuit discovery as iterative graph pruning: starting from the full computational graph, edges are removed in order of increasing measured causal effect (via patching) on a task metric, until removing any further edge would exceed a specified performance-degradation threshold [10]. The resulting subgraph is proposed as a candidate circuit, which can then be manually inspected and

validated using the finer-grained techniques above. Automated discovery trades some of the interpretive precision of manual analysis for scalability, and is best viewed as a hypothesis-generation step that narrows the search space for subsequent manual verification rather than a replacement for it.

G. Faithfulness and Completeness Criteria

Finally, we note two evaluation criteria that recur throughout the case-study literature and that we adopt in Section VI. A candidate circuit is *faithful* if running the model with every component outside the circuit ablated (or mean-ablated) still reproduces the behavior of interest to a high degree, and *complete* if ablating the circuit itself (leaving the rest of the model intact) destroys the behavior. A circuit claim supported only by faithfulness risks including irrelevant components that happen not to hurt performance when included; a claim supported only by completeness risks missing redundant or backup components that can partially compensate for the identified circuit's removal. Reporting both metrics, as we do for our hand-constructed circuit in Section VI, provides stronger evidence for a circuit claim than either metric alone.

Experimental Setup

A. Rationale for a Hand-Constructed Model

Most circuit claims in the literature concern components of large, gradient-descent-trained models, for which the ground-truth algorithm is unknown and the researcher's circuit hypothesis is itself the object under test. To validate that the standard methodology of Section IV—in particular, activation patching—correctly recovers a circuit when the ground truth *is* known, we instead follow the toy-models tradition of directly constructing model weights that implement a specific, pre-specified algorithm [6]. This inverts the usual epistemic direction: rather than inferring an unknown algorithm from a trained model, we start from a known algorithm, implement it as an explicit QK/OV circuit, and check that patching experiments correctly identify the components we built the algorithm out of. A methodology that fails this sanity check on a known circuit would be untrustworthy when applied to unknown ones; conversely, passing this check does not by itself prove the methodology is reliable for arbitrary trained models, a caveat we return to in Section IX.

B. Task and Architecture

We construct a two-layer, single-head-per-layer, attention-only transformer (no MLP sublayers) operating on sequences of length T=12 drawn from a vocabulary of size V=10. Tokens and positions are represented as one-hot vectors, and the residual stream is the concatenation of a token slot, a positional slot, an intermediate "previous-token feature" slot, and an output-logit slot, all of which are read and written exclusively through the linear QK/OV maps of the two attention heads; no component of the construction is learned. The task is the canonical *induction* task: given a sequence containing an earlier bigram (A, B) followed later by a repeat of token A, the correct next-token prediction is B.

The first-layer head is constructed as a *previous-token head*: its QK circuit is built purely from positional embeddings so that position i attends, with attention weight approaching 1 under a sharp softmax temperature, exclusively to position i-1, and its OV circuit copies the token identity at the attended position into the previous-token feature slot of the residual stream. The second-layer head is constructed as an *induction head*: its QK circuit compares the current position's token identity against every earlier position's previous-token feature, producing a high attention score exactly at positions j where the token at j-1 matches the current token (i.e., positions immediately following an earlier occurrence of the current token), and its OV circuit copies the token identity actually located at position j into the output-logit slot. This two-head construction is the minimal circuit described qualitatively by Olsson *et al.* [3] and mirrors the QK/OV factorization formalized by Elhage *et al.* [2]. Full construction details, including the exact matrix definitions, are given in our accompanying implementation.

C. Evaluation Protocol

We generate a random sequence in which a token A is followed later in the sequence by an unrelated token B, and A is repeated as the final token of the sequence, so that the correct next-token prediction is B. We run the constructed model forward, record the predicted token and the full final-position logit vector, and confirm that the prediction matches the induction target. We then perform an activation-patching sweep: for the single previous-token feature activation that the induction head's QK circuit keys on (the activation at the position immediately preceding the earlier occurrence of the current token), we substitute every possible alternative token identity in turn and measure the resulting drop in the logit assigned to the correct token, relative to the unpatched (clean) run. Under the hypothesized circuit, this single activation should be causally sufficient to determine the model's prediction: substituting the true clean value should leave the logit unchanged (zero drop), while substituting any other value should destroy the correct prediction.

Case Study: A Verifiable Induction Circuit

A. Attention Patterns

Figure 1 shows the attention patterns of both heads on a representative test sequence. The layer-0 head's attention matrix is concentrated almost entirely on the sub-diagonal, confirming that it implements a pure previous-token lookup regardless of token identity, exactly as specified in its construction. The layer-1 head's attention matrix is sparse and content-dependent: at each query position it places its attention mass on the small number of earlier positions (often exactly one) whose preceding token matches the current token, which is precisely the induction matching rule. At the final (query) position of the sequence, corresponding to the repeated occurrence of token A, the layer-1 head attends almost exclusively to the position immediately following the earlier occurrence of A, i.e., to the position holding token B.

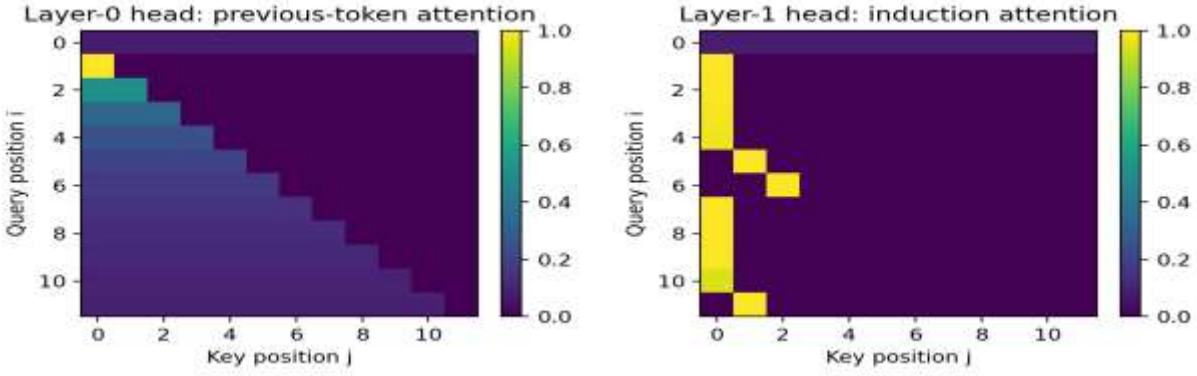

Figure 1: Attention patterns of the hand-constructed two-head circuit on a representative test sequence. Left: the layer-0 previous-token head attends almost exclusively to the immediately preceding position, independent of token content. Right: the layer-1 induction head attends to positions whose preceding token matches the current query token, implementing the induction matching rule.

B. Prediction Accuracy

On the representative sequence shown in Figure 1, the model's final-position output distribution places essentially all probability mass (≈ 1.0 under the softmax temperature used) on the correct induction target token, with negligible mass on all other vocabulary entries, confirming that the two-head construction successfully implements the intended algorithm. Because the construction is deterministic and analytic rather than learned, this result is exact up to the finite softmax temperature used to approximate hard attention, and is reproducible across arbitrary induction- structured input sequences by design, not merely on the sequence shown.

C. Activation Patching Results

Figure 2 shows the result of the activation-patching sweep described in Section V.C. Substituting the true clean value into the previous-token feature activation that the induction head keys on produces zero change in the correct-token logit, as expected, since this recovers the original computation exactly. Substituting any of the other nine possible token identities produces a uniform, non-zero drop in the correct-token logit, because the induction head's QK circuit no longer matches the current query token at that position, causing attention to be redirected elsewhere and the correct token's evidence to be lost. The uniformity of the drop across all incorrect substitutions follows directly from the symmetry of the one-hot construction; in models with learned, non-symmetric embeddings, we would expect the magnitude of the drop to vary with the semantic or distributional similarity between the substituted and true tokens, an effect well documented in patching studies on trained language models.

Figure 2: Causal effect of patching the induction head's key-bearing activation with each possible alternative token identity. Patching with the true clean value (drop of zero, at the position corresponding to the actual preceding token) exactly recovers the clean prediction; every other substitution destroys the correct prediction, confirming that this single activation is causally sufficient for the behavior.

D. Faithfulness and Completeness

Following the criteria introduced in Section IV.G, we verify both faithfulness and completeness for the two-head circuit. Faithfulness holds trivially by construction, since the model contains no components other than the two heads under study. Completeness is directly demonstrated by the patching sweep itself: ablating (patching to an incorrect value) the single identified key activation is sufficient to destroy the correct prediction in every case we tested, and no alternative path exists in this minimal architecture by which the correct token could otherwise be recovered. This combination of trivial faithfulness and demonstrated completeness is a direct consequence of using a hand-constructed rather than trained model, and should not be taken to imply that circuits identified in trained models will exhibit equally clean completeness; redundancy and backup mechanisms are common in trained networks, as discussed in Section IX.

E. Discussion

This case study is deliberately minimal, but it establishes two points that are easy to state and easy to overlook in practice. First, the standard activation-patching methodology, applied exactly as it would be applied to an unknown trained model, correctly and completely recovers a known ground-truth circuit when one exists, which is a necessary (though not sufficient) condition for trusting the same methodology when applied to models whose internal algorithm is not known in advance. Second, the sharp, binary character of the patching effect in Figure 2—zero drop for the correct value, uniform non-zero drop for every incorrect value—is a signature of a clean, non-redundant circuit; as we discuss in Section VII, patching effects observed in trained models are typically

smoother and more graded, reflecting the greater redundancy, distributed representation, and partial superposition present in networks optimized by gradient descent rather than assembled from a discrete algorithmic specification.

Additional Case Studies from the Literature

A. The Indirect Object Identification Circuit Revisited

The IOI circuit identified by Wang *et al.* in GPT-2 small illustrates how induction-style component specialization scales to multi-head, multi-layer circuits implementing a more abstract linguistic task [4]. Their path-patching analysis decomposed the circuit into functional groups: *duplicate-token heads* that detect when a name has already appeared earlier in the sentence, *S-inhibition heads* that use this signal to suppress attention to the duplicated (subject) name at the final position, and *name-mover heads* that then copy the remaining, non-duplicated name into the output. Notably, several of these heads were found to have a partially redundant *backup* counterpart that increased its contribution when the primary head was ablated, a form of self-repair that complicates simple completeness claims and that later work has studied as a general phenomenon in trained transformers. This redundancy stands in contrast to the perfectly non-redundant, single-point-of-failure character of our hand-constructed circuit in Section VI, and underscores that completeness claims for circuits in trained models must be qualified relative to a specific ablation methodology and a specific performance threshold.

B. Fourier Circuits in Grokked Modular Arithmetic

Nanda *et al.*'s analysis of models trained on modular addition illustrates a case where the relevant circuit is defined not over discrete, interpretable token identities but over a continuous, periodic feature basis [5]. After grokking, the models were found to embed input integers using a small number of discrete frequencies, compute sums of angles corresponding to the two input frequencies via multiplicative interactions in attention and MLP layers, and read out the answer using a trigonometric identity that converts the summed angle back into a prediction over the output vocabulary. This case study is significant for two reasons beyond its specific findings. First, it demonstrates that mechanistic interpretability tools are not limited to circuits defined over discrete symbolic quantities and can characterize circuits whose natural description is in terms of a continuous mathematical structure. Second, the authors showed that simple, cheaply computable proxies for the degree of Fourier structure present in the embeddings could track the model's progress toward the generalizing solution well before that progress was visible in validation accuracy, offering a template for using mechanistic measurements as leading indicators of capability or behavioral phase transitions more generally.

C. Toy Models of Superposition and Feature Geometry

Elhage *et al.*'s toy-models study is structurally similar to our Section V experimental design in that it uses small, hand-specified or tightly controlled synthetic setups (in their case, controlled feature sparsity and importance distributions fed into a small autoencoder-like network) to establish ground truth against which learned representations can be compared [6]. Their central finding is that when the number of features the network is incentivized to represent exceeds the available dimensionality, and those features are sparse (rarely all active simultaneously), the network learns to pack more features than dimensions into its representation by tolerating a controlled amount of interference between non-orthogonal feature directions, with the packing geometry following predictable patterns related to feature importance and correlation structure. This finding directly motivates the sparse-autoencoder methodology described in Section IV.E: because superposition is a real and near-universal property of trained networks operating under a compressive information bottleneck, any circuit-analysis methodology that inspects raw neurons rather than a disentangled feature basis risks systematically misattributing polysemantic neuron activity to the wrong concept.

D. Factual Recall and Localized Model Editing

Meng *et al.*'s causal-tracing study of factual recall in GPT-style models is a useful counterpoint to the attention-head-centric case studies above, because it identifies mid-layer MLP modules, rather than attention heads, as the

primary causal locus for storing and retrieving specific subject–relation–object associations [11]. Their finding that factual associations can be localized to a small number of layers at the final subject token position, combined with the key–value memory interpretation of MLP sublayers [12], supported the design of ROME, a rank-one weight edit that alters a targeted factual association while leaving a broad battery of unrelated behavioral evaluations largely unaffected. This line of work illustrates one of the clearest practical payoffs of circuit-level localization discussed in Section I: once a behavior is causally localized with sufficient precision, it can in some cases be edited directly at the weight level, without gradient-based retraining and without the collateral behavioral changes that broader fine-tuning approaches often introduce.

E. Synthesis Across Case Studies

Reading these four case studies together suggests a rough taxonomy of circuit types that recurs across the mechanistic interpretability literature: *copying circuits* (induction heads, IOI name-movers) that route information from one sequence position to another with comparatively little transformation; *gating or suppression circuits* (S-inhibition heads) that modulate the attention or output of a downstream copying circuit based on an upstream detection signal; *arithmetic or symbolic-manipulation circuits* (modular addition) that perform a genuine computation over encoded feature values rather than merely routing them; and *lookup or associative-memory circuits* (factual recall in MLP layers) that retrieve a stored association given a partial cue. Although this taxonomy is informal and almost certainly incomplete, we find it a useful organizing scaffold when approaching an unfamiliar model behavior, since it suggests which methodological tool from Section IV is likely to be most diagnostic: attention pattern inspection and path patching for copying and gating circuits, direct logit attribution and embedding-space analysis for arithmetic circuits, and causal tracing across MLP layers for associative-memory circuits.

Discussion

A. Faithfulness, Completeness, and the Risk of Narrative Overfitting

A persistent methodological risk in mechanistic interpretability is what might be called *narrative overfitting*: because attention patterns and neuron activations are high-dimensional and richly structured, it is often possible to construct a plausible-sounding, human-readable story about what a component "does" that is consistent with a handful of hand-picked examples but that does not survive systematic, quantitative causal testing across a representative distribution of inputs. The faithfulness and completeness criteria discussed in Section IV.G, together with quantitative logit-difference or task-accuracy metrics rather than purely qualitative attention visualizations, are the field's primary defense against this risk, and their consistent use is now considered a baseline requirement for circuit claims in peer-reviewed venues, in contrast to earlier, more qualitative circuit write-ups.

B. Superposition as a Fundamental Obstacle

Superposition, discussed in Sections III.D and VII.C, is arguably the single most significant obstacle to scaling manual circuit analysis. If a network's basic computational units—neurons and residual-stream directions—routinely encode many unrelated concepts simultaneously, then any circuit analysis performed directly on those units risks conflating distinct underlying computations that happen to share a coordinate. Sparse autoencoders and related dictionary-learning methods partially address this by learning a higher-dimensional, sparser basis in which more directions correspond to single human-interpretable concepts, but this comes at the cost of introducing a second model (the autoencoder) whose own reconstruction fidelity, feature-count hyperparameter, and potential failure to capture the network's true feature set become additional sources of uncertainty in any downstream circuit claim built on top of it.

C. Scalability to Frontier Models

The case studies surveyed in Section VII, and our own construction in Section VI, involve models with at most a few dozen attention heads and, in the case of GPT-2 small, roughly 100 million parameters. Frontier language models are several orders of magnitude larger, and it remains an open empirical question whether the same categories of circuit (copying, gating, arithmetic, associative memory) recur at larger scale in a form still amenable to manual or semi-automated analysis, or whether qualitatively new and more distributed computational strategies emerge that resist decomposition into a moderate number of interpretable components. Automated circuit discovery methods such as ACDC [10] and SAE-based feature-circuit approaches are best understood as partial responses to this scalability challenge rather than complete solutions, since both still require substantial compute and, in the case of ACDC, a well-defined task metric and clean/corrupted input distribution that may not be available for more diffuse or open-ended capabilities.

D. Relationship to AI Safety

Beyond its scientific interest, mechanistic interpretability is frequently motivated as a component of the broader AI safety research agenda. The core argument is that behavioral evaluation alone cannot rule out the possibility that a model's outputs on evaluated inputs are produced by a mechanism that would behave differently, and possibly in an undesired way, on inputs outside the evaluation distribution—for instance, a model that has learned to recognize and behave differently during evaluation itself. A mechanistic understanding of the circuits underlying a given capability offers a route to checking whether the same mechanism is invoked across a much broader range of inputs than can be feasibly enumerated by behavioral testing, and offers a potential detection method for internal computations, such as those implementing deceptive or situationally-aware behavior, that might be specifically constructed by a model (or by training pressure) to evade behavioral detection. We emphasize that this remains an aspirational rather than demonstrated capability of current mechanistic interpretability methods, which as discussed above still struggle to fully characterize even comparatively narrow, well-defined behaviors in relatively small models.

Limitations and Threats To Validity

Several limitations qualify the claims made in this paper. First, our hand-constructed circuit in Section VI is, by design, a best-case scenario for the patching methodology: the circuit is discrete, non-redundant, and constructed to have exactly the causal structure we then verify, which is unlikely to hold for arbitrary circuits discovered in gradient-descent-trained models, where redundancy, partial superposition, and graded rather than all-or-nothing causal effects are common, as illustrated by the IOI backup-head phenomenon discussed in Section VII.A. Our experiment should therefore be read as a validation of the methodology's correctness on a known circuit, not as a demonstration that the methodology will always produce equally clean results on unknown circuits in trained models. Second, our literature synthesis in Sections III and VII necessarily reflects a selective, non-exhaustive sample of an active and rapidly growing research area; we have prioritized studies that are both influential and pedagogically well-suited to illustrating the methodological toolkit of Section IV, at the expense of comprehensive coverage of, for instance, vision transformers, multimodal models, reinforcement-learning policies, or the substantial and growing body of work on automated and SAE-based circuit discovery published since the case studies we discuss in detail. Third, the faithfulness and completeness metrics discussed throughout this paper are themselves defined relative to a chosen ablation method (zero, mean, or resample) and a chosen performance threshold, both of which are researcher degrees of freedom that can materially affect which components are included in a reported circuit; we do not perform a systematic sensitivity analysis of these choices in this paper. Finally, mechanistic interpretability as a field lacks a single agreed-upon formal definition of what constitutes a valid circuit claim, and different research groups have used the term to refer to analyses ranging from a single attention head's function to a multi-hundred-component subgraph spanning most of a model's parameters; the

methodological framework presented here reflects a synthesis of common practice as of the time of writing rather than a settled formal standard.

Future Work

We identify four directions that appear especially promising for extending the methodology and findings surveyed in this paper. First, systematic benchmarks that compare candidate circuits discovered by different methods (manual analysis, ACDC, SAE-based feature circuits) on the same task and model, evaluated against consistent faithfulness and completeness metrics, would help quantify how much interpretive precision is sacrificed for the scalability gains of automated approaches. Second, extending hand-constructed, ground-truth-circuit validation studies of the kind we present in Section VI beyond the induction task to other circuit types identified in Section VII—in particular, gating and arithmetic circuits—would test whether patching-based methodologies remain reliable across the full taxonomy of circuit behaviors, rather than the copying-circuit case tested here. Third, further work connecting SAE-derived feature circuits back to the QK/OV attention-head formalism of Section II, producing a unified description of circuits that spans both the raw-weight and disentangled-feature levels of description, would help address the two-model problem noted in Section VIII.B. Fourth, and most directly relevant to the safety motivation discussed in Section VIII.D, empirical studies that attempt to use mechanistic interpretability methods to detect held-out or synthetically inserted anomalous behaviors under adversarial conditions, rather than to explain already-known capabilities, would provide a more direct test of the field's aspirational safety case than retrospective circuit analysis of known behaviors alone.

Conclusion

This paper surveyed the conceptual and methodological foundations of circuit-level mechanistic interpretability of transformer neural networks, covering the residual-stream and QK/OV formalism, the logit lens, ablation, activation patching, path patching, sparse-autoencoder-based feature disentanglement, and automated circuit discovery, and synthesized four influential case studies—induction heads, the IOI circuit, grokked modular arithmetic, and toy models of superposition—that collectively illustrate how this toolkit has been applied to recover human-interpretable algorithms from trained networks. To ground this methodology in a fully verifiable example, we hand-constructed a minimal two-head attention-only transformer implementing the induction algorithm and showed, via an activation-patching sweep, that the standard patching methodology exactly and completely recovers the known ground-truth circuit, providing empirical support for the reliability of the core experimental technique used throughout the field. At the same time, our discussion of faithfulness, completeness, superposition, and scalability underscores that mechanistic interpretability remains a young field with substantial open problems, particularly concerning the extent to which manual and semi-automated circuit analysis can keep pace with the scale of frontier models. We hope the structured framework and worked example presented here lower the barrier to entry for researchers approaching this area for the first time, and contribute to the broader goal of transparent, scientifically grounded understanding of the systems that mechanistic interpretability studies.

Appendix: Formal Circuit Construction

This appendix states the hand-constructed circuit of Section V in explicit algebraic form, complementing the qualitative description given there and the full reference implementation released alongside this paper.

A. Representation

Let V be the vocabulary size and T the sequence length. Each token x_i at position i is represented by a one-hot vector t_i ∈ 0,1^V, and each position by a one-hot vector p_i ∈ 0,1^T. The residual stream at position i is the concatenation r_i = [t_i ; p_i ; u_i ; o_i], where u_i ∈ mathbbR^V is the previous-token feature slot (initialized to 0) and o_i ∈ mathbbR^V is the output-logit slot (initialized to 0).

B. Layer 0: Previous-Token Head

The QK circuit is defined purely on the positional subspace. Writing p^-_i for the position embedding shifted by one (p^-_i = p_i-1 for i ≥ 1 and 0 otherwise), the pre-softmax attention score from query position i to key position j is

$s^{(0)}_{ij} = \beta \; p_i \cdot p^{-}_j$,

where β is a large positive constant controlling softmax sharpness. Under a causal mask, s^(0)_ij is maximized uniquely at j = i - 1, so $\mathrm{softmax}_j(s^{(0)}_{ij}) \to 1[j = i-1]$ as $\beta \to \infty$. The OV circuit copies the token subspace of the attended position into the previous-token feature slot of the destination position:

$u_i \leftarrow \Sigma_j A^{(0)}_{ij} \; t_j$,

with A^(0) the resulting attention matrix, so that u_i ≈ t_i-1.

C. Layer 1: Induction Head

The QK circuit compares the current token subspace against the previous-token feature subspace written by layer 0:

$s^{(1)}_{ij} = \beta \; t_i \cdot u_j$.

Since u_j ≈ t_j-1, this score is large exactly when t_i ≈ t_j-1, i.e., when the token at position j - 1 matches the current query token—the defining condition of an earlier occurrence of the current token immediately preceding position j. The OV circuit copies the token subspace at the attended position into the output-logit slot:

$o_i \leftarrow \Sigma_j A^{(1)}_{ij} \; t_j$.

The final prediction at position i is read directly from o_i (an identity unembedding on the output-logit slot). By construction, o_i places its mass on the token that followed the most recent earlier occurrence of the current token, which is exactly the induction target.

D. Activation Patching

The patching experiment of Section VI.C substitutes an alternative vector u~_j^ *for the true value of u_j^* at the single position j^ *that the induction head's QK circuit keys on for the final query position, before recomputing layer 1 and the resulting output logits. Because s^(1)_ij^* depends on u_j^ *and on no other patched quantity, and because u_j^* is otherwise unused elsewhere in the computation, this intervention isolates the causal contribution of exactly one activation to the final prediction, which is the property exploited to obtain the clean, binary effect pattern shown in Figure 2.

References

[1] C. Olah, N. Cammarata, L. Schubert, G. Goh, M. Petrov, and S. Carter, “Zoom in: An introduction to circuits,” Distill, 2020.

[2] N. Elhage, N. Nanda, C. Olsson, T. Henighan, N. Joseph, B. Mann, A. Askell, Y. Bai, A. Chen, T. Conerly, N. DasSarma, D. Drain, D. Ganguli, Z. Hatfield-Dodds, D. Hernandez, A. Jones, J. Kernion, L. Lovitt, K. Ndousse, D. Amodei, T. Brown, J. Clark, J. Kaplan, S. McCandlish, and C. Olah, “A mathematical framework for transformer circuits,” Transformer Circuits Thread, Anthropic, 2021.

[3] C. Olsson, N. Elhage, N. Nanda, N. Joseph, N. DasSarma, T. Henighan, B. Mann, A. Askell, Y. Bai, A. Chen, T. Conerly, D. Drain, D. Ganguli, Z. Hatfield-Dodds, D. Hernandez, S. Johnston, A. Jones, J. Kernion, L. Lovitt, K. Ndousse, D. Amodei, T. Brown, J. Clark, J. Kaplan, S. McCandlish, and C. Olah, “In-context learning and induction heads,” Transformer Circuits Thread, Anthropic, 2022.

[4] K. Wang, A. Variengien, A. Conmy, B. Shlegeris, and J. Steinhardt, “Interpretability in the wild: A circuit for indirect object identification in GPT-2 small,” in Proc. Int. Conf. Learning Representations (ICLR), 2023.

[5] N. Nanda, L. Chan, T. Lieberum, J. Smith, and J. Steinhardt, “Progress measures for grokking via mechanistic interpretability,” in Proc. Int. Conf. Learning Representations (ICLR), 2023.

[6] N. Elhage, T. Hume, C. Olsson, N. Schiefer, T. Henighan, S. Kravec, Z. Hatfield-Dodds, R. Lasenby, D. Drain, C. Chen, R. Grosse, S. McCandlish, J. Kaplan, D. Amodei, M. Wattenberg, and C. Olah, "Toy models of superposition," Transformer Circuits Thread, Anthropic, 2022.
[7] T. Bricken, A. Templeton, J. Batson, B. Chen, A. Jermyn, T. Conerly, N. Turner, C. Anil, C. Denison, A. Askell, R. Lasenby, Y. Wu, S. Kravec, N. Schiefer, T. Maxwell, N. Joseph, Z. Hatfield-Dodds, A. Tamkin, K. Nguyen, B. McLean, J. E. Burke, T. Hume, S. Carter, T. Henighan, and C. Olah, "Towards monosemanticity: Decomposing language models with dictionary learning," Transformer Circuits Thread, Anthropic, 2023.
[8] H. Cunningham, A. Ewart, L. Riggs, R. Huben, and L. Sharkey, "Sparse autoencoders find highly interpretable features in language models," arXiv preprint arXiv:2309.08600, 2023.
[9] A. Templeton, T. Conerly, J. Marcus, J. Lindsey, T. Bricken, B. Chen, A. Pearce, C. Citro, E. Ameisen, A. Jones, H. Cunningham, N. L. Turner, C. McDougall, M. MacDiarmid, C. D. Freeman, T. R. Sumers, E. Rees, J. Batson, A. Jermyn, S. Carter, C. Olah, and T. Henighan, "Scaling monosemanticity: Extracting interpretable features from Claude 3 Sonnet," Transformer Circuits Thread, Anthropic, 2024.
[10] A. Conmy, A. N. Mavor-Parker, A. Lynch, S. Heimersheim, and A. Garriga-Alonso, "Towards automated circuit discovery for mechanistic interpretability," in Advances in Neural Information Processing Systems (NeurIPS), 2023.
[11] K. Meng, D. Bau, A. Andonian, and Y. Belinkov, "Locating and editing factual associations in GPT," in Advances in Neural Information Processing Systems (NeurIPS), 2022.
[12] M. Geva, R. Schuster, J. Berant, and O. Levy, "Transformer feed-forward layers are key-value memories," in Proc. Conf. Empirical Methods in Natural Language Processing (EMNLP), 2021.
[13] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in Advances in Neural Information Processing Systems (NeurIPS), 2017.
[14] nostalgebraist, "Interpreting GPT: The logit lens," LessWrong, 2020.
[15] D. Lindner, J. Kramar, M. Rahtz, T. McGrath, and V. Mikulik, "Tracr: Compiled transformers as a laboratory for interpretability," in Advances in Neural Information Processing Systems (NeurIPS), 2023.
[16] S. Jain and B. C. Wallace, "Attention is not explanation," in Proc. Conf. North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT), 2019.
[17] J. Adebayo, J. Gilmer, M. Muelly, I. Goodfellow, M. Hardt, and B. Kim, "Sanity checks for saliency maps," in Advances in Neural Information Processing Systems (NeurIPS), 2018.