FROM NEURONS TO CONCEPTS: COARSE-TO-FINE NEURON-PATH DISCOVERY FOR INTERPRETABLE VISION TRANSFORMERS

Anonymous authors
Paper under double-blind review

ABSTRACT

Understanding *how* Vision Transformers (ViTs) compute their predictions remains challenging: most explanations highlight salient patches but do not reveal cross-layer internal information flow. Neuron-path discovery addresses this by tracing a prediction to a sequence of influential FFN neurons across layers, yet existing methods are hard to interpret (index-only neuron IDs) and expensive due to per-layer search over thousands of neurons. We present **Concept-Guided Coarse-to-Fine Neuron-Path Discovery**, a framework that injects human-interpretable visual concepts into neuron-path search while preserving neuron-level grounding. We first construct a **global concept bank** from attention-selected patch embeddings and partition FFN neurons into **concept-aligned groups** shared across layers. We then perform **coarse-to-fine path discovery**: (i) search for an influential *concept-level* path in a low-dimensional group-gated attribution space, and (ii) refine each selected concept group to a representative neuron to obtain a final neuron path. This design reduces the search space without sacrificing causal relevance. Experiments on ImageNet with multiple ViT variants show that our method finds neuron paths that are **more causally faithful under interventions** than prior neuron-path baselines, while achieving **5× practical speedups**. Moreover, the resulting concept paths expose coherent semantic trajectories across depth, enabling inspection of how visual evidence emerges and where depth-localized bottlenecks arise.

1 INTRODUCTION

Vision Transformers (ViTs) (Dosovitskiy, 2020; Touvron et al., 2021) have become a dominant architecture for visual recognition. Yet their predictions often remain difficult to explain in terms of *internal computation*, limiting debugging, robustness analysis, and trustworthy deployment.

Most existing ViT explanations are *input-level*—saliency maps, attention visualizations, and token/patch importance scores (Selvaraju et al., 2017; Zhou et al., 2016; Sundararajan et al., 2017; Vig, 2019; Abnar & Zuidema, 2020; Brocki et al., 2023; Chefer et al., 2021). While useful for showing *where* the model looks, they provide limited insight into *how* representations are transformed and composed across layers, i.e., the model's internal information flow (Lu et al., 2021; Dai et al., 2022; Foote et al., 2023).

A promising direction is *neuron-path discovery* (Wang et al., 2025), which models information flow as a sequence of influential FFN neurons (one per layer) whose joint attribution supports a prediction. This moves beyond isolated neuron importance by explicitly capturing cross-layer dependencies.

However, existing neuron-path methods face a central bottleneck for practical interpretability: their outputs are index-based (hard to interpret semantically) and their search is expensive (must score many neurons per layer under attribution-based objectives). This exposes a persistent tension between *faithfulness*, *human interpretability*, and *efficiency*: neuron-level paths are faithful but difficult to read and costly to compute, whereas higher-level semantic abstractions are easier to read but may lose mechanism-level grounding.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

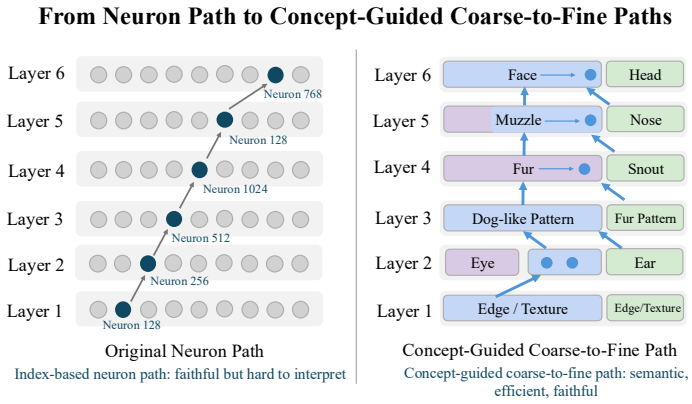


Figure 1: Overview of concept-guided coarse-to-fine neuron-path discovery for Vision Transformers.

In this work, we propose **Concept-Guided Coarse-to-Fine Neuron-Path Discovery**, which reconciles these objectives by integrating semantic concepts directly into neuron-path search. We first discover an influential *concept-level path* that describes how visual concepts propagate across layers, and then refine each concept node to a single influential neuron within the selected group. This coarse-to-fine strategy preserves neuron-level grounding while greatly reducing the search space and yielding semantically meaningful paths.

Our approach consists of three components. We construct a *global concept bank* from attention-selected image tokens, organize FFN neurons into *concept-aligned groups*, and perform coarse-to-fine discovery by searching over concept-level paths before refining to neuron-level paths.

We summarize our contributions as follows:

- We introduce a concept-guided framework that bridges neuron-level attribution and human-interpretable visual concepts for Vision Transformers.
- We propose a coarse-to-fine neuron-path discovery strategy that significantly reduces computational cost while preserving attribution fidelity.
- Extensive experiments across ViT variants show that our method achieves stronger intervention-based faithfulness, delivers **5**$\times$ practical speedups over neuron-path baselines, and yields semantically interpretable paths that reveal how visual concepts emerge across transformer layers.

2 Related Work

From Input-Level Explanations to Internal Mechanisms. Most interpretability methods for ViTs attribute predictions to input regions or tokens using attention- or gradient-based saliency, including attention visualization and rollout (Vig, 2019; Abnar & Zuidema, 2020; Brocki et al., 2023) and CAM/Grad-CAM or Integrated Gradients adaptations (Zhou et al., 2016; Selvaraju et al., 2017; Sundararajan et al., 2017; Chefer et al., 2021). While effective for evidence localization, these approaches primarily explain *where* the model attends and do not recover cross-layer internal information flow. Recent studies further examine the faithfulness of such explanations (Wu et al., 2024a;b; Zheng et al., 2024), highlighting limitations of attribution scores as indicators of causal influence, but remaining focused on input-level representations.

Paths, Structure, and Concept-Based Interpretability. A complementary line of work analyzes internal units via activation, probing, and attribution, identifying neurons or features associated with specific functions (Lu et al., 2020; Dai et al., 2022; Foote et al., 2023; Meng et al., 2022), and modeling cross-layer influence patterns (Lu et al., 2021). Most closely related, NeuronPath (Wang et al., 2025) models information flow in ViTs as influential neuron paths across layers using joint attribution, but yields index-based paths and incurs high computational cost. Other approaches introduce structured or single-path explanations through architectural or decoding constraints (Kim et al., 2022; Lao et al., 2024; Chowdhury et al., 2025), or employ human-interpretable concepts such as TCAV, Concept

Bottleneck Models, and concept-factorization or prototypical-concept explanations (Kim et al., 2018; Koh et al., 2020; Yuksekgonul et al., 2022; Kim et al., 2023; Hernandez et al., 2021; Mu & Andreas, 2020; Fel et al., 2022; Dreyer et al., 2023). These methods improve semantic accessibility or impose interpretability structure, but typically do not recover causally grounded, cross-layer information flow in pretrained models. Our method bridges these directions by using concepts to guide efficient discovery of cross-layer neuron paths, producing interpretable and intervention-ready explanations grounded in neuron-level computation.

Mechanistic Circuit Discovery vs. Path Probing. Mechanistic circuit discovery identifies subgraphs of a network that are causally responsible for a target behavior, with recent transformer-circuit work emphasizing automation, scalable attribution, edge pruning, and feature-level causal graphs (Conmy et al., 2023; Syed et al., 2023; Hanna et al., 2024; Bhaskar et al., 2024; Kramár et al., 2024; Marks et al., 2024); vision-model variants include pruning-based feature-preserving circuits, visual concept connectomes, automatic visual circuit discovery (Rajaram et al., 2024), and circuit-based neuron disambiguation. These methods recover *distributed* mechanisms over many units/edges. We instead pursue a complementary goal: extracting a *compact, semantically readable, depth-ordered* trace of evidence flow in a pretrained ViT. We deliberately constrain the explanation to one concept-aligned group per layer plus a neuron-level refinement, yielding a *path probe* rather than a full circuit—trading completeness for compactness, interpretability, and efficiency, and well-suited to diagnosing semantic trajectories and depth-localized bottlenecks. Full-circuit methods are therefore complementary rather than direct baselines for this path-probe setting.

3 Problem Setup and Preliminaries

We formalize influential neuron-path discovery in ViTs and introduce an *intervention-ready* scoring space for efficient cross-layer search. Notation follows Table 3.

3.1 Intervenable FFN Features in ViT Blocks

Let F be a pretrained ViT classifier. Given an input image x and a target class y^*, we denote the target score by $F_{y^*}(x)$. A ViT contains L transformer blocks. We focus on the FFN sublayer and denote by $\mathbf{w}^{(l)}(x) \in \mathbb{R}^D$ the post-activation FFN hidden vector of the `[CLS]` token at block l (Table 3). Each coordinate of $\mathbf{w}^{(l)}(x)$ is an internal feature channel that can be perturbed, making FFN features natural units for causal testing.

3.2 Cross-Layer Neuron-Path Discovery

Following NeuronPath (Wang et al., 2025), we seek *cross-layer* explanations as *information-flow paths* traversing transformer blocks. A neuron path is a length-L sequence selecting one FFN neuron per block,

$$\pi = (i_1, \ldots, i_L), \qquad i_l \in \{1, \ldots, D\}. \tag{1}$$

Given (x, y^*), our goal is to discover a path whose selected internal features jointly support the target score $F_{y^*}(x)$. To evaluate faithfulness, we rely on *interventions* on internal features. We will reparameterize such interventions using continuous *group gates* and perform attribution in the resulting gate space.

3.3 Gate Space and Intervention-Aligned Attribution

Motivation. Neuron-level scoring requires evaluating D candidates per layer, which is computationally expensive and yields index-only explanations. We therefore introduce a low-dimensional, explicitly intervenable gate space that supports efficient scoring aligned with intervention effects on $F_{y^*}(x)$.

Definition 3.1 (Gate space)**.** Let $\mathcal{G}^{(l)} = \{G_1^{(l)}, \ldots, G_{M_l}^{(l)}\}$ be a partition of FFN neurons into groups at block l (defined in Section 4). We attach a scalar gate $z_{l,m} \in [0, 1]$ to each group $G_m^{(l)}$ and collect all gates into a vector $z \in [0, 1]^{|\mathcal{I}|}$ indexed by

$$\mathcal{I} = \{(l, m) : l \in [L],\ m \in [M_l]\}. \tag{2}$$

We denote the target output under a gate configuration z by $F_{y^*}(x; z)$.

Definition 3.2 (Deletion baseline and reactivation)**.** Let $\mathbf{1}$ denote the all-active gate, i.e., $(\mathbf{1})_{l,m} = 1$. For any set $S \subseteq \mathcal{I}$, the deletion baseline $z^{\mathrm{del}}(S)$ deactivates exactly the groups in S while keeping all other groups active:

$$z^{\mathrm{del}}_{l,m}(S) = \begin{cases} 0, & (l,m) \in S, \\ 1, & (l,m) \notin S. \end{cases} \tag{3}$$

We interpret *reactivating* S as moving from $z^{\mathrm{del}}(S)$ to $\mathbf{1}$.

Definition 3.3 (Integrated Gradients in gate space)**.** Given a baseline z^0 and a target $\tilde{z}$, Integrated Gradients (IG) (Sundararajan et al., 2017) attributes the difference $F_{y^*}(x; \tilde{z}) - F_{y^*}(x; z^0)$ along the straight-line path $z(\alpha) = z^0 + \alpha(\tilde{z} - z^0)$. In vector form,

$$\mathrm{IG}(z^0 \to \tilde{z}) = (\tilde{z} - z^0) \odot \int_0^1 \nabla_z F_{y^*}(x; z(\alpha))\, d\alpha. \tag{4}$$

The (l,m)-th coordinate is

$$\mathrm{IG}_{l,m}(z^0 \to \tilde{z}) = (\tilde{z}_{l,m} - z^0_{l,m}) \int_0^1 \tfrac{\partial F_{y^*}(x; z(\alpha))}{\partial z_{l,m}}\, d\alpha, \quad (l,m) \in \mathcal{I}. \tag{5}$$

Property 1 (Completeness)**.** *IG satisfies completeness:*

$$\sum_{(l,m)\in\mathcal{I}} \mathrm{IG}_{l,m}(z^0 \to \tilde{z}) = F_{y^*}(x; \tilde{z}) - F_{y^*}(x; z^0). \tag{6}$$

Intervention-ready attribution convention. Throughout the paper, we quantify the effect of turning groups *on* by reactivation IG from deletion baselines. For any set $S \subseteq \mathcal{I}$, we define the reactivation attribution vector

$$\mathbf{a}(S) = \mathrm{IG}\big(z^{\mathrm{del}}(S) \to \mathbf{1}\big), \qquad a_{l,m}(S) = [\mathbf{a}(S)]_{l,m}. \tag{7}$$

3.4 Intervention-Based Faithfulness and GG-JAS Scoring

We assess faithfulness through interventions on internal features: removing or enhancing selected units should change the model behavior with respect to $F_{y^*}(x)$ (see Appendix G). Therefore, path discovery requires a scoring function aligned with the effect of interventions on $F_{y^*}(x)$.

Definition 3.4 (GG-JAS: Group-Gated Joint Attribution Score)**.** Given a set of gate coordinates $S \subseteq \mathcal{I}$, we define the group-gated joint attribution score (GG-JAS) as the total reactivation attribution assigned to coordinates in S:

$$\mathrm{GGJAS}(S) = \sum_{(l,m)\in S} a_{l,m}(S) = \sum_{(l,m)\in S} \mathrm{IG}_{l,m}\big(z^{\mathrm{del}}(S) \to \mathbf{1}\big). \tag{8}$$

From sets to paths. GG-JAS is defined over a set of gate coordinates $S \subseteq \mathcal{I}$, while our task is to discover a cross-layer path. We connect the two by mapping a (prefix) group path to the corresponding coordinate set in the gate space. For a length-l prefix $\gamma_{1:l} = (m_1, \ldots, m_l)$, define the prefix set $S(\gamma_{1:l}) = \{(t, m_t)\}_{t=1}^{l}$ (Table 3). For a full group path $\gamma = (m_1, \ldots, m_L)$, we similarly write $S(\gamma) = S(\gamma_{1:L}) = \{(l, m_l)\}_{l=1}^{L}$. Accordingly, the path-level score is

$$\mathrm{GGJAS}(\gamma) \;\triangleq\; \mathrm{GGJAS}(S(\gamma)). \tag{9}$$

In Section 4, we will instantiate S in Definition 3.4 in two ways: (i) *local* scoring with singleton sets $S = \{(l,m)\}$ for layer-wise group ranking, and (ii) *prefix-conditioned* scoring with prefix sets $S(\gamma_{1:l})$ to capture cross-layer interactions. These choices yield the variants $\mathrm{GGJAS}_{\mathrm{local}}$ and $\mathrm{GGJAS}_{\mathrm{cond}}$ (Table 3) used for greedy group-path discovery.

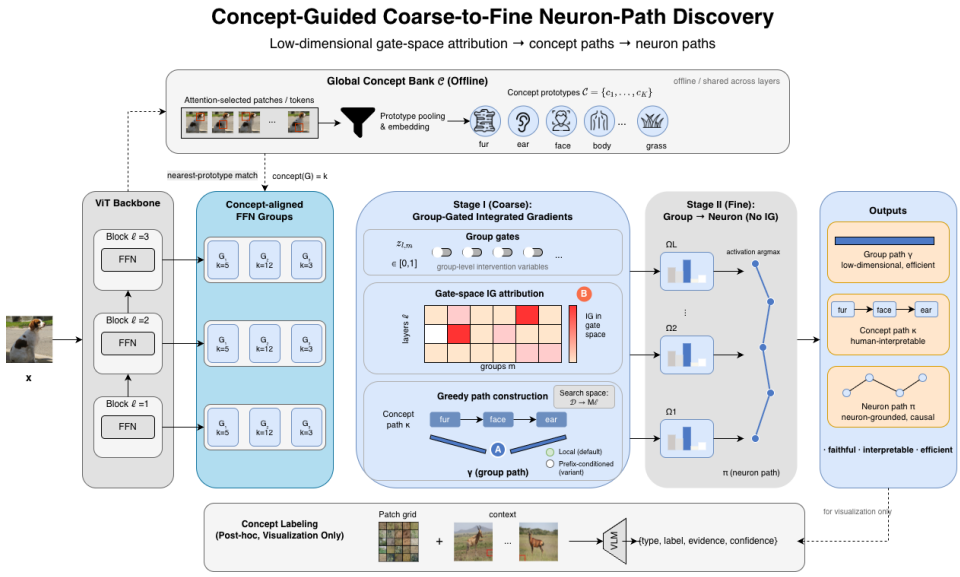


Figure 2: **Concept-guided coarse-to-fine neuron-path discovery for Vision Transformers.** Given (x, y^*), we (i) construct a global concept bank $\mathcal{C}$ and align FFN neurons into concept-aligned groups $\{G_m^{(l)}\}$; (ii) score groups in a low-dimensional gate space via GG-JAS and greedily select a group path γ (and concept path κ); (iii) refine each selected group to a representative neuron to obtain the final neuron path π.

4 Concept-Guided Coarse-to-Fine Neuron-Path Discovery

We propose a concept-guided coarse-to-fine framework for neuron-path discovery in Vision Transformers. Given an input x and target class y^*, our method outputs a group path $\gamma = (m_1, \ldots, m_L)$, the corresponding concept path $\kappa = (k_1, \ldots, k_L)$, and a refined neuron path $\pi = (i_1, \ldots, i_L)$ (Table 3).

Overview and key idea. Neuron-level scoring is expensive (large D) and produces index-only paths. We instead score *intervenable group gates* and search in the resulting low-dimensional gate space. Our scoring function is GG-JAS (Definition 3.4), which quantifies the effect of *reactivating* a specified set of gate coordinates $S \subseteq \mathcal{I}$ via IG from the deletion baseline (Definitions 3.2–3.3). We then select one group per layer greedily to form γ, and refine each selected group to a representative neuron with a lightweight activation rule.

4.1 Concept-Aligned FFN Neuron Groups

Our coarse stage operates on neuron *groups* rather than individual neurons. We assume a global concept bank $\mathcal{C} = \{c_1, \ldots, c_K\} \subset \mathbb{R}^d$ (Appendix C), which defines a shared semantic space across layers.

For each transformer block l, FFN neurons are partitioned into disjoint groups $\mathcal{G}^{(l)} = \{G_1^{(l)}, \ldots, G_{M_l}^{(l)}\}$ with $\bigcup_{m=1}^{M_l} G_m^{(l)} = [D]$ and $G_m^{(l)} \cap G_{m'}^{(l)} = \emptyset$ for $m \neq m'$. Neurons within a group behave similarly across data (Appendix D). We embed each group into the concept space and assign it a semantic label by nearest-prototype matching:

$$k = \text{concept}(G_m^{(l)}) \in [K]. \tag{10}$$

This alignment reduces the per-layer search space from D neurons to M_l groups and provides an interpretable label for each selected group along a path.

4.2 Stage I: Group-Gated Scoring and Greedy Search

Stage I selects one group per layer to form a group path $\gamma = (m_1, \ldots, m_L)$ and concept path $\kappa = (k_1, \ldots, k_L)$ where $k_l = \text{concept}(G_{m_l}^{(l)})$.

Group-gated model. We attach a scalar gate $z_{l,m} \in [0,1]$ to each group $G_m^{(l)}$. For a neuron index $j \in G_m^{(l)}$, the gated post-activation is

$$w_j^{(l)}(x;z) = z_{l,m}\, w_j^{(l)}(x). \tag{11}$$

Collecting all gates yields $z \in [0,1]^{|\mathcal{I}|}$ indexed by $\mathcal{I} = \{(l,m) : l \in [L], m \in [M_l]\}$ (Definition 3.1), and we denote the gated target score as $F_{y^*}(x;z)$.

Scoring objective: instantiate GG-JAS by choosing a set S. GG-JAS is defined for any set of gate coordinates $S \subseteq \mathcal{I}$ (Definition 3.4) as

$$\mathrm{GGJAS}(S) = \sum_{(l,m)\in S} a_{l,m}(S) = \sum_{(l,m)\in S} \mathrm{IG}_{l,m}\big(z^{\mathrm{del}}(S) \to \mathbf{1}\big)\,. \tag{12}$$

i.e., the total reactivation attribution assigned to the coordinates in S under the shared deletion baseline $z^{\mathrm{del}}(S)$. Thus, different scoring variants correspond to different choices of S.

Variants: local vs. prefix-conditioned scoring. We define two variants used for greedy search (Table 3).

(i) Local (layer-wise) score. For a candidate group $G_m^{(l)}$, use the singleton set $S = \{(l,m)\}$:

$$\mathrm{GGJAS}_{\mathrm{local}}(l,m) \triangleq \mathrm{GGJAS}(\{(l,m)\}) = a_{l,m}(\{(l,m)\})\,. \tag{13}$$

This measures the causal effect of turning on $G_m^{(l)}$ while keeping all other groups active.

(ii) Conditional (prefix-conditioned) score. For a prefix $\gamma_{1:l} = (m_1, \ldots, m_l)$, use the prefix set $S(\gamma_{1:l}) = \{(t, m_t)\}_{t=1}^{l}$:

$$\mathrm{GGJAS}_{\mathrm{cond}}(\gamma_{1:l}) \;\triangleq\; \mathrm{GGJAS}(S(\gamma_{1:l}))\,. \tag{14}$$

This captures cross-layer interactions by attributing the joint reactivation effect of the entire prefix under the shared deletion baseline $z^{\mathrm{del}}(S(\gamma_{1:l}))$.

Layer-progressive greedy selection. We construct γ progressively by selecting one group per layer.

For the prefix-conditioned variant, at layer l we evaluate each candidate extension $\gamma_{1:l} = (\gamma_{1:l-1}, m)$ and choose

$$m_l = \arg\max_{m\in[M_l]} \mathrm{GGJAS}_{\mathrm{cond}}\big(\gamma_{1:l-1}, m\big). \tag{15}$$

For the local variant:

$$m_l = \arg\max_{m\in[M_l]} \mathrm{GGJAS}_{\mathrm{local}}(l,m). \tag{16}$$

Unless otherwise specified, we adopt $\mathrm{GGJAS}_{\mathrm{local}}$ as the default coarse criterion due to stronger causal intervention effects in our experiments (Section 5).

Practical computation. In practice, IG is computed using a discrete approximation with T steps. Accordingly, we use $\widetilde{\mathrm{GGJAS}}(\gamma)$ to denote the discrete approximation of $\mathrm{GGJAS}(\gamma)$ (Table 3), and defer the discretization formula and implementation details to Appendix E.

4.3 Stage II: Refinement from Groups to Neurons

Stage I yields one selected group per layer. We refine each selected group to a concrete neuron to obtain a final neuron path. For layer l, restrict candidates to $\Omega_l = G_{m_l}^{(l)}$ and select the most active neuron:

$$i_l = \arg\max_{j\in\Omega_l} w_j^{(l)}(x). \tag{17}$$

This yields $\pi = (i_1, \ldots, i_L)$. We do not recompute neuron-level IG within each group; the coarse stage already identifies a functional subspace, and activation-based refinement is efficient and effective (Appendix I.3).

Algorithm 1 Concept-Guided Coarse-to-Fine Neuron-Path Discovery

Require: input–target pair (x, y^*); layerwise neuron groups $\{\mathcal{G}^{(l)} = \{G_m^{(l)}\}_{m=1}^{M_l}\}_{l=1}^{L}$; concept assignment $\mathrm{concept}(\cdot)$; attribution budget T; scoring rule $\mathrm{Score} \in \{\mathrm{GGJAS}_{\mathrm{local}}, \mathrm{GGJAS}_{\mathrm{cond}}\}$
Ensure: group path γ, concept path κ, neuron path π
1: Initialize empty paths γ, κ, π
2: **for** $l = 1$ to L **do**
3: **Coarse (group selection):** $m_l \leftarrow \arg\max_{m \in [M_l]} \mathrm{Score}(x, y^*, l, m, \gamma_{1:l-1}; T)$
4: Append m_l to γ and $\mathrm{concept}(G_{m_l}^{(l)})$ to κ
5: **Fine (neuron refinement):** $i_l \leftarrow \arg\max_{j \in G_{m_l}^{(l)}} w_j^{(l)}(x)$; append i_l to π
6: **end for**
7: **return** γ, κ, π

4.4 Complexity Analysis and Efficiency

Neuron-level path discovery scores D candidates per layer. In contrast, our coarse stage scores only M_l groups at layer l.

Under a discrete IG approximation with T steps, each candidate score requires T forward+backward evaluations of the gated model (Appendix E). Thus, the coarse-stage cost scales as

$$O\left(T \sum_{l=1}^{L} M_l \cdot \mathrm{Cost}(F)\right), \tag{18}$$

where $\mathrm{Cost}(F)$ denotes the cost of one forward+backward evaluation. When $M_l \ll D$, this yields substantial savings compared to neuron-level scoring, which scales as $O\Big(T \sum_{l=1}^{L} D \cdot \mathrm{Cost}(F)\Big)$. The fine stage requires only within-group activation comparisons and incurs negligible overhead.

5 Experiments

5.1 Setup

Data and models. All experiments are conducted on ImageNet-1k (Deng et al., 2009) and reported on the validation set. All evaluations use pretrained models without finetuning to isolate internal neuron-path behavior. We evaluate three ViT architectures with increasing capacity: ViT-B/16, ViT-B/32, and ViT-L/32 (Dosovitskiy, 2020).

Baselines. We compare against (i) **Activation** (selecting neurons by activation magnitude), (ii) **Influence Pattern** (Lu et al., 2021), and (iii) **NeuronPath** (Wang et al., 2025) (greedy neuron-path discovery by maximizing Joint Attribution Score, JAS, Appendix E.5).

Metrics. We evaluate methods along the three axes in our core claim: *causal faithfulness*, *efficiency*, and *semantic interpretability*. Causal faithfulness is measured primarily by intervention-based metrics: **RAD** (Removal Accuracy Deviation) and **EAD** (Enhancement Accuracy Deviation) under hard path-level removal (zeroing) and enhancement (doubling) (Appendix G). For reference, we also report **JAS**, but do not treat it as a primary faithfulness metric.

Fairness and shared resources. Our method uses a global concept bank and layerwise neuron grouping. For controlled evaluation, we construct a single concept bank and grouping and use them consistently across experiments (Appendix C, Appendix D). Neuron-level baselines still search in the original neuron space and do not use group-gated attribution during discovery.

5.2 Main Results: Faithfulness and Efficiency

Causal faithfulness under interventions. Table 1 reports intervention-based causal metrics on ImageNet-1k. Across all three ViT architectures, our concept-guided coarse-to-fine method achieves the largest **RAD** and **EAD**, i.e., the strongest accuracy changes under hard path-level removal/enhancement, indicating that it identifies *causally influential* neuron paths. For instance, on ViT-L/32 our prefix-free variant induces an 8.24% accuracy drop under path removal and an 8.39%

Table 1: Quantitative comparison of influential neuron-path discovery on ImageNet-1k (val). We report intervention-based causal metrics **RAD** (higher is better) and **EAD** (higher is better) as primary measures of faithfulness, and include **JAS** as a reference attribution score. PC denotes prefix-conditioned group attribution, PF denotes prefix-free (layer-wise) group attribution.

	ViT-B/16			ViT-B/32			ViT-L/32		
Method	RAD↑	EAD↑	JAS↑	RAD↑	EAD↑	JAS↑	RAD↑	EAD↑	JAS↑
Activation	0.07%	0.33%	-0.0034	0.15%	0.45%	0.0288	0.16%	0.86%	-0.0065
Influence Pattern (Lu et al., 2021)	0.50%	0.46%	0.0412	1.24%	0.83%	0.0841	1.41%	1.12%	0.1227
NeuronPath (Wang et al., 2025)	2.40%	2.04%	**0.4078**	3.81%	3.06%	**0.6610**	5.28%	5.02%	**1.0086**
Ours (PC + argmax)	3.91%	3.54%	0.2142	5.39%	4.74%	0.2967	6.72%	6.72%	0.5875
Ours (PF + argmax)	**4.36%**	**3.96%**	0.2464	**5.59%**	**4.89%**	0.2992	**8.24%**	**8.39%**	0.6707

Table 2: **Efficiency and faithfulness (RAD) under varying IG budgets** on ViT-L/32. Wall-clock time per image (mean ± std, seconds), peak GPU memory (GiB), and Removal Accuracy Drop (RAD, %, higher is better). Speedup is relative to NeuronPath under the same IG budget; PF+argmax achieves consistent ∼5× speedups while yielding higher RAD across all budgets.

Steps T	NeuronPath			Ours (PF + argmax)			
	Time(s)↓	Mem.↓	RAD↑	Time(s)↓	Mem.↓	Speed.↑	RAD↑
5	4.86 ± 0.45	4.20	5.5	0.95 ± 0.10	3.00	5.13	**8.2**
10	5.47 ± 0.15	5.03	5.7	1.09 ± 0.07	3.47	5.03	**8.4**
20	8.05 ± 0.05	6.63	5.7	1.52 ± 0.06	4.42	5.29	**8.3**
40	13.58 ± 0.06	9.80	5.8	2.48 ± 0.06	6.34	5.48	**8.4**

gain under path enhancement, outperforming NeuronPath by a clear margin. These results directly support our claim that the discovered paths reflect the model's causal computation rather than merely correlational saliency.

JAS-greedy does not imply faithfulness. NeuronPath optimizes the Joint Attribution Score (JAS) by construction, yet it is consistently below our best variant on RAD/EAD, while our method can achieve stronger interventions despite lower JAS. This reveals an objective mismatch: *greedily maximizing attribution scores does not reliably select the most causally decisive paths*. We provide controlled diagnostics within NeuronPath's own candidate space in Appendix I.1.

Efficiency of coarse-to-fine search. We measure end-to-end computational cost in a controlled setting (ViT-L/32; concept bank size $= 1024$; batch size $= 1$; mean ± std over 10000 images), reporting wall-clock runtime per image and peak GPU memory (Table 2). Across IG budgets $T \in \{5, 10, 20, 40\}$, our coarse-to-fine variant (PF+argmax) consistently reduces runtime and memory, achieving stable $\sim 5\times$ speedups over NeuronPath (e.g., $5.13\times$ at $T = 5$ and $5.48\times$ at $T = 40$). For ViT-L/32, the FFN hidden dimension is $D = 4096$ while coarse-stage attribution operates on $M_l \approx 1024$ concept-aligned groups per layer, suggesting a *theoretical* dimension reduction of $D/M_l \approx 4\times$; empirically, the observed speedups exceed this baseline. Importantly, these efficiency gains do not compromise faithfulness: PF+argmax consistently yields higher Remove Accuracy Drop (RAD) scores than NeuronPath across all budgets (Table 2). Additional results are reported in Appendix I.2.

5.3 Layer-wise Causal Decomposition

Path-level interventions verify that paths are causally influential, but do not reveal *where* influence concentrates. We therefore perform a layer-wise causal decomposition: for each input we intervene on a single layer l at a time using the same removal/enhancement operators as above, but evaluated with margin-based deviations $\mathrm{RAD}_{\text{margin}}(l), \mathrm{EAD}_{\text{margin}}(l)$ (Appendix G.2, more sensitive under small budgets), together with matched-size in-layer controls (random and top-activation neurons/groups).

Findings on ViT-L/32. Figure 3 shows layer-wise margin profiles for PF+argmax at neuron and concept-group granularity. Random controls remain near zero across layers, while intervening on a single discovered concept group matches single-neuron effects in mid-to-late layers under the same budget—concept-aligned grouping preserves causal signal. Causal effects concentrate in layers 7–20, with a pronounced spike at layer 22 where a single-unit intervention strongly modulates the decision margin, indicating a depth-localized information bottleneck.

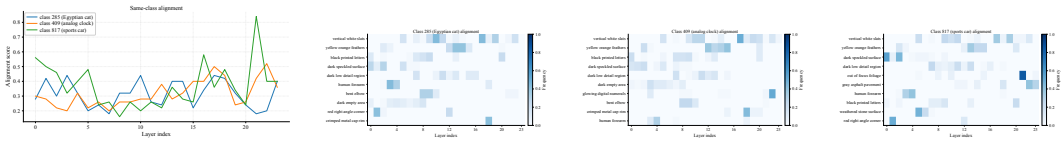
5.4 Semantic Interpretability of Concept Paths

We evaluate whether concept-guided paths yield interpretable semantic-evidence trajectories and enable dataset-level diagnostics not available from index-based neuron paths.



(a) EAD$_{\text{margin}}(l)$. (b) RAD$_{\text{margin}}(l)$.

Figure 3: Layer-wise causal effect profiles on ViT-L/32 under single-unit interventions, comparing PF+argmax at neuron and concept-group granularity with random and activation-matched in-layer controls. The pronounced spike at layer 22 indicates a depth-localized information bottleneck.



(a) Alignment curves (b) Class 285 (Egyptian cat) (c) Class 409 (analog clock) (d) Class 817 (sports car)

Figure 4: **Same-class semantic alignment across layers (ViT-L/32).** (a) Within-class alignment score $A_y(l) = \max_k \text{freq}_y(l,k)$ for three ImageNet classes. (b–d) Heatmaps of concept selection frequencies $\text{freq}_y(l,k)$ across layers (y-axis: top concepts; x-axis: layer index): darker cells = concepts repeatedly selected at the same depth.

Same-class semantic alignment across layers. Figure 4 reports within-class alignment $A_y(l) = \max_k \text{freq}_y(l,k)$ and per-class concept-frequency heatmaps for three ImageNet classes. Depth-dependent peaks appear consistently: same-class images repeatedly select a sparse set of concepts at matching layers, indicating class-aligned semantic routing at specific depths rather than arbitrary sequences; high alignment concentrating on background-like concepts can also surface shortcut reliance.

Diagnosing ViT decisions via concept paths. Translating an index-based neuron path into a high-level concept path exposes the *decision information flow* of a ViT in human-readable terms. Figure 11 illustrates this on a ViT-L/32 prediction of *Border collie* (class 232): RAD$_{\text{margin}}(l)$ identifies three depth-localized bottlenecks where a single concept-aligned unit dominates the decision margin—**L17** aligns with a *blurred green background* concept, **L21** with *long dog fur*, and **L23** with a *pointed animal ear*. This depth-resolved, concept-labeled trajectory turns the prediction into a diagnosable chain of evidence: which concepts the model relies on, at which depth, and how strongly each controls the output (more examples in Appendix K). Faithfulness is robust across coarse-/fine-stage variants and concept-construction choices (Appendix I.3, I.4).

6 Conclusion

We proposed **Concept-Guided Coarse-to-Fine Neuron-Path Discovery**, combining a global concept bank with concept-aligned groups for group-gated attribution and lightweight neuron refinement. Across ViT variants, the discovered paths are more causally faithful under intervention (higher RAD/EAD) and substantially faster than prior baselines, with concept paths making cross-layer information flow inspectable across inputs.

7 Limitation.

A key limitation of our method is its dependence on the quality of the learned concept bank. If the discovered concepts are noisy, background-biased, or poorly aligned with human semantics, the resulting concept paths may be misleading even when the selected neuron paths are causally influential. Future work should study more robust concept construction and evaluate concept stability across datasets and distribution shifts.

AI USE STATEMENT

In this work, we used a multimodal generative AI model (gpt-5.2) to assign post-hoc human-readable labels and generate brief evidence summaries for learned visual concepts from patch grids and highlighted context images. The model was queried using the fixed prompts, output schema, sampling procedure, and confidence-based selection protocol described in Appendix J. These AI-generated labels were used for qualitative semantic interpretation and visualization; they did not affect concept-bank construction, neuron grouping, neuron-path discovery, causal interventions, or the computation of the reported quantitative metrics.

We also used large language models to improve the clarity, grammar, and readability of the manuscript and to assist with manuscript-format and submission-compliance review. We did not use generative AI to originate the research question, formulate scientific claims, design the methodology or experiments, or determine quantitative results.

The authors reviewed and verified all AI-assisted text and all AI-generated labels included or analyzed in the paper. The authors take full responsibility for the final manuscript.

ETHICS STATEMENT

This work studies pretrained vision transformers using ImageNet-1k and does not involve human participants, user studies, or the collection of new personal or sensitive data. We do not introduce a new dataset. ImageNet may nevertheless contain societal and representation biases inherited from its data collection process, and some images may contain people. A multimodal generative AI model is used solely for post-hoc labeling of learned visual concepts. Such labels may reflect model biases, hallucinations, or spurious background correlations and should not be treated as ground-truth semantic annotations. Consequently, the resulting concept paths are intended as research and model-debugging tools rather than as a standalone basis for high-stakes decisions or claims about model safety, fairness, or reliability. We discuss these limitations and the potential for misleading semantic interpretations in Section 7 and Appendix J.

REPRODUCIBILITY STATEMENT

We provide the complete problem formulation and coarse-to-fine discovery algorithm in Sections 3–4 and Algorithm 1. Appendix C describes concept-bank construction, while Appendix D specifies neuron grouping and concept alignment. Appendix E provides the definition, derivation, and discrete implementation of GG-JAS, and Appendix G defines all intervention-based evaluation metrics. The model and data protocol, preprocessing, intervention rules, attribution configuration, hardware, and implementation details are documented in Appendix H. Appendix I reports additional comparisons, robustness analyses, and hyperparameter ablations. Finally, Appendix J provides the prompts, output schema, sampling procedure, confidence filtering, and examples used for multimodal concept labeling.

REFERENCES

Samira Abnar and Willem Zuidema. Quantifying attention flow in transformers. *arXiv preprint arXiv:2005.00928*, 2020.

Adithya Bhaskar, Alexander Wettig, Dan Friedman, and Danqi Chen. Finding transformer circuits with edge pruning. *ArXiv*, abs/2406.16778, 2024. URL https://api.semanticscholar.org/CorpusID:270702321.

Lennart Brocki, Jakub Binda, and Neo Christopher Chung. Class-discriminative attention maps for vision transformers. *arXiv preprint arXiv:2312.02364*, 2023.

Hila Chefer, Shir Gur, and Lior Wolf. Transformer interpretability beyond attention visualization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 782–791, 2021.

Arpita Chowdhury, Dipanjyoti Paul, Zheda Mai, Jianyang Gu, Ziheng Zhang, Kazi Sajeed Mehrab, Elizabeth G Campolongo, Daniel Rubenstein, Charles V Stewart, Anuj Karpatne, et al. Prompt-cam: Making vision transformers interpretable for fine-grained analysis. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 4375–4385, 2025.

Arthur Conmy, Augustine N. Mavor-Parker, Aengus Lynch, Stefan Heimersheim, and Adrià Garriga-Alonso. Towards automated circuit discovery for mechanistic interpretability. *ArXiv*, abs/2304.14997, 2023. URL https://api.semanticscholar.org/CorpusID:258418244.

Damai Dai, Li Dong, Yaru Hao, Zhifang Sui, Baobao Chang, and Furu Wei. Knowledge neurons in pretrained transformers. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 8493–8502, 2022.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, 2009. doi: 10.1109/CVPR.2009.5206848.

Alexey Dosovitskiy. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Maximilian Dreyer, Reduan Achtibat, Wojciech Samek, and Sebastian Lapuschkin. Understanding the (extra-)ordinary: Validating deep model decisions with prototypical concept-based explanations. *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 3491–3501, 2023. URL https://api.semanticscholar.org/CorpusID:265466851.

Thomas Fel, Agustin Picard, Louis Béthune, Thibaut Boissin, David Vigouroux, Julien Colin, Rémi Cadene, and Thomas Serre. Craft: Concept recursive activation factorization for explainability. *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2711–2721, 2022. URL https://api.semanticscholar.org/CorpusID:253708233.

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

Michael Hanna, Sandro Pezzelle, and Yonatan Belinkov. Have faith in faithfulness: Going beyond circuit overlap when finding model mechanisms. *ArXiv*, abs/2403.17806, 2024. URL https://api.semanticscholar.org/CorpusID:268691935.

Evan Hernandez, Sarah Schwettmann, David Bau, Teona Bagashvili, Antonio Torralba, and Jacob Andreas. Natural language descriptions of deep visual features. In *International Conference on Learning Representations*, 2021.

Been Kim, Martin Wattenberg, Justin Gilmer, Carrie Cai, James Wexler, Fernanda Viegas, et al. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav). In *International conference on machine learning*, pp. 2668–2677. PMLR, 2018.

Eunji Kim, Dahuin Jung, Sangha Park, Siwon Kim, and Sungroh Yoon. Probabilistic concept bottleneck models. *arXiv preprint arXiv:2306.01574*, 2023.

Sangwon Kim, Jaeyeal Nam, and Byoung Chul Ko. Vit-net: Interpretable vision transformers with neural tree decoder. In *International conference on machine learning*, pp. 11162–11172. PMLR, 2022.

Pang Wei Koh, Thao Nguyen, Yew Siang Tang, Stephen Mussmann, Emma Pierson, Been Kim, and Percy Liang. Concept bottleneck models. In *International conference on machine learning*, pp. 5338–5348. PMLR, 2020.

János Kramár, Tom Lieberum, Rohin Shah, and Neel Nanda. Atp*: An efficient and scalable method for localizing llm behaviour to components. *ArXiv*, abs/2403.00745, 2024. URL https://api.semanticscholar.org/CorpusID:268230363.

Danning Lao, Qi Liu, Jiazi Bu, Junchi Yan, and Wei Shen. Vitree: Single-path neural tree for step-wise interpretable fine-grained visual categorization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 2866–2873, 2024.

Kaiji Lu, Piotr Mardziel, Klas Leino, Matt Fredrikson, and Anupam Datta. Influence paths for characterizing subject-verb number agreement in lstm language models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 4748–4757, 2020.

Kaiji Lu, Zifan Wang, Piotr Mardziel, and Anupam Datta. Influence patterns for explaining information flow in bert. *Advances in Neural Information Processing Systems*, 34:4461–4474, 2021.

Samuel Marks, Can Rager, Eric J. Michaud, Yonatan Belinkov, David Bau, and Aaron Mueller. Sparse feature circuits: Discovering and editing interpretable causal graphs in language models. *ArXiv*, abs/2403.19647, 2024. URL https://api.semanticscholar.org/CorpusID:268732732.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in gpt. *Advances in Neural Information Processing Systems 35*, 2022. URL https://api.semanticscholar.org/CorpusID:255825985.

Jesse Mu and Jacob Andreas. Compositional explanations of neurons. *Advances in Neural Information Processing Systems*, 33:17153–17163, 2020.

Achyuta Rajaram, Neil Chowdhury, Antonio Torralba, Jacob Andreas, and Sarah Schwettmann. Automatic discovery of visual circuits. *ArXiv*, abs/2404.14349, 2024. URL https://api.semanticscholar.org/CorpusID:266646733.

Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pp. 618–626, 2017.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *International conference on machine learning*, pp. 3319–3328. PMLR, 2017.

Aaquib Syed, Can Rager, and Arthur Conmy. Attribution patching outperforms automated circuit discovery. *ArXiv*, abs/2310.10348, 2023. URL https://api.semanticscholar.org/CorpusID:264147090.

Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International conference on machine learning*, pp. 10347–10357. PMLR, 2021.

Jesse Vig. A multiscale visualization of attention in the transformer model. *arXiv preprint arXiv:1906.05714*, 2019.

Yifan Wang, Yifei Liu, Yingdong Shi, Changming Li, Anqi Pang, Sibei Yang, Jingyi Yu, and Kan Ren. Discovering influential neuron path in vision transformers. *arXiv preprint arXiv:2503.09046*, 2025.

Junyi Wu, Bin Duan, Weitai Kang, Hao Tang, and Yan Yan. Token transformation matters: Towards faithful post-hoc explanation for vision transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10926–10935, 2024a.

Junyi Wu, Weitai Kang, Hao Tang, Yuan Hong, and Yan Yan. On the faithfulness of vision transformer explanations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10936–10945, 2024b.

Mert Yuksekgonul, Maggie Wang, and James Zou. Post-hoc concept bottleneck models. *arXiv preprint arXiv:2205.15480*, 2022.

Xu Zheng, Farhad Shirani, Zhuomin Chen, Chaohao Lin, Wei Cheng, Wenbo Guo, and Dongsheng Luo. F-fidelity: A robust framework for faithfulness evaluation of explainable ai. *arXiv preprint arXiv:2410.02970*, 2024.

Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2921–2929, 2016.

A Appendix Overview

This appendix provides notation, additional technical details, derivations, and experimental analyses that complement the main paper. We organize the appendix to separate infrastructure components, theoretical formulations, and empirical validations of design choices.

B Notation and Symbols

Table 3 summarizes the core notation used throughout the paper. We only include symbols that are reused across multiple sections or are critical for understanding the proposed method.

C CONCEPT BANK CONSTRUCTION

This section describes the construction of the global concept bank $\mathcal{C} = \{c_1, \ldots, c_K\} \subset \mathbb{R}^d$ used throughout the paper (Table 3). The concept bank provides a unified semantic reference space shared across layers and inputs, and serves as the foundation for concept-aligned neuron grouping and concept-guided path discovery.

C.1 ATTENTION-BASED PATCH SELECTION

Given a dataset $\mathcal{D} = \{(x_n, y_n)\}_{n=1}^N$, we identify task-relevant visual regions in each image using class-to-patch attention. For an input image x, we extract attention probability matrices $A^{(l,h)}(x)$ at a fixed transformer block $l_a \in [L]$ across heads h, and aggregate them to obtain class-to-patch attention scores $\alpha_i^{(l_a)}(x)$ for patch index i.

We then select a fixed number of patches per image by retaining the top-k_p patch indices:

$$\mathcal{P}_{k_p}(x) = \mathrm{TopK}_i \left(\alpha_i^{(l_a)}(x)\right).$$

This attention-based filtering removes background or weakly relevant regions and focuses concept discovery on visual evidence most directly contributing to the model's prediction.

C.2 PATCH EMBEDDING EXTRACTION

For each selected patch index $i \in \mathcal{P}_{k_p}(x)$, we extract a d-dimensional patch embedding $e(x, i) \in \mathbb{R}^d$ from a fixed extractor block $l_e \in [L]$. The extractor block is chosen so that all patch embeddings lie in the same feature space as the concept prototypes.

Aggregating across all images in $\mathcal{D}$ yields a patch embedding pool:

$$\mathcal{E} = \left\{ e(x_n, i) \;\middle|\; n = 1, \ldots, N,\ i \in \mathcal{P}_{k_p}(x_n) \right\}.$$

This pool serves as the input for unsupervised concept discovery.

C.3 UNSUPERVISED CONCEPT DISCOVERY

We construct the concept bank by clustering the embedding pool $\mathcal{E}$ into K clusters. In practice, we apply minibatch k-means to optimize

$$\min_{\{c_k\}_{k=1}^K} \sum_{e \in \mathcal{E}} \min_{k \in [K]} \|e - c_k\|_2^2,$$

where $c_k \in \mathbb{R}^d$ denotes the prototype of the k-th cluster. When cosine similarity is preferred, patch embeddings are ℓ_2-normalized prior to clustering, which is equivalent to spherical k-means.

Let $\{C_1, \ldots, C_K\}$ denote the resulting partition of $\mathcal{E}$. Each concept prototype is computed as the mean of its assigned embeddings:

$$c_k = \frac{1}{|C_k|} \sum_{e \in C_k} e.$$

The resulting concept bank is $\mathcal{C} = \{c_1, \ldots, c_K\} \subset \mathbb{R}^d$.

Properties of the concept bank. The constructed concept bank is *global* and shared across all transformer layers. It defines a unified embedding space that is used to embed both image patches and neuron groups, and is defined independently of any particular FFN layer or neuron, making $\mathcal{C}$ a suitable intermediary between input-level visual evidence and internal neuron-group representations.

D NEURON GROUP CONSTRUCTION AND CONCEPT ALIGNMENT

This section describes how FFN neurons are grouped within each transformer block and aligned to the global concept bank $\mathcal{C}$ (Appendix C). All notation follows Table 3. This pipeline is *post-hoc*: it operates on a pretrained ViT checkpoint and does not modify training.

Inputs and outputs. The procedure takes as input the concept bank $\mathcal{C}$, the pretrained model, and the same dataset $\mathcal{D} = \{(x_n, y_n)\}_{n=1}^{N}$ used for concept construction. The outputs include: (i) a within-layer partition $\mathcal{G}^{(l)}$ of FFN neurons for each block l, (ii) a concept-space embedding $h(G_m^{(l)}) \in \mathbb{R}^d$ for each group, and (iii) a concept assignment $\mathrm{concept}(G_m^{(l)}) \in [K]$.

D.1 Neuron Activation Signatures

To characterize neuron behavior across inputs, we construct a dataset-level *activation signature* for each FFN neuron.

Fix a transformer block $l \in [L]$. For an input image x, recall that $\mathbf{w}^{(l)}(x) \in \mathbb{R}^D$ denotes the post-activation FFN vector of the `[CLS]` token at block l (Table 3), and let $w_j^{(l)}(x)$ denote its j-th coordinate for neuron $j \in [D]$.

To incorporate patch-level evidence, we also consider neuron activations on patch tokens. Let $w_j^{(l)}(x, i)$ denote the post-activation value of neuron j at patch token i. Following Appendix C, let $\mathcal{P}_{k_p}(x)$ denote the set of attention-selected patch indices for x.

We aggregate neuron activations over the selected patches to obtain a single scalar response per image, e.g.,

$$r_{l,j}(x) = \mathrm{Agg}_{i \in \mathcal{P}_{k_p}(x)}\, w_j^{(l)}(x, i),$$

where Agg is mean aggregation by default (or max aggregation). Stacking these responses over the dataset yields the activation signature $s_{l,j} \in \mathbb{R}^N$ for neuron j at block l:

$$s_{l,j} = \big(r_{l,j}(x_1), \ldots, r_{l,j}(x_N)\big).$$

These activation signatures serve as the basis for grouping neurons within each block.

D.2 Within-Layer Neuron Grouping

Within each transformer block l, we cluster neurons according to the similarity of their activation signatures $\{s_{l,j}\}_{j=1}^{D}$. This produces a partition of the FFN neurons:

$$\mathcal{G}^{(l)} = \{G_1^{(l)}, \ldots, G_{M_l}^{(l)}\}, \qquad \bigcup_{m=1}^{M_l} G_m^{(l)} = [D], \qquad G_m^{(l)} \cap G_{m'}^{(l)} = \emptyset \ (m \neq m').$$

Neurons within the same group exhibit similar activation patterns across the dataset.

Clustering algorithm. We use minibatch k-means on activation signatures for scalability. The number of groups per block M_l is treated as a hyperparameter and is typically fixed across layers.

Practical computation. To avoid repeated forward passes, FFN activations for all blocks are captured in a single forward pass per batch using model hooks. The grouping pipeline can be implemented in two passes over the dataset: first to compute activation signatures, and second to identify top-activating patches for each group.

D.3 Group Embedding in Concept Space

To align neuron groups with the concept bank, we embed each group into the same d-dimensional concept space as the patch embeddings and concept prototypes (Table 3).

For a group $G_m^{(l)}$, we identify a set of patches that strongly activate the group. Let $\mathcal{P}_G(G_m^{(l)})$ denote the selected set of (x, i) pairs (image and patch index) for this group, obtained by ranking patches using a group-level activation score (e.g., aggregating $w_j^{(l)}(x, i)$ over $j \in G_m^{(l)}$) and retaining the top ones.

Using the corresponding patch embeddings $e(x, i) \in \mathbb{R}^d$ from Appendix C, we define the group embedding:

$$h(G_m^{(l)}) = \frac{1}{|\mathcal{P}_G(G_m^{(l)})|} \sum_{(x,i) \in \mathcal{P}_G(G_m^{(l)})} e(x, i) \in \mathbb{R}^d.$$

By construction, $h(G_m^{(l)})$ lies in the same embedding space as the concept prototypes $\{c_k\}_{k=1}^K$.

D.4 Concept Alignment

Each neuron group is aligned to the global concept bank using cosine similarity in the shared concept space. Define the alignment score

$$\mathrm{Score}(G_m^{(l)}, c_k) = \cos\Big(h(G_m^{(l)}),\, c_k\Big), \qquad c_k \in \mathcal{C}.$$

The top-1 concept assignment is then

$$\mathrm{concept}(G_m^{(l)}) = \arg\max_{k\in[K]} \mathrm{Score}(G_m^{(l)}, c_k).$$

For analysis, we also retain the top-k concepts and their similarity scores for each group.

Neuron-to-concept propagation. All neurons inherit the concept label of their assigned group. This yields a concept-aligned partition of FFN neurons within each layer, which is used by the main method for coarse group selection and for constructing interpretable concept paths.

Remarks. This procedure decouples *group formation* (activation similarity within a layer) from *semantic labeling* (alignment to the global concept bank), enabling data-driven discovery of functional groups while maintaining a consistent semantic vocabulary across layers.

E Group-Gated JAS (GG-JAS): Theory and Derivation

This appendix presents the theoretical formulation of the *Group-Gated Joint Attribution Score* (GG-JAS) used in the main text and clarifies its relationship to the joint attribution score (JAS) in NeuronPath (Wang et al., 2025). We follow Table 3 and the definitions in Section 3.

E.1 Group-Gated Reparameterization

Consider a transformer model with L blocks. For block l, let $\mathbf{w}^{(l)}(x) \in \mathbb{R}^D$ denote the *ungated* post-activation FFN hidden vector of the class token (Table 3). We partition FFN neurons at each block into M_l disjoint groups $\mathcal{G}^{(l)} = \{G_1^{(l)}, \ldots, G_{M_l}^{(l)}\}$.

Group gates. For each group $G_m^{(l)}$, we introduce a scalar gate $z_{l,m} \in [0,1]$ that jointly scales all coordinates in the group. For any neuron index $j \in G_m^{(l)}$, the gated post-activation is

$$w_j^{(l)}(x; z) = z_{l,m}\, w_j^{(l)}(x), \qquad \forall j \in G_m^{(l)}. \tag{19}$$

Although (19) is pointwise multiplicative, note that $w_j^{(l)}(x)$ itself depends on earlier-layer gates through the forward pass. All group gates are concatenated into $z \in [0,1]^{|\mathcal{I}|}$ indexed by $\mathcal{I} = \{(l,m) : l \in [L], m \in [M_l]\}$, yielding the gated output $F_{y^*}(x; z)$.

E.2 IG with Deletion Baselines in Gate Space

We use Integrated Gradients (IG) (Sundararajan et al., 2017) in the gate space to quantify the causal effect of *reactivating* selected groups.

Deletion baselines. For any set $S \subseteq \mathcal{I}$, the deletion baseline $z^{\mathrm{del}}(S)$ deactivates exactly the coordinates in S while keeping all other coordinates active:

$$z_{l,m}^{\mathrm{del}}(S) = \begin{cases} 0, & (l,m) \in S, \\ 1, & (l,m) \notin S. \end{cases} \tag{20}$$

Let $\mathbf{1}$ denote the all-active gate. We consider the straight-line path

$$z(\alpha) = z^{\mathrm{del}}(S) + \alpha(\mathbf{1} - z^{\mathrm{del}}(S)), \qquad \alpha \in [0,1]. \tag{21}$$

IG attribution. Assuming $F_{y^*}(x; z)$ is (a.e.) differentiable along (21), IG from $z^{\mathrm{del}}(S)$ to $\mathbf{1}$ is

$$\mathrm{IG}\big(z^{\mathrm{del}}(S)\to\mathbf{1}\big) = (\mathbf{1} - z^{\mathrm{del}}(S)) \odot \int_0^1 \nabla_z F_{y^*}(x; z(\alpha))\, d\alpha, \tag{22}$$

with coordinate form

$$\mathrm{IG}_{l,m}\big(z^{\mathrm{del}}(S)\to\mathbf{1}\big) = (1 - z^{\mathrm{del}}_{l,m}(S)) \int_0^1 \frac{\partial F_{y^*}(x; z(\alpha))}{\partial z_{l,m}}\, d\alpha, \qquad (l,m) \in \mathcal{I}. \tag{23}$$

Completeness. IG satisfies completeness in the gate space:

$$\sum_{(l,m)\in\mathcal{I}} \mathrm{IG}_{l,m}\big(z^{\mathrm{del}}(S)\to\mathbf{1}\big) = F_{y^*}(x; \mathbf{1}) - F_{y^*}(x; z^{\mathrm{del}}(S)). \tag{24}$$

Moreover, since $(\mathbf{1} - z^{\mathrm{del}}(S))_{l,m} = 0$ for $(l,m) \notin S$, only coordinates in S can receive nonzero IG attribution under (23).

E.3 Definition of GG-JAS and Special Cases

Set-form GG-JAS. Define the attribution vector $\mathbf{a}(S) = \mathrm{IG}\big(z^{\mathrm{del}}(S)\to\mathbf{1}\big)$ and $a_{l,m}(S) = \mathrm{IG}_{l,m}\big(z^{\mathrm{del}}(S)\to\mathbf{1}\big)$. The GG-JAS score of a set $S \subseteq \mathcal{I}$ is

$$\mathrm{GGJAS}(S) = \sum_{(l,m)\in S} a_{l,m}(S) = \sum_{(l,m)\in S} \mathrm{IG}_{l,m}\big(z^{\mathrm{del}}(S)\to\mathbf{1}\big)\,. \tag{25}$$

By the support property noted above, (25) is also equal to the full IG sum over $\mathcal{I}$ under this baseline choice.

Path / prefix / local as special cases. A group path is $\gamma = (m_1, \ldots, m_L)$ (Table 3) with path set $S(\gamma) = \{(l, m_l)\}_{l=1}^{L}$, and prefix set $S(\gamma_{1:l}) = \{(t, m_t)\}_{t=1}^{l}$. We recover the variants used in the main text:

$$\begin{aligned} \mathrm{GGJAS}(\gamma) &= \mathrm{GGJAS}\big(S(\gamma)\big), \\ \mathrm{GGJAS}_{\mathrm{cond}}(\gamma_{1:l}) &= \mathrm{GGJAS}\big(S(\gamma_{1:l})\big), \\ \mathrm{GGJAS}_{\mathrm{local}}(l,m) &= \mathrm{GGJAS}\big(\{(l,m)\}\big). \end{aligned} \tag{26}$$

E.4 Discrete Approximation (Implementation)

In practice, we approximate the integral in IG using T Riemann steps $\alpha_t = \frac{t}{T}$. For any set $S \subseteq \mathcal{I}$, the discrete approximation of $a_{l,m}(S)$ in (23) is

$$\widetilde{a}_{l,m}(S) = (1 - z^{\mathrm{del}}_{l,m}(S)) \cdot \frac{1}{T} \sum_{t=1}^{T} \left.\frac{\partial F_{y^*}(x; z)}{\partial z_{l,m}}\right|_{z=z(\alpha_t)}, \tag{27}$$

where $z(\alpha_t)$ is defined in (21). Accordingly,

$$\widetilde{\mathrm{GGJAS}}(S) = \sum_{(l,m)\in S} \widetilde{a}_{l,m}(S). \tag{28}$$

This is the quantity computed in Stage I (Section 4.2).

E.5 Relation to NeuronPath JAS

NeuronPath (Wang et al., 2025) defines the *Joint Attribution Score* (JAS) for a selected cross-layer neuron set/path by integrating gradients along a scaling path that restores the selected neuron activations from *zero effect* to *full effect*, while leaving all non-selected units *active*. Below we express this procedure as Integrated Gradients (IG) under a *neuron-gate* reparameterization, and show that our GG-JAS is the corresponding group-wise generalization in the gate space of Section 3.3.

Neuron-gate reparameterization. Fix an input x and target class y^*. Let $\mathbf{w}^{(l)}(x) \in \mathbb{R}^D$ be the ungated post-activation FFN vector at block l (Table 3). Consider a selected neuron set $S \subseteq \{(l, j) : l \in [L],\ j \in [D]\}$ (e.g., one neuron per layer along a path). Introduce a *neuron gate* $g_{l,j} \in [0, 1]$ for $(l, j) \in S$ and define the gated activation

$$w_j^{(l)}(x; g) = g_{l,j}\, w_j^{(l)}(x), \qquad (l, j) \in S, \tag{29}$$

while keeping all non-selected coordinates active (equivalently, their gates are fixed to 1). Let $F_{y^*}(x; g)$ denote the resulting output under this neuron gating.

Deletion baseline and reactivation path. Define the deletion baseline in the neuron-gate space by turning off exactly the selected neurons:

$$g_{l,j}^{\mathrm{del}}(S) = \begin{cases} 0, & (l, j) \in S, \\ 1, & (l, j) \notin S, \end{cases} \qquad \mathbf{1}_{l,j} = 1, \tag{30}$$

and consider the straight-line reactivation path $g(\alpha) = g^{\mathrm{del}}(S) + \alpha(\mathbf{1} - g^{\mathrm{del}}(S))$.

JAS as IG under the deletion-baseline semantics. Applying IG in the neuron-gate space attributes the reactivation effect $F_{y^*}(x; \mathbf{1}) - F_{y^*}(x; g^{\mathrm{del}}(S))$ to the selected coordinates $(l, j) \in S$:

$$\mathrm{IG}_{l,j}\big(g^{\mathrm{del}}(S) \to \mathbf{1}\big) = (1 - g_{l,j}^{\mathrm{del}}(S)) \int_0^1 \frac{\partial F_{y^*}(x; g(\alpha))}{\partial g_{l,j}}\, d\alpha. \tag{31}$$

Moreover, by the chain rule applied to (29),

$$\frac{\partial F_{y^*}(x; g)}{\partial g_{l,j}} = w_j^{(l)}(x)\, \frac{\partial F_{y^*}(x; g)}{\partial w_j^{(l)}}, \qquad (l, j) \in S. \tag{32}$$

Substituting (32) into (31) yields the activation-weighted integrated-gradients form used by NeuronPath to score a selected neuron set ("zero effect" $\to$ "full effect").

From neuron-wise JAS to group-wise GG-JAS. Our GG-JAS replaces neuron-wise gates $g_{l,j}$ with group-wise gates $z_{l,m}$ (Table 3) that jointly scale all neurons in a group $G_m^{(l)}$. Under the same deletion-baseline semantics (turning off exactly the selected coordinates and reactivating them along a straight line), GG-JAS computes IG in the group-gate space. When each group is a singleton neuron (i.e., $M_l = D$ and $G_m^{(l)} = \{m\}$), the group-gate space coincides with the neuron-gate space and GG-JAS reduces to NeuronPath's JAS as a special case.

F Information-Theoretic View of Layer-wise Information Bottlenecks

This section provides an information-theoretic interpretation of the sharp single-layer causal spike observed in Figures 3a and 3b. We use the notation in Appendix B.

Representations as a Markov chain across depth. Let $\mathbf{w}^{(l)}(x) \in \mathbb{R}^D$ denote the post-activation FFN hidden vector of the class token at block l. Let X denote the (random) input and Y the (random) label. A standard abstraction of forward computation is the depth-wise Markov chain

$$X \to \mathbf{w}^{(1)}(X) \to \cdots \to \mathbf{w}^{(L)}(X) \to \hat{Y},$$

where the prediction is given by the logit argmax

$$\hat{Y} \;=\; \arg\max_c F_c(X).$$

By the data processing inequality, for any l,

$$I(Y; \mathbf{w}^{(1)}(X)) \geq \cdots \geq I(Y; \mathbf{w}^{(l)}(X)) \geq \cdots \geq I(Y; \mathbf{w}^{(L)}(X)),$$

i.e., label-relevant information is progressively transformed and (in general) cannot increase through deterministic processing.

Decision margin as a label-relevant statistic. For an input (x, y^*), define the (logit) margin

$$\mathrm{Margin}(x) = F_{y^*}(x) - \max_{c \neq y^*} F_c(x).$$

This scalar is a sufficient statistic for whether the prediction is correct under a fixed argmax rule, and it tightly tracks the classifier's confidence. Changes in $\mathrm{Margin}(x)$ therefore provide a fine-grained probe of how internal perturbations affect the decision boundary, even when accuracy does not change.

Layer-local bottleneck via conditional information flow. A layer l acts as an *information bottleneck* for classification if (i) a small subset of coordinates of $\mathbf{w}^{(l)}(x)$ carries a disproportionate amount of label-relevant information, and (ii) downstream layers rely heavily on those coordinates to produce the final decision. One way to formalize this is through conditional mutual information under a small intervention budget. Let $U^{(l)}(x)$ denote a low-dimensional statistic of $\mathbf{w}^{(l)}(x)$, e.g., a single neuron value $w_i^{(l)}(x)$ or a group-level aggregate associated with a neuron group $G_m^{(l)}$. Let $\mathbf{w}^{(l)}_{\setminus U}(x)$ denote the remaining coordinates. Then layer l is bottleneck-like when, for some small $U^{(l)}$,

$$I\Big(Y; U^{(l)}(X) \mid \mathbf{w}^{(l)}_{\setminus U}(X)\Big) \quad \text{is large,}$$

meaning that $U^{(l)}(X)$ contains label information that is not redundant with the rest of the layer representation.

Why a large $\mathrm{RAD}_{\text{margin}}(l)$ / $\mathrm{EAD}_{\text{margin}}(l)$ indicates a bottleneck. Our layer-wise intervention modifies only $U^{(l)}$ at depth l (removal sets it to zero; enhancement scales it), while leaving all other layers untouched. Equivalently, when $U^{(l)}$ corresponds to a neuron group $G_m^{(l)}$, this can be viewed as a layer-local gate intervention on the associated gate coordinate $z_{l,m}$ (Table 3). Denote the resulting perturbed model by $\widetilde{F}$ and the perturbed margin by $\widetilde{\mathrm{Margin}}(x)$. The margin-based deviation

$$\Delta\mathrm{Margin}(x) = \widetilde{\mathrm{Margin}}(x) - \mathrm{Margin}(x)$$

quantifies how sensitive the final decision statistic is to the information carried by $U^{(l)}(x)$. If $\mathbb{E}[\Delta\mathrm{Margin}(X)]$ has large magnitude for a single-layer single-unit intervention, then the downstream computation must be amplifying or routing label-relevant information through that unit; otherwise, the perturbation would be absorbed by redundancy in $\mathbf{w}^{(l)}(X)$ or by later layers. Thus, a sharp spike in $\mathrm{RAD}_{\text{margin}}(l)$ and $\mathrm{EAD}_{\text{margin}}(l)$ at a particular depth indicates that a low-dimensional $U^{(l)}$ has unusually high *effective influence* on the decision margin, consistent with a depth-localized bottleneck.

Connection to redundancy vs. synergy across layers. A complementary view is to ask whether the causal effect of a full path is *additive* across layers or *synergistic*. If influence is distributed, then single-layer interventions yield small effects, but jointly intervening across multiple layers yields a large path-level effect; this corresponds to label information being encoded in a redundant or synergistic manner across layers. In contrast, a bottleneck layer concentrates non-redundant label information into a small subspace, so single-layer interventions at that depth are already strong. Empirically, the pronounced layer-22 spike (Figures 3b and 3a) suggests that ViT-L/32 consolidates decision-critical evidence into a small set of FFN units around this depth, after which later layers predominantly refine or read out this compressed signal.

Group-level bottlenecks and concept alignment. For a neuron group $G_m^{(l)}$ with group embedding $h(G_m^{(l)})$ and assigned concept $\mathrm{concept}(G_m^{(l)})$, the above interpretation applies with $U^{(l)}(x)$ defined as the group activation (or an aggregate statistic of neurons within the group). The observation that neuron-level and group-level spikes are comparable in the middle-to-late layers supports the hypothesis that our concept-guided grouping preserves the low-dimensional label-relevant subspace responsible for the bottleneck effect, while providing a more semantically interpretable handle on it.

Takeaway. From an information-theoretic perspective, the layer-wise causal spike is evidence that the model's label-relevant information flow becomes locally concentrated at a specific depth. At this bottleneck layer, a small intervention budget (one neuron or one concept group) suffices to substantially shift the decision margin, indicating a non-redundant, decision-critical subspace in $\mathbf{w}^{(l)}(X)$. This provides a mechanistic justification for using layer-wise margin-based deviation to diagnose information bottlenecks in deep transformers.

G Evaluation Metrics

We follow the accuracy deviation metric introduced in NeuronPath (Wang et al., 2025) and adopt the same intervention protocol to evaluate the causal effect of discovered neuron paths. The only difference lies in the granularity of the intervened neuron set, which in our experiments may correspond to a single neuron, a neuron path across layers, or a group-level (concept-aligned) path.

G.1 Accuracy-Based Metrics

G.1.1 Accuracy under intervention

Let $F : \mathbb{R}^d \to \mathbb{R}^C$ denote a pretrained classifier, and let $\mathcal{P}_x$ denote a set of selected neurons (or neuron groups) for input x. Given a dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{|\mathcal{D}|}$, the original classification accuracy is defined as

$$\mathrm{Acc} = \frac{1}{|\mathcal{D}|} \sum_{i=1}^{|\mathcal{D}|} \mathbb{I}(\hat{y}_i = y_i),$$

where $\hat{y}_i = \arg\max F(x_i)$ denotes the predicted class.

Following NeuronPath, we intervene on the neurons in $\mathcal{P}_{x_i}$ by either *removing* or *enhancing* their activations. Specifically, path removal is implemented by zeroing out the activation values of the selected neurons (or neuron groups), while path enhancement is implemented by doubling their activation values. After intervention, the model output becomes $\widetilde{F}(x_i)$, yielding a new prediction $\widetilde{y}_i = \arg\max \widetilde{F}(x_i)$. The resulting classification accuracy is

$$\widetilde{\mathrm{Acc}} = \frac{1}{|\mathcal{D}|} \sum_{i=1}^{|\mathcal{D}|} \mathbb{I}(\widetilde{y}_i = y_i).$$

G.1.2 Accuracy deviation

The accuracy deviation induced by neuron path intervention is defined as

$$\Delta\mathrm{Acc} = \widetilde{\mathrm{Acc}} - \mathrm{Acc}.$$

This quantity measures the signed change in predictive performance caused by intervening on the selected neurons.

G.1.3 Removal- and enhancement-based variants

When the intervention corresponds to *path removal*, we define the Removal-based Accuracy Deviation (RAD) as

$$\mathrm{RAD} = -\Delta\mathrm{Acc}.$$

A larger RAD indicates that deactivating the selected neurons leads to a more severe degradation in performance, implying stronger causal importance of the discovered path.

When the intervention corresponds to *path enhancement*, we define the Enhancement-based Accuracy Deviation (EAD) as

$$\mathrm{EAD} = \Delta\mathrm{Acc}.$$

A larger EAD indicates that selectively amplifying the selected neurons results in a greater improvement in model performance.

G.2 Logit-Based Deviation Metrics

Accuracy-based deviation is inherently discrete and may be insensitive to substantial changes in model confidence that do not flip the final prediction. To obtain a more fine-grained measure of causal effect, we additionally consider logit-based deviation metrics.

G.2.1 Target-logit deviation

For an input x_i with ground-truth label y_i, let $F_{y_i}(x_i)$ denote the logit corresponding to the target class. After intervention, the corresponding logit becomes $\widetilde{F}_{y_i}(x_i)$. We define the target-logit deviation induced by intervention as

$$\Delta\mathrm{Logit}_i = \widetilde{F}_{y_i}(x_i) - F_{y_i}(x_i).$$

Logit-based removal and enhancement metrics are defined by averaging over the dataset:

$$\mathrm{RAD}_{\text{logit}} = -\frac{1}{|\mathcal{D}|}\sum_{i=1}^{|\mathcal{D}|}\Delta\mathrm{Logit}_i, \qquad \mathrm{EAD}_{\text{logit}} = \frac{1}{|\mathcal{D}|}\sum_{i=1}^{|\mathcal{D}|}\Delta\mathrm{Logit}_i.$$

G.2.2 Logit-margin deviation

In addition to the target-class logit, we also consider deviation in the logit margin, which captures how intervention shifts the decision boundary relative to competing classes. For each input x_i, we define the logit margin as

$$\mathrm{Margin}(x_i) = F_{y_i}(x_i) - \max_{c \neq y_i} F_c(x_i),$$

and analogously define $\widetilde{\mathrm{Margin}}(x_i)$ using $\widetilde{F}(x_i)$.

The logit-margin deviation is then

$$\Delta\mathrm{Margin}_i = \widetilde{\mathrm{Margin}}(x_i) - \mathrm{Margin}(x_i).$$

Margin-based removal and enhancement metrics are defined as

$$\mathrm{RAD}_{\text{margin}} = -\frac{1}{|\mathcal{D}|}\sum_{i=1}^{|\mathcal{D}|}\Delta\mathrm{Margin}_i, \qquad \mathrm{EAD}_{\text{margin}} = \frac{1}{|\mathcal{D}|}\sum_{i=1}^{|\mathcal{D}|}\Delta\mathrm{Margin}_i.$$

H Implementation and Reproducibility Details

Hardware and software. All experiments are conducted on a single NVIDIA RTX PRO 6000 GPU. We implement all methods in PyTorch with CUDA enabled. Unless otherwise specified, all experiments are run with fixed random seeds for dataset sampling and clustering, and results are stable across repeated runs.

Model and data protocol. We evaluate three pretrained Vision Transformer backbones on the ImageNet validation set: ViT-B/16, ViT-B/32, and ViT-L/32 (Dosovitskiy, 2020). All models are pretrained on ImageNet-1k (Deng et al., 2009) and evaluated without any fine-tuning. Inputs are resized and center-cropped to 224×224, and normalized using the standard ImageNet mean and standard deviation. All analyses are performed post hoc on frozen model checkpoints.

Attention-based patch selection and embedding extraction. For each input image, we compute class-to-patch attention scores by aggregating attention weights across all heads (mean aggregation) from a fixed attention extractor block. Unless otherwise specified, both the attention extractor block and the patch embedding extractor block are chosen as the *last* transformer block. We select the top-k_p patches per image based on attention scores, excluding the class token, with $k_p = 8$. The selected patches are used consistently for patch-pool construction, concept discovery, and neuron-group embedding.

Concept bank and neuron grouping. The global concept bank is constructed as described in Appendix C. Within each transformer block l, FFN neurons are clustered based on activation signatures (Appendix D). We fix the number of neuron groups per block to match the size of the global concept bank, i.e.,

$$M_l = K \qquad \text{for all layers } l,$$

where K is the number of concepts. This enforces a consistent grouping granularity across layers and aligns group capacity with the global concept vocabulary.

Injection site and intervention rules. All causal interventions and attribution gates are applied at the FFN activation level. Specifically, interventions operate on the post-activation FFN hidden values (after the nonlinearity and before the second linear projection). Unless otherwise stated, interventions are applied to the class-token FFN activations at each transformer block.

For neuron-, group-, or path-level *removal*, the selected activations are set to zero. For *enhancement*, the selected activations are multiplied by a factor of 2. No other activations are modified.

Attribution configuration. Integrated Gradients (IG) is used to compute all attribution-based scores. Following NeuronPath (Wang et al., 2025), we approximate IG using $T = 20$ discrete steps with interpolation coefficients $\alpha_t = t/T$. For group-gated attribution, the baseline gate vector is $z^0 = \mathbf{0}$ and the integration path follows a straight-line interpolation $z(\alpha) = \alpha z^\gamma$. All attribution and intervention evaluations are performed with batch size 1 to avoid batching effects on attribution cost.

Implementation details. FFN activations for all transformer blocks are captured using forward hooks. Interventions are applied on-the-fly during the forward pass by editing the corresponding activation tensors, ensuring that causal effects are evaluated under the same inference pipeline as attribution computation. To reduce redundant computation, activation collection and grouping are performed using cached forward passes when possible.

I Additional Experimental Results

This section provides additional analyses and ablation studies that further support the claims in the main paper. In particular, we include (i) diagnostics showing that attribution scores are not sufficient indicators of causal faithfulness, (ii) additional efficiency scaling results, (iii) a full comparison of coarse-to-fine variants to justify our default design choices, and (iv) hyperparameter sensitivity analyses. Unless otherwise specified, all experimental protocols (models, data preprocessing, intervention rules, and attribution configuration) follow Appendix H.

I.1 JAS is Not a Sufficient Indicator of Faithful Neuron Paths

In NeuronPath (Wang et al., 2025), the Joint Attribution Score (JAS) is designed to rank neuron paths by aggregating attribution signals across layers. While JAS provides a principled attribution-based measure of neuron importance, it is not explicitly optimized to reflect causal influence under intervention. In this subsection, we present two complementary pieces of evidence showing that JAS, although informative, is not a sufficient indicator of neuron path faithfulness.

Weak rank correlation with intervention-based metrics. We analyze the statistical relationship between the Joint Attribution Score (JAS) and intervention-based faithfulness metrics at the class level. For each image, we run NeuronPath (JAS-Greedy) to identify a neuron path and compute its JAS score, together with its causal effects measured under removal and enhancement interventions. Causal effects are quantified using the intervention metrics described in Appendix G (including logit- and margin-based variants in Appendix G.2).

Across the ImageNet-1k validation set, we observe only moderate rank-level correlation between JAS and intervention-based faithfulness metrics. Specifically, the Spearman correlation between JAS and removal-based effects is $\rho \approx 0.48$, while the correlation with enhancement-based effects is $\rho \approx 0.38$. Despite being statistically significant, these correlations indicate that JAS explains only a limited fraction of the ordering of causally influential classes.

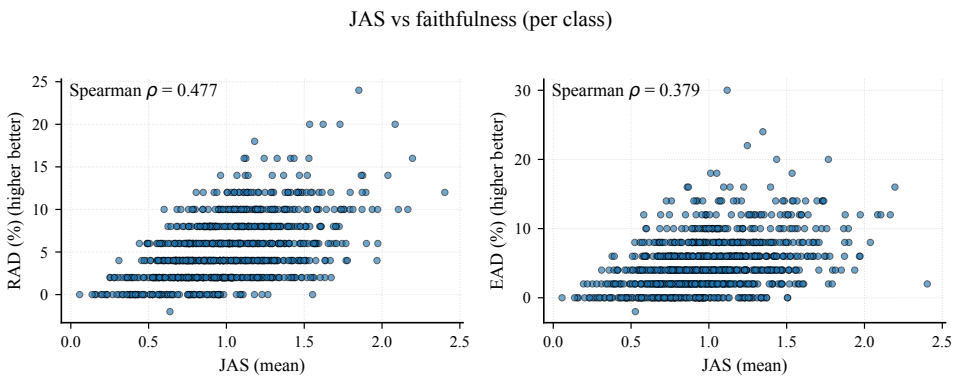


Figure 5: Relationship between Joint Attribution Score (JAS) and intervention-based causal effects at the class level. Each point corresponds to one ImageNet-1k class, with metrics aggregated over all validation images of that class. **Left:** Removal Accuracy Deviation (RAD; higher is better) versus JAS. **Right:** Enhancement Accuracy Deviation (EAD) versus JAS. Spearman rank correlations are reported in each panel. While JAS exhibits a moderate monotonic trend with intervention effects on average, substantial dispersion is observed, indicating that attribution-based ranking does not reliably reflect causal influence under intervention.

Overall, these results suggest that JAS captures only a partial aspect of neuron-path importance, and that ranking paths solely by attribution scores is insufficient to reliably prioritize those with the strongest behavioral impact under causal intervention.

Attribution-optimal paths are not intervention-optimal. While the above correlation analysis reveals a statistical mismatch between attribution scores and causal effects, it does not yet establish whether this discrepancy manifests at the level of individual neuron path selection. In particular, a natural question remains: *given the candidate paths explored during NeuronPath's greedy search, does maximizing JAS also select the path with the strongest causal influence?*

To answer this question, we design a controlled diagnostic experiment that directly compares attribution-optimal and intervention-optimal paths *within the same local candidate space*. We randomly sample **1,000** images from the ImageNet-1k validation set. For each image, we follow the original greedy search procedure of NeuronPath (Wang et al., 2025) to construct a JAS-maximizing neuron path. During this process, at each layer we additionally record the top-k candidate neurons evaluated by the JAS objective, excluding the greedy-selected one. In all experiments, we set $k = 5$.

This procedure yields a compact candidate pool that is entirely determined by NeuronPath's own greedy search and requires no additional forward or backward passes beyond those already used for JAS evaluation. From this pool, we consider two types of paths: (i) the JAS-greedy path selected by NeuronPath's original objective, and (ii) an *oracle path*, defined as the candidate path that maximizes intervention-based causal effect.

Since accuracy-based metrics are inherently discrete and sensitive to prediction flips, we primarily report logit-based and margin-based variants of RAD and EAD (Appendix G.2), which provide a smoother and more sensitive measure of causal influence. Algorithm 2 summarizes the oracle path construction procedure used in this experiment.

As summarized in Table 4, and following the oracle selection procedure in Algorithm 2, we find that across $1{,}000$ ImageNet-1k validation images, the JAS-greedy path is never the most causally effective one under margin-based intervention metrics. In every case, there exists at least one alternative path within the same candidate set that produces a strictly larger margin change under path removal. The average gap between the oracle path and the JAS-greedy path is 0.10, with a median gap of 0.08, indicating a consistent and non-negligible difference in causal influence.

We observe the same trend under enhancement interventions. On average, oracle paths yield larger logit- and margin-based gains than their JAS-greedy counterparts, despite being selected from the same local candidate space. Together, these results demonstrate a systematic objective mismatch:

Algorithm 2 Oracle Path Selection within JAS-Greedy Candidate Space

Input: model F, input (x, y), number of layers L, top-k
Output: JAS-greedy path π^g, oracle path $\pi^\star$
Step 1: JAS-greedy path and candidate pool
for $l = 1$ **to** L **do**
  Compute JAS scores for all neurons at layer l given prefix
  Select top-k neurons $\textsc{TopK}^{(l)}$ by JAS
  $\pi_l^g \leftarrow \textsc{TopK}^{(l)}[1]$
end for
Step 2: Single-layer swap candidates
Initialize candidate set $\mathcal{C} = \{\pi^g\}$
for $l = 1$ **to** L **do**
  for $r = 2$ **to** k **do**
    Construct path π by replacing π_l^g with $\textsc{TopK}^{(l)}[r]$
    Add π to $\mathcal{C}$
  end for
end for
Step 3: Oracle selection by intervention
for each $\pi \in \mathcal{C}$ **do**
  Compute removal-based causal effect $\mathrm{RAD}_{\text{margin}}(\pi)$
end for
$\pi^\star \leftarrow \arg\max_{\pi \in \mathcal{C}} \mathrm{RAD}_{\text{margin}}(\pi)$

even within the candidate set explored by NeuronPath's own greedy search, maximizing JAS does not guarantee the selection of the most causally faithful neuron path.

I.2 Efficiency Scaling with IG Budget

This subsection provides additional evidence on how computational efficiency and faithfulness scale with the Integrated Gradients (IG) step budget T, complementing the main-text analysis in Section 5.2.

Runtime scaling. We vary the IG step budget $T \in \{5, 10, 20, 40\}$ and measure the end-to-end runtime per image for NeuronPath and our coarse-to-fine variants on ViT-L/32 (concept bank size $= 1024$; batch size $= 1$). As shown in Table 5, **PF+argmax** consistently achieves the lowest runtime across all budgets. In particular, PF+argmax yields a $5.13\times$ speedup at $T = 5$ (0.95 s vs. 4.86 s) and maintains comparable gains at larger budgets (e.g., $5.48\times$ at $T = 40$), indicating favorable scaling behavior as attribution cost increases.

Faithfulness under efficient search. A natural concern is whether the observed efficiency gains come at the expense of explanation quality. To assess this, we examine intervention-based faithfulness using Remove Accuracy Drop (RAD) across all IG budgets. As reported in Table 5, PF+argmax consistently attains higher RAD scores than NeuronPath, indicating a stronger causal effect under removal intervention. For example, at $T = 5$, PF+argmax improves RAD from 5.5% to 8.2%, and this advantage remains stable as T increases. These results confirm that the coarse-to-fine approximation improves computational efficiency without degrading—and in practice strengthening—the faithfulness of the discovered neuron paths.

Full efficiency and faithfulness results. For completeness, Table 5 reports runtime, peak memory usage, and intervention metric (RAD) for all methods and IG budgets.

I.3 Full Comparison of Coarse-to-Fine Variants

We conduct a controlled comparison of coarse-to-fine design choices to understand which components are critical for discovering causally faithful neuron paths and to justify our default configuration (PF + argmax) used in the main paper. All experiments in this subsection follow Appendix H and are

performed on a pretrained ViT-L/32 model evaluated on ImageNet-1k (validation set), using a fixed concept bank size of 2048. Results are summarized in Table 6.

Effect of coarse-stage attribution. Comparing prefix-conditioned (PC) and prefix-free (PF) group attribution reveals a consistent advantage of the prefix-free strategy. Under both fine-stage choices, PF variants induce larger removal and enhancement effects than their PC counterparts. This suggests that conditioning group attribution on previously selected prefixes is not necessary, and may even dilute the attribution signal at the group level. Marginal, layer-wise group attribution provides a stronger and more stable coarse signal for identifying influential paths.

Effect of fine-stage refinement. Within a fixed coarse strategy, replacing fine-grained JAS-based neuron selection with simple activation argmax consistently improves causal intervention effects. For example, under prefix-free coarse attribution, switching from JAS to argmax refinement improves RAD from -4.40% to -8.24%, and EAD from 4.33% to 8.39%. This indicates that once the correct neuron group is identified, neurons within the group are sufficiently correlated such that activation magnitude serves as an effective and computationally efficient proxy for fine selection.

Comparison with NeuronPath. While NeuronPath achieves the highest Joint Attribution Score (JAS) by construction, its intervention-based effects are consistently weaker than those of our best coarse-to-fine configuration. Notably, the prefix-free group attribution combined with activation argmax refinement outperforms NeuronPath by a large margin on both RAD and EAD, despite having a substantially lower JAS. This further reinforces the conclusion that attribution score magnitude alone is not a reliable indicator of causal faithfulness.

Summary. Overall, causal faithfulness is primarily determined at the coarse stage by accurately identifying influential neuron groups. Prefix-free group attribution provides the strongest coarse signal, and simple activation-based refinement is sufficient—and often preferable— for selecting representative neurons within each group.

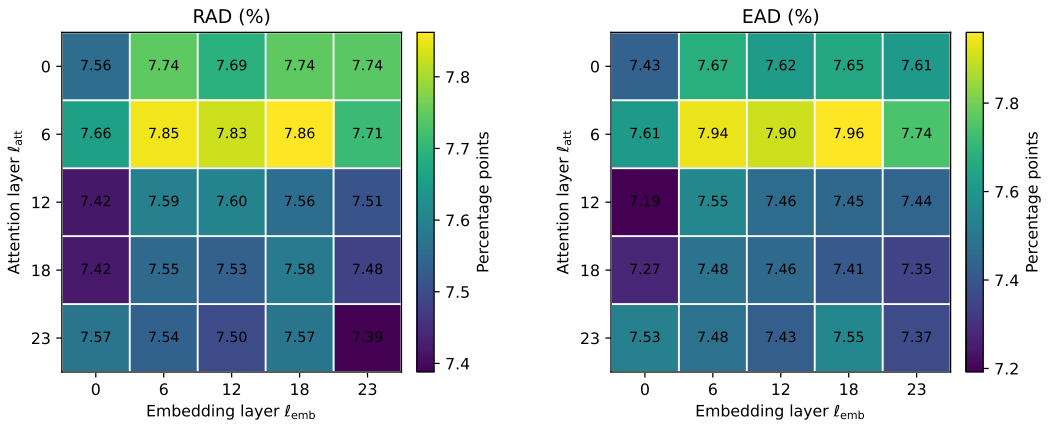
I.4 Hyperparameter Sensitivity

This subsection presents targeted ablations that examine the robustness of our concept-guided neuron-path discovery pipeline with respect to *hyperparameters involved in concept bank construction*. Specifically, we study sensitivity to: (i) the transformer layers used for attention-based patch selection and embedding extraction, (ii) the concept bank size K, and (iii) the neuron grouping granularity M. All experiments follow the same intervention protocol and faithfulness metrics (RAD/EAD and their continuous variants) as described in Appendix G. Unless otherwise specified, we keep all other components of the pipeline fixed, including the coarse-to-fine discovery procedure and intervention settings, and vary only the factor under study.

Ablation A: Sensitivity to Concept-Construction Layers. We first analyze the sensitivity of concept-guided neuron-path discovery to the choice of transformer layers used for concept construction. All experiments are conducted on **ViT-L/32**. We vary the attention extraction layer ℓ_{att} (used for patch selection) and the embedding extraction layer ℓ_{emb} (used for concept clustering) over a fixed candidate set $S = \{0, 6, 12, 18, 23\}$. This results in $|S|^2 = 25$ layer-pair configurations, including both *tied* settings ($\ell_{\text{att}} = \ell_{\text{emb}}$) and *untied* settings where the two layers differ. Throughout this ablation, we fix the concept bank size to $K = 512$ and the number of neuron groups per layer to $M = 512$, and keep all other components of the discovery and intervention pipeline unchanged.

Across all layer-pair configurations, both removal- and enhancement-based accuracy deviations exhibit smooth and consistent patterns. As summarized in Table 7 and visualized in Fig. 6, no single attention or embedding layer dominates the intervention effect. Both tied ($\ell_{\text{att}} = \ell_{\text{emb}}$) and untied configurations yield comparable causal responses. Overall, these results indicate that the proposed pipeline is stable with respect to the choice of concept-construction layers.

Ablation B: Sensitivity to Concept Bank Size. Next, we study the effect of the concept bank size K on causal faithfulness. By varying the number of concepts while keeping the discovery procedure and neuron grouping strategy fixed, this ablation probes the trade-off between semantic granularity



(a) RAD (%) (b) EAD (%)

Figure 6: Sensitivity of intervention-based faithfulness to the choice of concept-construction layers on ViT-L/32. Heatmaps show (left) removal-based accuracy deviation (RAD) and (right) enhancement-based accuracy deviation (EAD) as functions of the attention extraction layer ℓ_{att} and embedding extraction layer ℓ_{emb}. Each cell corresponds to one $(\ell_{\text{att}}, \ell_{\text{emb}}) \in S \times S$ configuration.

and stability. In particular, it tests whether faithful neuron paths require a finely granular concept vocabulary, or whether performance remains stable over a broad range of concept resolutions.

As reported in Table 8, accuracy-based removal- and enhancement-based deviations (RAD and EAD) increase smoothly with the concept bank size K. Even relatively small concept banks already yield substantial causal effects, while larger K provides a gradual and consistent improvement, reflecting finer semantic granularity. Importantly, no sharp degradation is observed at either lower or higher K, indicating that the discovery pipeline is stable across a wide range of concept resolutions and does not rely on a finely tuned concept vocabulary.

Ablation C: Sensitivity to Neuron Group Granularity. Finally, we analyze the impact of neuron grouping granularity by varying the number of neuron groups per layer M while holding the concept bank fixed. This ablation isolates the effect of the coarse-level search space size, independently of concept construction. It evaluates how sensitive the coarse-to-fine discovery process is to overly coarse or overly fine groupings, and whether a broad intermediate range of M yields stable causal faithfulness.

Table 9 reports accuracy-based RAD and EAD under varying grouping granularity M with a fixed concept bank size ($K = 512$). Across the examined range of M, both removal- and enhancement-based effects remain consistently strong, without sharp degradation at either coarse or fine grouping levels. This suggests that the coarse-to-fine discovery process is robust to the size of the group-level search space, and does not depend on a narrowly tuned neuron grouping granularity.

J Concept Labeling via Multimodal LLM Prompting

This appendix describes a post-hoc procedure for assigning human-readable semantic labels to the unlabeled concepts in the concept bank (Appendix C). Concept labeling is used solely for interpretability and visualization.

J.1 Overview

Given a learned concept c_k, our goal is to produce a concise semantic description that captures the dominant visual pattern associated with the concept. We formulate this as a multimodal prompting task for a large vision–language model (VLM), which receives curated visual evidence and returns a structured label prediction.

System Prompt. You are an expert visual concept labeling assistant. You will see a grid of top-activating image patches for one concept from a vision model, plus several full images with the activating patch highlighted. Use only visual evidence from the images. Choose exactly one type: `object_part`, `texture_pattern`, or `abstract`. Provide a short label (2–5 words), lower case, noun phrase, no punctuation. Do not use dataset class names or proper nouns. Provide a brief evidence summary describing recurring visual cues, including any global context. Provide a confidence score between 0 and 1. Return only JSON with keys: `type`, `label`, `evidence`, `confidence`.

User Prompt. Concept ID: `{concept_id}`
Image 1 is a grid of the top-activating patches for this concept (earlier patches activate more). Images 2+ are full images with the activating patch highlighted. Label the concept based only on what you see.

Figure 7: Multimodal prompt used for concept labeling. Visual inputs consist of a patch grid and context images.

J.2 Visual Evidence for Labeling

For each concept, we construct a compact multimodal input consisting of representative visual evidence:

- **Patch grid.** A single image showing a grid of the top-N image patches most strongly associated with the concept, ordered by activation strength.
- **Context images.** Several full images in which the activating patch is highlighted with a bounding box, providing global scene context.

All images are resized to a fixed resolution and jointly provided to the VLM as visual input. This evidence is intended to expose both local appearance and broader contextual cues of the concept.

J.3 Multimodal Prompting Protocol

Labeling is performed via a single multimodal query composed of: (i) a system prompt specifying labeling constraints and output format, (ii) a user prompt identifying the concept and describing the images, and (iii) the visual evidence images.

The model is instructed to return a JSON object with four fields: `type`, `label`, `evidence`, and `confidence`. The `type` field must take exactly one of the following values: `object_part`, `texture_pattern`, or `abstract`. All outputs are validated against a fixed JSON schema.

J.4 Multiple Sampling and Selection

To improve robustness, we optionally query the VLM multiple times per concept. Across sampling rounds, previously generated labels are included in an "avoid list" to reduce duplication. Candidate labels are normalized (lowercasing and whitespace normalization), and the final concept label is selected as the highest-confidence prediction among unique candidates.

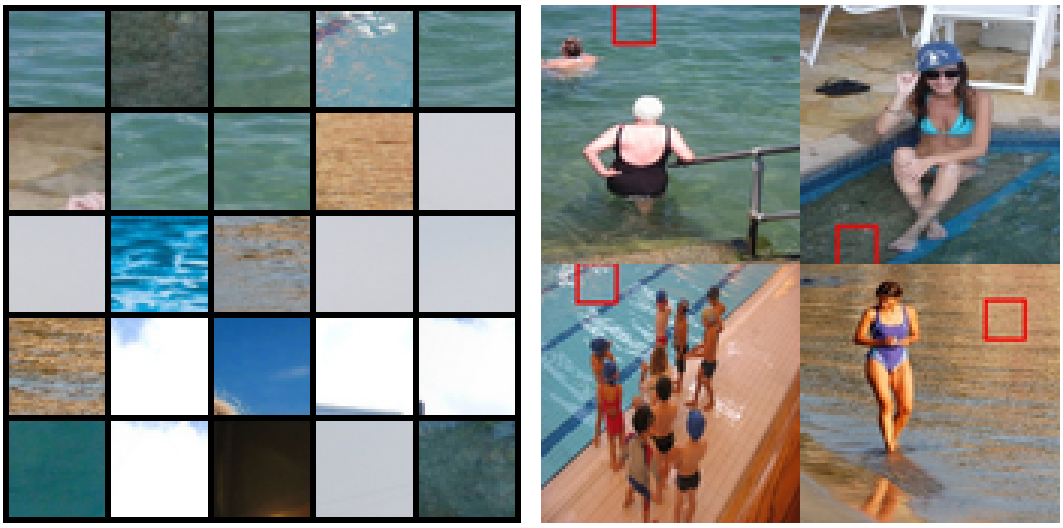
Concepts whose maximum confidence falls below a predefined threshold are flagged for optional manual review.

J.5 Prompts

J.6 Labeling Setting

All concept labels are produced via a vision–language model using the multimodal prompting protocol described above. For each concept, we construct a patch-grid image containing the top-$N = 25$ activating patches arranged in a 5×5 grid, and provide 4 additional full context images where the activating patch is highlighted with a red bounding box (line width $= 3$ pixels). All patch grids and context images are resized to 224×224 resolution before being passed to the model.

The model is queried with the fixed system prompt and user prompt template (Appendix J) and is constrained to produce a JSON object validated against a predefined schema. We sample 5 label



Patch grid (top-N) **Context images (highlighted patch)**

Concept ID: 14 **Type:** `texture_pattern` **Label:** `rippling water surface` **Confidence:** 0.86
Evidence: Top patches repeatedly show blue green wavy surfaces with specular highlights and small ripples. Highlighted regions in full images fall on pool or sea water areas rather than people or objects.

Figure 8: Concept visualization example corresponding to texture_pattern concept: '**rippling water surface**'

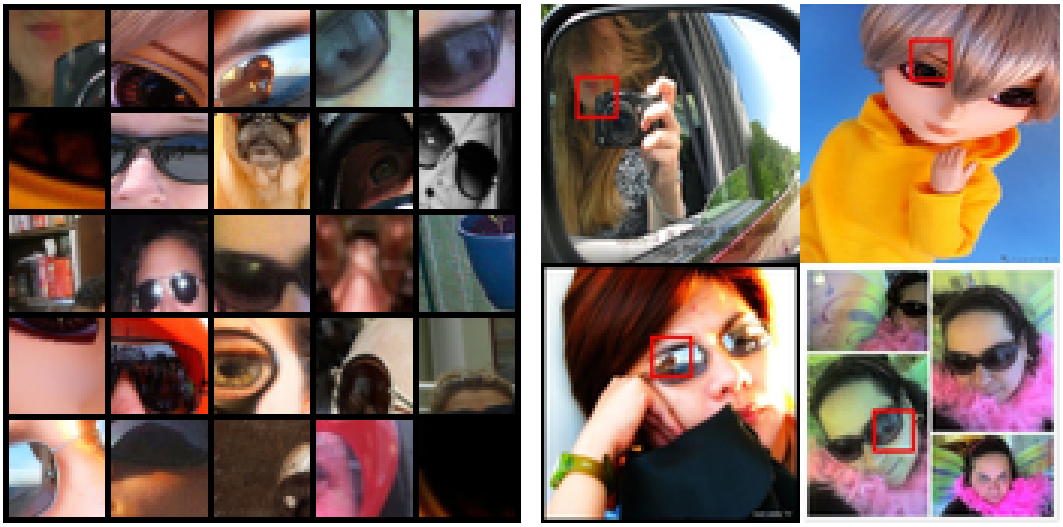
candidates per concept and select the final label as the highest-confidence prediction among unique candidates. Concepts whose maximum confidence is below 0.5 are flagged for optional manual review.

J.7 Examples

We present three representative concept labeling examples corresponding to the three concept types considered in this work: `texture_pattern`, `object_part`, and `abstract`. All examples are obtained using a ViT-L/32 vision backbone, concept bank size $K = 256$ and the same vision–language model (`gpt-5.2`) under the labeling setting described in Appendix J.6. For each concept, we visualize the patch-grid evidence (top-activating patches) together with four context images in which the activating region is highlighted. The annotated concept label is shown along with the model's structured output (type, evidence summary, and confidence).

K Additional Semantic Flow and Bottleneck Examples

This section provides additional qualitative examples of cross-layer semantic flow and bottleneck alignment for ViT-L/32, complementing the representative case shown in the main text. Figures 12–15 visualize the discovered concept paths across depth together with the corresponding layer-wise causal profiles measured by $\mathrm{RAD}_{\mathrm{margin}}(l)$ for several ImageNet classes (class 5 *electric ray*, class 147, class 274, and class 287). Across these examples, we consistently observe a coherent semantic progression from lower-level visual patterns to higher-level class-discriminative concepts, while causal influence is concentrated at a small number of depth-localized bottleneck layers. These additional cases demonstrate that the alignment between semantic structure and causal control points is a recurring phenomenon rather than an artifact of a single selected instance.

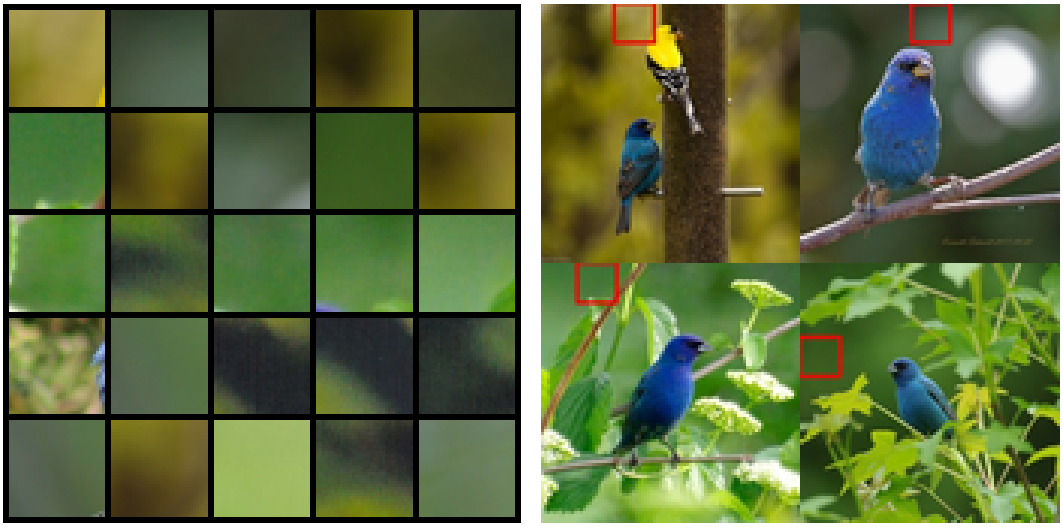


Patch grid (top-N) **Context images (highlighted patch)**

Concept ID: 4 **Type:** `object_part` **Label:** `eyeglass lens reflection` **Confidence:** 0.74

Evidence: Top patches repeatedly show close-ups of eyeglass lenses and frames with shiny glare, including mirrored reflections of a camera/scene; highlighted regions in full images coincide with sunglasses/eyewear areas and reflective lens surfaces.

Figure 9: Concept visualization example corresponding to Object-part concept: '**eyeglass lens reflection**'



Patch grid (top-N) **Context images (highlighted patch)**

Concept ID: 32 **Type:** `abstract` **Label:** `green foliage bokeh` **Confidence:** 0.78

Evidence: Top patches are mostly soft green and yellow blurred areas with smooth gradients and occasional circular highlights. In the full images, the highlighted regions fall on out-of-focus leafy backgrounds rather than on the birds or sharp objects, indicating a background blur/foliage context.

Figure 10: Concept visualization example corresponding to abstract concept: '**green foliage bokeh**'

Category	Symbol	Meaning
Data and model scale		
Data	$\mathcal{D} = \{(x_n, y_n)\}_{n=1}^{N}$	Dataset for concept/group construction.
Model	L	# transformer blocks.
Group count	M_l	# groups at block l.
Dim	d	ViT hidden size (feature dim).
Dim	D	FFN inner dim (# neurons).
FFN post-activation features and targets		
Act (ungated)	$\mathbf{w}^{(l)}(x) \in \mathbb{R}^D$	Post-activation FFN vector of `[CLS]` at block l.
Act (gated)	$\mathbf{w}^{(l)}(x; z) \in \mathbb{R}^D$	Gated version of $\mathbf{w}^{(l)}(x)$ under z.
Target	$F_{y^*}(x) \in \mathbb{R}$	Target score/logit for class y^*.
Target (gated)	$F_{y^*}(x; z)$	Target output under gating z.
Concept bank		
Concept	$\mathcal{C} = \{c_1, \ldots, c_K\} \subset \mathbb{R}^d$	Concept prototypes in $\mathbb{R}^d$.
Hyperparam	K	# concepts.
Neuron groups		
Groups	$\mathcal{G}^{(l)} = \{G_1^{(l)}, \ldots, G_{M_l}^{(l)}\}$	Partition of neurons at block l.
Group	$G_m^{(l)} \subseteq [D]$	m-th group at block l.
Embed	$h(G_m^{(l)}) \in \mathbb{R}^d$	Group embedding in concept space.
Assign	$\mathrm{concept}(G_m^{(l)}) \in [K]$	Assigned concept index.
Coarse-to-fine paths		
Group path	$\gamma = (m_1, \ldots, m_L)$	Selected group per block.
Prefix	$\gamma_{1:l} = (m_1, \ldots, m_l)$	Length-l prefix of γ.
Concept path	$\kappa = (k_1, \ldots, k_L)$	Concept labels along γ.
Neuron path	$\pi = (i_1, \ldots, i_L)$	Refined neuron path.
Candidate set	$\Omega_l = G_{m_l}^{(l)}$	Neuron candidates at layer l.
Group gates and IG-based attribution		
Gate	$z_{l,m} \in [0, 1]$	Gate for group $G_m^{(l)}$.
Index set	$\mathcal{I} = \{(l, m) : l \in [L], m \in [M_l]\}$	Gate coordinate index set.
Gate vector	$z \in [0, 1]^{\lvert\mathcal{I}\rvert}$	All gates concatenated.
All-active gate	$\mathbf{1}$	All-on gate: $(\mathbf{1})_{l,m} = 1$.
Set	$S \subseteq \mathcal{I}$	Selected gate coordinates.
Deletion baseline	$z^{\mathrm{del}}(S)$	Gates off on S, on elsewhere.
Prefix set	$S(\gamma_{1:l}) = \{(t, m_t)\}_{t=1}^{l}$	Coordinates on prefix $\gamma_{1:l}$.
Path set	$S(\gamma) = S(\gamma_{1:L})$	Coordinates on full path γ.
IG	$\mathrm{IG}(z^0 \to \tilde{z})$	Integrated Gradients in gate space.
Reactivation attrib.	$\mathbf{a}(S)$	$\mathbf{a}(S) = \mathrm{IG}(z^{\mathrm{del}}(S) \to \mathbf{1})$.
Reactivation attrib.	$a_{l,m}(S)$	$a_{l,m}(S) = \mathrm{IG}_{l,m}(z^{\mathrm{del}}(S) \to \mathbf{1})$.
IG steps	T	# steps for discrete IG.
GG-JAS scoring variants		
Cond. score	$\mathrm{GGJAS}_{\mathrm{cond}}(\gamma_{1:l})$	Prefix-conditioned GG-JAS.
Local score	$\mathrm{GGJAS}_{\mathrm{local}}(l, m)$	Layer-wise GG-JAS for $G_m^{(l)}$.
Path score	$\mathrm{GGJAS}(\gamma)$	$\mathrm{GGJAS}(S(\gamma))$.
Discrete	$\widetilde{\mathrm{GGJAS}}(\gamma)$	Discrete $\mathrm{GGJAS}(\gamma)$ with T steps.

Table 3: Core notation used throughout the paper.

Table 4: Comparison between JAS-greedy paths and oracle paths selected from the same candidate set explored during greedy search. Oracle paths are chosen to maximize margin-based causal effects under intervention. Across $1{,}000$ ImageNet validation images, oracle paths consistently induce stronger removal and enhancement effects than JAS-greedy paths, demonstrating that maximizing attribution score does not guarantee maximal causal faithfulness.

Metric	**JAS-Greedy**	**Oracle**
Removal-based intervention		
RAD (logit)	0.437	**0.474**
RAD (margin)	0.673	**0.777**
Enhancement-based intervention		
EAD (logit)	0.402	**0.419**
EAD (margin)	0.716	**0.768**
Oracle advantage (margin-based)		
Mean gap	–	**0.104**
Median gap	–	**0.077**
Win rate (%)	–	**100**

Table 5: **Efficiency and faithfulness (RAD)** results on ViT-L/32 (concept bank size $= 1024$; batch size $= 1$; mean over 1000 images). Peak memory is reported in GiB (mean). Speedup is relative to NeuronPath at the same IG budget. **RAD is reported in % (higher is better)**.

IG steps T	Method	Runtime↓	Mem.↓	Speedup↑	RAD↑
5	NeuronPath	4.858±0.453	4.20	1.00	5.5%
	Ours (PC + argmax)	1.791±0.155	3.00	2.71	7.2%
	Ours (PF + argmax)	0.947±0.098	3.00	5.13	**8.2**%
10	NeuronPath	5.465±0.147	5.03	1.00	5.7%
	Ours (PC + argmax)	2.135±0.076	3.47	2.56	7.6%
	Ours (PF + argmax)	1.087±0.074	3.47	5.03	**8.4**%
20	NeuronPath	8.054±0.050	6.63	1.00	5.7%
	Ours (PC + argmax)	3.221±0.053	4.42	2.50	7.4%
	Ours (PF + argmax)	1.523±0.057	4.42	5.29	**8.3**%
40	NeuronPath	13.578±0.055	9.80	1.00	5.8%
	Ours (PC + argmax)	5.599±0.055	6.34	2.42	7.3%
	Ours (PF + argmax)	2.478±0.058	6.34	5.48	**8.4**%

Table 6: Full comparison of coarse-to-fine variants under the **Concept-Guided Coarse-to-Fine Neuron-Path Discovery** framework on ViT-L/32. PC denotes *prefix-conditioned* Group-Gated JAS, while PF denotes *prefix-free (layer-wise)* Group-Gated JAS. All results are reported with a fixed concept bank size of 2048.

Method	Coarse Attribution	Fine Selection	RAD↑	EAD↑	JAS↑
NeuronPath (Wang et al., 2025)	Neuron-level JAS	JAS	5.28%	5.02%	**1.0086**
Ours (PC)	Prefix-conditioned Group-Gated JAS	JAS	3.76%	3.70%	0.3451
Ours (PC)	Prefix-conditioned Group-Gated JAS	Argmax (activation)	6.72%	6.72%	0.5875
Ours (PF)	Prefix-free (layer-wise) Group-Gated JAS	JAS	4.40%	4.33%	0.3681
Ours (PF)	**Prefix-free (layer-wise) Group-Gated JAS**	**Argmax (activation)**	**8.24%**	**8.39%**	0.6707

Table 7: Sensitivity to concept-construction layers ($\ell_{\text{att}}, \ell_{\text{emb}}$) on ViT-L/32 under the default PF+argmax setting. We sweep attention and embedding extraction layers over $S = \{0, 6, 12, 18, 23\}$ while fixing the concept bank size $K = 512$, the number of neuron groups per layer $M = 512$, and the IG step $T = 20$. RAD= $|a_{\text{remove}} - a_{\text{base}}|$, EAD= $|a_{\text{enhance}} - a_{\text{base}}|$. All accuracy-related values are reported in percentage (%).

ℓ_{att}	ℓ_{emb}	Base Acc. (%)	Remove Acc. (%)	Enhance Acc. (%)	RAD (%)	EAD (%)
0	0	61.15	53.59	68.59	7.56	7.43
0	6	61.15	53.41	68.83	7.74	7.67
0	12	61.15	53.46	68.77	7.69	7.62
0	18	61.15	53.41	68.81	7.74	7.65
0	23	61.15	53.41	68.76	7.74	7.61
6	0	61.15	53.49	68.76	7.66	7.61
6	6	61.15	53.30	69.09	7.85	7.94
6	12	61.15	53.33	69.05	7.83	7.90
6	18	61.15	53.29	69.11	7.86	7.96
6	23	61.15	53.44	68.89	7.71	7.74
12	0	61.15	53.73	68.34	7.42	7.19
12	6	61.15	53.56	68.70	7.59	7.55
12	12	61.15	53.55	68.62	7.60	7.46
12	18	61.15	53.59	68.60	7.56	7.45
12	23	61.15	53.64	68.59	7.51	7.44
18	0	61.15	53.73	68.43	7.42	7.27
18	6	61.15	53.60	68.64	7.55	7.48
18	12	61.15	53.62	68.61	7.53	7.46
18	18	61.15	53.57	68.56	7.58	7.41
18	23	61.15	53.67	68.50	7.48	7.35
23	0	61.15	53.58	68.68	7.57	7.53
23	6	61.15	53.62	68.63	7.54	7.48
23	12	61.15	53.66	68.58	7.50	7.43
23	18	61.15	53.58	68.70	7.57	7.55
23	23	61.15	53.76	68.52	7.39	7.37

Table 8: Sensitivity of accuracy-based causal faithfulness to the concept bank size K on ViT-L/32. We report removal-based and enhancement-based accuracy deviation (RAD / EAD, in %). All other components of the discovery and intervention pipeline are fixed.

Concept Bank Size K	RAD (%)	EAD (%)
128	6.60	6.25
256	7.07	6.90
512	7.39	7.37
1024	7.77	7.81
2048	8.43	8.56

Table 9: Sensitivity to grouping granularity M with a fixed concept bank size ($K = 512$) on ViT-L/32. We report accuracy-based removal- and enhancement-based accuracy deviation (RAD / EAD, in %). All results are computed over $n = 50{,}000$ ImageNet validation images.

Group Count M	RAD (%)	EAD (%)
128	7.8	7.7
256	7.0	6.7
512	8.2	8.1
1024	7.3	7.0
2048	7.4	7.3

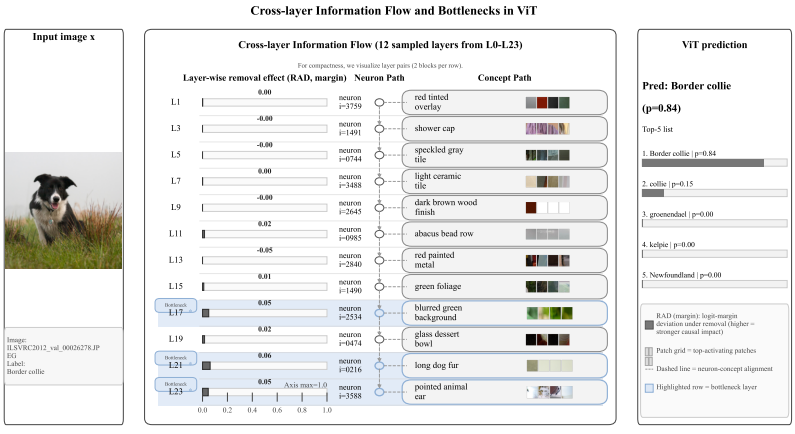


Figure 11: **Cross-layer semantic evolution and bottleneck alignment for class *Border collie* (ViT-L/32).** 12 sampled layers from L0–L23. Left: input and prediction. Middle: per-layer single-unit removal effect $\mathrm{RAD}_{\mathrm{margin}}(l)$, discovered neuron indices, and the aligned concept path with top-activating patch grids; highlighted rows (L17, L21, L23) mark bottleneck layers where a single concept-aligned unit strongly modulates the decision margin. Right: predicted class distribution.

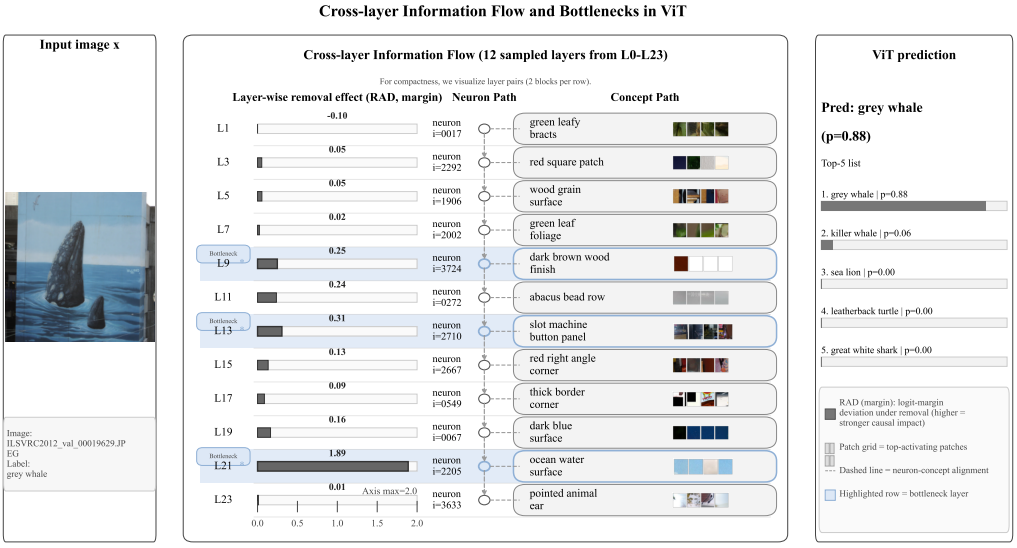


Figure 12: Additional example of cross-layer semantic flow and bottleneck alignment (class 147, ViT-L/32).

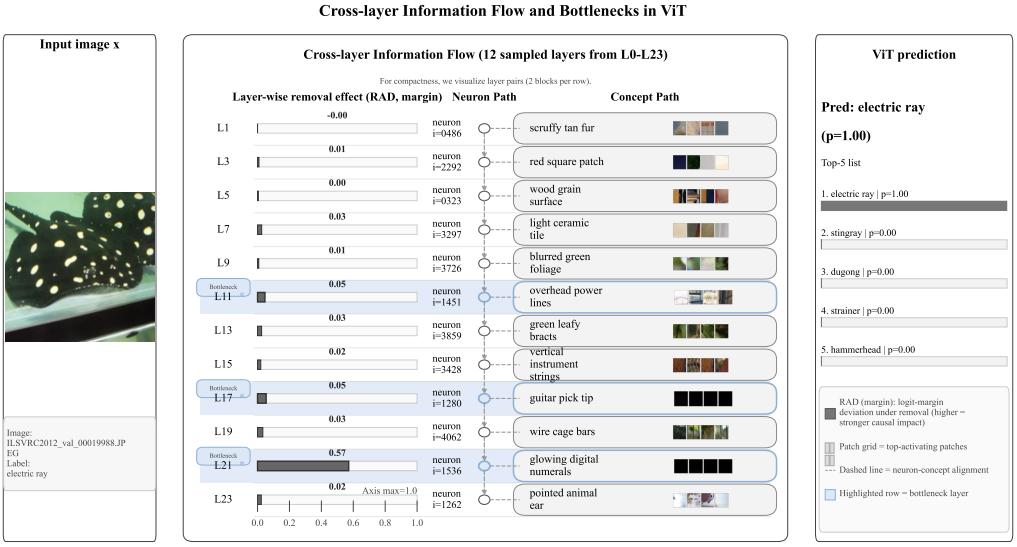


Figure 13: Additional example of cross-layer semantic flow and bottleneck alignment (class 5, *electric ray*, ViT-L/32).

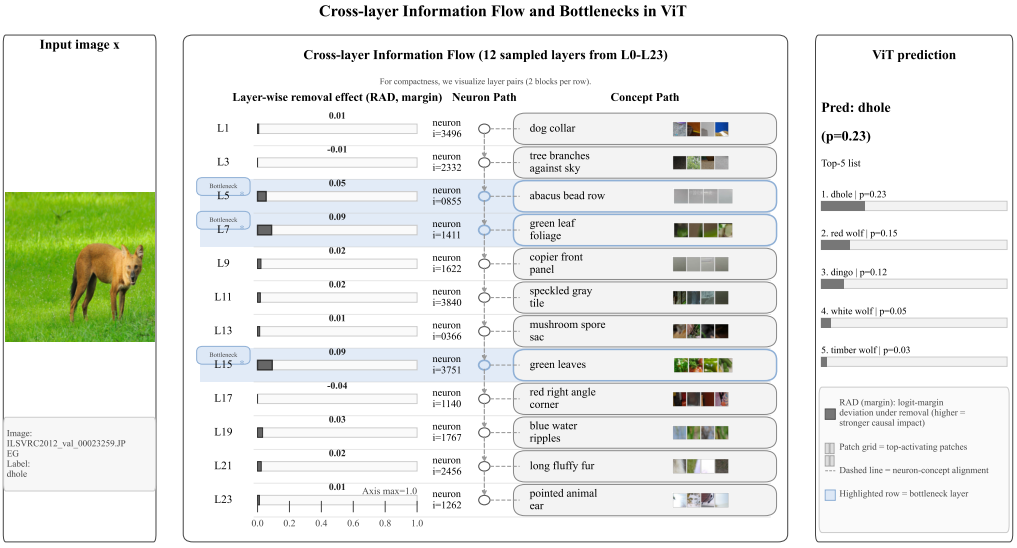


Figure 14: Additional example of cross-layer semantic flow and bottleneck alignment (class 274, ViT-L/32).

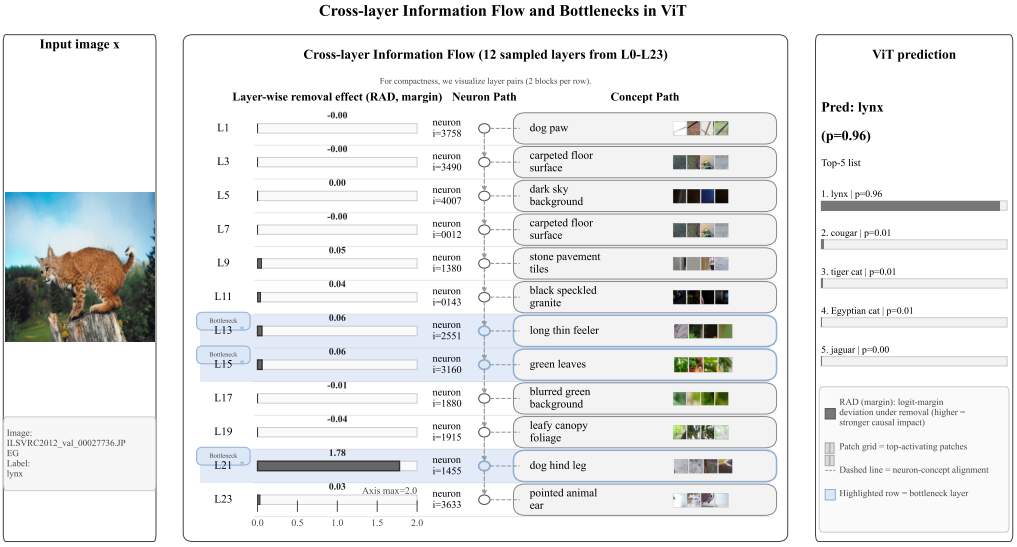


Figure 15: Additional example of cross-layer semantic flow and bottleneck alignment (class 287, ViT-L/32).