SPICE: Simple Polysemantic Feature Interpretation via Clustering-based Explanation

Sehyun Lee [1] **Dahee Kwon** [1] **Damin Lee** [1] **Jaesik Choi** [1]

Abstract

One of the pivotal recent challenges in neural network interpretability is polysemanticity, where a single neuron is activated by multiple, often unrelated concepts, hindering clear functional understanding. Although prior work has explored this phenomenon, existing approaches remain architecture-specific and depend on manual heuristics such as a fixed number of concept clusters (K), limiting their generality and scalability—especially for modern Transformer-based models. To address these limitations, we introduce SPICE (**S**imple **P**olysemantic Feature **I**nterpretation via **C**lustering-based **E**xplanation), a generalizable framework for analyzing polysemanticity in deep vision architectures. SPICE avoids architecture-dependent propagation rules, enabling the first systematic comparison of polysemanticity across both CNNs and Transformers, and automatically determines the number of concept clusters per neuron, eliminating reliance on a preset K and supporting scalable analysis for large models. Using SPICE, we conduct a comprehensive investigation into how polysemanticity emerges, varies across depth and architecture, and forms through distinct computational pathways. Code: https://github.com/sehyun-lee288/spice-icml26-mech-interp.

1. Introduction

Deep learning models have achieved remarkable performance across a wide range of applications, but their rapidly increasing complexity and scale have made it ever more difficult to understand their underlying mechanisms (Arrieta et al., 2020). As a result, interpretability research—which seeks to clarify the internal representations and computational processes learned by models—has become increasingly important and is now an active area of study. In particular, mechanistic interpretability, which examines the functions of neurons and modules in storing and processing knowledge, has emerged as one of the most prominent research directions in the field (Olah et al., 2017; 2020; Bereska & Gavves, 2024; Elhage et al., 2021).

In the vision domain, such research has been actively pursued, with approaches such as (Bau et al., 2017) and (Oikarinen & Weng, 2023) analyzing models at the level of individual neurons by identifying shared visual features among the samples that strongly activate them. This approach rests on the implicit assumption that each neuron corresponds to a single function. Recent studies, however, have shown that multiple concepts often coexist within a single neuron—a phenomenon known as ***polysemanticity*** (Elhage et al., 2022; Dreyer et al., 2024). As a result, single-neuron analysis faces fundamental limitations in faithfulness, and such interpretations alone are insufficient to fully explain the underlying mechanisms of deep models.

To account for polysemanticity, several approaches have been proposed. While training interpretable Sparse Autoencoders (SAEs) offers one promising path, such methods do not directly reveal the intrinsic structural flow of the network (Cunningham et al., 2023; Thasarathan et al., 2025). To address this, attribution clustering has emerged as a key technique for operationalizing circuit-based tracing by grouping a neuron's attribution footprints to separate its encoded concepts. However, existing attribution clustering methods suffer from two critical limitations. First, they exhibit a fundamental lack of generality. To date, these analyses have been almost exclusively confined to CNNs, failing to provide a unified framework applicable to modern Transformer-based architectures. Second, they lack scalability due to their reliance on manual heuristics. Specifically, these approaches often require pre-defining a fixed number of concept clusters (K) for each neuron-a subjective and laborious process that is infeasible for today's large-scale models. Consequently, developing a generalizable and scalable method to analyze polysemanticity across diverse

[1]Kim Jaechul Graduate School of AI, Korea Advanced Institute of Science and Technology (KAIST). Correspondence to: Jaesik Choi <jaesik.choi@kaist.ac.kr>.

Proceedings of the 43rd International Conference on Machine Learning, Seoul, South Korea. PMLR 306, 2026. Copyright 2026 by the author(s).

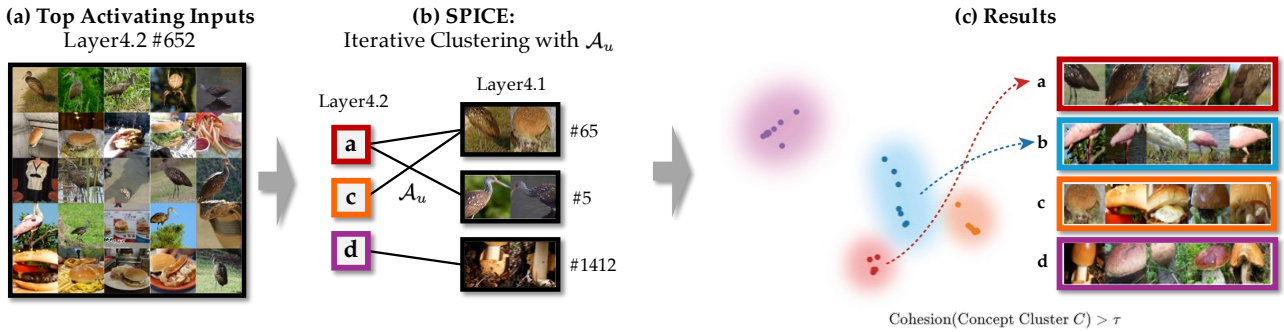

Figure 1. **An overview of the proposed method (SPICE). (a)** A polysemantic neuron exhibits mixed selectivity across diverse image inputs. **(b)** SPICE traces the computational pathways of these activations by computing attribution footprints ($\mathcal{A}_u$) from the preceding layer. **(c)** We iteratively cluster these footprints to disentangle the neuron's representations. By strictly separating clusters that satisfy the condition Cohesion$(C) > \tau$, SPICE successfully decomposes the entangled inputs into highly coherent, interpretable conceptual groups.

architectures remains a key unsolved problem.

Therefore, we introduce SPICE (**S**imple **P**olysemantic Feature **I**nterpretation via **C**lustering-based **E**xplanation), a method that disentangles polysemantic neurons with an emphasis on generality and scalability. SPICE is both simple and effective, addressing two major limitations of prior work. (1) We introduce a scalable and flexible attribution-clustering framework for dissecting polysemantic neurons. By adaptively determining the optimal number of concept clusters (K), SPICE eliminates the reliance on manual heuristics. Furthermore, extensive evaluations and ablations demonstrate that our method achieves highly robust disentanglement performance across diverse architectures, extreme activation ranges, and various attribution methods. (2) Using this framework, we provide an in-depth analysis of polysemanticity to explore the representational differences across models. Our findings highlight distinct inductive biases between CNNs and Transformers, and illustrate the diverse computational pathways that underlie concept formation.

Using the proposed method, we conduct an in-depth investigation into the formation patterns of polysemanticity and their internal structure. Beyond extending interpretability, this analysis offers novel insights into the fundamental principles by which models organize and represent concepts.

2. Related Work

Mechanistic Interpretability A significant branch of interpretability research focuses on identifying human-understandable concepts and mechanisms within neural networks (Bereska & Gavves, 2024). This line of work seeks to move beyond input–output correlations and explain how a model computes internally. Early studies focused on associating individual neurons with specific visual concepts (Bau et al., 2017; Olah et al., 2017; 2020). Since then, research has broadened to include methods for automatically discovering concepts with limited human supervision (Kalibhat et al., 2023; Oikarinen & Weng, 2023), as well as approaches that analyze circuits—pathways of functionally related neurons—to uncover higher-level computational structures (Cao et al., 2025; Wang et al., 2022; Kowal et al., 2024). Our work builds on this tradition, aiming to provide a scalable and systematic analysis of a neuron's internal conceptual structure.

Polysemanticity as a Challenge Early interpretability methods often relied on an implicit monosemantic assumption—that each neuron corresponds to a single coherent concept. However, polysemanticity is essentially a fundamental property of deep networks, arising from data statistics and superposition constraints (Marshall & Kirchner, 2024; Elhage et al., 2022). To account for this, Sparse Autoencoders (SAEs) have emerged as a powerful tool for disentangling compressed features (Cunningham et al., 2023; Thasarathan et al., 2025) and remain a highly active direction. They are, however, complementary to attribution-based analyses in scope: rather than tracing how concepts arise along the network's intrinsic computation flow, SAEs decompose representations into a learned dictionary, and recent work has noted practical considerations such as training stability and reconstruction–interpretability trade-offs that motivate methods grounded in the model's own attribution structure (**?**). Furthermore, while several works leverage Vision-Language Models (VLMs) to probe polysemantic concepts (Achiam et al., 2023; Zhai et al., 2023; Oikarinen & Weng, 2024; Yu et al., 2025), these approaches are fundamentally bottlenecked by representation misalignment and the sensitivity to the chosen external model.

Disentanglement via Attribution Approach To directly analyze the internal structure of polysemantic neurons, several approaches leverage attribution clustering—grouping a

neuron's attribution footprints to separate its encoded concepts (Dreyer et al., 2024; Hesse et al., 2025; Yu et al., 2025). These methods are powerful, but the literature exhibits two critical limitations that our work addresses. First, a lack of generality across architectures. To date, these analyses (Dreyer et al., 2024; Hesse et al., 2025) have been almost exclusively confined to CNN architectures. A unified framework to systematically analyze and compare polysemanticity in modern Transformer-based models has been notably absent. Second, prior work lacks scalability in concept discovery, largely due to reliance on manual or restrictive heuristics. Pioneering work like PURE (Dreyer et al., 2024) requires manually specifying the number of concepts (K) for each neuron, while methods such as (Hesse et al., 2025) perform explicit disentanglement but are constrained to separating only a small, fixed number of components rather than discovering an arbitrary, data-driven set of clusters. Both approaches are subjective, time-consuming, and practically infeasible for large-scale analysis. Therefore, a framework that is both generalizable and scalable for attribution-based analysis has remained a key unsolved problem, which we address in this paper.

3. Method

The visual concept captured by a single neuron is often inferred from the shared features of the samples that most strongly activate it (Bau et al., 2017; Kalibhat et al., 2023; Oikarinen & Weng, 2023). While this approach can reveal what a neuron has learned, it becomes challenging to disentangle multiple concepts when a neuron is polysemantic—i.e., when several distinct concepts coexist within the same unit. To analyze this polysemanticity within a model's learned visual representations, our goal is therefore to cluster the multiple visual concepts encoded by a single neuron. To this end, we introduce **S**imple **P**olysemantic Feature **I**nterpretation via **C**lustering-based **E**xplanation (SPICE).

Identifying Representational Footprints via Attribution Since deep networks build representations hierarchically across layers, understanding an internal visual concept requires tracing its compositional origins along this information flow. We therefore characterize each neuron by a *representational footprint* computed via attribution methods (Shrikumar et al., 2017; Selvaraju et al., 2017; Simonyan et al., 2014), under the assumption that concepts sharing similar footprints are treated as 'related' by the model.

Formally, given a dataset $\mathcal{D} = \{x_i\}$ and a network f, we first interpret the concept of a neuron u^l in layer l through its set of most highly activating samples, $\mathcal{D}_{u^l} \subset \mathcal{D}$. We define a *neuron* based on the architecture: for convolution-based models, we consider each channel as a neuron, as it acts as a distinct feature detector across spatial locations [1]. For transformer-based models, we consider each hidden dimension (channel) of a representation as a neuron, as it maintains a distinct representation for each input patch.

Then, to trace the origin of the visual concept in u^l, we quantify how each neuron u_j^{l-1} in the preceding layer, contributes to the activation of neuron u^l, denoted a_{u^l}. The attribution $A(u_j^{l-1} \to u^l)$ was obtained following the Input×Gradient method (Shrikumar et al., 2017). For a given input x, the attribution is calculated as:

$$\underbrace{A(u_j^{l-1} \to u^l)}_{\in \mathbb{R}^{H\times W} \text{ or } \mathbb{R}^P} = a_{u_j^{l-1}}(x) \odot \frac{\partial a_{u^l}(x)}{\partial a_{u_j^{l-1}}(x)} \tag{1}$$

where $\odot$ denotes the element-wise product, and the output is a per-location scalar map of shape $H \times W$ for CNNs or a per-patch vector of length P for Transformers. We aggregate these values into a single scalar $A_j^{l-1} \in \mathbb{R}$ by summing over all spatial/patch locations, so that the per-neuron footprint set used for clustering is $\mathcal{A}_u = \{A_j^{l-1} \in \mathbb{R} \mid j = 1, \ldots, d\} \in \mathbb{R}^d$.

We also observe that, as the depth of a deep vision model increases, the average similarity among attribution footprints $\mathcal{A}_u$ (averaged across neurons u in the layer) becomes substantially higher. We hypothesize that this effect stems from the increasing anisotropy of the feature space, a phenomenon widely reported in prior work on large models (Godey et al., 2024; Razzhigaev et al., 2024; Bansal et al., 2018; Elhage et al., 2023). To mitigate this, we normalize attribution matrices along the neuron dimension before clustering, which counteracts the anisotropy and lets the clustering focus on relative semantic patterns rather than absolute magnitudes. A detailed analysis of the normalization effect is provided in Appendix B.3. For convenience, we will abbreviate $A(u_j^{l-1} \to u^l)$ into A_j^{l-1}.

Disentangling Concepts by Clustering Footprints Having defined the representational footprints, we now introduce our method, namely SPICE. Rather than assuming a fixed number of underlying concepts, we propose an iterative algorithm that progressively identifies and separates cohesive groups of footprints based on their pairwise cosine similarity. As detailed in Algorithm 1, the process begins by attempting to partition the current set of attributions $\mathcal{A}_u = \{A_j^{l-1} | j = [1, ..., d]\}$, where d denotes the number of neurons in layer $l-1$, into $k = 2$ clusters. A cluster $\mathcal{C}$ is considered a cohesive concept if its internal coherence, which we define as the average pairwise cosine similarity between its members, exceeds an adaptive threshold τ. Cohesive concepts are accepted as final and removed from the working set $\mathcal{A}'$. The algorithm then dynamically adjusts

[1] The term 'neuron' is often used interchangeably with 'feature' in the literature, or 'channel' in the context of CNNs.

Algorithm 1 SPICE: DisentangleConcepts($\mathcal{A}_u, \tau$)

Input: attribution footprints $\mathcal{A}_u$, cohesion threshold τ.
Output: disjoint concepts $\mathcal{C}^*$ (each a set of footprints).
▷ *Initialize working variables*
$\mathcal{A}' \leftarrow \mathcal{A}_u$
$\mathcal{C}^* \leftarrow \emptyset$
$k \leftarrow 2$
while $|\mathcal{A}'| \geq k$ **do**
  ▷ *Search for cohesive clusters of size k*
  Let $\mathcal{C}_{cohesive}$ be the set of all clusters $C \in$ Cluster($\mathcal{A}', k$) where Cohesion(C) $> \tau$.
  if $\mathcal{C}_{cohesive} \neq \emptyset$ **then**
    ▷ *Found cohesive clusters: update*
    $\mathcal{C}^* \leftarrow \mathcal{C}^* \cup \mathcal{C}_{cohesive}$
    $\mathcal{A}' \leftarrow \mathcal{A}' \setminus \bigcup_{C \in \mathcal{C}_{cohesive}} C$
    $k \leftarrow \max(2, k - (|\mathcal{C}_{cohesive}| - 1))$
  else
    ▷ *No cohesive cluster: increase search size*
    $k \leftarrow k + 1$
  end if
end while
▷ *Treat remaining footprints as individual concepts*
$\mathcal{C}^* \leftarrow \mathcal{C}^* \cup \{\{a\} \mid a \in \mathcal{A}'\}$
return $\mathcal{C}^*$

where Cohesion(C) $= \text{mean}_{a_i, a_j \in C}(\text{sim}(a_i, a_j))$.

the number of clusters k and repeats this process on the remaining attributions until no further cohesive concepts can be disentangled.

A key component is the dynamic calculation of this cohesion threshold τ. A fixed, universal threshold would fail to adapt to the diverse similarity distributions exhibited by the footprints of different neurons. We therefore compute τ adaptively from the set of footprints currently under analysis, $\mathcal{A}'$. Our empirical analysis reveals that the character of the similarity distribution systematically changes with network depth, as illustrated in Figure 7(right). Early layers typically exhibit a unimodal, long-tailed distribution of similarities. In contrast, later layers often display a bimodal distribution, where the emergence of a second peak suggests the formation of distinct, highly cohesive conceptual groups.

To create a robust criterion that accommodates both scenarios, we define the threshold τ as the maximum of two candidates: (1) the 95th percentile of the pairwise similarity values, and (2) the value corresponding to the second peak of a Kernel Density Estimate (KDE) fitted to the similarity distribution. The percentile provides a robust baseline, particularly for the unimodal distributions in early layers, while the second peak offers a more semantically meaningful partition for the well-formed concepts in later layers.

Table 1. Separability comparison of baselines versus **Ours** across models and layers. As PURE was originally designed for CNNs, we adapted its CRP implementation for Transformer models for this comparison.

Layer	PURE*	LE	CPE	Ours
ResNet-50				
L2.1	$1.018_{\pm 0.031}$	$\mathbf{1.112}_{\pm 0.122}$	$0.996_{\pm 0.014}$	$1.092_{\pm 0.056}$
L3.3	$0.999_{\pm 0.012}$	$1.108_{\pm 0.153}$	$0.995_{\pm 0.007}$	$\mathbf{1.160}_{\pm 0.081}$
L4.2	$1.009_{\pm 0.071}$	$1.187_{\pm 0.221}$	$0.998_{\pm 0.011}$	$\mathbf{1.291}_{\pm 0.224}$
ViT-B				
B2	$1.002_{\pm 0.010}$	$\mathbf{1.199}_{\pm 0.181}$	$0.994_{\pm 0.009}$	$1.137_{\pm 0.062}$
B6	$1.012_{\pm 0.013}$	$1.185_{\pm 0.153}$	$0.994_{\pm 0.009}$	$\mathbf{1.240}_{\pm 0.089}$
B11	$1.035_{\pm 0.025}$	$1.357_{\pm 0.181}$	$0.998_{\pm 0.018}$	$\mathbf{1.577}_{\pm 0.089}$
DenseNet				
L2.1	$1.012_{\pm 0.028}$	$\mathbf{1.090}_{\pm 0.158}$	$0.994_{\pm 0.007}$	$1.071_{\pm 0.059}$
L3.12	$1.053_{\pm 0.054}$	$1.062_{\pm 0.139}$	$0.994_{\pm 0.005}$	$\mathbf{1.114}_{\pm 0.060}$
L4.16	$1.025_{\pm 0.035}$	$\mathbf{1.191}_{\pm 0.116}$	$0.994_{\pm 0.009}$	$1.033_{\pm 0.032}$
CLIP ViT-B				
B2	$1.007_{\pm 0.018}$	$1.088_{\pm 0.082}$	$0.994_{\pm 0.008}$	$\mathbf{1.090}_{\pm 0.051}$
B6	$1.008_{\pm 0.016}$	$1.165_{\pm 0.148}$	$0.997_{\pm 0.009}$	$\mathbf{1.212}_{\pm 0.090}$
B11	$1.021_{\pm 0.040}$	$1.244_{\pm 0.192}$	$0.998_{\pm 0.013}$	$\mathbf{1.410}_{\pm 0.102}$

This dual-criterion approach ensures that our definition of a concept is contextually grounded across the network's hierarchy. The sensitivity analysis on the threshold parameter τ is in Appendix B.2.

4. Experiment

In this section, we present both the qualitative and quantitative evaluation of our proposed method, along with a comprehensive analysis. Detailed experimental settings and ablation studies are provided in Appendix A and B, respectively.

4.1. Quantitative Comparison with Baselines

To demonstrate the effectiveness of our method in disentangling and explaining multiple concepts encoded within a single neuron, we extensively evaluate SPICE across four diverse architectures (ResNet-50, ViT-B, DenseNet, and CLIP ViT-B). We compare our approach against several recent baselines, including PURE (Dreyer et al., 2024), CPE (Yu et al., 2025), and LE (Oikarinen & Weng, 2024). While PURE was originally restricted to CNNs due to its reliance on CRP grammar, we extended and adapted its implementation for Transformer architectures to ensure a comprehensive evaluation. Moreover, unlike SPICE-which adaptively determines the optimal number of clusters per neuron-PURE requires the number of clusters to be specified in advance. Accordingly, we configure it by selecting the optimal number of clusters using the silhouette score.

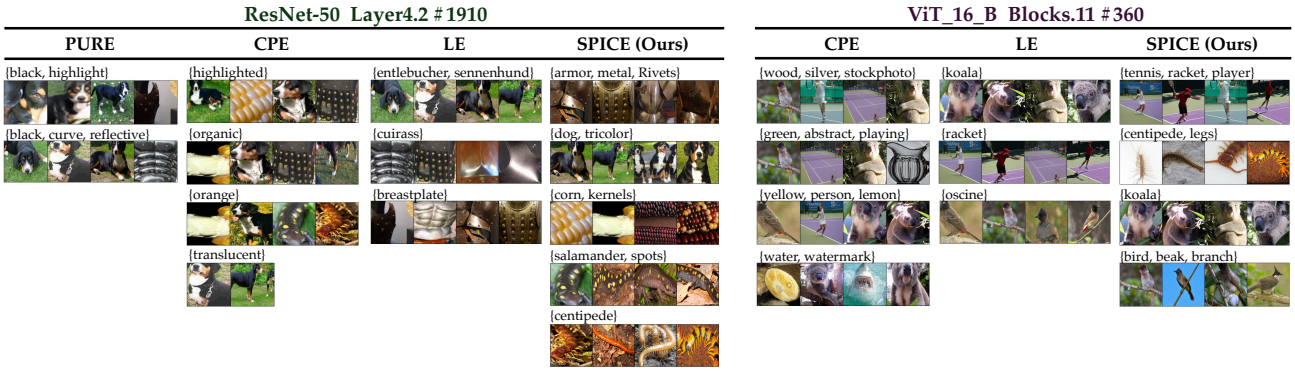

Figure 2. Qualitative comparisons with various baselines. Each row shows four exemplar images representing the visual characteristics of a discovered cluster, accompanied by its corresponding textual description.

4.1.1. Separability based on External Semantic Space.

To quantitatively evaluate the effectiveness of our concept disentanglement, we first adopt *separability* as our primary metric. Following the established protocol of PURE, we utilize CLIP as the external verification tool to measure these similarities, grounded in the premise that visually similar images cluster closely in the hidden space. Concretely, for each neuron u with discovered cluster set $\mathcal{C}_u = \{C_1, \dots, C_{K_u}\}$, we define

$$\text{sep}(u) = \frac{\overline{\text{sim}}_{\text{intra}}(u)}{\overline{\text{sim}}_{\text{inter}}(u)}, \quad \text{Separability} = \underset{u}{\text{mean}}\, \text{sep}(u),$$

where $\overline{\text{sim}}_{\text{intra}}(u)$ averages CLIP cosine similarity over *all* within-cluster image pairs and $\overline{\text{sim}}_{\text{inter}}(u)$ averages over *all* cross-cluster pairs (i.e. a ratio-of-means, not a mean-of-ratios). A higher score indicates more successful and distinct concept separation.

As shown in Table 1, SPICE exhibits dominant separability scores across various networks and depths. While text-prior-based methods like LE show competitive performance in early layers (e.g., B2), their performance notably degrades in deeper layers (e.g., L4.2, B11) where concepts become highly abstract and polysemantic. In contrast, SPICE maintains robust disentanglement, significantly outperforming all baselines in the deep layers. The slight deviation in early layers occurs because they typically capture low-level features (e.g., textures, colors) that are less aligned with CLIP's high-level semantic space. This phenomenon is visually supported in Figure 2, and a more detailed analysis is provided in Appendix F.

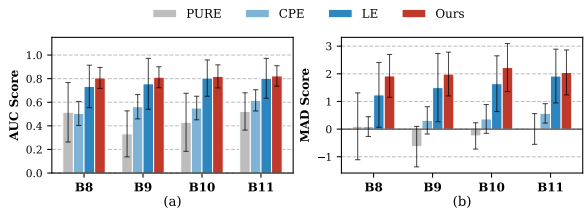

Figure 3. Simulation-based Semantic Evaluation in ViT-B

4.1.2. Simulated-based Evaluation.

To further rigorously validate the semantic faithfulness of the discovered concepts beyond embedding-based metrics, we conduct a generative evaluation using the CoSy protocol (Kopf et al., 2024). For each cluster, we generate a textual concept description following the CPE pipeline of (Yu et al., 2025) (using GPT-4.1, `gpt-4.1-2025-04-14`, with their Chain-of-Explanation prompts), and then synthesize verification images with SDXL (`stable-diffusion-xl-base-1.0`) using the prompt "*realistic photo of a close up of* {concept}". AUC and MAD are computed *per cluster* and then averaged across clusters and neurons, so that the metric is invariant to the cluster count K each method produces.

As illustrated in Figure 3, SPICE consistently achieves the highest AUC and MAD scores across multiple blocks of ViT-B compared to PURE, CPE, and LE. It is worth noting that while inherent text-visual misalignment exists in deep vision models, SPICE achieves higher validity because it intrinsically discovers concept clusters from internal activations independent of text. By doing so, it effectively bypasses the bottleneck of text-defined priors that fundamentally limits baselines like LE and CPE, ensuring a more faithful recovery of the model's actual learned concepts.

4.2. Qualitative Comparison with Baselines

To visually demonstrate the effectiveness of our method in disentangling and explaining multiple concepts encoded within a single neuron, we perform a qualitative comparison against the aforementioned baselines (PURE (Dreyer et al.,

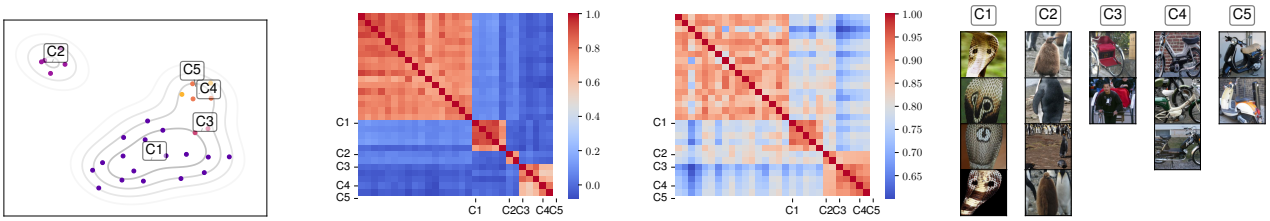

Figure 4. Dissecting a polysemantic neuron (#652) from ViT-B block 11. **(a)** UMAP embedding reveals distinct conceptual clusters within the neuron's representation footprints. **(b)** A similarity heatmap quantifies the low inter-cluster and high intra-cluster similarity. **(c)** A CLIP similarity heatmap quantitatively shows that concepts are semantically disparate. **(d)** Exemplar images confirm the diverse and disparate visual concepts represented by each cluster; such as snakeskin (C1), penguins (C2), and various two-wheeled vehicles (C3-C5).

2024), CPE (Yu et al., 2025), and LE (Oikarinen & Weng, 2024)).

In Figure 2, each row illustrates representative images belonging to a single concept cluster discovered by each method. Across both ResNet-50 and ViT-B/16, SPICE discovers the most comprehensive set of polysemantic clusters, indicating its strong capability to fully recover the diverse concepts encoded within a neuron. Moreover, while several baselines exhibit notable visual inconsistency and noise within their clusters, SPICE consistently yields semantically aligned and coherent groupings. Similar to LE and CPE, SPICE naturally integrates with vision-language models (Achiam et al., 2023) to provide accurate textual descriptions for each cluster. Taken together, these qualitative results demonstrate that SPICE offers the most reliable visual characterization of neuron-level polysemanticity. More examples are shown in Appendix F.

4.3. In-depth Analysis

Using our framework, we conduct a comprehensive analysis of polysemanticity. We first visually dissect a single neuron, and then trace the computational pathways that give rise to these diverse concepts. We then broaden our scope to explore architectural inductive biases across models, and finally validate the robustness of our findings through rigorous ablation studies.

4.3.1. Dissecting a Polysemantic Neuron.

To understand how polysemanticity physically manifests, we conduct a detailed case study of Neuron #652 from the last block of ViT-B. As illustrated in Figure 4, we decompose its activations into distinct conceptual clusters. The UMAP projection (Figure 4a) and exemplar images (Figure 4d) reveal clear semantic heterogeneity, with the neuron responding to entirely unrelated concepts such as snakeskin textures (C1), penguins (C2), and two-wheeled vehicles (C3-C5). The heatmaps (Figure 4b, c) further validate that these clusters are structurally well-separated within the model's internal space and possess low semantic overlap, confirming the precise disentanglement by SPICE.

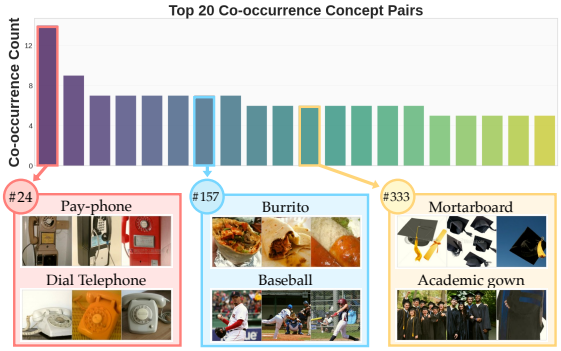

Figure 5. Top 20 co-occurrence concept pairs in ViT-B.

4.3.2. Formation Mechanisms and Concept Pathways

Having observed that a single neuron can perfectly disentangle into distinct concepts, we next investigate *how* these specific concepts are formed. By measuring label co-occurrences in the final ViT-B block (Figure 5), we find that polysemantic neurons arise heterogeneously: some encode highly related concepts (e.g., variants of telephones in neuron #24), while others combine entirely unrelated categories (e.g., burritos and baseballs in neuron #157).

To determine if these distinct forms of polysemanticity stem from different mechanisms, we trace their upstream computational pathways (Figure 6). For neuron #13, which encodes unrelated concepts, we observe largely disjoint upstream connections, with only an 18% overlap among the top-50 contributing neurons. Conversely, neuron #371, encoding related concepts, shares 42% of its attribution pathways, indicating a shared formation circuit. These human-aligned interpretations are further validated by VLM-generated descriptions (Achiam et al., 2023; Yu et al., 2025). Ultimately, these findings reveal that polysemantic neurons are formed through diverse processes—ranging

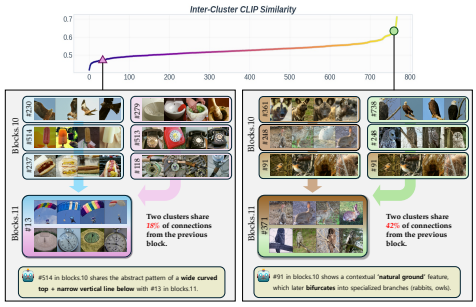

Figure 6. (**Left**) Inter-cluster CLIP similarity, where the x-axis denotes neurons in ViT-B block 11 sorted by similarity. (**Middle**, **Right**) Example of a polysemantic neuron with low inter-cluster similarity, and a high inter-cluster similarity, respectively.

from overlapping semantic circuits to entirely disjoint pathways—which fundamentally dictate their interpretability.

4.3.3. Architectural Inductive Bias.

Building on these neuron-level mechanisms, we broaden our analysis to examine how macroscopic architectural designs influence polysemantic behavior across the entire network (Figure 7). First, regarding the number of disentangled concepts (left), CNN-based models (ResNet, ConvNeXt) maintain numerous concepts before a sharp collapse in the final blocks, whereas Vision Transformers (ViT) show a gradual decrease. Second, assessing internal cohesion via intra-cluster similarity (center), CNNs exhibit consistent stability interrupted only by spatial downsampling transitions. Conversely, Transformers display a characteristic U-shaped trend, exploring higher conceptual diversity in middle layers before converging.

These behavioral differences stem from the evolution of feature-similarity distributions (Figure 7, right). While all models begin with unimodal distributions indicating limited separation in early layers, they transition to distinctly bimodal distributions later, reflecting clear concept splits. This multimodality shift, quantified via Hartigan's Dip Test (see Appendix C), is most pronounced in ViTs. Ultimately, these contrasting trajectories highlight distinct inductive biases: CNNs preserve cohesive concepts driven by local processing, while ViTs undergo mid-layer diversification and late-stage reconsolidation via global self-attention.

4.3.4. Ablation Studies

The validity of the aforementioned scientific insights hinges on the robustness of the SPICE framework. Here, we thoroughly evaluate our core design choices.

Robustness across Activation Ranges. Standard interpretability evaluation protocols often restrict their analysis to the highly activated range (e.g., Top-10%) of a neuron.

Table 2. Separability across Activation Ranges (Top/Mid/Bot)

Method	B2			B4			B6			B8			B10		
	T	M	B	T	M	B	T	M	B	T	M	B	T	M	B
PURE	1.02	1.00	1.02	1.02	1.00	1.02	1.03	1.00	1.01	1.03	1.00	1.01	1.07	1.00	1.01
CPE	0.99	0.99	0.99	1.01	0.99	0.99	0.99	0.99	1.00	1.01	0.99	0.99	1.01	0.99	0.99
LE	**1.20**	**1.11**	1.12	1.15	**1.15**	1.13	1.19	**1.18**	1.13	**1.21**	1.14	1.13	1.33	1.13	1.04
Ours	1.14	1.09	**1.12**	**1.19**	1.14	**1.16**	**1.24**	1.18	**1.21**	1.20	**1.31**	**1.29**	**1.48**	**1.27**	**1.41**

To ensure our method's robustness and comprehensively address the evaluation scope, we expand our quantitative analysis to cover the **Top**, **Middle**, and **Bottom** activation ranges (100 samples each) across varying depths. As shown in Table 2, SPICE consistently maintains high separability scores significantly above the PURE and CPE baselines across all depths and ranges. Notably, while LE achieves competitive scores in early layers (e.g., B2), its performance degrades substantially in deeper layers (B10) and lower activation ranges (Bottom: 1.04). In contrast, SPICE demonstrates superior robustness, achieving a dominant score of 1.41 even in the B10 Bottom range. This confirms that SPICE effectively disentangles concepts regardless of activation magnitude, successfully operating even in highly polysemantic and less activated spaces where existing text-prior-based methods fail.

Robustness to Attribution Methods. To validate our core design choices, we first examine whether SPICE's disentanglement relies on the specific attribution method used to compute the representational footprints. As shown in Figure 8 (Top), SPICE consistently outperforms the PURE baseline regardless of the attribution method employed—including Integrated Gradients (Sundararajan et al., 2017), GradientShap (Lundberg & Lee, 2017), Guided Backpropagation (Springenberg et al., 2015), and Input $\times$ Gradient. This confirms that the significant performance gain stems from our framework itself, rather than an artifact of a specific metric. We adopt Input $\times$ Gradient as our default choice due to its efficiency.

Adaptive vs. Static Thresholding. Finally, we evaluate the necessity of our adaptive cohesion threshold compared to a naive static one. We emphasize that any clustering algorithm can trivially partition every neuron's footprints into k groups (e.g., $k = 2$); the question is whether the resulting clusters are *cohesive enough to be treated as concepts*. SPICE therefore retains only clusters passing $\text{Cohesion}(C) > \tau$, and we report *coverage* as the fraction of 100 randomly sampled neurons for which at least one such cluster is found (neurons with no qualifying cluster are excluded from the separability average). As illustrated in Figure 8 (Bottom), a static threshold rule, where τ for each neuron is set to a fixed percentile of its own pairwise similarities (e.g., top-10%, i.e. the 90th percentile), yields high raw separability but coverage collapses (e.g., only 6 of the 100 sampled neurons retained), so the score reflects a small

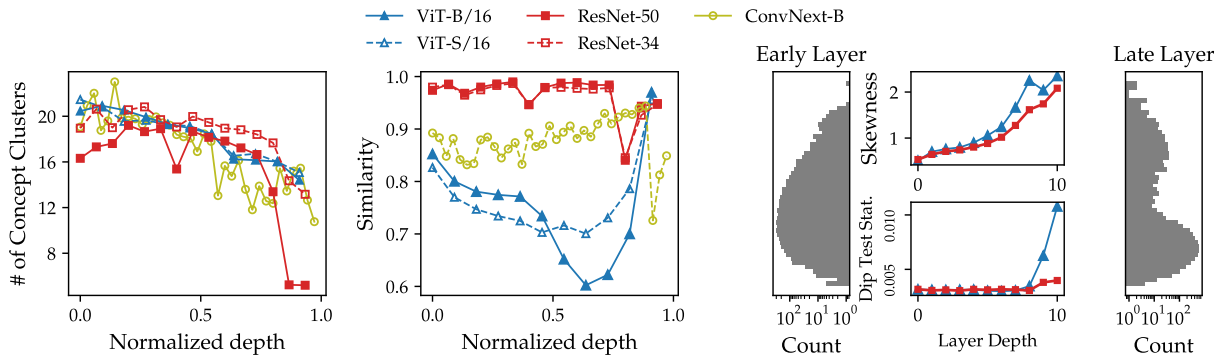

Figure 7. Polysemanticity differs across network architectures. **(Left)** The number of concepts per neuron across layers. **(Center)** Intra-cluster similarity across layers. **(Right)** A structural evolution where the similarity distribution shifts from unimodal to bimodal.

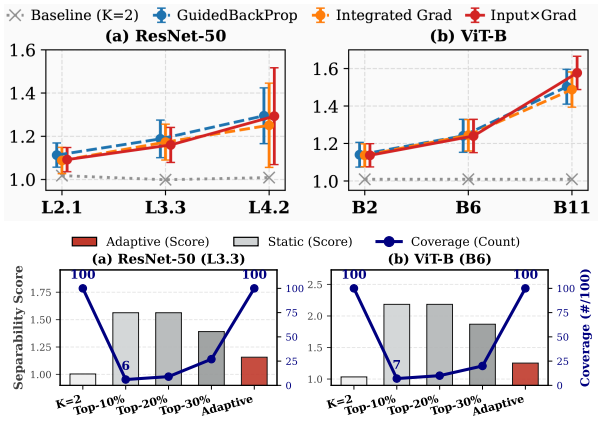

Figure 8. **Ablation Studies.** **(Top)** Robustness to attribution methods. **(Bottom)** Adaptive vs. static cohesion thresholding, evaluated on a fixed pool of 100 randomly sampled neurons. The x-axis is *coverage*, the fraction of these 100 neurons for which at least one cluster C with $\text{Cohesion}(C) > \tau$ is found; neurons that yield no such cluster are excluded from the separability average. Each curve sweeps the threshold strictness: a stricter τ raises separability but admits fewer neurons.

set of easy outliers. SPICE's adaptive, per-neuron threshold (the max of the 95th percentile and the KDE second-peak; Sec. 3) reaches near 100% coverage while maintaining competitive separability, showing that the choice of τ, not the choice of k, is the bottleneck. We further confirm in Appendix B.5 that this near-full coverage does not come from artificial over-segmentation.

5. Limitations

While SPICE generally yields superior separability, the utility of more advanced attribution methods warrants further investigation. Moreover, the lower separability observed in the early layers of ViT-B highlights a structural constraint of the CLIP verification protocol itself as early layers capture low-level features that are not easily distinguishable within CLIP's high-level semantic space. This underscores that numerical assessments of concept quality heavily depend on the external semantic space used for verification. Despite its generality, our method presents practical constraints: generating human-understandable labels relies on external VLMs, and the iterative clustering process is more computationally intensive than single-pass methods. These challenges motivate clear avenues for future work, such as developing strategies to harness VLM knowledge while compensating for representational misalignment. Ultimately, the framework can be extended toward circuit-level disentanglement, moving beyond single neurons to systematically trace the formation mechanisms of concepts along their pathways.

6. Conclusion

In this work, we addressed the challenge of polysemanticity—the tendency of a single neuron to encode multiple, often unrelated concepts—which obscures the functional interpretation in deep models. We introduce SPICE (Simple Polysemantic Feature Interpretation via Clustering-based Explanation), a simple and generalizable framework that overcomes the architectural constraints and heuristic assumptions (e.g., fixed K) of prior approaches. SPICE effectively disentangles the visual concepts encoded within individual neurons, enabling the first systematic comparison of polysemanticity across both CNNs and Transformers. By automatically determining the number of concept clusters per neuron, it eliminates reliance on manual hyperparameters and provides a scalable analysis pipeline suitable for large modern architectures. Using SPICE, we conduct a comprehensive analysis across diverse deep vision architectures. We believe this work highlights the importance of shifting interpretability from single-neuron analyses to the concept clusters they encode, providing a more faithful lens through which to understand the internal mechanisms of deep visual models.

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

References

Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F. L., Almeida, D., Altenschmidt, J., Altman, S., Anadkat, S., et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Achtibat, R., Dreyer, M., Eisenbraun, I., Bosse, S., Wiegand, T., Samek, W., and Lapuschkin, S. From attribution maps to human-understandable explanations through concept relevance propagation. *Nature Machine Intelligence*, 5 (9):1006–1019, 2023.

Arrieta, A. B., Díaz-Rodríguez, N., Del Ser, J., Bennetot, A., Tabik, S., Barbado, A., García, S., Gil-López, S., Molina, D., Benjamins, R., et al. Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai. *Information fusion*, 58:82–115, 2020.

Ball, G. H. and Hall, D. J. Isodata, a novel method of data analysis and pattern classification. 1965.

Bansal, N., Chen, X., and Wang, Z. Can we gain more from orthogonality regularizations in training deep networks? *Advances in Neural Information Processing Systems*, 31, 2018.

Bau, D., Zhou, B., Khosla, A., Oliva, A., and Torralba, A. Network dissection: Quantifying interpretability of deep visual representations. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 6541–6549, 2017.

Bereska, L. and Gavves, E. Mechanistic interpretability for ai safety–a review. *arXiv preprint arXiv:2404.14082*, 2024.

Bossard, L., Guillaumin, M., and Van Gool, L. Food-101–mining discriminative components with random forests. In *European conference on computer vision*, pp. 446–461. Springer, 2014.

Campello, R. J., Moulavi, D., and Sander, J. Density-based clustering based on hierarchical density estimates. In *Pacific-Asia conference on knowledge discovery and data mining*, pp. 160–172. Springer, 2013.

Cao, T. M., Hoang-Xuan, N., Pham, H. H., Nguyen, P. L., and Thai, M. T. Neurflow: Interpreting neural networks through neuron groups and functional interactions. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. URL https://openreview.net/forum?id=GdbQyFOUlJ.

Cunningham, H., Ewart, A., Riggs, L., Huben, R., and Sharkey, L. Sparse autoencoders find highly interpretable features in language models. *arXiv preprint arXiv:2309.08600*, 2023.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., and Houlsby, N. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=YicbFdNTTy.

Dreyer, M., Purelku, E., Vielhaben, J., Samek, W., and Lapuschkin, S. Pure: Turning polysemantic neurons into pure features by identifying relevant circuits. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8212–8217, 2024.

Elhage, N., Nanda, N., Olsson, C., Henighan, T., Joseph, N., Mann, B., Askell, A., Bai, Y., Chen, A., Conerly, T., et al. A mathematical framework for transformer circuits. *Transformer Circuits Thread*, 1(1):12, 2021.

Elhage, N., Hume, T., Olsson, C., Schiefer, N., Henighan, T., Kravec, S., Hatfield-Dodds, Z., Lasenby, R., Drain, D., Chen, C., et al. Toy models of superposition. *arXiv preprint arXiv:2209.10652*, 2022.

Elhage, N., Lasenby, R., and Olah, C. Privileged bases in the transformer residual stream. *Transformer Circuits Thread*, pp. 24, 2023.

Godey, N., de la Clergerie, É., and Sagot, B. Anisotropy is inherent to self-attention in transformers. *arXiv preprint arXiv:2401.12143*, 2024.

Hartigan, J. A. and Hartigan, P. M. The dip test of unimodality. *The annals of Statistics*, pp. 70–84, 1985.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2015. URL https://api.semanticscholar.org/CorpusID:206594692.

Hesse, R., Fischer, J., Schaub-Meyer, S., and Roth, S. Disentangling polysemantic channels in convolutional neural networks. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 4799–4803, 2025.

Huang, G., Liu, Z., van der Maaten, L., and Weinberger, K. Q. Densely connected convolutional networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pp. 2261–2269. IEEE Computer Society, 2017. doi: 10.1109/CVPR.2017.243. URL https://doi.org/10.1109/CVPR.2017.243.

Kalibhat, N., Bhardwaj, S., Bruss, C. B., Firooz, H., Sanjabi, M., and Feizi, S. Identifying interpretable subspaces in image representations. In *International Conference on Machine Learning*, pp. 15623–15638. PMLR, 2023.

Kopf, L., Bommer, P. L., Hedström, A., Lapuschkin, S., Höhne, M. M., and Bykov, K. Cosy: Evaluating textual explanations of neurons. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=R0bnWrpIeN.

Kowal, M., Wildes, R. P., and Derpanis, K. G. Visual concept connectome (vcc): Open world concept discovery and their interlayer connections in deep models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10895–10905, 2024.

Liu, Z., Mao, H., Wu, C.-Y., Feichtenhofer, C., Darrell, T., and Xie, S. A convnet for the 2020s. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 11976–11986, 2022.

Lundberg, S. M. and Lee, S. A unified approach to interpreting model predictions. In Guyon, I., von Luxburg, U., Bengio, S., Wallach, H. M., Fergus, R., Vishwanathan, S. V. N., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pp. 4765–4774, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/8a20a8621978632d76c43dfd28b67767-Abstract.html.

Marshall, S. C. and Kirchner, J. H. Understanding polysemanticity in neural networks through coding theory. *arXiv preprint arXiv:2401.17975*, 2024.

Oikarinen, T. and Weng, T.-W. Linear explanations for individual neurons. In *International Conference on Machine Learning*, 2024.

Oikarinen, T. P. and Weng, T. Clip-dissect: Automatic description of neuron representations in deep vision networks. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. URL https://openreview.net/forum?id=iPWiwWHc1V.

Olah, C., Mordvintsev, A., and Schubert, L. Feature visualization. *Distill*, 2(11):e7, 2017.

Olah, C., Cammarata, N., Schubert, L., Goh, G., Petrov, M., and Carter, S. Zoom in: An introduction to circuits. *Distill*, 5(3):e00024–001, 2020.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., and Sutskever, I. Learning transferable visual models from natural language supervision. In Meila, M. and Zhang, T. (eds.), *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pp. 8748–8763. PMLR, 2021. URL http://proceedings.mlr.press/v139/radford21a.html.

Razzhigaev, A., Mikhalchuk, M., Goncharova, E., Oseledets, I., Dimitrov, D., and Kuznetsov, A. The shape of learning: Anisotropy and intrinsic dimensions in transformer-based models. In *Findings of the Association for Computational Linguistics: EACL 2024*, pp. 868–874, 2024.

Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., and Batra, D. 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.

Shrikumar, A., Greenside, P., and Kundaje, A. Learning important features through propagating activation differences. In *International conference on machine learning*, pp. 3145–3153. PMlR, 2017.

Siméoni, O., Vo, H. V., Seitzer, M., Baldassarre, F., Oquab, M., Jose, C., Khalidov, V., Szafraniec, M., Yi, S., Ramamonjisoa, M., et al. Dinov3. *arXiv preprint arXiv:2508.10104*, 2025.

Simonyan, K., Vedaldi, A., and Zisserman, A. Visualising image classification models and saliency maps. *Deep Inside Convolutional Networks*, 2(2), 2014.

Springenberg, J. T., Dosovitskiy, A., Brox, T., and Riedmiller, M. A. Striving for simplicity: The all convolutional net. In Bengio, Y. and LeCun, Y. (eds.), *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Workshop Track Proceedings*, 2015. URL http://arxiv.org/abs/1412.6806.

Sundararajan, M., Taly, A., and Yan, Q. Axiomatic attribution for deep networks. In Precup, D. and Teh, Y. W. (eds.), *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW,*

Australia, 6-11 August 2017, volume 70 of *Proceedings of Machine Learning Research*, pp. 3319–3328. PMLR, 2017. URL http://proceedings.mlr.press/v70/sundararajan17a.html.

Thasarathan, H., Forsyth, J., Fel, T., Kowal, M., and Derpanis, K. G. Universal sparse autoencoders: Interpretable cross-model concept alignment. In *Forty-second International Conference on Machine Learning*, 2025.

Wang, K., Variengien, A., Conmy, A., Shlegeris, B., and Steinhardt, J. Interpretability in the wild: a circuit for indirect object identification in gpt-2 small. *arXiv preprint arXiv:2211.00593*, 2022.

Yu, W., Wang, Q., Liu, C., Li, D., and Hu, Q. Coe: Chain-of-explanation via automatic visual concept circuit description and polysemanticity quantification. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 4364–4374, 2025.

Zhai, X., Mustafa, B., Kolesnikov, A., and Beyer, L. Sigmoid loss for language image pre-training, 2023. URL https://arxiv.org/abs/2303.15343.

A. Experimental Settings

Models We apply our framework to widely-used models: ViT (Dosovitskiy et al., 2021), ResNet (He et al., 2015), ConvNeXT (Liu et al., 2022), DenseNet (Huang et al., 2017), and CLIP ViT (Radford et al., 2021). All models are pretrained on ImageNet (Deng et al., 2009), except CLIP (pretrained on image-text pairs). We analyzed specific layers across these architectures: all bottlenecks in ResNet-50 residual blocks, various dense blocks in DenseNet, all 12 transformer blocks in ViT-B and CLIP ViT-B's visual encoder, and all blocks across the 4 stages of ConvNeXt.

Baselines We compare our method against three established baselines: PURE (Dreyer et al., 2024), LE (Oikarinen & Weng, 2024), and CPE (Yu et al., 2025). For all baselines and our method, we conduct the analysis using the top-100 highly activated samples for each neuron to ensure a consistent evaluation of local semantics.

PURE is a pioneering approach that virtually disentangles polysemantic neurons using attribution methods. While its underlying Concept Relevance Propagation (CRP) (Achtibat et al., 2023) was originally designed specifically for CNNs, we adapted its implementation for Transformer models to ensure a comprehensive comparison across diverse architectures. Furthermore, PURE originally requires manual specification of the number of concepts. To ensure comparability, we instead automatically determine the optimal number of concepts (K) using the Silhouette Score, searching over the candidate set $\{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 20, 30, 40, 50\}$. **LE** computes sparse coefficients via linear approximation over a predefined concept set across the entire activation range. While we adhere to the original protocol by fitting the model over the full activation spectrum, we restrict the final evaluation to the concepts associated with the highly activated range to ensure a fair comparison. Since LE allows for the assignment of multiple concepts to a single sample, we allocated concepts accordingly for each sample to facilitate direct comparison. **CPE** (Yu et al., 2025) is evaluated following its original Chain-of-Explanation (COE) framework. We use the acronym CPE (Concept Polysemanticity Entropy) in the main text to emphasize its core metric. Concept extraction is performed using gpt-4 (Achiam et al., 2023).

B. Ablation Studies and Sensitivity Analysis

This section evaluates the impact of individual components within our framework, including advanced attribution techniques, threshold, normalization, and different clustering methods.

B.1. Robustness to Attribution Methods

To qualitatively complement the quantitative robustness results presented in the main text, we provide visual comparisons of the concept clusters discovered using different gradient-based attribution methods, including Integrated Gradients (Sundararajan et al., 2017), GradientShap (Lundberg & Lee, 2017), and Guided Backprop (Springenberg et al., 2015).

As illustrated in Figure 9a (ViT-B/16) and Figure 9b (ResNet-50), our proposed approach (using Input $\times$ Gradient) consistently yields visually coherent groupings. For instance, in the ViT-B/16 example, GradientShap tends to produce semantically mixed clusters, whereas SPICE with Input $\times$ Gradient maintains highly cohesive semantic boundaries. Similarly, in the ResNet-50 example, Guided Backprop results in visibly entangled concepts. These qualitative results align with our quantitative findings, confirming that while our default choice of Input × Gradient provides reliable and distinct conceptual disentanglement, the SPICE framework is broadly applicable across various attribution techniques

B.2. Threshold (τ) Sensitivity

In our proposed method, we construct flexible concept clusters tailored to each model and dataset by selecting an appropriate threshold. Specifically, we adopt the larger value between the second peak of the similarity distribution (estimated using KDE) and the score at the top-p$\%$ quantile as the threshold. To examine the sensitivity of this hyperparameter, we varied the value of p and measured its impact on both the number of clusters formed and the within-cluster similarity. The results are shown in Figure 10. We conducted experiments with p = [0.99, 0.95, 0.9, 0.85] on ResNet50 and ViT-Base-16 models. As shown in Figure 10-(a), ViT consistently produced more concept clusters on average compared to ResNet50, which aligns with prior findings that polysemantic neurons are more prevalent and complex in transformer architectures. Interestingly, as the threshold became looser, the number of clusters decreased, whereas exhibited the fewest clusters at p=0.99. This behavior arises because, at such a strict threshold, some clusters fail to form due to the distances between samples in the representation space being larger than the cutoff. Figure 10-(b) reports the within-cluster similarity scores across thresholds, all of which remain significantly high ($>$0.9). A slight downward trend is observed as the threshold decreases, indicating

that looser thresholds admit some noisier samples. Taken together, these results show that p=0.95 provides the best trade-off, reliably producing meaningful clusters across architectures while maintaining the highest within-cluster similarity.

B.3. Effect of Normalization

We investigate the effect of normalization on the quality of the discovered concepts. We apply normalization independently along each attribution dimension, analogous to feature-wise standardization commonly used in time-series analysis. Raw attribution vectors often exhibit significant heterogeneity in scale and variance across different dimensions. Without adjustment, dimensions with inherently larger magnitudes may disproportionately dominate the clustering process, potentially obscuring the semantic structures we aim to identify. By normalizing per dimension, we ensure that the clustering is driven

(a) Neuron #330 in Block 11 of ViT-B/16 under different attribution methods.

(b) Neuron #1540 in Layer 4.2 of ResNet-50 under different attribution methods.

Figure 9. Comparison of concept clusters under different attribution methods for ViT-B/16 and ResNet-50.

by the relative distribution of importance (the semantic pattern) rather than absolute magnitude biases.

Table 3. Model Layer Attribution Normalization and Separability

Model	Layer	Normalization	Separability
ResNet-50	L2.1		1.058
ResNet-50	L2.1	✓	**1.092**
ResNet-50	L4.2		1.017
ResNet-50	L4.2	✓	**1.293**
ViT-B	B2		1.065
ViT-B	B2	✓	**1.137**
ViT-B	B11		1.507
ViT-B	B11	✓	**1.577**

B.4. Choice of Clustering Method

We compare the effectiveness of three different clustering algorithms in generating well-separated concepts: K-Means (our default), ISODATA (Ball & Hall, 1965), and HDBSCAN (Campello et al., 2013). ISODATA adaptively splits and merges samples based on the inter-cluster distance and the intra-cluster variance. To enable a fair comparison while preserving the spirit of our attribution-based clustering method, we adapt ISODATA by performing cluster merging based on our proposed score and cluster splitting based on the silhouette score. We set the splitting threshold to the critical value at which the behavior begins to deviate from our method, and used this setting to examine whether the direction of changes induced by the splitting logic is reasonable.

As illustrated in Figure 11, our proposed method (SPICE) demonstrates superior performance in isolating semantically consistent concepts across various network depths. Unlike ISODATA and HDBSCAN, which often produce noisy or entangled clusters, our approach effectively disentangles distinct visual concepts within a single neuron. Notably, SPICE maintains high semantic coherence for each identified cluster while successfully capturing a diverse range of features, demonstrating its robustness across different layers of the model.

B.5. Correlation between Cluster Count and Separability

To validate our clustering algorithm and address potential concerns regarding over-segmentation, we analyzed the relationship between the number of discovered concepts (k) and their separability scores. We observed no significant correlation (Spearman $\rho = -0.041, p = 0.682$), alongside negligible differences across groups with varying k (ANOVA $p = 0.717$). This lack of correlation is crucial: it implies that SPICE does not artificially over-cluster features to inflate performance. Instead, the consistent separability across varying k confirms that our method adaptively discovers the optimal number of concepts based on the neuron's intrinsic polysemanticity, maintaining high validity for both simple and highly complex neurons.

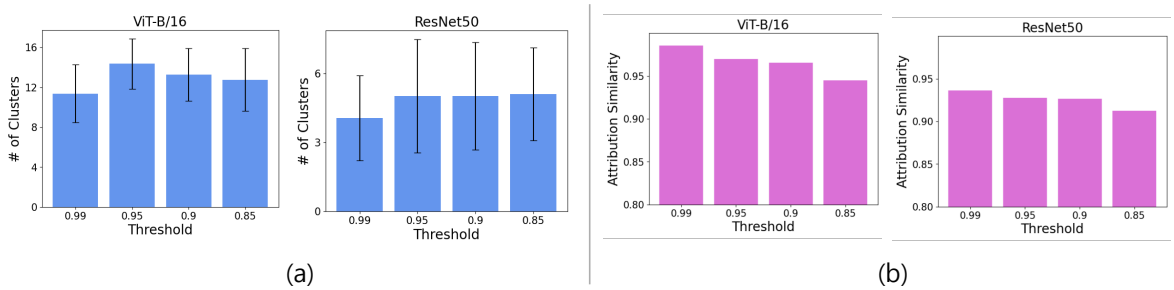

Figure 10. Sensitivity analysis on the threshold of the last layer/block of ViT-B/16 and ResNet-50. Left **(a)** shows the number of concept clusters, and right **(b)** shows the attribution similarity of concept clusters at different thresholds.

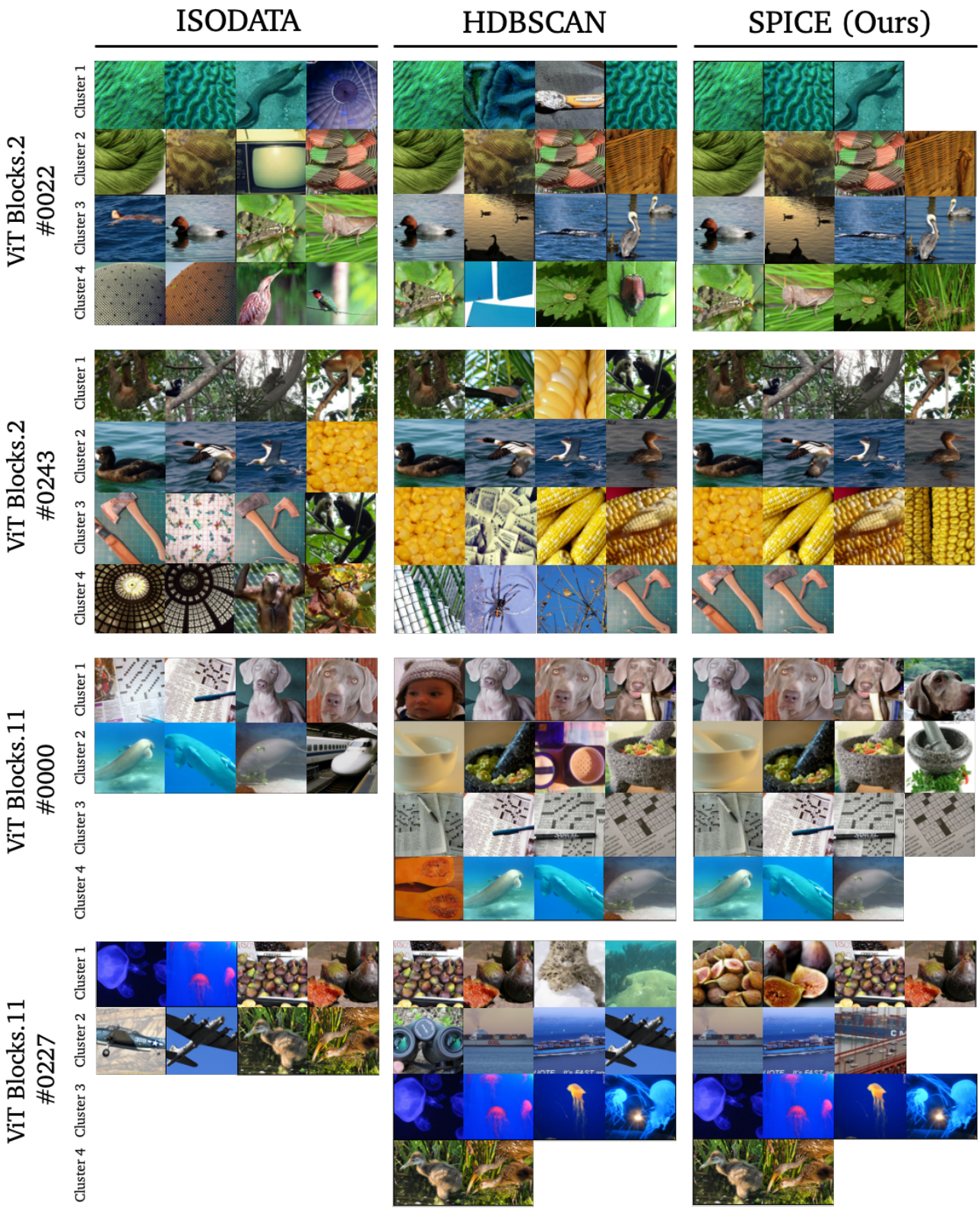

Figure 11. Ablation study of the clustering components in our method, comparing the use of ISODATA, HDBSCAN, and K-Means for computing cluster means. The last column shows the default version of our algorithm, which employs iterative K-Means.

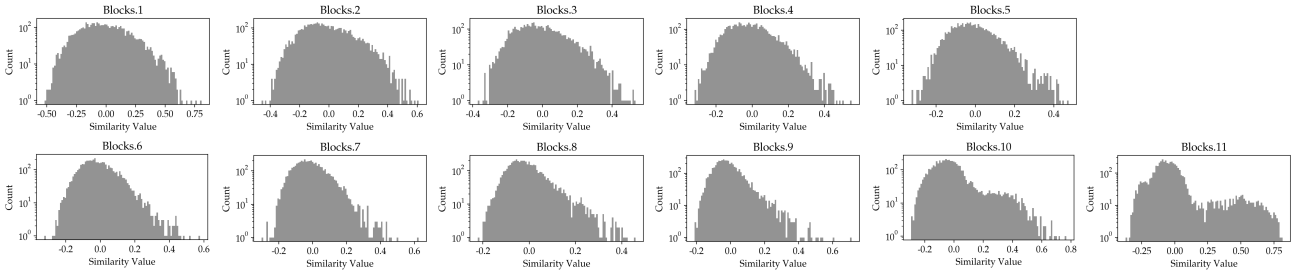

Figure 12. Changes in the distribution of attribution similarity across ViT-B's blocks, shown from the first block (top-left) to the last block (bottom-right).

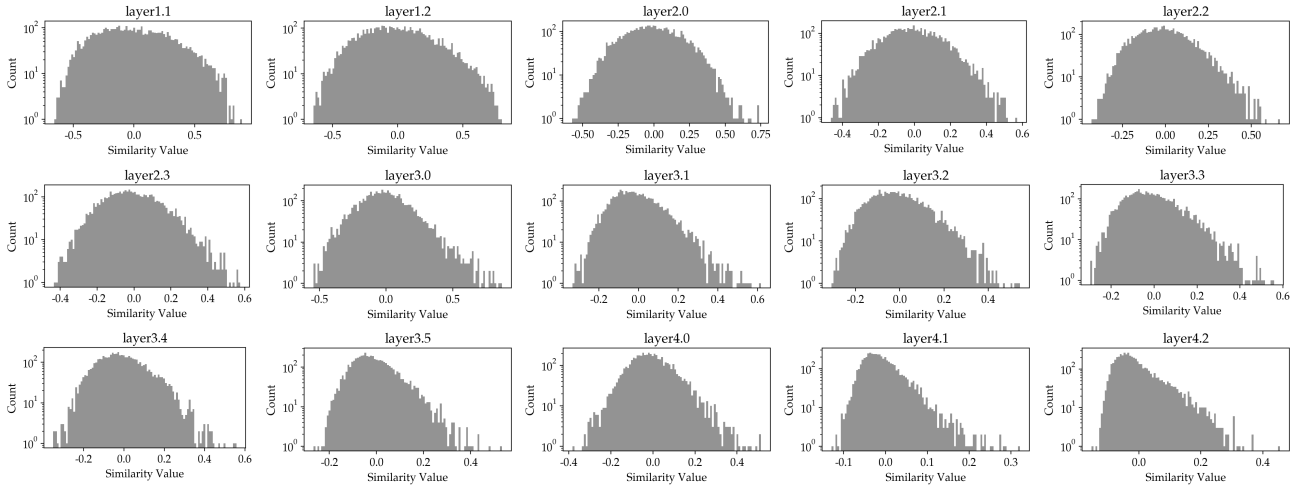

Figure 13. Changes in the distribution of attribution similarity across ResNet-50's layers, shown from the first layer (top-left) to the last layer (bottom-right).

C. Structural Evolution of Concepts Across Network Depth

In this section, we provide a detailed analysis of how the structure of attribution similarity evolves across layers, justifying the necessity of our dual-criterion thresholding strategy defined in Section 3 in the main text. Figures 12 and 13 visualize these distributions for ViT-Base and ResNet-50, respectively, while Table 4 quantifies their unimodality using Hartigan's Dip Test (Hartigan & Hartigan, 1985). As shown in Figure 12, ViT-Base exhibits a clear transition. Early layers are robustly unimodal, but deeper layers (10–11) begin to develop a bimodal structure for a significant subset of neurons, forming a distinct secondary peak in the high-similarity region. This is statistically captured by the Dip Test p-values in Table 4; while the averages drop to 0.60 and 0.28, the accompanying high variance ($\pm 0.42, \pm 0.40$) is highly indicative. Mathematically, this broad distribution implies that rather than every neuron becoming universally multimodal, a substantial subpopulation of neurons clearly diverges from strict unimodality, signaling the emergence of highly compressed, polysemantic concepts. In contrast, ResNet-50 remains consistently unimodal ($p > 0.9$) across all depths. While deeper layers in ResNet become right-skewed, they do not form a separable secondary mode.

These observations highlight why a single thresholding method is insufficient. For bimodal distributions (e.g., deep ViT layers), a fixed percentile ignores the true distribution of the data. Instead, the local maximum (the second peak) identified by KDE pinpoints the exact center of a semantically cohesive cluster. Since we measure cosine similarity, leveraging this high-similarity peak as the threshold ensures that only the most strongly related pathways are grouped, leaving out the noisy middle ground. Conversely, for unimodal distributions (e.g., ResNet or early ViT), the absence of a secondary peak makes KDE thresholding invalid. To dynamically accommodate both structural behaviors, we set the threshold as $\tau = \max(\text{95th percentile}, \text{KDE 2nd peak})$. This selectively applies strict, peak-driven thresholding for highly polysemantic layers while safely defaulting to a 95th-percentile baseline for unimodal spaces.

Table 4. Comparison of distribution statistics between ViT-Base and ResNet-50. ResNet-50 shows consistently high Hartigan's Dip Test p-values (> 0.9), indicating stable unimodal distributions across all depths. In contrast, ViT-Base exhibits a precipitous drop in mean p-values accompanied by exceptionally high variance in its final layers (**bold**). This broad distribution indicates the emergence of a substantial subpopulation of highly multimodal (polysemantic) neurons, rather than a universal shift across the entire layer. Depth is measured in a block-wise or layer-wise manner, depending on the model architecture.

Depth	ViT-Base	ResNet50
Layer 1	0.98 ± 0.06	0.95 ± 0.17
Layer 2	0.98 ± 0.07	0.97 ± 0.13
Layer 3	0.98 ± 0.09	0.96 ± 0.11
Layer 4	0.98 ± 0.06	0.97 ± 0.12
Layer 5	0.97 ± 0.09	0.97 ± 0.08
Layer 6	0.98 ± 0.06	0.97 ± 0.08
Layer 7	0.99 ± 0.06	0.97 ± 0.11
Layer 8	0.98 ± 0.08	0.98 ± 0.08
Layer 9	0.94 ± 0.19	0.97 ± 0.09
Layer 10	$\mathbf{0.60 \pm 0.42}$	0.97 ± 0.09
Layer 11	$\mathbf{0.28 \pm 0.40}$	0.97 ± 0.10
Layer 12	—	0.97 ± 0.10
Layer 13	—	0.96 ± 0.16
Layer 14	—	0.97 ± 0.12
Layer 15	—	0.98 ± 0.09

D. Running Time Analysis

In this section, we provide a formal derivation of the theoretical worst-case time complexity of our iterative clustering algorithm and contrast it with the empirical running time observed in practice.

D.1. Theoretical Worst-Case Complexity

Each iteration of the algorithm involves K-Means clustering, which scales as $\mathcal{O}(k \cdot N \cdot d)$, and pairwise similarity validation, which incurs a cost of $\mathcal{O}(N^2)$. Treating the feature dimension d as a constant, the per-iteration computational cost is dominated by the $\mathcal{O}(N^2)$ validation step. The theoretical worst-case scenario occurs if the algorithm fails to satisfy the cohesion threshold at every step, causing the number of search clusters k to increment linearly up to N without reducing the working set. Summing the $\mathcal{O}(N^2)$ cost across $\mathcal{O}(N)$ iterations results in a theoretical upper bound of $\mathcal{O}(N^3)$.

D.2. Empirical Analysis

Despite the cubic theoretical bound, the worst-case scenario (complete rejection of clusters) rarely occurs in real-world experiments. This is primarily because the input samples are derived from the high activations of a single neuron, which inherently exhibit strong semantic coherence. Since these samples tend to cluster tightly around distinct concepts in the feature space, the algorithm can rapidly identify and accept valid groups satisfying the similarity threshold. This prevents the pathological case of repeated rejections and allows the algorithm to efficiently prune the search space.

To verify this practical scalability, we measured the wall-clock time across varying sample sizes N. Table 5 reports the observed time and a complexity check metric, $T/(N \log_{10} N)$. As shown in the table, this metric stabilizes around 4.0×10^{-4} for $N \geq 1,000$. This stability confirms that, aided by the inherent structure of neuron activations, the algorithm follows an effective complexity of $\mathcal{O}(N \log N)$.

E. Training Dynamics and Transfer Learning

To investigate the evolution of feature representations during transfer learning, we fine-tuned a ViT-Large (ViT-L) model pre-trained on the LVD-1689M dataset using the DINOv3 (Siméoni et al., 2025) framework. The downstream task was performed on the Food-101 dataset (Bossard et al., 2014). To ensure parameter efficiency and focus on the adaptation of high-level semantic features, we updated only the Multi-Layer Perceptron (MLP) weights of the final Transformer block and

Table 5. Computational Efficiency of Iterative Divisive Clustering

Sample Size (N)	**Observed Time (s)**	**Complexity Check** (Time/$(N \log_{10} N)$)
100	1.45	0.003149
200	4.63	0.004369
500	1.89	0.000608
1,000	3.01	0.000436
2,000	5.94	0.000391
5,000	17.71	0.000416

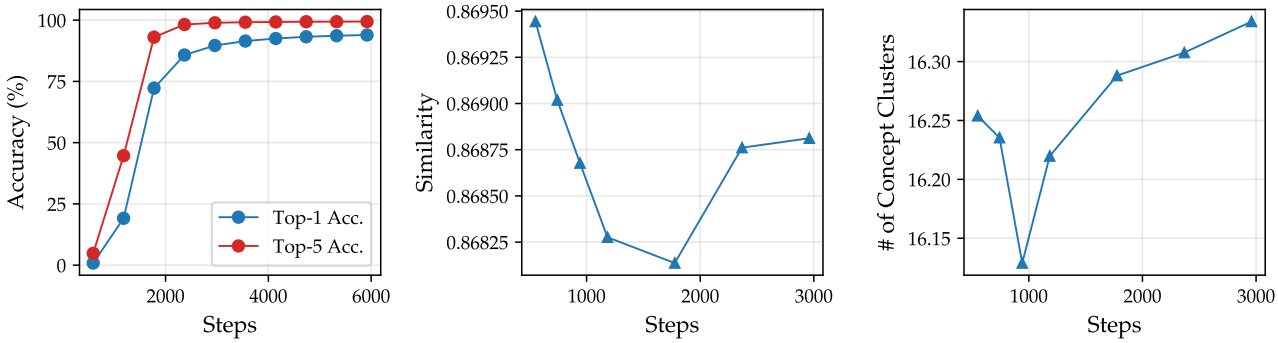

Figure 14. Training accuracy plotted against training steps during transfer learning of ViT-L. The left panel displays Top-1 and Top-5 classification accuracy. The middle and right panels illustrate the attribution similarity and the average number of concept clusters, respectively.

the linear classification head. The model was trained for 6 epochs. To capture the rapid dynamic changes characteristic of the early training phase, metrics were logged every 50 steps during the first two epochs.

We begin our analysis of transfer-learning dynamics by examining how individual feature representations evolve during fine-tuning. Specifically, we first focus on neuron-level semantic changes to understand how pre-trained visual concepts adapt to the new task domain. Figure 15 presents a representative example—Neuron #550—which exhibited the most extensive concept expansion during fine-tuning. Initially sensitive to visual attributes such as the green color of *Edamame* or the brown, textured appearance of *Pancakes*, this neuron gradually broadened its activations to include semantically distinct yet visually similar categories such as *Greek Salad*, *Guacamole*, and *Falafel*. This case illustrates how transfer learning often repurposes existing filters to cover new visual clusters rather than forming entirely new features.

The global training behaviors are summarized in Figure 14, which reports accuracy metrics, attribution similarity, and the evolution of concept clusters. When we assigned representative concepts to each neuron based on the majority class labels of the activating samples, we found that, across the 1,024 neurons in the targeted MLP layer, 284 neurons (approximately 28%) preserved their original concept clusters between the initial and final checkpoints. This substantial stability indicates that DINOv3 pre-training yields robust visual features that remain largely suitable for the downstream task. Given that the feature dimensionality (1,024) far exceeds the number of target classes (101), the model retains a large portion of its

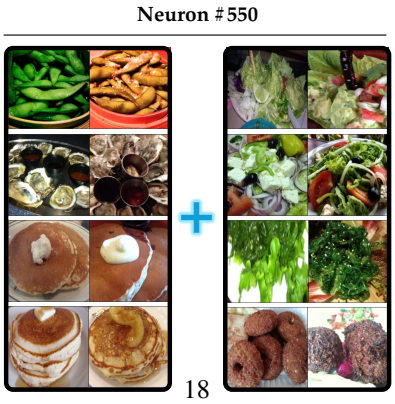

Figure 15. Visualization of concept expansion in Neuron #550. The left panel shows the top-activating concepts at initialization, while the right panel shows the new concepts added after fine-tuning.

pre-trained representational space rather than reorganizing it wholesale.

F. Diverse Models and Layers Analysis

We extend the analysis to ensure our findings are not architecture-specific. Figures 16, 18, and 17 demonstrate that our method produces semantically coherent concepts across different feature abstraction levels (early, middle, and late layers). The consistent improvement observed across this diverse set of backbones confirms the general applicability of our framework in enhancing interpretability for modern deep learning architectures.

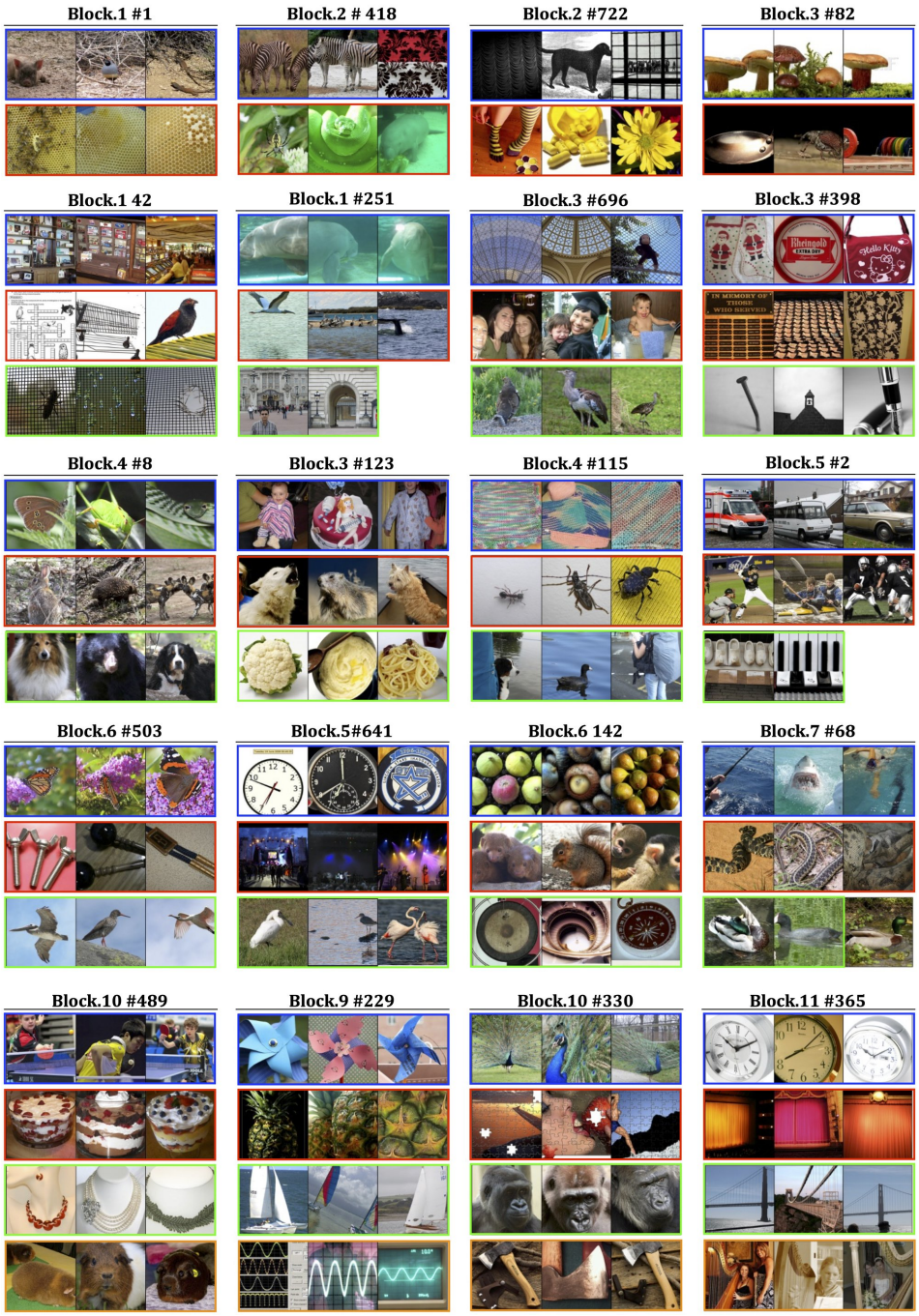

Figure 16. Examples of polysemantic neurons of ViT-B with different blocks. Each row in a neuron represents a distinct concept cluster.

Figure 17. Examples of polysemantic neurons of ResNet-50 with different blocks. Each row in a neuron represents a distinct concept cluster.

Figure 18. Examples of polysemantic neurons of ConvNeXt-B with different blocks. Each row in a neuron represents a distinct concept cluster.