Seeing Through Circuits: Faithful Mechanistic Interpretability for Vision Transformers

Nina Żukowska, Wolfgang Stammer, Bernt Schiele, and Jonas Fischer

Max Planck Institute for Informatics, Saarland Informatics Campus, Saarbrücken, Germany

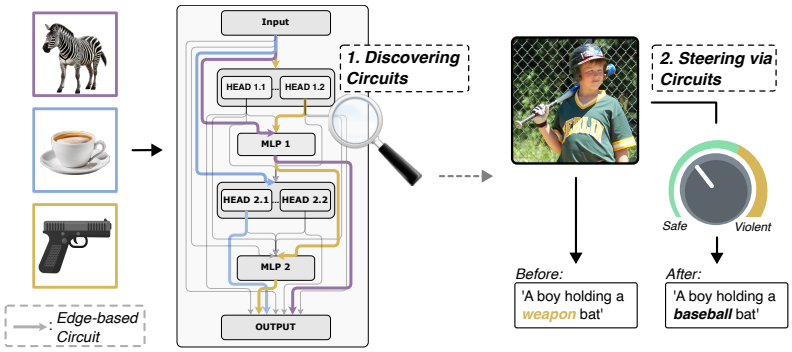

Fig. 1: Discovering Visual Mechanistic Circuits in computation graphs. Left: We discover sparse, edge-based circuits in Vision Transformers, where each circuit (colored paths) captures the computational subgraph responsible for recognizing a specific class. **Right:** These circuits enable *mechanistic steering*. By deactivating a discovered circuit (e.g., one associated with violent content) we can correct model behavior, such as shifting a CLIP model's interpretation of an image.

Abstract. Transparency of neural networks' internal reasoning is at the heart of interpretability research, adding to trust, safety, and understanding of these models. The field of mechanistic interpretability has recently focused on studying task-specific computational graphs, defined by connections (*edges*) between model components. Such edge-based circuits have been defined in the context of large language models, yet vision-based approaches so far only consider *neuron*-based circuits. These tell *which* information is encoded, but not *how* it is routed through the complex wiring of a neural network. In this work, we investigate whether useful mechanistic circuits can be identified through computational graphs in vision transformers. We propose an effective method for Automatic Visual Circuit Discovery (Vi-CD) that recovers class-specific circuits for classification, identifies circuits underlying typographic attacks in CLIP, and discovers circuits that lend themselves for steering to correct harmful model behaviour. Overall, we find that insightful and actionable edge-based circuits can be recovered from vision transformers, adding transparency to the internal computations of these models.

Keywords: Mechanistic Interpretability · Visual Circuits · Vision Transformers · Typographic Attacks · Activation Steering

1 Introduction

Modern vision models have achieved remarkable performance across a wide range of tasks, from image classification to open-vocabulary recognition. Yet despite their success, these models remain largely opaque: it is difficult to understand *why* a model arrives at a particular prediction, and consequently difficult to control or correct its behavior when it fails. This opacity has practical consequences beyond scientific curiosity. Models are vulnerable to adversarial attacks such as typographic attacks on CLIP [20, 37, 39], susceptible to shortcut learning [15, 32, 33], and difficult to debug when deployed in safety-critical settings [4].

Mechanistic interpretability aims to address these challenges by reverse-engineering the internal computations of neural networks. A central concept in this field is the *circuit*: a sparse subgraph of a model's computational graph that implements a specific behavior of interest [7, 14, 17, 25, 36]. In transformer language models, circuit discovery has converged on an *edge-centric* formulation, where models are represented as directed graphs over residual-stream connections between attention heads and MLP blocks, and the goal is to identify a minimal subgraph responsible for a given behavior [5, 9]. A key requirement for such circuits is *faithfulness*: a circuit is faithful if restricting the model's computation to only that circuit preserves its original task performance. Faithfulness ensures that the discovered circuit genuinely accounts for the model's behavior, rather than merely correlating with it.

In vision models, however, circuit discovery has remained largely at the neuron, channel, or feature level [7, 13, 31]. While such node-based circuits reveal *which* components are important for a given task, they do not capture *how* information flows between them. Edge-based circuits address this limitation by explicitly modeling the connections between components, offering several advantages: they can disentangle the contributions of polysemantic neurons that participate in multiple circuits, they enable causal interventions via edge-level activation patching, and they support more precise notions of faithfulness since individual connections can be included or excluded without ablating entire components. Despite these benefits, edge-based circuits have been little investigated in the vision domain.

In this work, we investigate whether the advantages of edge-based circuit discovery, so far demonstrated primarily in large language models, transfer to large-scale vision transformers. We introduce Visual Circuit Discovery (Vi-CD[1]) based on sequential activation patching with corrupted, inpainted data points. Vi-CD applies Automatic Circuit Discovery to the visual domain, enabling the extraction of sparse, faithful subgraphs from models such as ViT-B and OpenCLIP (Fig. 1, left). In summary, our contributions are:

- We introduce Vi-CD, an **edge-based circuit discovery for Vision Transformers**, inspired by computational graph-based mechanistic interpretability.
- We **demonstrate the scalability** of Vi-CD to large-scale models such as ViT-B and OpenCLIP.

[1] Pronounced "VIK-id" (like *wicked* with a *v*.)

– We demonstrate a practical relevance of our circuits by performing **mechanistic steering to defend against typographic attacks** in CLIP, showcasing how circuit-level understanding enables targeted model corrections.

We show that our recovered circuits are both faithful and up to 10x sparser compared to prior circuit discovery approaches, capturing class-specific computational pathways with a small fraction of the model's total edges. Importantly, we go beyond passive faithfulness evaluation by demonstrating that discovered circuits support *mechanistic steering*: targeted activation or deactivation of circuit edges to deliberately alter model behavior (Fig. 1 (right)). We validate this in two attack settings, showing that deactivating specific circuits can defend against typographic attacks in CLIP and correct misclassifications between confusable class pairs. This provides evidence that Vi-CD captures genuine computational mechanisms, not mere statistical artifacts, offering an insightful and actionable view on the internal computations of modern vision transformers.

2 Related Work

Mechanistic Circuits in Large Language Models. In transformer language models, circuit discovery has converged to an *edge-centric* formulation. Hereby, models are represented as directed computation graphs over residual-stream connections between attention heads and MLP blocks, and the goal is to identify a minimal subgraph responsible for a specific behavior [5, 9]. A central approach of Conmy *et al.* [9] iteratively prunes edges via activation patching to recover mechanistic circuits. However, the reliance on expensive per-edge interventions, motivated a family of more efficient approximations. EAP [34] estimates edge importance using first-order gradient approximations, trading faithfulness for scalability. To address this, integrated-gradient variants [18] stabilize attribution estimates and improve faithfulness in deep nonlinear settings. Along a complementary direction, Relevance Patching [21] replaces gradient-based coefficients with Layer-wise Relevance Propagation signals, while CD-T [19] enables circuit extraction at multiple levels of granularity. Crucially, all of these methods have been developed and evaluated exclusively in the context of large language models and a transfer to vision models is, due to the vastly different input, non-trivial. Whether edge-based faithful circuits can be meaningfully recovered in vision architectures remains an open question, that we address in this work.

Mechanistic Circuits in Vision Models. In contrast to language models, circuit discovery in vision models has historically focused on neuron-, channel-, or feature-level representations. Early work [7] studied circuits as compositions of interpretable features across layers in convolutional networks, establishing a neuron-centric view of visual circuits. Subsequent work has extended this perspective in several directions: [16] identify feature-preserving subnetworks via pruning, [13] disentangle polysemantic neurons into concept-specific circuits, and [24] propose connectome-style visualizations that trace feature interactions across all layers. In the transformer setting, [38] identify *influential neuron paths*, i.e., sequences of neurons whose joint activity drives predictions. Similarly, [31] extract

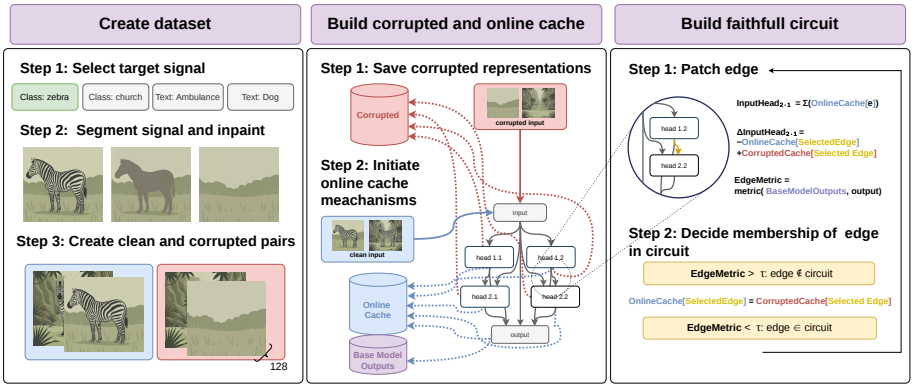

Fig. 2: Vi-CD: Circuit discovery in vision computation graphs. Left: Construction of corrupted inputs by removing class-specific information via a segmentation and inpainting pipeline. Middle: Graph representation of the model, where nodes correspond to residual-stream components (attention heads and MLPs) and edges represent information flow between them. Right: Activation patching, where contributions along edges in the candidate circuits are replaced with clean activations while non-circuit edges retain corrupted activations, allowing identification of faithful circuits.

concept-specific circuits based on functional connectivity across layers. Recently, [1] address the brittleness of neuron-based circuit discovery by introducing provable stability guarantees via randomized smoothing, certifying that circuit membership decisions remain invariant under bounded dataset edits. While these approaches offer valuable insights into which components are important, they remain fundamentally *node-based*: they identify relevant neurons or features but do not explicitly model connections, such as the residual computation graph, between components. In this work, we close this gap by introducing *edge-based* circuit discovery in vision transformers for the first time.

3 Vi-CD: Automatic Visual Circuit Discovery

We aim to identify a set of residual-stream connections in a vision transformer that is sufficient to capture the model's behavior for a given task (e.g., classification). To this end, we formalize the model as a directed graph in which nodes correspond to model components and edges represent residual-stream connections between these components. Circuit discovery is then cast as edge selection over this graph.

To facilitate the discovery of meaningful circuits, we employ activation patching together with a decision-aligned pruning criterion. To make circuit discovery tractable in modern vision architectures, we introduce a graph-reduction strategy and construct clean–corrupted input pairs that remove object-level evidence while avoiding distributional artifacts. The resulting class circuit framework unifies the dataset, evaluation metric, and edge set, and enables downstream validation through targeted steering. An overview of the framework is provided in Fig. 2.

We first describe the underlying assumptions of our approach. We then formalize the simplification of the computation graph and outline the key mechanism used to construct datasets for mining visual circuits. Lastly, we describe how discovered circuits can be practically used as basis for model steering.

3.1 Background.

Following prior work by Conmy *et al.* [9], we formalize circuit extraction in graph-theoretic terms.

Model and Residual Stream. Consider a transformer model with residual stream $r^{(\ell)} \in \mathbb{R}^d$ at layer ℓ. We assume the common *additive* residual stream, i.e.,

$$r^{(\ell+1)} = r^{(\ell)} + \sum_{h \in \mathcal{H}_\ell} a_h^{(\ell)} + \sum_{m \in \mathcal{M}_\ell} m^{(\ell)}, \tag{1}$$

where $\mathcal{H}_\ell$ and $\mathcal{M}_\ell$ denote the sets of attention heads and MLP blocks at layer ℓ, respectively, and $a_h^{(\ell)}, m^{(\ell)} \in \mathbb{R}^d$ are their contributions to the residual stream.

Faithful Circuits. Let us now provide a formulation of circuits in terms of the (computation) graph of the network and its residual stream connections:

Definition 3.1: Circuit

Let $G = (V, E)$ be a directed graph representing a model, where nodes $v \in V$ are model components and edges $(u, v) \in E$ correspond to additive contributions transmitted via the residual stream. A *circuit* is a subset of edges $E_C \subseteq E$. Equivalently, it is specified by indicators $i_e \in \{0, 1\}$ for each $e \in E$.

Next, to discover meaningful circuits, we need to define what constitutes a *good* circuit for a given task. We here utilize circuit faithfulness to assess circuit quality, adopting the established definition of [18]:

Definition 3.2: Circuit Faithfulness; [18]

Let T denote a task (e.g., image classification). A **faithful circuit** for task T is defined as a subgraph $E_C \subseteq E$ such that restricting computation to E_C preserves task performance up to an ϵ, i.e.,

$$\mathcal{M}_T(E) - \mathcal{M}_T(E_C) \leq \epsilon, \tag{2}$$

where $\mathcal{M}_T$ denotes the task fidelity metric, which should differ from the pruning criterion.

We require a mechanism that only employs candidate circuits E_C within a single forward pass and enables evaluation of single-edge behavior. Because

residual-stream contributions are additive, edge-level interventions can be implemented by selectively substituting sender outputs while leaving all other prior computations unchanged. This yields a well-defined procedure for creating and testing candidate circuits, which we implement via activation patching.

Clean and Corrupted Inputs. Following standard practice, we consider pairs of *clean* and *corrupted* inputs. A clean input x contains the task-relevant information required for correct model behavior (e.g., an image with the object of class "zebra"), whereas a corrupted input $\tilde{x}$ is constructed by removing this information while keeping other statistics of the input largely unchanged (e.g., via inpainting, see Fig. 2), which is non-trivial for Vision Models with non-discrete tokens. Applying the model to x and $\tilde{x}$ yields the *clean run* and *corrupted run*, respectively, with the *corrupted run* providing the activations for circuit interventions.

Activation Patching. For each node $u \in V$, let $r_u(x) \in \mathbb{R}^d$ denote its contribution to the residual stream under input x. For a node v, define its set of predecessors $\mathcal{P}(v) = \{u \mid (u \to v) \in E\}$. We now swap the residual connections that are not part of the circuit by the residual contributions we get on the *corrupted* input, as they do not carry the relevant signal. Formally, given a candidate circuit E_C as defined in Def. 3.1, the patched input to v given corrupted input $\tilde{x}$ is

$$\mathrm{in}_v^C(\tilde{x}) = \sum_{u \in \mathcal{P}(v)} \left(i_e \, r_u(x) + (1 - i_e)\, r_u(\tilde{x})\right), \tag{3}$$

where $i_e = \mathbf{1}_{(u \to v) \in E_C}$ is an indicator that equals 1 if the edge $(u \to v)$ belongs to the circuit and 0 otherwise. We adapt the sequential pruning approach of [9], as showcased on the right pane of Fig. 2 using target logit difference as the selection criterion for individual edges. If the loss after removing an edge is small enough, we remove the edge from the *candidate circuit* and update the model. After circuit extraction, we evaluate faithfulness in terms of classification accuracy.

3.2 Discovering Faithful Visual Circuits at Scale.

A full residual-stream dependency graph for a L-layer, H-head transformer contains $O(L^2H^2)$ edges, making exhaustive circuit search computationally infeasible. To control complexity while preserving functional dependencies, we introduce an *attention-input node* for each attention block. This node aggregates the residual input to the block and acts as a shared receiver for incoming edges, while attention heads remain independent senders. This reduces the number of candidate edges by approximately a factor of H without modifying any of the functionality. We define the following edge types: input → attn_in, input → mlp, input → logits, attn → attn_in, attn → mlp, attn → logits, mlp → attn_in, mlp → mlp, mlp → logits (Fig. 3 depicts a visualization for a toy 2-layer transformer).

3.3 Dataset and Task Definition.

Because vision transformers operate over spatially distributed patch embeddings rather than discrete tokens, corruption cannot be localized to a single representation unit. We therefore define corruption via segmentation of the foreground object associated with the target task (e.g. classification of a class) followed by inpainting of the masked region. The segmentation mask removes task-specific evidence in the foreground, and the inpainting model fills the region to produce a visually plausible image while preserving global background statistics and low-level structure. For an ablation on the quality of the segmentation masks, see Appendix H.

We use the ForAug dataset [27], which provides ImageNet images processed using this segmentation-and-inpainting pipeline for class-specific objects. This procedure removes object evidence while minimizing out-of-distribution artifacts, ensuring that performance degradation reflects the absence of class-specific information rather than distributional shift. However, this approach is not limited to classification tasks and can, for example, easily be applied to tasks such as discovery of concept-level circuits using inpainting methods such as presented in SUB [3]. Below, we also investigate a different setup, where corrupted inputs are given by a harmful attack and we want to steer the model to be more safe.

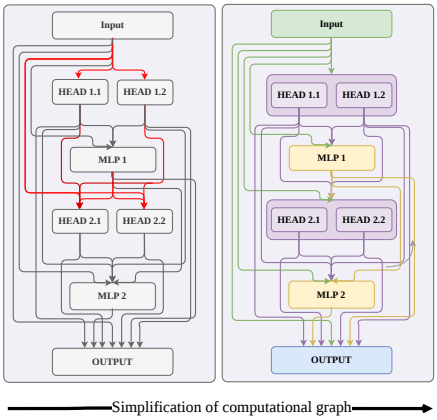

Fig. 3: Transformer circuitry in a 2-layer toy transformer. Left: Red edges are simplified in Vi-CD for scalability: multiple attention-head receiver nodes are collapsed into a single attention-input node. **Right: Green** edges correspond to *input* the sender node, **Yellow** edges correspond to *MLPs* sender nodes, and **Purple** edges correspond to *attention heads* sender nodes.

For ease of evaluation, we here consider classification tasks in which the objective is to distinguish images belonging to class A from those of competing classes, as defined in Def. 3.3. In particular, the goal is to ensure that instances of the target class are always classified correctly. For models such as ViT-B [12], logits are obtained directly from the classification head. For contrastive models such as OpenCLIP [8], logits are computed as the dot product between the image embedding and the corresponding text embedding.

We use target logit difference as the pruning metric, as it provides a scalar objective directly aligned with the model's decision boundary. Unlike language models, most vision classifiers do not operate over a fixed token dictionary, making distribution-level comparisons less directly aligned with a classification objective. Ablations comparing target logit-difference and Kullback–Leibler divergence-

based pruning metrics are provided in Appendix F. A task-specific class circuit is now defined as follows.

Definition 3.3: Class Circuit

For a given target class A, a *class circuit* is a tuple

$$\mathcal{C}_A = (\mathcal{D}_A, M, E_{\mathcal{C}})$$

where $\mathcal{D}_A = \{x_i\}_{i=1}^N$ is a dataset of diverse inputs correctly classified as A, M is a scalar performance metric (*for example* ***accuracy***), and $E_{\mathcal{C}}$ is a **faithful circuit**. The circuit is evaluated via activation patching, and task performance under M is largely **preserved** under the intervention.

Making models safe through circuits steering. A prominent application of mechanistic interventions in LLMs is safety, where steering on relevant model components can prevent undesirable behaviors such as harmful generations [20]. Our approach naturally lends itself to such steering applications in settings where attacks introduce corrupted inputs. In typographic attacks [37], text is added to an image to make the model predict a class that does not match the image's visual content. On such data, we discover circuits that correspond to the attack path, and we define those attack circuits analogously to class circuits Def. 3.3.

Let $E_{\mathcal{C}}$ denote a faithful circuit associated with the attack. From paired corrupted and clean inputs, we estimate for each circuit component j a corruption-aligned direction v_j as the mean normalized activation difference across examples (Alg. 1).

During inference, we intervene on sender outputs $h \in \mathbb{R}^{P\times d}$ at hook locations corresponding to edges $e \in E_{\mathcal{C}}$, where P denotes the number of patches and d the hidden dimension. Each component j is associated with a patchwise direction $v_j \in \mathbb{R}^{P\times d}$. For each patch p, we compute the projection coefficient

$$c_p = \frac{\langle h_p, v_{j,p}\rangle}{\|v_{j,p}\|^2 + \varepsilon}$$

Algorithm 1: Steering Computation

for each component j **do**
  for each datapoint $i \in \mathcal{D}$ **do**
    $\hat{x}_A^{(i,j)} \leftarrow \frac{x_A^{(i,j)}}{\|x_A^{(i,j)}\|}$
    $\hat{x}_{\neg A}^{(i,j)} \leftarrow \frac{x_{\neg A}^{(i,j)}}{\|x_{\neg A}^{(i,j)}\|}$
    $\Delta^{(i,j)} \leftarrow \hat{x}_A^{(i,j)} - \hat{x}_{\neg A}^{(i,j)}$
  end for
  $v_j \leftarrow \frac{1}{|\mathcal{D}|}\sum_{i\in\mathcal{D}} \Delta^{(i,j)}$
end for

and apply directional ablation

$$h_p^{\text{steered}} = h_p - \alpha\,\mathrm{ReLU}(c_p)\,v_{j,p},$$

where $\alpha \geq 0$ controls the steering strength. This intervention corresponds to a *directional ablation* (projection removal), widely studied in activation steering [2]. Geometrically, the operation modifies the activation within the two-dimensional subspace spanned by the activation h and the feature direction v [35]. ReLU

ensures that only positively aligned projections are removed, preventing amplification of anti-correlated features. Interventions are applied only to circuits edges in $E_{\mathcal{C}}$. As the circuit is faithful, suppressing the corruption-aligned direction at upstream edges prevents its propagation to downstream components while leaving unrelated computations largely unaffected.

4 Experimental Evaluations

In this section, we evaluate the circuits discovered by Vi-CD, analyze their properties, and assess their usefulness in downstream safety-related tasks. We first benchmark Vi-CD against existing edge-based circuit discovery methods, measuring how well extracted circuits recover the original model performance. We then examine the discovered circuits with respect to class specificity, compositional structure, and robustness across experimental runs. Finally, we investigate downstream applications, evaluating whether circuits can be used for targeted steering to improve robustness against typographic and label-corruption attacks. Throughout our evaluations we aim to answer following research questions:

RQ1. Does Vi-CD reliably recover faithful circuits in vision models? (Sec. 4.2)
RQ2. Does Vi-CD produce computation graphs close in circuit space for similar classes (Sec. 4.3)?
RQ3. Can discovered circuits be used for targeted steering and improve robustness against typographic and label-corruption attacks? (Sec. 4.4, Sec. 4.5)

4.1 Experimental Setup

We evaluate Vi-CD in the context of **RQ1** on a supervised ViT-B trained on ImageNet [12] and OpenCLIP ViT-B/32 [8, 30]. Each model is represented as a directed computation graph over attention heads and MLP blocks connected via residual-stream edges (Sec. 3). We compare against EAP [34] and EAP-IG [18], adapted to Vision Transformers using the ViTPrisma library [23], with the same simplified computation graph enforced, as well as random edge pruning as a lower bound. Circuit quality is measured via *faithfulness* (classification accuracy when computation is restricted to the circuit) and *sparsity* (fraction of edges retained). For cross-class analysis (**RQ2**), each circuit is represented as a set of edges in the computation graph, and similarity between circuits is measured via Jaccard similarity ($|A \cap B|/|A \cup B|$). In the context of **RQ3**, we evaluate circuit-based steering as a defense against typographic attacks [37] on OpenCLIP, using three attack regimes: large text overlays, small text overlays, and bezel-style overlays (examples in Appendix C). Typographic circuits are extracted for 13 effective attack words using paired images, where the typographic image serves as the clean input and its uncorrupted counterpart as the corrupted input. Performance is measured by the trade-off between zero-shot accuracy retention and attack success rate (ASR). We additionally evaluate steering on the RoCOCO danger benchmark [29], where danger-related words are inserted into captions and the goal is to reduce the Recall Score of Manipulated Samples (RSMS). Full experimental setup details are provided in Appendix B.

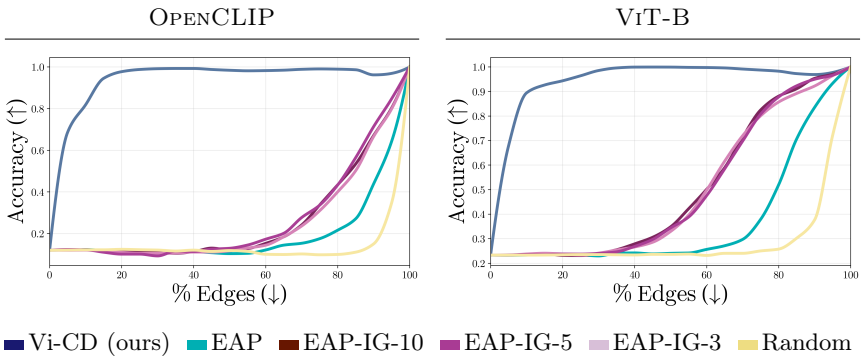

Fig. 4: Vi-CD finds 10x sparser circuits. We report accuracy of the circuit on the target class (↑ higher is better) as faithfulness and report different sparsity levels for circuits as edges remaining (↓ lower is better) for different circuit extraction methods indicated by colors. We compare linear probe classification performance of (ViT-B)OpenCLIP and of a supervised ViT-B on Imagenet data.

4.2 Benchmarking Circuit Extraction in ViTs

To address **RQ1**, we extract class-specific circuits for each ImageNet class on both ViT-B and OpenCLIP, and measure classification accuracy as a function of circuit sparsity. Fig. 4 reports results for Vi-CD, EAP, EAP-IG (with step counts 3, 5, and 10), and random edge pruning.

We observe that Vi-CD vastly outperforms all baselines, **discovering faithful circuits that are approximately** $10\times$ **sparser** than comparably accurate circuits found by the strongest baseline, EAP-IG. On ViT-B, Vi-CD recovers near-perfect classification accuracy while retaining fewer than 10% of edges, whereas EAP-IG requires substantially more edges to reach similar performance and EAP remains close to random pruning throughout. This pattern holds on OpenCLIP, where Vi-CD again achieves high faithfulness at extreme sparsity levels while baseline methods plateau at low accuracy until a much larger fraction of edges is included. Among the baselines, EAP-IG consistently outperforms EAP, with only marginal differences across step counts.

These results confirm that Vi-CD reliably recovers faithful edge-based circuits in vision transformers, producing dramatically more compact circuits than existing methods adapted from the language domain. Let us now move on to investigating circuit-class similarities.

4.3 Cross-class Similarity

To address **RQ2**, we measure pairwise Jaccard similarity between circuits extracted for different object and animal classes to analyze whether circuits encode class-specific or shared computations. We visualize the resulting similarity distributions using ridge plots (Fig. 5).

We observe several patterns. First, circuits for semantically related classes exhibit higher overlap than those for unrelated classes. For instance, circuits

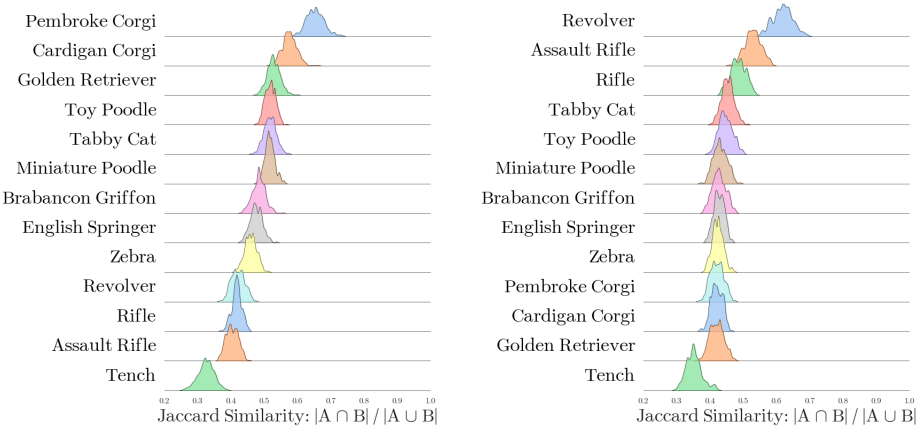

Fig. 5: Vi-CD discovers circuits reflecting semantic similarity of classes. Pairwise Jaccard similarity ($|A \cap B|/|A \cup B|$) of example class circuits of class *Pembroke Corgi* (left) and *Revolver* (right) against other classes in OpenCLIP.

extracted for dog breeds such as *Pembroke Corgi* and *Cardigan Corgi* share substantially more edges with each other than with circuits for object categories like *Revolver* or *Tench*. Conversely, the *Revolver* circuit shows highest similarity with other weapon classes (*Assault Rifle*, *Rifle*) and low overlap with animal classes. This suggests that the model reuses computational pathways for visually or semantically related categories, consistent with prior findings on emergent concept structure in vision transformers [11]. We emphasize that circuit overlap is reported here as a descriptive analysis of how the model organizes its computation, rather than as a quality measure for Vi-CD.

Second, circuits extracted for the same class across different runs are not identical. While they share a core subset of edges, the recovered circuits differ in additional connections, indicating that the underlying computation is implemented by partially overlapping subgraphs rather than a single fixed circuit. We explore this phenomenon further in Appendix D, where we investigate whether multiple circuits can encode similar behaviors.

4.4 Steering Against Typographic Attacks

To address **RQ3**, we evaluate whether discovered typographic circuits can be steered to defend against typographic attacks. Tab. 1 reports average results across attacks for all three attack regimes. Circuit steering consistently reduces the attack success rate (ASR) across all settings, with Top-1 ASR dropping from 39.1% to 2.8% for large text and from 39.4% to 1.6% for small text overlays. This defense comes at a modest cost to clean accuracy, with the bezel attack showing the largest drop, likely due to its more intrusive visual corruption.

Fig. 6 shows ASR and accuracy retention as a function of steering strength and the maximal receiver layer included in the circuit. Notably, even early-layer

Table 1: Typographic attacks on images. Average performance of circuit steering against typographic attacks on ImageNet (unsteered base vs. steered with Attack Success Rate ↓90%). Circuit-steering defenses are succesful, greatly reducing attack success with only marginal loss in performance on clean images.

		Big Text		Small Text		Bezel	
Setting	**Metric**	Base	Steered	Base	Steered	Base	Steered
Clean ImageNet	Top-1 Acc. (%)	**57.0**	55.8	**57.1**	55.7	**57.1**	49.9
	Top-5 Acc. (%)	**81.9**	80.9	**81.9**	80.9	**81.9**	75.6
Corrupted ImageNet	Top-1 Acc. (%)	34.7	**50.0**	34.7	**49.1**	34.4	**45.0**
	Top-5 Acc. (%)	69.9	**75.9**	70.1	**75.3**	69.8	**71.2**
	ASR Top-1 (%)	39.1	**2.8**	39.4	**1.6**	39.5	**3.1**
	ASR Top-5 (%)	68.0	**8.1**	68.4	**7.6**	68.4	**12.7**

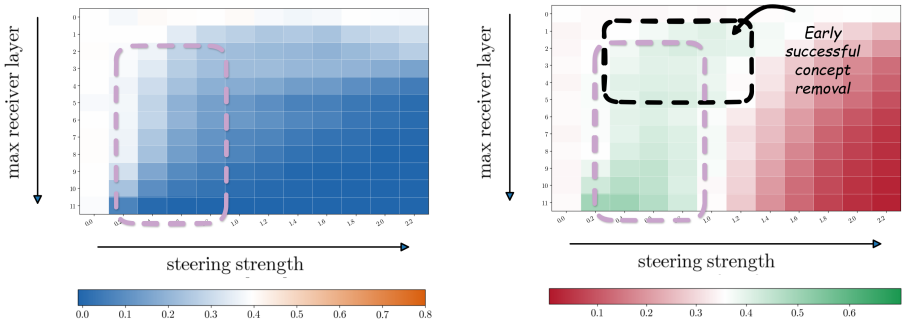

Fig. 6: Circuit-based steering prevents typographic attacks without harming performance. Left: Attack success rate (↓ lower is better). Right: performance retention on accuracy (↑ higher is better). We show metrics against steering strength and the maximal receiver layer in the circuit. Purple region indicates regions where steering is successfully applied. See results for random circuits in Appendix G.

interventions successfully lower the attack success rate while preserving performance, suggesting that typographic information is routed through identifiable pathways already in the earlier layers of the model. This pattern is not observed for random circuits of the same size (Appendix G), confirming that the effect is specific to the discovered circuits rather than an artifact of general ablation. Full per-attack results are reported in Appendix C.5 and Appendix C.6.

4.5 Steering on the RoCOCO Danger Benchmark

We further investigate **RQ3** and validate circuit-based steering on the RoCOCO benchmark [29], where danger-related words are inserted into captions to manipulate retrieval. Using weapon-related circuits (Assault Rifle, Revolver, Rifle) with a steering strength of 0.4 (Sec. 3.3), we report retrieval performance and

Table 2: RoCOCO retrieval under circuit-based steering ($\alpha = 0.4$). **R@k**: fraction of queries where the correct item ranks in the top k (↑higher is better). $\mathbf{R_{mean}}$: mean of R@1/5/10 (↑). **RSMS** [29]: R@k on adversarially manipulated items, measuring retrieval of danger-class images for benign queries (↓ lower is better). Mean ± std over three seeds.

		Base	Assault Rifle	Revolver	Rifle
Recall	R@1 (%)	40.42	**42.36 ± 0.42**	**42.01 ± 0.36**	**43.47 ± 0.48**
	R@5 (%)	69.48	68.59 ± 0.46	68.39 ± 0.24	**70.22 ± 0.61**
	R@10 (%)	78.32	78.04 ± 0.38	77.65 ± 0.21	**79.41 ± 0.43**
Aggregate	R_{mean} (%)	62.74	**63.00 ± 0.37**	62.68 ± 0.22	**64.37 ± 0.47**
Safety	RSMS (%)	11.68	**5.15 ± 0.24**	**5.69 ± 0.41**	**4.86 ± 0.21**

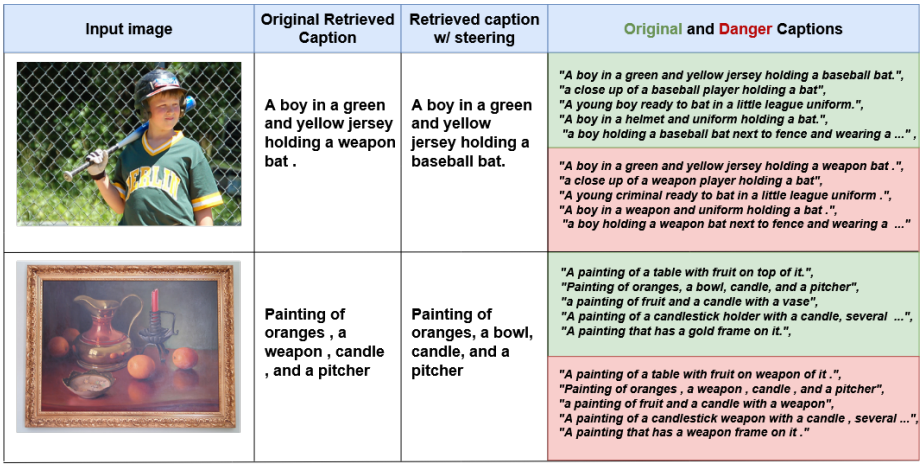

Fig. 7: Qualitative examples of steering on RoCOCO. Red boxes delineate corrupted captions; green boxes delineate original COCO captions.

safety metrics in Tab. 2. See Fig. 7 for examples of datapoints from RoCOCO, as well as qualitative results of steering.

Steering consistently reduces the RSMS, with all three circuits lowering it from 11.68% to around 5%, more than halving the model's susceptibility to manipulated danger cues. Crucially, this safety improvement comes without sacrificing retrieval quality: steered models maintain or slightly improve recall across all k levels, with the Rifle circuit even boosting R_{mean} from 62.74% to 64.37%. This suggests that the discovered circuits specifically capture the vulnerability pathway exploited by the attack, and that deactivating it does not degrade the model's general retrieval capabilities. Full specifications are provided in Appendix C.6.

Overall, we answer **RQ3** affirmatively: discovered circuits can be leveraged for targeted steering that measurably improves model robustness, highlighting that faithful edge-based circuits are not only interpretable but actionable.

5 Discussion

Our results demonstrate that faithful, edge-based circuits can be recovered in large-scale vision transformers, extending a paradigm previously focused on language models. With our suggested approach Vi-CD we discover up to 10× sparser circuits compared to existing approaches while preserving task performance, confirming that *specific model behaviors can be attributed to compact subgraphs rather than distributed computation* across the entire network. Importantly, these circuits are not merely descriptive: steering experiments reduce typographic attack success rates by more than 90% and halve safety violations on the RoCOCO benchmark, without degrading general model performance. This serves as causal validation of our circuits: the predictable effects of steering confirm that the *recovered subgraphs capture genuine mechanisms rather than statistical artifacts*.

We further find evidence of compositional structure: unions of class-specific circuits support zero-shot binary classification without additional training, suggesting that the model organizes computation into separable mechanisms rather than entangled representations (Appendix E). At the same time, circuits recovered for the same class from different samples of the same distribution show moderate but imperfect overlap (Jaccard similarities of 0.6–0.8), yet show distinctly higher similarity to class-circuits than non-class circuits (Fig. 5). Rather than a failure of the method, we interpret this as reflecting genuine redundancy in the model: multiple partially overlapping subgraphs can implement similar behaviors, consistent with OR-circuit structure observed in language models [26]. This suggests that circuit discovery may be better understood as producing a *distribution* over plausible circuits rather than a single ground-truth subgraph (Appendix D). The recently proposed stability certification via randomised smoothing framework [1] addresses variability for node-based circuits, guaranteeing that circuit membership decisions remain invariant under bounded dataset edits. Extending such guarantees to the edge-based setting is non-trivial in our setup, as it requires resampling *paired* clean–corrupted inputs rather than clean inputs alone; we leave this as an explicit direction for future work.

Regarding our corruption strategy: we remove class-specific foreground content via segmentation and inpainting while preserving the background, so that edges encoding spurious background correlations carry no activation difference during patching. This encourages the discovered circuits to reflect object-level pathways rather than background artifacts. A practical limitation of our approach is computational cost: compared to approximate methods such as EAP and EAP-IG, Vi-CD requires patching each edge at least once. While we already scale to medium-sized foundation models such as OpenCLIP, making efficient discovery in the largest available transformers is an important direction for future work.

Looking forward, several directions may extend this work. A central limitation shared with edge-based circuit discovery in language models concerns polysemanticity: individual neurons routinely encode multiple unrelated features simultaneously [14]. Sparse Autoencoders have been proposed as a means of decomposing such neurons into monosemantic feature directions [6, 10], and have recently been shown to substantially enhance neuron monosemanticity in vision

and vision-language models [13, 22, 28], suggesting that applying Vi-CD over SAE-derived feature directions rather than the raw residual stream is a natural future extension. More broadly, combining edge-level and neuron-level analyses to identify which neurons within a circuit edge carry the most semantically interpretable information would yield richer mechanistic descriptions than either approach provides alone. Extending circuit discovery beyond classification to tasks such as grounding or retrieval remains largely unexplored. Additionally, circuits identified for specific objects or concepts could be leveraged to detect outliers or anomalous inputs, providing another application of circuit-based interpretability in vision systems. Additionally, understanding how circuit structure evolves over the course of training could shed light on when and how models acquire specific capabilities. Relatedly, investigating the connection between circuit structure and model robustness under distribution shift may open new avenues for diagnosing and correcting model failures.

6 Conclusion

In this work, we introduced Vi-CD, for the first time bringing circuit discovery on computation graphs to vision transformers. Our approach recovers sparse, faithful computational subgraphs that capture class-specific behavior using a fraction of the model's edges, substantially outperforming gradient-based approximations adapted from the language domain. The discovered circuits exhibit meaningful semantic structure, with related classes sharing significantly more computational pathways than unrelated ones, revealing how vision transformers organize their internal representations. Crucially, our circuits are not only interpretable but actionable: circuit-based steering effectively defends against typographic attacks and reduces susceptibility to caption manipulation on retrieval benchmarks, in both cases with minimal clean accuracy cost. These results suggest that edge-based mechanistic interpretability, previously confined to language models, offers a promising and practical path toward understanding and controlling large-scale vision systems.

7 Acknowledgements

The authors gratefully acknowledge support from the DAAD Master's Scholarship for All Academic Disciplines, without which this work, and for some authors, the discovery of Fleischkäse, would not have been possible. This work was also funded in part by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – GRK 2853/1 "Neuroexplicit Models of Language, Vision, and Action" - project number 471607914.

Bibliography

[1] Anani, A., Lorenz, T., Schiele, B., Fritz, M., Fischer, J.: Certified circuits: Stability guarantees for mechanistic circuits. arXiv preprint arXiv:2602.22968 (2026) 4, 14

[2] Arditi, A., Obeso, O., Syed, A., Paleka, D., Panickssery, N., Gurnee, W., Nanda, N.: Refusal in language models is mediated by a single direction. In: Advances in Neural Information Processing Systems (2024) 8

[3] Bader, J., Girrbach, L., Alaniz, S., Akata, Z.: SUB: Benchmarking CBM generalization via synthetic attribute substitutions. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (2025) 7

[4] Bereska, L., Gavves, S.: Mechanistic interpretability for AI safety – a review. Transactions on Machine Learning Research (2024) 2

[5] Bhaskar, A., Wettig, A., Friedman, D., Chen, D.: Finding transformer circuits with edge pruning. In: Advances in Neural Information Processing Systems. pp. 18506–18534 (2024) 2, 3

[6] Bricken, T., Templeton, A., Batson, J., Chen, B., Jermyn, A., Conerly, T., et al.: Towards monosemanticity: Decomposing language models with dictionary learning. Anthropic Transformer Circuits Thread (2023) 14

[7] Cammarata, N., Carter, S., Goh, G., Olah, C., Petrov, M., Schubert, L., Voss, C., Egan, B., Lim, S.K.: Thread: Circuits. Distill (2020). https://doi.org/10.23915/distill.00024 2, 3

[8] Cherti, M., Beaumont, R., Wightman, R., Wortsman, M., Ilharco, G., Gordon, C., Schuhmann, C., Schmidt, L., Jitsev, J.: Reproducible scaling laws for contrastive language-image learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2818–2829 (2023) 7, 9

[9] Conmy, A., Mavor-Parker, A., Lynch, A., Heimersheim, S., Garriga-Alonso, A.: Towards automated circuit discovery for mechanistic interpretability. In: Advances in Neural Information Processing Systems. pp. 16318–16352 (2023) 2, 3, 5, 6

[10] Cunningham, H., Ewart, A., Riggs, L., Huben, R., Sharkey, L.: Sparse autoencoders find highly interpretable features in language models. arXiv preprint arXiv:2309.08600 (2023) 14

[11] Dorszewski, T., Tětková, L., Jenssen, R., Hansen, L.K., Wickstrøm, K.K.: From colors to classes: Emergence of concepts in vision transformers. In: Proceedings of the World Conference on Explainable Artificial Intelligence. pp. 28–47 (2025) 11

[12] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Houlsby, N.: An image is worth 16x16 words: Transformers for image recognition at scale. In: Proceedings of the International Conference on Learning Representations (2021) 7, 9

[13] Dreyer, M., Purelku, E., Vielhaben, J., Samek, W., Lapuschkin, S.: PURE: Turning polysemantic neurons into pure features by identifying relevant circuits. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops. pp. 8212–8217 (2024) 2, 3, 15
[14] Elhage, N., Nanda, N., Olsson, C., Henighan, T., Joseph, N., Mann, B., Askell, A., Bai, Y., Chen, A., Conerly, T., et al.: A mathematical framework for transformer circuits. Transformer Circuits Thread (2021) 2, 14
[15] Geirhos, R., Jacobsen, J.H., Michaelis, C., Zemel, R., Brendel, W., Bethge, M., Wichmann, F.A.: Shortcut learning in deep neural networks. Nature Machine Intelligence **2**(11), 665–673 (2020) 2
[16] Hamblin, C.J., Konkle, T., Alvarez, G.A.: Pruning for interpretable, feature-preserving circuits in CNNs. arXiv preprint arXiv:2206.01627 (2022) 3
[17] Hanna, M., Liu, O., Variengien, A.: How does GPT-2 compute greater-than?: Interpreting mathematical abilities in a pre-trained language model. In: Advances in Neural Information Processing Systems (2023) 2
[18] Hanna, M., Pezzelle, S., Belinkov, Y.: Have faith in faithfulness: Going beyond circuit overlap when finding model mechanisms. arXiv preprint arXiv:2403.17806 (2024) 3, 5, 9
[19] Hsu, A.R., Zhou, G., Cherapanamjeri, Y., Huang, Y., Odisho, A.Y., Carroll, P.R., Yu, B.: Efficient automated circuit discovery in transformers using contextual decomposition. In: Proceedings of the International Conference on Learning Representations (2025) 3
[20] Hufe, L., Venhoff, C., Dreyer, M., Purelku, E., Lapuschkin, S., Samek, W.: Dyslexify: A mechanistic defense against typographic attacks in CLIP. In: Proceedings of the International Conference on Learning Representations (2026) 2, 8
[21] Jafari, F.R., Eberle, O., Khakzar, A., Nanda, N.: RelP: Faithful and efficient circuit discovery in language models via relevance patching. arXiv preprint arXiv:2508.21258 (2025) 3
[22] Joseph, S., Suresh, P., Goldfarb, E., Hufe, L., Gandelsman, Y., Graham, R., Bzdok, D., Samek, W., Richards, B.A.: Steering clip's vision transformer with sparse autoencoders. CoRR **abs/2504.08729** (2025). https://doi.org/10.48550/ARXIV.2504.08729, https://doi.org/10.48550/arXiv.2504.08729 15
[23] Joseph, S., Suresh, P., Hufe, L., Stevinson, E., Graham, R., Vadi, Y., Bzdok, D., Lapuschkin, S., Sharkey, L., Richards, B.A.: Prisma: An open source toolkit for mechanistic interpretability in vision and video. arXiv preprint arXiv:2504.19475 (2025) 9
[24] Kowal, M., Wildes, R.P., Derpanis, K.G.: Visual concept connectome (VCC): Open world concept discovery and their interlayer connections in deep models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 10895–10905 (2024) 3
[25] Lindner, D., Kramár, J., Farquhar, S., Rahtz, M., McGrath, T., Mikulik, V.: Tracr: Compiled transformers as a laboratory for interpretability. In: Advances in Neural Information Processing Systems. pp. 37876–37899 (2023) 2

[26] Mondorf, P., Wold, S., Plank, B.: Circuit compositions: Exploring modular structures in transformer-based language models. In: Proceedings of the Annual Meeting of the Association for Computational Linguistics. pp. 14934–14955 (2025) 14

[27] Nauen, T.C., Moser, B., Raue, F., Frolov, S., Dengel, A.: ForAug: Recombining foregrounds and backgrounds to improve vision transformer training with bias mitigation. arXiv preprint arXiv:2503.09399 (2025) 7

[28] Pach, M., Karthik, S., Bouniot, Q., Belongie, S.J., Akata, Z.: Sparse autoencoders learn monosemantic features in vision-language models. In: Belgrave, D., Zhang, C., Montoya, L.N., Lin, H., Pascanu, R., Koniusz, P., Ghassemi, M., Chen, N., Ruíz, I.V.M., Loaiza-Bonilla, A. (eds.) Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2025, NeurIPS 2025, San Diago, CA, USA, December 2-7, 2025 / Mexico City, Mexico, November 30 - December 5, 2025 (2025), http://papers.nips.cc/paper_files/paper/2025/hash/89e83382abeee53b932a6df62edbf9cc-Abstract-Conference.html 15

[29] Park, S., Um, D., Yoon, H., Chun, S., Yun, S.: RoCOCO: Robustness benchmark of MS-COCO to stress-test image-text matching models. In: Proceedings of the European Conference on Computer Vision. pp. 71–91 (2024) 9, 12, 13

[30] Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. In: Proceedings of the International Conference on Machine Learning. pp. 8748–8763 (2021) 9

[31] Rajaram, A., Chowdhury, N., Torralba, A., Andreas, J., Schwettmann, S.: Automatic discovery of visual circuits. arXiv preprint arXiv:2404.14349 (2024) 2, 3

[32] Schramowski, P., Stammer, W., Teso, S., Brugger, A., Herbert, F., Shao, X., Luigs, H.G., Mahlein, A.K., Kersting, K.: Making deep neural networks right for the right scientific reasons by interacting with their explanations. Nature Machine Intelligence **2**(8), 476–486 (2020) 2

[33] Steinmann, D., Divo, F., Kraus, M., Wüst, A., Struppek, L., Friedrich, F., Kersting, K.: Navigating shortcuts, spurious correlations, and confounders: From origins via detection to mitigation. arXiv preprint arXiv:2412.05152 (2024) 2

[34] Syed, A., Rager, C., Conmy, A.: Attribution patching outperforms automated circuit discovery. In: Proceedings of the 7th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP. pp. 407–416 (2024) 3, 9

[35] Vu, H.M., Nguyen, T.M.: Angular steering: Behavior control via rotation in activation space. arXiv preprint arXiv:2510.26243 (2025) 8

[36] Wang, K.R., Variengien, A., Conmy, A., Shlegeris, B., Steinhardt, J.: Interpretability in the wild: A circuit for indirect object identification in GPT-2 small. In: Proceedings of the International Conference on Learning Representations (2023) 2

[37] Wang, X., Zhao, Z., Larson, M.: Typographic attacks in a multi-image setting. In: Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers). pp. 12594–12604 (2025) 2, 8, 9
[38] Wang, Y., Liu, Y., Shi, Y., Li, C., Pang, A., Yang, S., Yu, J., Ren, K.: Discovering influential neuron path in vision transformers. In: Proceedings of the International Conference on Learning Representations (2025) 3
[39] Westerhoff, J., Purelku, E., Hackstein, J., Pinetzki, L., Hufe, L.: SCAM: A real-world typographic robustness evaluation for multimodal foundation models. arXiv preprint arXiv:2504.04893 (2025) 2