Can Cross-Layer Transcoders Replace Vision Transformer Activations? An Interpretable Perspective on Vision

Gerasimos Chatzoudis[1] Konstantinos D. Polyzos[2] Zhuowei Li[1*]
Difei Gu[1] Gemma E. Moran[1] Hao Wang[1] Dimitris N. Metaxas[1]

[1]Rutgers University [2]University of California San Diego

gc745@scarletmail.rutgers.edu kpolyzos@ucsd.edu zl502@cs.rutgers.edu
difei.gu@rutgers.edu gm845@stat.rutgers.edu hw488@cs.rutgers.edu dnm@cs.rutgers.edu

Abstract

Understanding the internal activations of Vision Transformers (ViTs) is critical for building interpretable and trustworthy models. While Sparse Autoencoders (SAEs) have been used to extract human-interpretable features, they operate on individual layers and fail to capture the cross-layer computational structure of Transformers, as well as the relative significance of each layer in forming the last-layer representation. Alternatively, we introduce the adoption of Cross-Layer Transcoders (CLTs) as reliable, sparse, and depth-aware proxy models for MLP blocks in ViTs. CLTs use an encoder–decoder scheme to reconstruct each post-MLP activation from learned sparse embeddings of preceding layers, yielding a linear decomposition that transforms the final representation of ViTs from an opaque embedding into an additive, layer-resolved construction that enables faithful attribution and process-level interpretability. We train CLTs on CLIP ViT-B/32 and ViT-B/16 across CIFAR-100, COCO, and ImageNet-100. We show that CLTs achieve high reconstruction fidelity with post-MLP activations while preserving and even improving, in some cases, CLIP zero-shot classification accuracy. In terms of interpretability, we show that the cross-layer contribution scores provide faithful attribution, revealing that the final representation is concentrated in a smaller set of dominant layer-wise terms whose removal degrades performance and whose retention largely preserves it. These results showcase the significance of adopting CLTs as an alternative interpretable proxy of ViTs in the vision domain.

1. Introduction

Foundation Models (FMs) have shown strong generalization performance in diverse tasks ranging from classification to open-ended generation [1, 18, 21, 28, 36]. Despite their empirical success, these models remain largely opaque, limiting interpretability as well as controllability and reliability in practice. Particularly on the vision domain, understanding the internal representations of the widely-used Vision Transformers (ViTs) remains an open challenge for building interpretable, controllable, and verifiable models. Although existing approaches, including Sparse Autoencoders [5, 19, 32] have aimed to show that transformer activations can be organized into human-interpretable components, they operate *locally*, learning features only within a single layer. Consequently, they fall short in capturing cross-layer interdependencies or the relative contribution of each layer to the final-layer representation, providing limited insight into how information is transformed *across* network depth.

Aiming to advance interpretable vision models and inspired by their use in the language domain, we propose a novel interpretable perspective on vision by leveraging Cross-Layer Transcoders (CLTs). Specifically, our goal is to judiciously integrate CLTs as an alternative framework for analyzing the internal structure of transformer representations within Vision Transformers (ViTs). CLTs aim to reconstruct the post-MLP activations at a target layer from sparse features computed in earlier layers enabling a directed, cross-layer decomposition of each representation. This allows for a proper assessment of the contribution of each source layer to a given target layer. When CLTs are accurate enough to replace the original MLP blocks, the resulting CLT-based 'replacement' model can serve as a reliable proxy for analyzing the original transformer without compromising the ViT model performance on the downstream task.

While CLTs have been used to identify attribution graphs in Large Language Models (LLMs) [2], to the best of our knowledge their proper adoption for vision models as interpretable alternatives has not been explored yet. Unlike autoregressive language models in the language domain, where all tokens share a uniform sequential structure, ViTs

*Work done outside of Amazon.

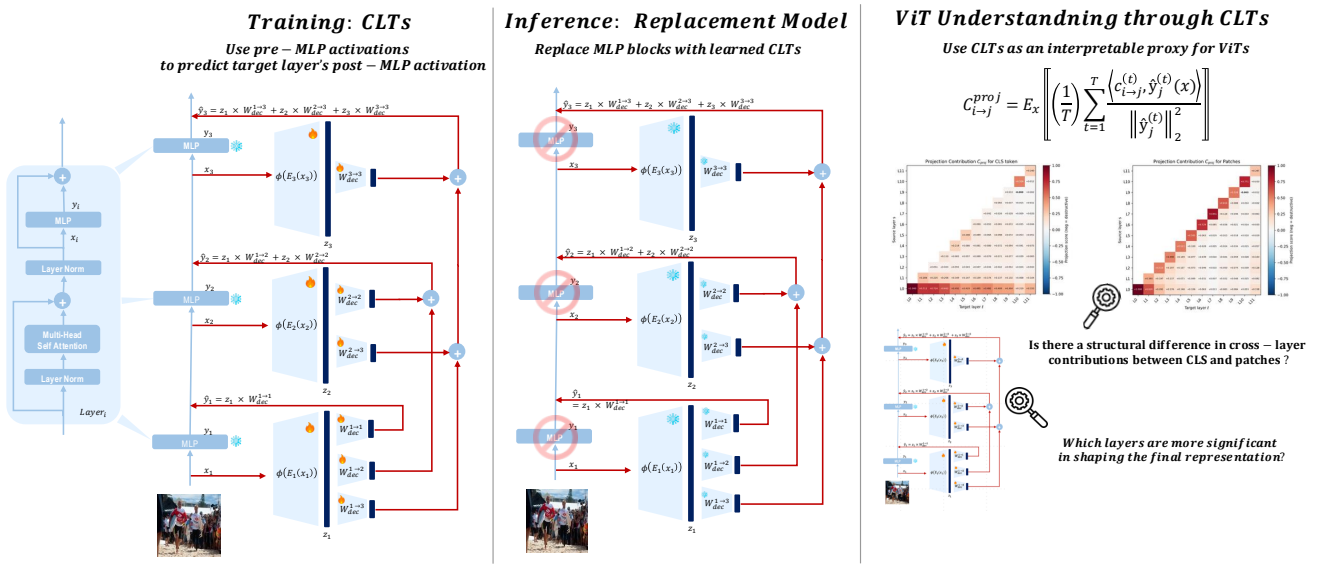


Figure 1. Overview of the Cross-Layer Transcoder (CLT) framework. **Left**: Each CLT encodes LN2 activations into sparse codes z_ℓ and reconstructs MLP outputs y_ℓ via triangular decoders. **Middle**: During inference, CLTs replace MLPs across layers, preserving zero-shot performance. **Right**: Using CLTs as an interpretable proxy for ViTs to understand the cross-layer contributions of different token types and identify the most significant layers in shaping the ViT's final representations.

operate over spatially structured patch tokens with additional degrees of freedom, including varying patch granularity, two-dimensional spatial dependencies, and, in some ViTs, a global `[CLS]` token that serves a fundamentally different computational role. These fundamental differences of the language and vision domains naturally render the practical utility of CLTs to the vision domain a distinct and nontrivial question that is yet to be explored. To that end, in this paper we investigate the practical utility and benefits of CLTs in ViTs by addressing the following open research questions:

RQ1: Functional Replacement. *Can CLTs functionally replace MLP blocks in Vision Transformers as an alternative interpretable proxy*? We aim to evaluate whether CLTs can faithfully reconstruct post-MLP activations from sparse features of earlier layers, and whether such replacements preserve downstream classification performance under different sparsity schemes and replacement strategies within CLTs.

RQ2: Interpretable Cross-Layer Contribution. *Can CLTs provide faithful attribution as well as process-level interpretability?* We aim to investigate whether the sparse, depth-ordered structure of CLTs supports meaningful interpretation of internal ViT representations by quantifying layer-wise influence using cross-layer contribution scores. Specifically, our goal is to provide both quantitative and example-based evidence of how different layers shape the final representation, yielding interpretable cross-layer contributions.

Driven by these open research questions, the contributions of the present work can be summarized as follows:

- **CLTs for Vision.** We investigate the effectiveness of Cross-Layer Transcoders in Vision Transformers, enabling sparse, cross-layer reconstruction of post-MLP activations from earlier layers.
- **Accurate MLP Replacement in key regimes.** CLTs can replace MLP blocks, especially in later layers or for [CLS] tokens across all layers, preserving and even improving in some cases zero-shot classification performance. Consequently, CLTs serve as a reliable proxy for the original ViT to investigate how earlier layers contribute across depth of the model.
- **Interpretability via Cross-Layer Attribution.** Projection-based contribution scores show that patch tokens exhibit strongly diagonal-dominant attribution, with each layer primarily explaining its own post-MLP output, while the `[CLS]` token draws credit broadly across depth, aggregating information from many preceding layers.
- **Faithful attribution reveals concentrated credit across depth.** Ablation experiments demonstrate that the final-layer representation is predominantly shaped by a small subset of source layers: retaining only the top-4 out of 12 layers recovers original model's accuracy, while removing the single highest-scored layer causes substantial degradation. This corroborates that the projection-based scores faithfully identify the layers that are most significant in forming the output representation.

2. Related Work

Mechanistic Interpretability and Sparse Autoencoders. A wide range of methods have been proposed to interpret vision models, including feature visualization [26, 31, 37] and network dissection [3, 25]. More recently, Mechanistic Interpretability has emerged as a systematic approach to analyzing and understanding neural networks [11, 27]. A central challenge in this area is the presence of polysemantic neurons, i.e., units that respond to multiple, seemingly unrelated inputs, arising from superposition. Superposition is the phenomenon where networks encode more features than the available dimensions allow, forcing different concepts to share the same activations [12]. Recent efforts have leveraged SAEs to uncover interpretable features within LLMs [8, 14, 34]. Sparse Autoencoders (SAEs) have been employed to alleviate superposition by learning sparse, overcomplete representations of internal model activations via dictionary learning [5, 30].

Although SAEs have been predominantly studied in the context of language models, recent efforts explore SAEs' utility in the vision domain. In particular, SAEs have been applied to feature analysis, generative modeling, and concept disentanglement in the visual domain [4, 13, 32, 33, 35] . Much of this work has centered on interpretability, revealing latent structure in deep visual representations. More recent studies have extended these insights to practical downstream applications, including the use of sparse features to analyze CLIP's susceptibility to typographic attacks [17], improve classification accuracy via class-conditioned latent masking and feature selection [19], and guide model predictions through visual sparse steering [6].

Feature Circuits and Cross-Layer Transcoders Sparse Autoencoders have also been applied to uncover feature circuits, i.e., directed graphs that capture how interpretable features activate, interact, and causally contribute to a model's final output [7, 8, 15, 22]. In parallel to Sparse Autoencoders, recent work has investigated more structured sparsity-based models for interpretability. Transcoders, for example, are sparse, feedforward modules that operate within a single layer, reconstructing post-MLP activations from pre-MLP inputs using learned dictionaries [10].

While Transcoders focus on modeling transformations within a layer, other approaches extend this idea across layers to capture cross-layer causal structure. Specifically, Crosscoders generalize this idea by learning sparse codes that jointly reconstruct representations across multiple layers, or even across different models, enabling applications such as feature sharing and model diffing [20, 23, 24]. Cross-Layer Transcoders (CLTs) represent a specialized instance of Crosscoders, forming a triangular, depth-ordered architecture that reconstructs each MLP output from the sparse codes of all preceding layers [2]. CLTs have been applied in LLMs to construct attribution graphs and replacement models that reveal how early-layer features influence downstream computations [2]. Our work explores the effectiveness of CLTs in the context of Vision Transformers (ViTs), presenting a CLT-based model for ViT activations and investigating both functional replacement and visual interpretability.

3. Method

3.1. Cross-Layer Transcoder (CLT)

Transformer Preliminaries. Let a Vision Transformer (ViT) with L layers operate on T tokens of width d. In layer i, let $x_i \in \mathbb{R}^{T\times d}$ denote the post-attention (LN2), pre-MLP activations and let the MLP produce $y_i = \mathrm{MLP}_i(x_i) \in \mathbb{R}^{T\times d}$. We target these y_i for linear reconstruction using sparse features extracted from *earlier* layers, enabling both attribution across depth and drop-in replacement of MLPs.

Sparse Feature Encoding (per Layer). Each layer i has a learned linear encoder with the weights $E_i \in \mathbb{R}^{d\times m}$ (typically $m > d$, overcomplete). We denote the embedding of token $t \in \{1, \dots, T\}$ as $x_{i,t} \in \mathbb{R}^d$. For each token, we map $x_{i,t}$ to a sparse feature vector

$$z_{i,t} = \phi(x_{i,t}E_i) \in \mathbb{R}^m \tag{1}$$

where $\phi(\cdot)$ is a non-linear function operating on the sparse feature space. We support three non-linear functions $\phi(\cdot)$:

- **JumpReLU (learned thresholds)** [29]: for each token,

$$z_{i,t} = (x_{i,t}E_i) \odot \mathbf{1}\big[x_{i,t}E_i > \tau_i\big] \tag{2}$$

  where $\tau_i \in \mathbb{R}^m_{\geq 0}$ are learned per-feature thresholds shared across tokens. The thresholds τ_i are learned jointly with the encoders and decoders using a straight-through estimator.
- **ReLU-Top-k** [14]: for each token,

$$z_{i,t} = \mathrm{TopK}\big(\max(0, x_{i,t}E_i),\ k\big) \tag{3}$$

  which keeps the top-k positive features (by value) per token and sets all others to zero.
- **Abs-Top-k**, similar to [38]: for each token, let $u_{i,t} = x_{i,t}E_i$, and define

$$z_{i,t} = \mathrm{sign}(u_{i,t}) \odot \mathrm{TopK}\big(|u_{i,t}|,\ k\big) \tag{4}$$

  which keeps the top-k features by absolute value per token while preserving their sign.

Cross-Layer Transport. For each destination layer j, we reconstruct the MLP output from all sources $i \leq j$:

$$\hat{y}_j = \sum_{i=1}^{j} z_i W_{\text{dec}}^{i\to j}, \qquad W_{\text{dec}}^{i\to j} \in \mathbb{R}^{m\times d} \tag{5}$$

Here $z_i \in \mathbb{R}^{T\times m}$ denotes the matrix of token-wise sparse codes at layer i, whose t-th row is the per-token code $z_{i,t}^\top$. This additive, token-wise reconstruction enforces a strictly causal structure: each target layer j is reconstructed only from activations in source layers $i \le j$, thereby preventing information leakage from future layers. The resulting triangular decoder architecture naturally defines a directed attribution graph across depth, where each edge $i \to j$ corresponds to a learned contribution from layer i toward reconstructing y_j.

Training Objective. Given supervision pairs $\{(x_j, y_j)\}_{j=1}^L$, we minimize the total reconstruction loss:

$$\mathcal{L} = \sum_{\ell=1}^{L} \|\hat{y}_\ell - y_\ell\|_2^2 + \lambda \sum_{\ell=1}^{L} \mathcal{R}_{\text{sparse}}(z_\ell) \tag{6}$$

where $\hat{y}_\ell$ is the CLT reconstruction at layer ℓ and z_ℓ are the feature activations at that layer. The sparsity term $\mathcal{R}_{\text{sparse}}(z_\ell)$ depends on the choice of the non-linear function $\phi(\cdot)$:

- **JumpReLU**: Following the CLT setup in [2], we use a decoder-norm weighted Tanh sparsity penalty applied per feature. Let $z_\ell \in \mathbb{R}^{B\times T\times m_\ell}$ denote the activations at layer ℓ, where B is the batch size, T the number of tokens per example, and m_ℓ the number of features at that layer. For each feature j, we define $W_{\text{dec},j}^{(\ell)}$ as the concatenation of all decoder vectors that read from feature j at layer ℓ. Our sparsity term is then:

$$\mathcal{R}_{\text{sparse}}(z_\ell) = \mathbb{E}_{b,t,j}\left[\tanh\left(c\,\|W_{\text{dec},j}^{(\ell)}\|_2 \cdot |z_{\ell,b,t,j}|\right)\right] \tag{7}$$

  where $c > 0$ is a hyperparameter controlling the sharpness of the penalty and $\mathbb{E}_{b,t,j}[\cdot]$ denotes the empirical average over all batch, token, and feature indices. This encourages features with large decoder norm $\|W_{\text{dec},j}^{(\ell)}\|_2$ to be used more sparingly, while the $\tanh(\cdot)$ keeps the penalty bounded and provides smooth gradients.
- **ReLU Top-k** and **Abs Top-k**: No additional regularization is used, as sparsity is enforced directly by the Top-K operator.

3.2. Cross-Layer Attribution

The CLT yields a faithful decomposition of each target layer's representation as a linear sum of contributions from all preceding layers:

$$\hat{y}_j = \sum_{i\le j} c_{i\to j}, \qquad c_{i\to j} = z_i\, W_{\text{dec}}^{i\to j}, \tag{8}$$

where $c_{i\to j} \in \mathbb{R}^{T\times D}$ is the decoded contribution from source layer i toward reconstructing the MLP output at target layer j, with T denoting the number of tokens and D the hidden dimension.

A natural question is which source layers are most responsible for shaping the reconstruction at each target layer. To quantify this, we project each layer's contribution onto the full reconstruction, yielding a per-token, per-layer attribution score that measures the fraction of the output representation explained by each source layer:

$$C_{i\to j}^{\text{proj}} = \mathbb{E}_x\left[\frac{1}{T}\sum_{t=1}^{T} \frac{\langle c_{i\to j}^{(t)}(x),\ \hat{y}_j^{(t)}(x)\rangle}{\|\hat{y}_j^{(t)}(x)\|_2^2}\right], \tag{9}$$

where $c_{i\to j}^{(t)}, \hat{y}_j^{(t)} \in \mathbb{R}^D$ denote the contribution and reconstruction vectors at token t, and $\langle\cdot,\cdot\rangle$ is the inner product over the feature dimension. By construction, the per-source scores decompose each layer's output into signed additive contributions from all preceding layers, providing an attribution over depth. Scores close to zero indicate source layers with negligible influence on the target, while negative scores identify layers whose contributions partially oppose the aggregate reconstruction, a form of inter-layer redundancy that the CLT makes explicitly visible. This decomposition enables us to inspect, at any token and target layer, how the representation is assembled across the network's depth.

4. Replacing MLP blocks with Cross-Layer Transcoders

Cross-layer Transcoders aim to reconstruct the representations of a target layer from the pre-MLP representations of previous layers. Reconstructing the target layer representations from earlier ones allows us to construct a *Replacement Model*, where the MLP block outputs are replaced by their corresponding CLT approximations. Cross-Layer Transcoders can then be used as a *proxy* to investigate the original model's internal representations. In this section, we first aim to train Cross-Layer Transcoders for Vision Transformers and explore whether we can faithfully reconstruct late-layer post-MLP representations. We then examine how this replacement affects the model's performance on downstream classification tasks.

4.1. Reconstruction of MLP Representations

We train Cross-Layer Transcoders (CLTs) on CIFAR-100, COCO, and IMAGENET-100 using CLIP ViT-B/32 and ViT-B/16 backbones. For each model–dataset pair, we instantiate three sparsity variants: JUMPRELU, RELU-TOP-k, and ABS-TOP-k, with k=128. CLTs are trained on *all tokens* for 10 epochs with an expansion factor of 16, using AdamW with learning rate 2×10^{-4}. Additional training details are provided in the Supplementary material.

To evaluate how faithfully CLTs approximate post-MLP representations across depth, we compare, for each layer ℓ, the reconstructed output $\hat{y}_\ell$ to the teacher activation y_ℓ at every token and compute Mean Squared Error (MSE), R^2,

Figure 2. Reconstruction performance of Cross-Layer Transcoders (CLTs) across transformer layers on CIFAR-100 using ViT-B/32. We compare three sparsity variants, i.e., JUMPRELU, RELU-TOP-k, and ABS-TOP-k (with k=128) using (left) cosine similarity, (center) mean squared error (MSE per token, log scale), and (right) variance explained (R^2). Metrics are averaged over all tokens in the test set.

and cosine similarity. For each model–dataset pair, these quantities are first averaged over tokens and layers. Table 1 summarizes the reconstruction quality for CLIP ViT-B/32 and ViT-B/16 on CIFAR-100, COCO, and IMAGENET-100, comparing the three sparsity variants: JUMPRELU, RELU-TOP-k, and ABS-TOP-k.

Across all settings, CLTs achieve strong reconstruction fidelity with the teacher representations: layer-averaged cosine similarities lie in the 0.92–0.97 range and R^2 values are typically between 0.89 and 0.95, indicating that most of the variance in post-MLP activations is captured by CLTs. ViT-B/16 is consistently easier to approximate than ViT-B/32 (e.g., on CIFAR-100, the per-layer R^2 improves from roughly 0.89–0.91 to 0.92–0.94), and on B/32 backbones RELU-TOP-k enjoys the lowest MSE and highest cosine.

Remark. We hypothesize that the reconstruction gap between the two backbones stems from patch granularity: ViT-B/16 uses smaller patches, distributing visual information across a larger token set and yielding simpler per-token activations, whereas ViT-B/32 compresses the same image into fewer, larger patches, producing richer per-token representations that are inherently harder to approximate.

To further examine how reconstruction quality varies with depth, Figure 2 plots per-layer MSE, R^2, and cosine similarity for ViT-B/32 on CIFAR-100. We observe a clear depth-dependent pattern: very early layers and the final layer are reconstructed almost perfectly (e.g., for RELU-TOP-k, $R^2 \geq 0.96$ and cosine ≥ 0.98 at layers 0 and 11), while intermediate blocks incur higher error (with R^2 dipping to $\approx$0.86 and cosine to $\approx$0.92 around layers 7–9). Among sparsifiers, RELU-TOP-k is consistently strongest in the mid–late regime, whereas JUMPRELU and ABS-TOP-k exhibit sharper degradation in the deepest middle layers (e.g., R^2 near 0.80–0.84 and slightly lower cosine).

4.2. Cascaded Replacement Models

In the previous section, we showed that CLTs can faithfully approximate post-MLP activations across transformer depth. We now ask a functional question: *can CLTs replace MLPs without degrading the ViT's zero-shot classification performance?* In the baseline ViT, block ℓ computes $y_\ell = \mathrm{MLP}_\ell(x_\ell)$ and updates the representation via the residual connection $x_{\ell+1} = x_\ell + y_\ell$. Within a contiguous replacement range $\ell_1 \to \ell_2$, we substitute MLP outputs with CLT reconstructions:

$$y'_\ell = \begin{cases} \hat{y}_\ell, & \ell \in [\ell_1, \ell_2] \\ y_\ell, & \text{otherwise} \end{cases}, \qquad x'_{\ell+1} = x'_\ell + y'_\ell \tag{10}$$

with $x'_0 = x_0$ and

$$\hat{y}_\ell = \sum_{i \leq \ell} z'_i \, W_{\mathrm{dec}}^{i \to \ell}, \qquad z'_i = \phi(x'_i \, E_i) \tag{11}$$

where each modified output y'_ℓ propagates forward through the residual connection, allowing reconstruction errors to compound across depth. We refer to this procedure as *Cascaded Replacement*, and use the $\ell_1 \to \ell_2$ notation notation throughout to denote the contiguous block of substituted layers. In particular, we progressively widen the replacement window toward the final block, evaluating how faithfully CLTs preserve zero-shot accuracy as substitution depth grows.

Table 1. Reconstruction performance of CLTs, averaged over all post-MLP layers across datasets and backbones. We report Mean Squared Error (MSE ↓), R^2 (↑), and Cosine similarity (↑) for different sparsity variants.

Dataset	Backbone	Sparsity	MSE ↓	R^2 ↑	Cosine ↑
CIFAR-100	ViT-B/32	JUMPRELU	0.0032	0.8894	0.9411
		RELU-TOP-k	**0.0024**	**0.9099**	**0.9525**
		ABS-TOP-k	0.0030	0.8893	0.9418
	ViT-B/16	JUMPRELU	0.0018	0.9366	0.9571
		RELU-TOP-k	**0.0017**	**0.9369**	**0.9564**
		ABS-TOP-k	0.0022	0.9202	0.9464
COCO	ViT-B/32	JUMPRELU	0.0026	0.8971	0.9456
		RELU-TOP-k	**0.0018**	**0.9119**	**0.9542**
		ABS-TOP-k	0.0028	0.8934	0.9442
	ViT-B/16	JUMPRELU	**0.0011**	**0.9541**	**0.9715**
		RELU-TOP-k	0.0012	0.9375	0.9609
		ABS-TOP-k	0.0026	0.8945	0.9336
IMAGENET-100	ViT-B/32	JUMPRELU	0.0026	0.8976	0.9457
		RELU-TOP-k	**0.0022**	**0.8995**	**0.9473**
		ABS-TOP-k	0.0036	0.8504	0.9209
	ViT-B/16	JUMPRELU	**0.0011**	**0.9505**	**0.9691**
		RELU-TOP-k	0.0012	0.9450	0.9654
		ABS-TOP-k	0.0024	0.9081	0.9421

Table 2. Top-1 zero-shot accuracy (%) for CLIP ViT-B/32 with ReLU-Top-k sparsity (k=128) under cascaded MLP replacement.

Dataset	Model	Tokens	Baseline	3→11	7→11	10→11	0→11
CIFAR-100	Baseline		61.65				
	Transcoders	CLS		61.41	61.37	61.29	61.40
	CLTs	CLS		61.69	61.86	61.39	61.74
	Transcoders	All		56.16	61.10	61.48	53.89
	CLTs	All		54.84	62.08	61.20	51.12
COCO	Baseline		43.12				
	Transcoders	CLS		43.10	42.98	43.00	43.22
	CLTs	CLS		43.22	43.24	43.10	43.36
	Transcoders	All		41.14	42.96	43.00	39.54
	CLTs	All		41.34	43.10	43.10	39.12
ImageNet-100	Baseline		80.42				
	Transcoders	CLS		80.38	80.48	80.84	80.46
	CLTs	CLS		80.78	80.72	80.72	80.86
	Transcoders	All		76.54	80.02	80.74	70.10
	CLTs	All		75.36	80.44	80.58	68.90

4.2.1. Cross-Layer vs. Per-Layer Transcoders

We first compare Cross-Layer Transcoders to standard per-layer Transcoders trained with the same ReLU-Top-k sparsity on ViT-B/32. Standard Transcoders are a degenerate case of Cross-Layer Transcoders where information flow is restricted to intralayer activations only. In Table 2, we observe that Cross-Layer Transcoders match or slightly outperform Transcoders in the [CLS]-only setting, while exhibiting comparable behaviour for All-token routing. Both architectures remain close to the baseline for late-layer substitution (7→11, 10→11), but degrade when cascading across all layers and all tokens, highlighting the difficulty of fully replacing early-layer patch computations. Overall, Transcoders and Cross-Layer Transcoders exhibit very similar zero-shot behavior under MLP replacement. However, only Cross-Layer Transcoders expose explicit cross-layer contributions via their triangular decoder ($c_{i\to j} = z_i W_{\text{dec}}^{i\to j}$), which is crucial for our cross-layer attribution analysis in Section 5. For this reason, in the remainder of the paper we focus on Cross-Layer Transcoders.

4.2.2. Cross-Layer Transcoder Ablations

In Table 3, we report top-1 zero-shot classification accuracy for CLIP ViT-B/32 and ViT-B/16 on CIFAR-100, COCO, and IMAGENET-100 under different cascaded replacement ranges (3→11, 7→11, 10→11, and *All*, where *All* denotes replacing all MLP layers 0→11). In all cases, CLTs were trained offline against the frozen teacher model (teacher-forcing), but at test time, cascaded substitution uses the CLT outputs as inputs to subsequent layers, allowing us to directly measure how reconstruction errors accumulate along the depth of the network. We also compare the cascading effect of CLT substitution under different token-routing modes, i.e., `[CLS]` only, patch tokens only, and all tokens, which allows us to disentangle its impact on the global classification token versus spatial tokens.

To assess functional fidelity, we compare the baseline ViT and its CLT-substituted counterpart on the same test splits used for training CLTs, using standard CLIP zero-shot classification. Following CLIP, we compute cosine similarity between the image embedding (taken from the final `[CLS]` representation) and a set of text embeddings for class-name prompts, and assign the label with highest similarity. Table 3 reports top-1 accuracy for each dataset, backbone, token-routing mode, and layer range, across the three sparsifiers: JR (JumpReLU), RTK (ReLU-Top-k), and ATK (Abs-Top-k).

Results. Across all datasets and backbones, CLT substitution is remarkably stable for `[CLS]`-only replacement: even when all layers are replaced (column *All*), top-1 accuracies remain essentially at baseline (e.g., CIFAR-100 ViT-B/32 CLS: 61.33–61.74 vs. 61.65; IMAGENET-100 ViT-B/16 CLS: 83.28–84.54 vs. 84.34). Replacing only the last few layers (10→11) is also safe for patch and all tokens, with accuracies nearly indistinguishable from the original model. In contrast, fully cascading CLTs across all layers and all tokens leads to clear accuracy degradation, particularly when routing patch tokens. For example, on CIFAR-100 ViT-B/32, patch-token RTK drops from 61.65 (baseline) to 51.12 when all MLPs are replaced (column *All*); on COCO ViT-B/16, patch-token ATK falls from 43.56 to 35.46; and on IMAGENET-100 ViT-B/16 with all tokens, ATK drops from 84.34 to 70.78.

Overall, ViT-B/16 proves more robust to substitution than ViT-B/32, mirroring the higher reconstruction quality in Table 1. The routing breakdown (`[CLS]` vs. patches vs. all tokens) thus reveals a consistent pattern: CLTs provide reliable functional approximations for the global classification `[CLS]` token and for late-layer MLPs, while early-layer patch substitutions accumulate error more aggressively.

5. Interpreting Vision Transformer Activations via Cross-Layer Sparse Features

Having established that CLTs can faithfully replace MLP blocks (Section 4), we now exploit the structure of the CLT decomposition itself as an interpretability tool. Because each target-layer reconstruction is expressed as an additive sum of decoded contributions from all preceding layers (Eq. 8), CLTs provide a form of structured attribution unavailable from standard per-layer methods, decomposing each layer's reconstruction into signed, per-source contribution scores that quantify which layers matter most for a given target representation. We first analyze the resulting cross-layer attribution structure (Section 5.1) and then validate its faithfulness through projection-based ablation (Section 5.2).

Table 3. Top-1 classification accuracy (%) across datasets, backbones, and token types. Columns correspond to layer range and sparsity combinations: JR = JumpReLU, RTK = ReLU-Top-k, ATK = Abs-Top-k. For reference, baseline top-1 accuracies (in %) are: CIFAR-100 (B/32: 61.65, B/16: 65.97), COCO (B/32: 43.12, B/16: 43.56), and ImageNet-100 (B/32: 80.42, B/16: 84.34).

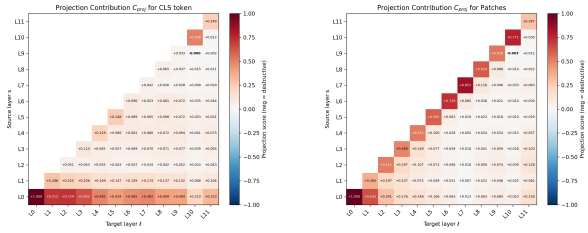
Dataset	**Backbone**	**Tokens**	3→11			7→11			10→11			All		
			JR	RTK	ATK	JR	RTK	ATK	JR	RTK	ATK	JR	RTK	ATK
CIFAR-100														
	ViT-B/32	CLS	61.38	61.69	61.58	61.41	61.86	61.26	61.06	61.39	61.16	61.33	61.74	61.43
		Patches	53.65	54.20	55.83	61.51	61.69	63.18	61.49	61.25	61.56	49.63	51.12	48.68
		All	54.10	54.84	55.69	61.54	62.08	62.85	61.01	61.20	61.11	49.40	51.12	48.41
	ViT-B/16	CLS	66.10	65.81	65.39	66.15	66.15	65.86	66.06	65.90	65.90	66.04	65.92	65.38
		Patches	63.54	60.76	63.31	66.05	65.48	67.24	65.97	65.82	66.13	62.40	58.57	56.59
		All	63.71	60.75	62.87	66.22	65.56	67.08	65.91	65.98	65.89	62.45	58.82	56.72
COCO														
	ViT-B/32	CLS	43.08	43.22	43.12	43.26	43.24	43.20	43.04	43.10	42.90	43.12	43.36	43.00
		Patches	40.58	40.92	41.48	42.98	43.14	42.80	43.14	43.18	43.06	39.04	38.94	39.68
		All	41.00	41.34	41.46	43.14	43.10	42.76	42.78	43.10	42.92	38.60	39.12	40.06
	ViT-B/16	CLS	43.68	43.64	42.64	43.52	43.64	43.34	43.52	43.88	43.38	43.62	43.62	42.76
		Patches	43.00	42.34	39.46	43.26	43.36	43.30	43.62	43.64	43.72	42.72	42.00	35.46
		All	42.78	42.66	38.18	43.62	43.58	43.48	43.42	43.92	43.60	43.00	42.08	34.16
IMAGENET-100														
	ViT-B/32	CLS	80.86	80.78	80.16	80.64	80.72	80.16	80.56	80.72	80.18	80.92	80.86	80.26
		Patches	75.58	75.26	72.46	80.34	80.18	80.64	80.52	80.46	80.46	71.96	68.74	60.26
		All	75.12	75.36	71.44	80.18	80.44	79.38	80.38	80.58	80.18	71.60	68.90	60.26
	ViT-B/16	CLS	84.56	84.08	83.74	84.66	84.16	84.40	84.56	84.38	84.34	84.54	84.02	83.28
		Patches	83.16	82.76	79.36	83.94	84.24	83.84	84.20	84.22	84.58	83.04	81.40	73.06
		All	83.00	82.64	78.08	84.28	84.32	83.34	84.40	84.36	84.74	83.12	80.94	70.78

5.1. Cross-Layer Contribution Scores

To understand how the final-layer representation is assembled across depth, we evaluate the projection-based attribution scores $C^{\text{proj}}_{i\to j}$ (Eq. 9) for every target layer j, computing the fraction of each target's reconstruction explained by each source layer $i \le j$. Scores are computed separately for `[CLS]` and patch tokens by restricting the token average in Eq. 9 to the corresponding positions, and are then averaged over the validation set. The resulting heatmaps directly reveal how attribution is partitioned across source layers.

Figure 3 visualizes $C^{\text{proj}}_{i\to L}$ for `[CLS]` (left) and patch tokens (right), revealing two qualitatively distinct regimes. For patch tokens, the attribution matrix is strongly diagonal-dominant: each layer explains the majority of its own post-MLP output, with only modest credit diffusing to immediate neighbors. This locality is consistent with the view that patch-level computations are dominated by within-layer transformations that progressively refine spatial features. The `[CLS]` token exhibits a markedly different structure. Credit is distributed broadly across depth, with several early layers receiving scores comparable to or exceeding the self-layer term $C^{\text{proj}}_{L\to L}$. This indicates that the final `[CLS]` representation does not emerge primarily from the last MLP block but is instead a depth-integrated aggregation of semantic signals accumulated over many preceding layers, consistent with its architectural role as a global summary token.

We also observe that even for patch tokens, the earliest layer (L0) retains a non-negligible fraction of credit at every target depth, which may indicate that low-level features such as edges and textures persist in deeper representations rather than being fully overwritten. Additionally, the contrast between the diagonal patch structure and the distributed `[CLS]` structure offers an intuition for the asymmetric substitution behavior in Table 3: `[CLS]`-only replacement may succeed because attribution is spread across many source layers, making the reconstruction less sensitive to errors at any single layer, whereas patch tokens depend predominantly on their own layer and are thus more affected by per-layer approximation errors.



(a) `[CLS]` token (b) Patch tokens

Figure 3. Cross-layer attribution scores $C^{\text{proj}}_{i\to j}$ for ViT-B/16 on CIFAR-100, averaged over the validation set. Scores along each column yield a partition of attribution across depth. Patch tokens (right) exhibit strong diagonal dominance, while `[CLS]` (left) draws attribution broadly across source layers.

Table 4. Faithful attribution via projection-based ablation on the final layer of ViT-B/32. Accuracy (%) and KL divergence from the unmodified logits are reported for [CLS] and all tokens ([CLS]+patch).

Model	Dataset	Tokens	Baseline		Full CLT		Drop Top-1		Keep Top-4	
			Acc↑	KL↓	Acc↑	KL↓	Acc↑	KL↓	Acc↑	KL↓
ViT-B/32	cifar100	all (cls+patch)	61.65%	-	61.31%	0.0104	59.68%	0.0800	59.96%	0.0714
		cls	61.65%	-	61.31%	0.0104	59.91%	0.0704	60.72%	0.0582
	coco	all (cls+patch)	43.12%	-	43.26%	0.0103	42.98%	0.1619	43.02%	0.0156
		cls	43.12%	-	43.26%	0.0103	42.78%	0.1157	43.00%	0.0155
	imagenet100	all (cls+patch)	80.42%	-	80.54%	0.0073	74.94%	0.2320	80.56%	0.0109
		cls	80.42%	-	80.54%	0.0073	74.86%	0.2733	80.48%	0.0106

5.2. Faithful Attribution

To validate that the cross-layer attribution scores faithfully reflect each source layer's functional importance, we perform projection-based ablation experiments on the final-layer reconstruction. Specifically, for each input we rank the source layers by their attribution score $C_{i\to j}^{\text{proj}}$ at the final target layer $j{=}L$ and evaluate three ablation conditions: (i) **Full CLT**, which retains all layer contributions and measures the baseline fidelity of the decomposition; (ii) **Drop Top-1**, which removes the single highest-scored source layer; and (iii) **Keep Top-4**, which retains only the four highest-scored layers and zeros out all remaining contributions. For each condition, we substitute the ablated reconstruction $\tilde{y}_L$ into the CLIP vision encoder in place of the original MLP output and report zero-shot classification accuracy and KL divergence from the unmodified logit distribution. All rankings are computed per instance, ensuring that the ablation respects input-dependent variation in layer importance. We evaluate on both the [CLS] token alone and all tokens.

Results are presented in Table 4. The Full CLT reconstruction closely preserves baseline accuracy across all datasets, with KL divergence below 0.011 in every setting, confirming that the additive decomposition is near-lossless. Removing the single highest-ranked source layer (Drop Top-1) produces consistent accuracy drops, most notably on ImageNet-100, where accuracy falls from 80.54% to 74.94% (-5.6 %) with a corresponding KL increase to 0.232, demonstrating that the top-scoring layer captures a disproportionate share of the classification-relevant signal. Conversely, retaining only the top-4 source layers (Keep Top-4) recovers near-baseline performance, with accuracy within 0.2 % and KL divergence under 0.02 on ImageNet-100. This dual evidence of *necessity* (removal of high-scoring layers degrades performance) and *sufficiency* (retention of a small subset preserves it) confirms that the projection-based scores provide a faithful assessment of each layer's contribution to the final representation. The consistency of these trends across datasets and token configurations further supports the generality of the attribution mechanism.

6. Conclusion and Future Work

In this paper, we introduced the novel adoption of CLTs as sparse, depth-aware, interpretable proxy models for MLP blocks in Vision Transformers. Across three datasets and two CLIP backbones, we have shown that CLTs achieve high reconstruction fidelity while preserving, and in some cases even improving, zero-shot classification accuracy, particularly when substituting the [CLS] token or considering late-layer blocks CLT-based approximations. Beyond functional replacement, the innate linear decomposition structure of CLTs yields signed, per-source attribution scores that reveal a striking asymmetry: patch tokens are governed predominantly by within-layer transformations, whereas the [CLS] token aggregates semantic signal broadly across depth; an architectural insight that cannot be seen in standard per-layer methods such as conventional transcoders or SAEs. In addition, the novel adoption of CLTs in the vision domain provides a faithful attribution: the final representation is concentrated in a small subset of dominant source layers whose removal degrades performance and whose retention largely preserves it. Together, these results and insights position CLTs as a principled framework for structured, cross-layer interpretability in vision models without compromising zero-shot classification performance.

The depth-resolved sparse features learned by CLTs give rise to three distinct research directions that belong to our future research agenda. First, the layer-wise decomposition provides a principled substrate for hierarchical concept discovery, revealing how visual primitives in early layers give rise to semantic abstractions in deeper ones and tracing their composition across depth through the CLT attribution structure. Second, CLTs currently model only the MLP pathway; incorporating attention heads into the cross-layer decomposition can potentially yield a unified sparse proxy that accounts for both feature transformation and token interaction, enabling end-to-end circuit analysis for Vision Transformers. Finally, we aim to explore CLTs as an interpretable proxy for a broader range of vision tasks and for diverse Multimodal Large Language Models.

Acknowledgments

We would like to thank the reviewers and the area chair for their evaluation and feedback. This work was supported in part by NSF grants 2212302, 2212301, 2212303, AFOSR 23RT0630, and NIH 2R01HL127661.

References

[1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[2] Emmanuel Ameisen, Jack Lindsey, Adam Pearce, Wes Gurnee, Nicholas L Turner, Brian Chen, Craig Citro, David Abrahams, Shan Carter, Basil Hosmer, et al. Circuit tracing: Revealing computational graphs in language models. *Transformer Circuits Thread*, 6:16318–16352, 2025.

[3] David Bau, Bolei Zhou, Aditya Khosla, Aude Oliva, and Antonio Torralba. Network dissection: Quantifying interpretability of deep visual representations. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6541–6549, 2017.

[4] Usha Bhalla, Alex Oesterling, Suraj Srinivas, Flavio Calmon, and Himabindu Lakkaraju. Interpreting clip with sparse linear concept embeddings (splice). *Advances in Neural Information Processing Systems*, 37:84298–84328, 2024.

[5] Trenton Bricken, Adly Templeton, Joshua Batson, Brian Chen, Adam Jermyn, Tom Conerly, Nick Turner, Cem Anil, Carson Denison, Amanda Askell, et al. Towards monosemanticity: Decomposing language models with dictionary learning. *Transformer Circuits Thread*, 2(5):6, 2023.

[6] Gerasimos Chatzoudis, Zhuowei Li, Gemma E Moran, Hao Wang, and Dimitris N Metaxas. Visual sparse steering: Improving zero-shot image classification with sparsity guided steering vectors. *arXiv preprint arXiv:2506.01247*, 2025.

[7] Arthur Conmy, Augustine Mavor-Parker, Aengus Lynch, Stefan Heimersheim, and Adrià Garriga-Alonso. Towards automated circuit discovery for mechanistic interpretability. *Advances in Neural Information Processing Systems*, 36:16318–16352, 2023.

[8] Hoagy Cunningham, Aidan Ewart, Logan Riggs, Robert Huben, and Lee Sharkey. Sparse autoencoders find highly interpretable features in language models. *arXiv preprint arXiv:2309.08600*, 2023.

[9] Matthijs Douze, Alexandr Guzhva, Chengqi Deng, Jeff Johnson, Gergely Szilvasy, Pierre-Emmanuel Mazaré, Maria Lomeli, Lucas Hosseini, and Hervé Jégou. The faiss library. *IEEE Transactions on Big Data*, 2025.

[10] Jacob Dunefsky, Philippe Chlenski, and Neel Nanda. Transcoders find interpretable llm feature circuits. *Advances in Neural Information Processing Systems*, 37:24375–24410, 2024.

[11] Nelson Elhage, Neel Nanda, Catherine Olsson, Tom Henighan, Nicholas Joseph, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, et al. A mathematical framework for transformer circuits. *Transformer Circuits Thread*, 1(1):12, 2021.

[12] Nelson Elhage, Tristan Hume, Catherine Olsson, Nicholas Schiefer, Tom Henighan, Shauna Kravec, Zac Hatfield-Dodds, Robert Lasenby, Dawn Drain, Carol Chen, et al. Toy models of superposition. *arXiv preprint arXiv:2209.10652*, 2022.

[13] Thomas Fel, Ekdeep Singh Lubana, Jacob S Prince, Matthew Kowal, Victor Boutin, Isabel Papadimitriou, Binxu Wang, Martin Wattenberg, Demba Ba, and Talia Konkle. Archetypal sae: Adaptive and stable dictionary learning for concept extraction in large vision models. *arXiv preprint arXiv:2502.12892*, 2025.

[14] Leo Gao, Tom Dupré la Tour, Henk Tillman, Gabriel Goh, Rajan Troll, Alec Radford, Ilya Sutskever, Jan Leike, and Jeffrey Wu. Scaling and evaluating sparse autoencoders. *arXiv preprint arXiv:2406.04093*, 2024.

[15] Michael Hanna, Mateusz Piotrowski, Jack Lindsey, and Emmanuel Ameisen. Circuit-tracer: A new library for finding feature circuits. In *Proceedings of the 8th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*, pages 239–249, 2025.

[16] Jeff Johnson, Matthijs Douze, and Hervé Jégou. Billion-scale similarity search with gpus. *IEEE transactions on big data*, 7 (3):535–547, 2019.

[17] Sonia Joseph, Praneet Suresh, Ethan Goldfarb, Lorenz Hufe, Yossi Gandelsman, Robert Graham, Danilo Bzdok, Wojciech Samek, and Blake Aaron Richards. Steering clip's vision transformer with sparse autoencoders. In *Mechanistic Interpretability for Vision at CVPR 2025 (Non-proceedings Track)*, 2025.

[18] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4015–4026, 2023.

[19] Hyesu Lim, Jinho Choi, Jaegul Choo, and Steffen Schneider. Sparse autoencoders reveal selective remapping of visual concepts during adaptation. *arXiv preprint arXiv:2412.05276*, 2024.

[20] Jack Lindsey, Adly Templeton, Jonathan Marcus, Thomas Conerly, Joshua Batson, and Christopher Olah. Sparse crosscoders for cross-layer features and model diffing. *Transformer Circuits Thread*, pages 3982–3992, 2024.

[21] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36:34892–34916, 2023.

[22] 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 preprint arXiv:2403.19647*, 2024.

[23] Julian Minder, Clément Dumas, Bilal Chughtai, and Neel Nanda. Robustly identifying concepts introduced during chat fine-tuning using crosscoders. In *Sparsity in LLMs (SLLM): Deep Dive into Mixture of Experts, Quantization, Hardware, and Inference*.

[24] Julian Minder, Clément Dumas, Caden Juang, Bilal Chugtai, and Neel Nanda. Overcoming sparsity artifacts in crosscoders to interpret chat-tuning. *arXiv preprint arXiv:2504.02922*, 2025.

[25] Tuomas Oikarinen and Tsui-Wei Weng. Clip-dissect: Automatic description of neuron representations in deep vision networks. *arXiv preprint arXiv:2204.10965*, 2022.

[26] Chris Olah, Alexander Mordvintsev, and Ludwig Schubert. Feature visualization. *Distill*, 2(11):e7, 2017.

[27] Chris Olah, Nick Cammarata, Ludwig Schubert, Gabriel Goh, Michael Petrov, and Shan Carter. Zoom in: An introduction to circuits. *Distill*, 5(3):e00024–001, 2020.

[28] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PmLR, 2021.

[29] Senthooran Rajamanoharan, Tom Lieberum, Nicolas Sonnerat, Arthur Conmy, Vikrant Varma, János Kramár, and Neel Nanda. Jumping ahead: Improving reconstruction fidelity with jumprelu sparse autoencoders. *arXiv preprint arXiv:2407.14435*, 2024.

[30] Lee Sharkey, Dan Braun, and Beren Millidge. Taking features out of superposition with sparse autoencoders. In *AI Alignment Forum*, pages 12–13, 2022.

[31] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps, corr abs/1312.6034. *arXiv preprint arXiv:1312.6034*, 2014.

[32] Samuel Stevens, Wei-Lun Chao, Tanya Berger-Wolf, and Yu Su. Sparse autoencoders for scientifically rigorous interpretation of vision models. *arXiv e-prints*, pages arXiv–2502, 2025.

[33] Viacheslav Surkov, Chris Wendler, Antonio Mari, Mikhail Terekhov, Justin Deschenaux, Robert West, Caglar Gulcehre, and David Bau. One-step is enough: Sparse autoencoders for text-to-image diffusion models. *arXiv preprint arXiv:2410.22366*, 2024.

[34] Adly Templeton. *Scaling monosemanticity: Extracting interpretable features from claude 3 sonnet*. Anthropic, 2024.

[35] Harrish Thasarathan, Julian Forsyth, Thomas Fel, Matthew Kowal, and Konstantinos G Derpanis. Universal sparse autoencoders: Interpretable cross-model concept alignment. In *Forty-second International Conference on Machine Learning*, 2025.

[36] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

[37] Matthew D. Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *Computer Vision – ECCV 2014*, pages 818–833, Cham, 2014. Springer International Publishing.

[38] Xudong Zhu, Mohammad Mahdi Khalili, and Zhihui Zhu. Abstopk: Rethinking sparse autoencoders for bidirectional features. *arXiv preprint arXiv:2510.00404*, 2025.

Can Cross-Layer Transcoders Replace Vision Transformer Activations? An Interpretable Perspective on Vision

Supplementary Material

7. Visually Explainable Cross-Layer Contribution

The contribution scores $C_{s\to\ell}$ quantify *which* layers matter, but not *what* they represent. To obtain example-based explanations of cross-layer influence, we provide some evidence of a retrieval-based framework that operates directly on the CLT sparse codes. Recall that a target post-MLP representation $\hat{y}_L$ at the final layer can be decomposed as

$$\hat{y}_L = W^{\text{dec}}_{0\to L} z_0 + \cdots + W^{\text{dec}}_{L\to L} z_L$$

where each term corresponds to the contribution of a previous layer's sparse representation z_i, transformed by its decoder $W^{\text{dec}}_{i\to L}$. Instead of inspecting individual neurons z_{ik} in isolation, we treat the full sparse vector z_i as a feature and use it for layer-wise retrieval.

Concretely, we construct an external corpus from the training images of each dataset. For each image j and layer i, we pass the image through the CLT and extract its sparse codes $z_i^{(j)}$. We then aggregate token-level features (e.g., by averaging over spatial tokens) into a global sparse descriptor $z^{(j)}_{i,\text{agg}}$ and index these descriptors into a per-layer FAISS database D_i [9, 16]. At test time, for a given input image, we compute its sparse activations $\{z_i\}$ and use them to retrieve the top-K most similar training images from each layer's index:

$$\mathcal{N}_i(z_i) = \arg\operatorname*{topK}_j \text{sim}\left(z_i, z^{(j)}_{i,\text{agg}}\right), \quad \text{for } i \le \ell \qquad (12)$$

This mirrors the reconstruction process $\hat{y}^\ell = \sum_{i\le\ell} z_i W^{i\to\ell}_{\text{dec}}$ by surfacing images that share similar latent activations at each contributing layer. Unlike decoding, this retrieval provides interpretable, layer-specific visual evidence of what each layer's sparse features contribute to the final representation.

Figures 4 and 5 illustrate our retrieval-based framework for a single query image, showing the top-3 retrieved training samples per layer. For the `[CLS]` token, retrievals exhibit a depth-wise progression from low-level to high-level semantics: shallow layers primarily group images by generic color and layout statistics; mid-level layers emphasize similar configurations of people and objects; and the deepest layers retrieve highly class-consistent examples (e.g., images depicting the same activity, i.e., surfing), exhibiting robustness to viewpoint and background variation. This depth-wise semantic sharpening aligns with our cross-layer contribution analysis and with the strong functional performance of CLS-only substitution: as features become more class-specific with depth, CLT-based surrogates can both reconstruct and visually explain the representations driving the final decision.

Table 5. CLT surrogate faithfulness metrics.

	Pred. Distrib.			Top-k Agree		Embedding			Prompt Sens.	
Layers	ΔAcc	Flip↓	KL↓	Top-1	Top-5	Cos	CKA	Spear.	r	KL↓
0-11$^{\text{CLS}}$	+0.07	10.4%	.034	89.6	88.5	.984	.948	.929	.995	.036
7-11$^{\text{CLS}}$	+0.01	9.5%	.030	90.5	89.0	.985	.954	.937	.992	.031
10-11$^{\text{CLS}}$	−0.07	7.2%	.018	92.8	91.2	.991	.979	.973	.994	.019
11$^{\text{CLS}}$	−0.04	5.4%	.010	94.6	93.2	.996	.991	.991	.997	.010
0-11$^{\text{Patches}}$	−11.1	35.2%	.412	64.8	70.0	.957	.817	.759	.942	.383
7-11$^{\text{Patches}}$	+0.05	15.0%	.079	85.0	85.8	.990	.972	.957	.989	.073
10-11$^{\text{Patches}}$	−0.13	2.5%	.002	97.5	96.8	.999	.999	.998	.996	.002

8. Additional Metrics for CLTs' faithfulness

To provide further evidence of CLT's faithfulness to the original model, we evaluate distributional alignment (KL, flip rate), top-k agreement, embedding geometry (cos/CKA/Spearman), and prompt sensitivity across 18 templates for the ViT-B/32 model on CIFAR100. Table 5 shows that in the regimes the CLTs faithfully reconstruct the original model's representations (CLS across layers; late-layer replacement), the surrogate closely matches the teacher beyond accuracy (e.g., KL< 0.035, CKA> 0.94, prompt-trend $r > 0.98$). These results also validate the faithful CLT late-layer patch replacement (e.g., 10–11^P: KL$= 0.002$, flip$= 2.5\%$, CKA$= 0.999$). In contrast, early patch cascades degrade (e.g., 0–11^P: KL$= 0.412$, flip$= 35.2\%$), as identified in Table 3.

9. Training Details

Teacher models and datasets. All Cross-Layer Transcoders (CLTs) are trained on top of frozen CLIP image encoders with ViT-B/32 and ViT-B/16 backbones. For each backbone, we consider three datasets: CIFAR-100, COCO, and ImageNet-100. For every (dataset, backbone) pair we train three separate CLT variants, one for each sparsifier: JumpReLU, ReLU-Top-k, and Abs-Top-k (with $k = 128$). CLTs only access the internal activations of the teacher ViT and do not modify or finetune the underlying CLIP parameters.

Supervision and targets. Let $x_\ell \in \mathbb{R}^{T\times d}$ denote the post-attention (LN2), pre-MLP activations at layer ℓ, and let $y_\ell = \text{MLP}_\ell(x_\ell)$ be the corresponding post-MLP outputs. For each image, we run the frozen teacher ViT once and cache

Figure 4. Layerwise visual retrieval using CLT sparse codes of CLS for a test image. Each row shows the top-3 retrieved training samples across transformer layers, revealing the semantic evolution of representations.

Figure 5. Layerwise visual retrieval using CLT sparse codes of Patches for a test image. Each row shows the top-3 retrieved training samples across transformer layers, revealing the semantic evolution of representations.

(x_ℓ, y_ℓ) for all layers $\ell = 0, \ldots, 11$ and all tokens (both [CLS] and patch tokens). CLTs are trained to reconstruct y_ℓ from sparse features computed from $\{x_i\}_{i \le \ell}$, using teacher-forcing at training time; i.e., all CLT inputs come from the unmodified teacher trajectory.

Optimization hyperparameters. For all datasets, backbones, and sparsifiers we train CLTs with the AdamW optimizer, learning rate 2×10^{-4}, and an expansion factor of 16. Each CLT is trained for 10 epochs over the corresponding dataset, using all tokens (both [CLS] and patch tokens) in the loss. Hyperparameters are shared across datasets and backbones.

10. Reconstruction Accuracy across Layers

In Figures 6–10, we report the reconstruction quality of Cross-Layer Transcoders (CLTs) across all transformer layers on three datasets (CIFAR-100, COCO, and ImageNet-100) and two CLIP backbones (ViT-B/32 and ViT-B/16). For each configuration, we compare three sparsity variants, i.e., JUMPRELU, RELU-TOP-k, and ABS-TOP-k, using cosine similarity, mean squared error (log scale), and variance explained (R^2), averaged over all tokens in the test set.

11. Classification Accuracy under Cascaded CLT Replacement

We report top-1 classification accuracy (%) under cascaded CLT replacement across all layers ($s \to 11$) in Figures 6–23. For each dataset (CIFAR-100, COCO, ImageNet-100) and ViT backbone (ViT-B/32, ViT-B/16), we evaluate three sparsity mechanisms: JUMPRELU (JR), RELU-TOP-k (RTK), and ABS-TOP-k (ATK), under three token settings (CLS-only, patch-only, and all tokens). The replacement is performed progressively from early to late layers ($s = 0$ to $s = 11$), and results are compared to the frozen ViT baseline. We observe that across all datasets and backbones, CLS-token replacement achieves near-identical or slightly better accuracy compared to the original model. This shows that CLS tokens are robust to replacement. Regarding patch tokens, accuracy improves significantly as more layers are replaced, especially in the later layers.

12. Cross-Layer Contribution Scores

To better understand the internal attribution structure of Cross-Layer Transcoders (CLTs), we visualize in Figures 11–16 the contribution scores $C_{s \to \ell}$, which quantify the influence of each source layer s on the reconstruction of activations at target layer ℓ. These heatmaps reveal a clear depth-aware structure across all datasets and backbones, where contributions are strongest from temporally proximal layers.

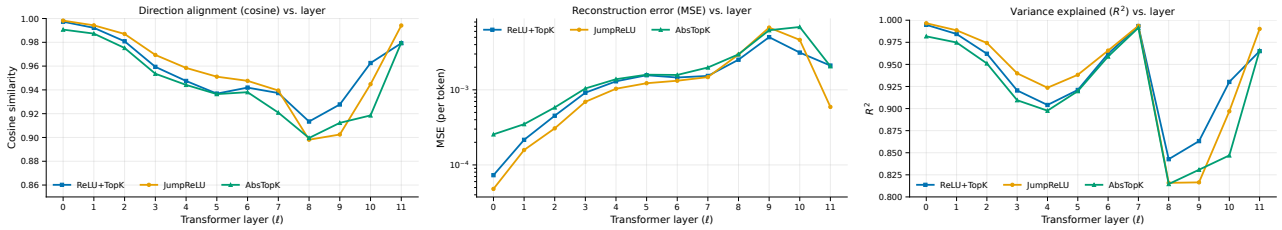


Figure 6. Reconstruction performance of CLTs across transformer layers on CIFAR-100 using ViT-B/16. We report cosine similarity (left), MSE per token in log scale (center), and R^2 (right) for JUMPRELU, RELU-TOP-k, and ABS-TOP-k sparsity variants (k=128), averaged across all tokens in the validation set.

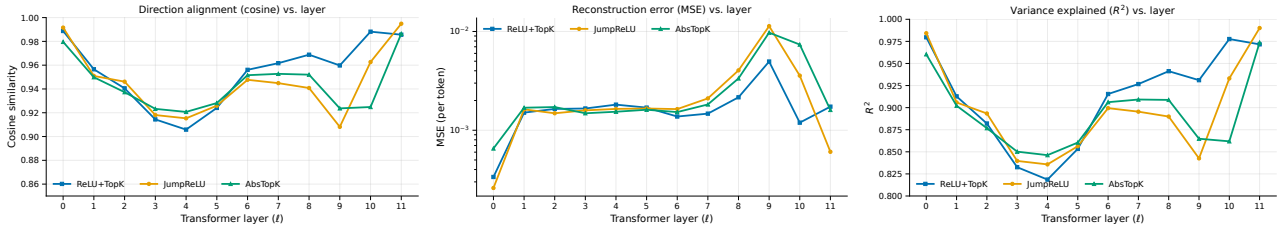


Figure 7. Reconstruction performance of CLTs across transformer layers on COCO using ViT-B/32. We show cosine similarity (left), MSE per token (log scale, center), and R^2 (right) for the three sparsity variants JUMPRELU, RELU-TOP-k, and ABS-TOP-k (k=128), averaged across all tokens in the validation set.

Table 6. Top-1 classification accuracy (%) on CIFAR-100 for ViT-B/32 with CLS tokens, across layer ranges $s \to 11$. JR = JumpReLU, RTK = ReLU-Top-k, ATK = Abs-Top-k. Baseline top-1 accuracies (in %) are ViT-B/32: 61.65 and ViT-B/16: 65.97.

Range	JR	RTK	ATK
0→11	61.33	**61.74**	61.43
1→11	61.43	**61.76**	61.56
2→11	61.39	**61.72**	61.61
3→11	61.38	**61.69**	61.58
4→11	61.20	**61.76**	61.66
5→11	61.21	**61.59**	61.48
6→11	61.43	**61.58**	61.39
7→11	61.41	**61.86**	61.26
8→11	61.62	**61.90**	61.23
9→11	61.31	**61.85**	61.10
10→11	61.06	**61.39**	61.16
11→11	61.23	**61.31**	61.03

Notably, CLS tokens exhibit more distributed contributions spanning earlier layers, while patch tokens show sharper, more localized attribution.

Table 7. Top-1 classification accuracy (%) on CIFAR-100 for ViT-B/32 with Patches tokens, across layer ranges $s \to 11$. JR = JumpReLU, RTK = ReLU-Top-k, ATK = Abs-Top-k. Baseline top-1 accuracies (in %) are ViT-B/32: 61.65 and ViT-B/16: 65.97.

Range	JR	RTK	ATK
0→11	49.63	**51.12**	48.68
1→11	49.92	51.61	**52.40**
2→11	51.95	52.86	**53.46**
3→11	53.65	54.20	**55.83**
4→11	56.59	57.02	**58.86**
5→11	58.89	59.52	**61.38**
6→11	60.39	61.16	**63.06**
7→11	61.51	61.69	**63.18**
8→11	61.83	62.04	**63.53**
9→11	61.82	61.73	**62.57**
10→11	61.49	61.25	**61.56**
11→11	**61.65**	**61.65**	**61.65**

Table 8. Top-1 classification accuracy (%) on CIFAR-100 for ViT-B/32 with All tokens, across layer ranges $s \to 11$. JR = JumpReLU, RTK = ReLU-Top-k, ATK = Abs-Top-k. Baseline top-1 accuracies (in %) are ViT-B/32: 61.65 and ViT-B/16: 65.97.

Range	JR	RTK	ATK
0→11	49.40	**51.12**	48.41
1→11	49.90	51.91	**51.97**
2→11	51.84	**53.51**	53.16
3→11	54.10	54.84	**55.69**
4→11	56.62	57.32	**58.23**
5→11	59.21	59.96	**61.19**
6→11	60.54	61.60	**62.95**
7→11	61.54	62.08	**62.85**
8→11	61.69	62.32	**63.01**
9→11	61.63	61.83	**62.07**
10→11	61.01	**61.20**	61.11
11→11	61.23	**61.31**	61.03

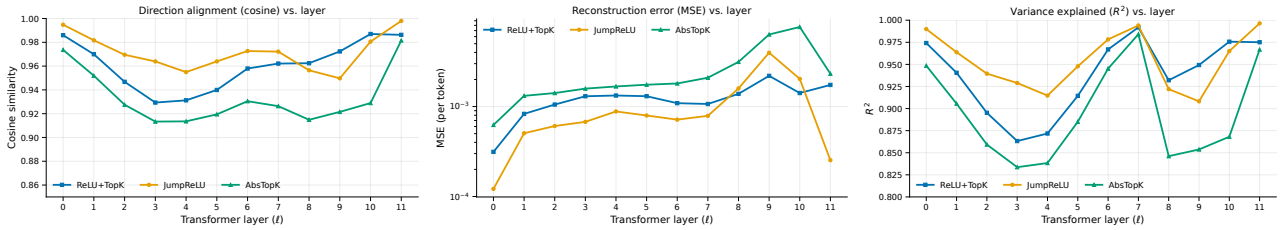


Figure 8. Reconstruction performance of CLTs across transformer layers on COCO using ViT-B/16. As in the main paper, we compare JUMPRELU, RELU-TOP-k, and ABS-TOP-k (k=128) with cosine similarity (left), MSE per token (log scale, center), and R^2 (right), averaged over the validation set.

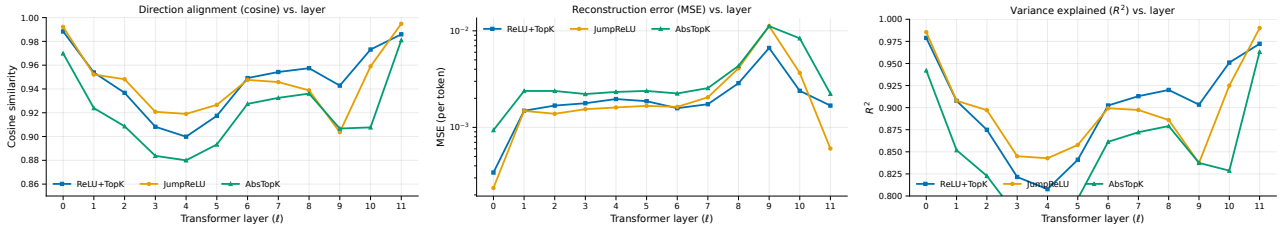


Figure 9. Reconstruction performance of CLTs across transformer layers on ImageNet-100 using ViT-B/32. We plot cosine similarity (left), MSE per token in log scale (center), and R^2 (right) for the three sparsity variants JUMPRELU, RELU-TOP-k, and ABS-TOP-k (k=128), averaged across all tokens in the validation set.

Table 9. Top-1 classification accuracy (%) on CIFAR-100 for ViT-B/16 with CLS tokens, across layer ranges $s \to 11$. JR = JumpReLU, RTK = ReLU-Top-k, ATK = Abs-Top-k. Baseline top-1 accuracies (in %) are ViT-B/32: 61.65 and ViT-B/16: 65.97.

Range	JR	RTK	ATK
0→11	**66.04**	65.92	65.38
1→11	**66.05**	65.82	65.62
2→11	**66.02**	65.78	65.73
3→11	**66.10**	65.81	65.39
4→11	**66.05**	65.87	65.61
5→11	**66.15**	65.80	65.65
6→11	**66.16**	66.08	65.71
7→11	**66.15**	**66.15**	65.86
8→11	**66.12**	65.98	66.02
9→11	**66.12**	65.93	66.01
10→11	**66.06**	65.90	65.90
11→11	65.87	**66.00**	65.65

Table 10. Top-1 classification accuracy (%) on CIFAR-100 for ViT-B/16 with Patches tokens, across layer ranges $s \to 11$. JR = JumpReLU, RTK = ReLU-Top-k, ATK = Abs-Top-k. Baseline top-1 accuracies (in %) are ViT-B/32: 61.65 and ViT-B/16: 65.97.

Range	JR	RTK	ATK
0→11	**62.40**	58.57	56.59
1→11	**62.58**	59.40	60.68
2→11	**63.15**	60.07	61.97
3→11	**63.54**	60.76	63.31
4→11	64.29	62.32	**64.85**
5→11	65.24	63.50	**66.11**
6→11	65.72	64.76	**66.85**
7→11	66.05	65.48	**67.24**
8→11	65.68	65.79	**66.79**
9→11	65.65	65.50	**66.06**
10→11	65.97	65.82	**66.13**
11→11	**65.97**	**65.97**	**65.97**

Table 11. Top-1 classification accuracy (%) on CIFAR-100 for ViT-B/16 with All tokens, across layer ranges $s \to 11$. JR = JumpReLU, RTK = ReLU-Top-k, ATK = Abs-Top-k. Baseline top-1 accuracies (in %) are ViT-B/32: 61.65 and ViT-B/16: 65.97.

Range	JR	RTK	ATK
0→11	**62.45**	58.82	56.72
1→11	**62.70**	59.23	60.30
2→11	**62.95**	59.74	61.54
3→11	**63.71**	60.75	62.87
4→11	64.14	62.16	**64.57**
5→11	65.20	63.60	**65.86**
6→11	66.08	64.73	**66.31**
7→11	66.22	65.56	**67.08**
8→11	65.72	65.84	**66.65**
9→11	65.77	65.51	**66.05**
10→11	65.91	**65.98**	65.89
11→11	65.87	**66.00**	65.65

Table 12. Top-1 classification accuracy (%) on COCO for ViT-B/32 with CLS tokens, across layer ranges $s \to 11$. JR = JumpReLU, RTK = ReLU-Top-k, ATK = Abs-Top-k. Baseline top-1 accuracies (in %) are ViT-B/32: 43.12 and ViT-B/16: 43.56.

Range	JR	RTK	ATK
0→11	43.12	**43.36**	43.00
1→11	43.14	**43.32**	43.08
2→11	43.10	**43.26**	42.96
3→11	43.08	**43.22**	43.12
4→11	43.18	**43.22**	43.06
5→11	43.22	**43.30**	43.06
6→11	**43.30**	43.24	43.22
7→11	**43.26**	43.24	43.20
8→11	43.32	**43.40**	43.26
9→11	43.16	**43.34**	42.92
10→11	43.04	**43.10**	42.90
11→11	43.24	**43.26**	43.00

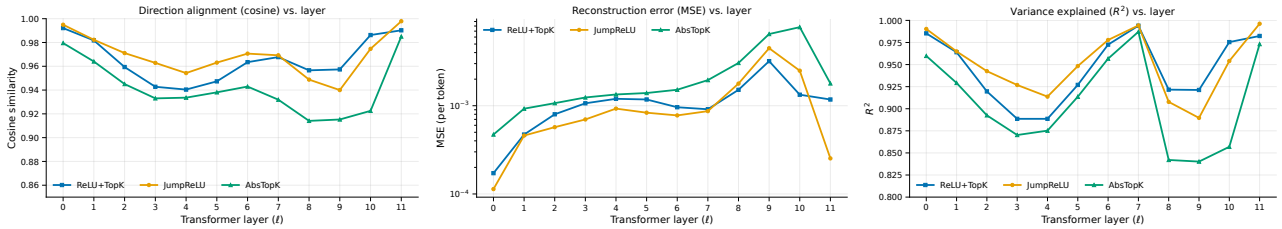
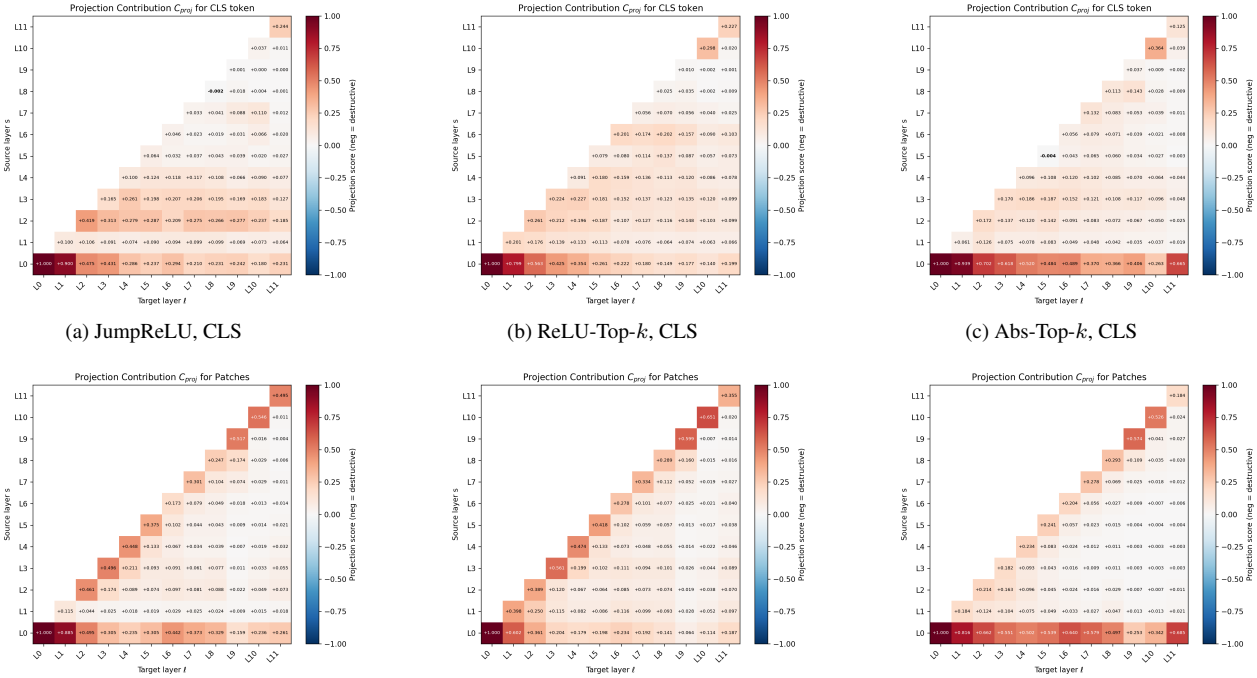


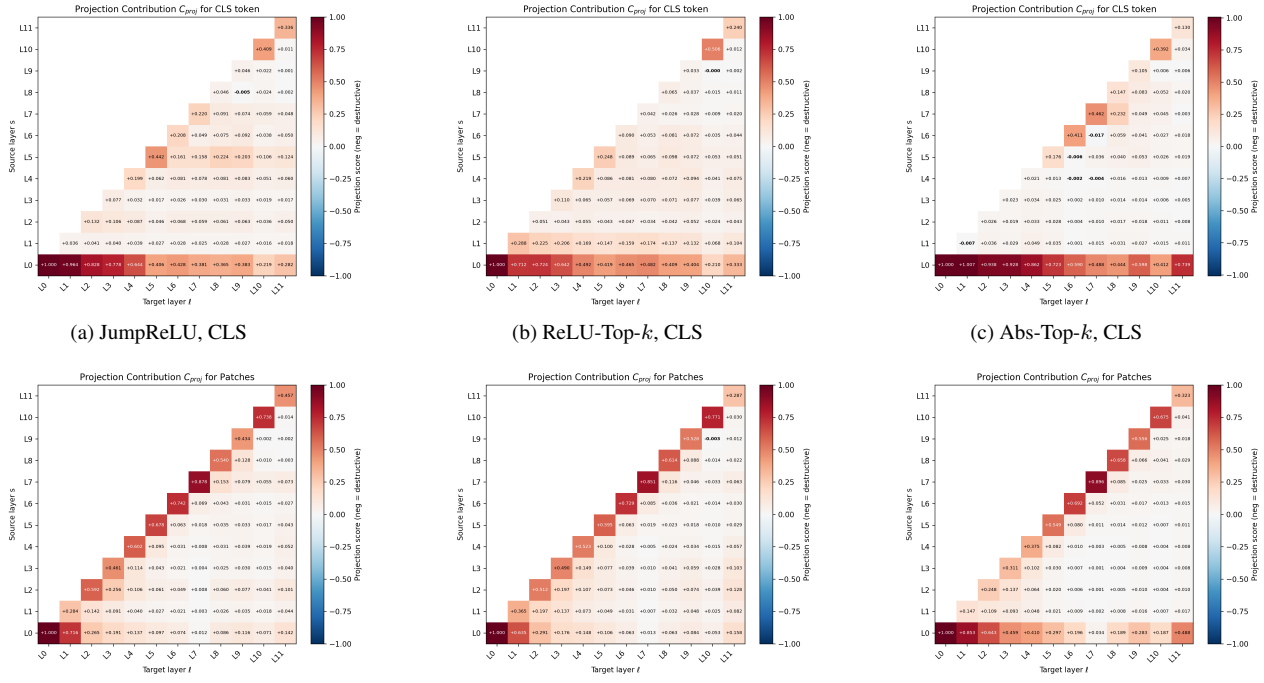
Figure 10. Reconstruction performance of CLTs across transformer layers on ImageNet-100 using ViT-B/16. Cosine similarity (left), MSE per token (log scale, center), and R^2 (right) are reported for JUMPRELU, RELU-TOP-k, and ABS-TOP-k sparsity (k=128), averaged across all tokens in the test set.



(a) JumpReLU, CLS (b) ReLU-Top-k, CLS (c) Abs-Top-k, CLS

(d) JumpReLU, patches (e) ReLU-Top-k, patches (f) Abs-Top-k, patches

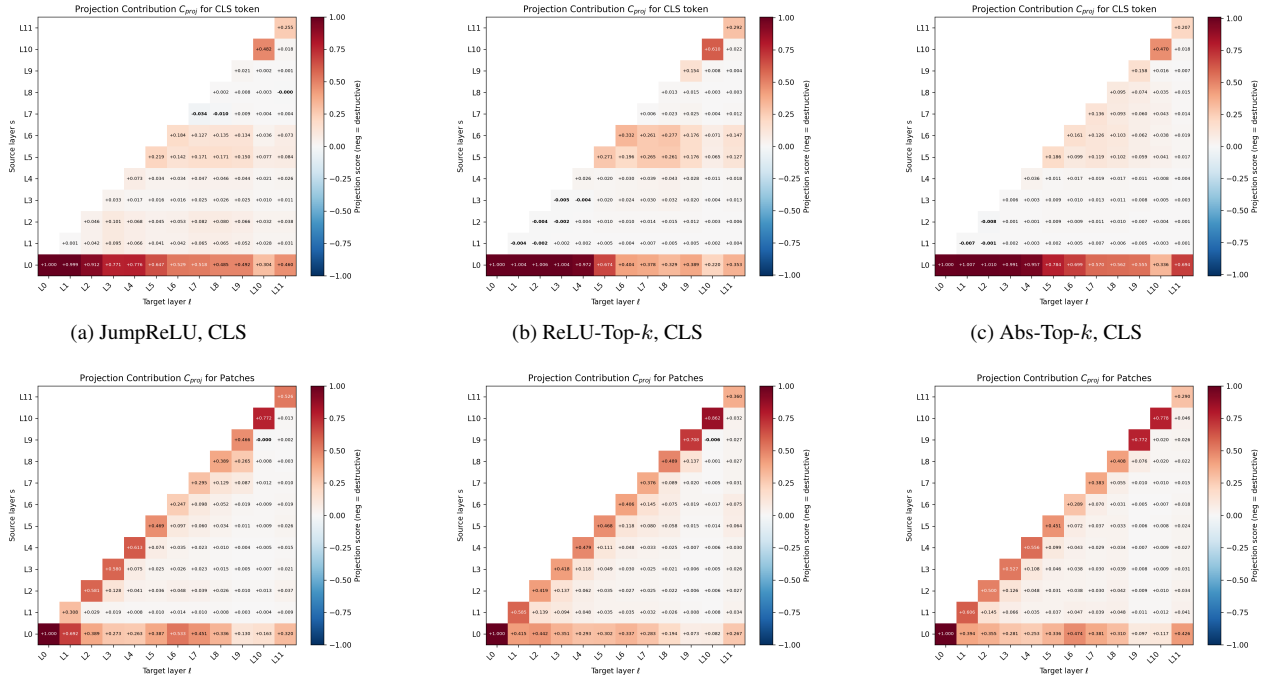
Figure 11. Cross-layer contribution scores $C_{s\to\ell}$ on CIFAR-100 with ViT-B/32. Columns vary the sparsifier (JumpReLU, ReLU-Top-k, Abs-Top-k) and rows show CLS (top) and patch tokens (bottom). Each heatmap visualizes the proportional contribution of source layer s to the reconstructed activation at target layer ℓ, averaged over the validation set.



(a) JumpReLU, CLS (b) ReLU-Top-k, CLS (c) Abs-Top-k, CLS

(d) JumpReLU, patches (e) ReLU-Top-k, patches (f) Abs-Top-k, patches

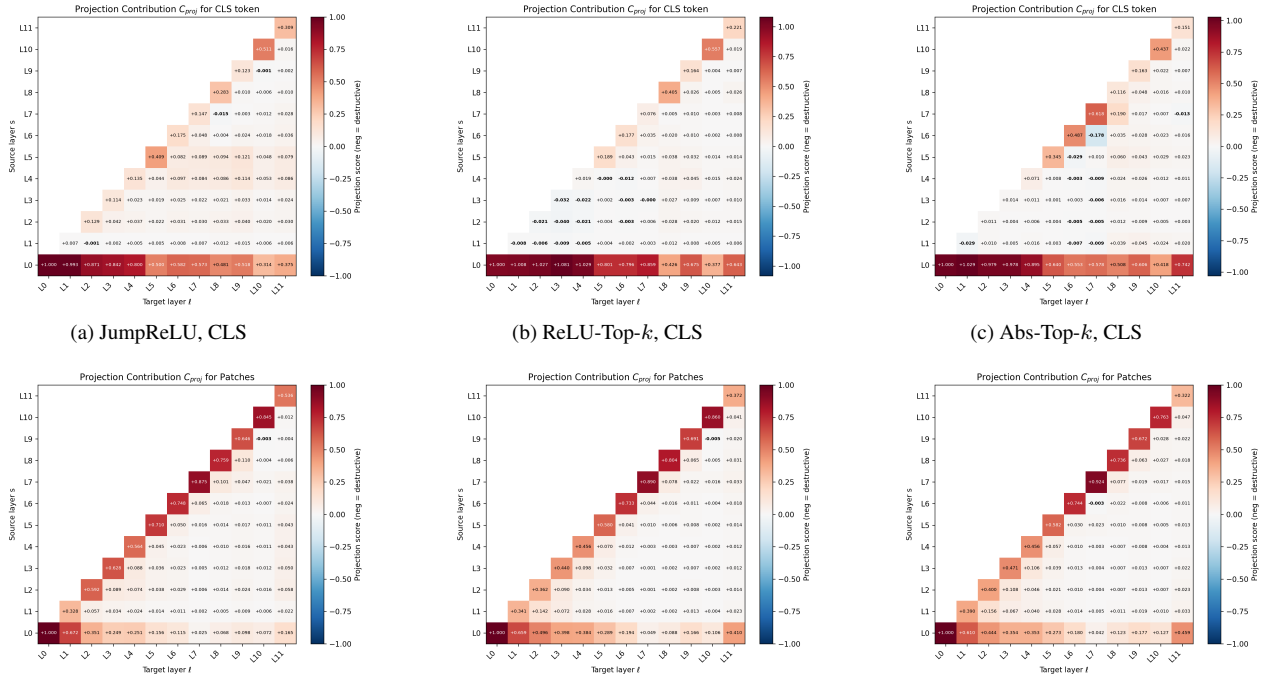
Figure 12. Cross-layer contribution scores $C_{s\to\ell}$ on CIFAR-100 with ViT-B/16. Columns vary the sparsifier (JumpReLU, ReLU-Top-k, Abs-Top-k) and rows show CLS (top) and patch tokens (bottom). Each heatmap visualizes the proportional contribution of source layer s to the reconstructed activation at target layer ℓ, averaged over the validation set.



(a) JumpReLU, CLS (b) ReLU-Top-k, CLS (c) Abs-Top-k, CLS

(d) JumpReLU, patches (e) ReLU-Top-k, patches (f) Abs-Top-k, patches

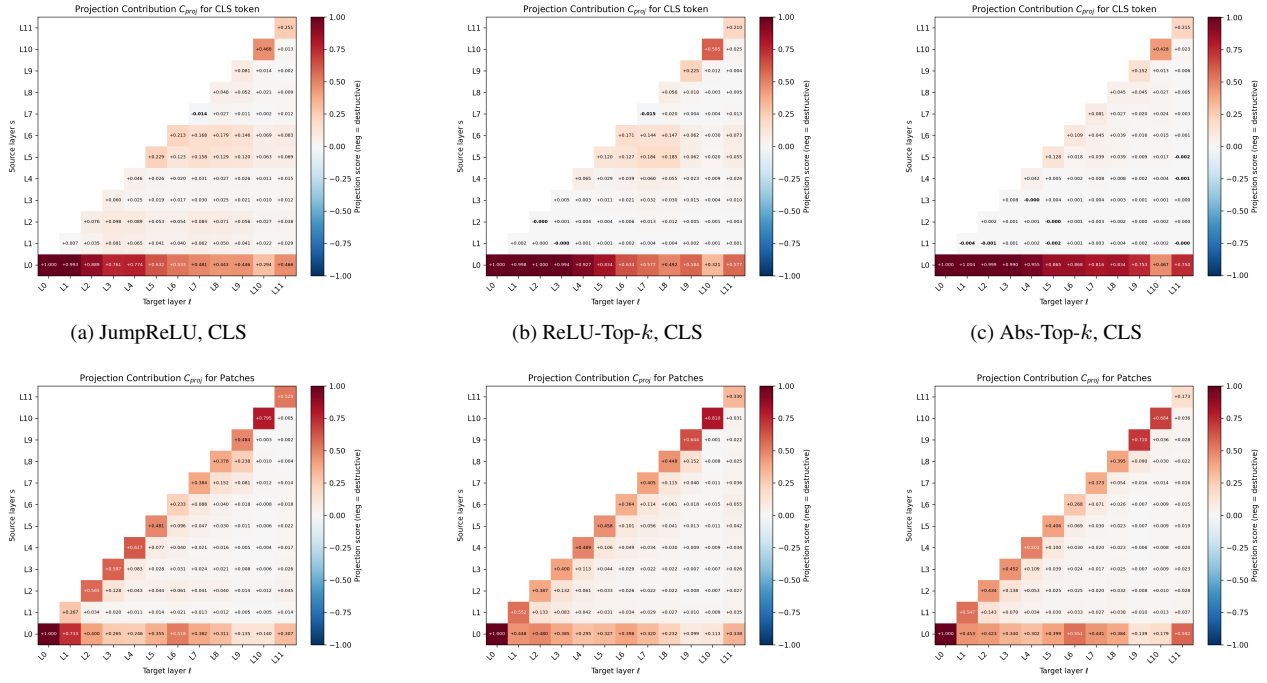
Figure 13. Cross-layer contribution scores $C_{s\to\ell}$ on COCO with ViT-B/32. Columns vary the sparsifier (JumpReLU, ReLU-Top-k, Abs-Top-k) and rows show CLS (top) and patch tokens (bottom). Each heatmap visualizes the proportional contribution of source layer s to the reconstructed activation at target layer ℓ, averaged over the validation set.



(a) JumpReLU, CLS (b) ReLU-Top-k, CLS (c) Abs-Top-k, CLS

(d) JumpReLU, patches (e) ReLU-Top-k, patches (f) Abs-Top-k, patches

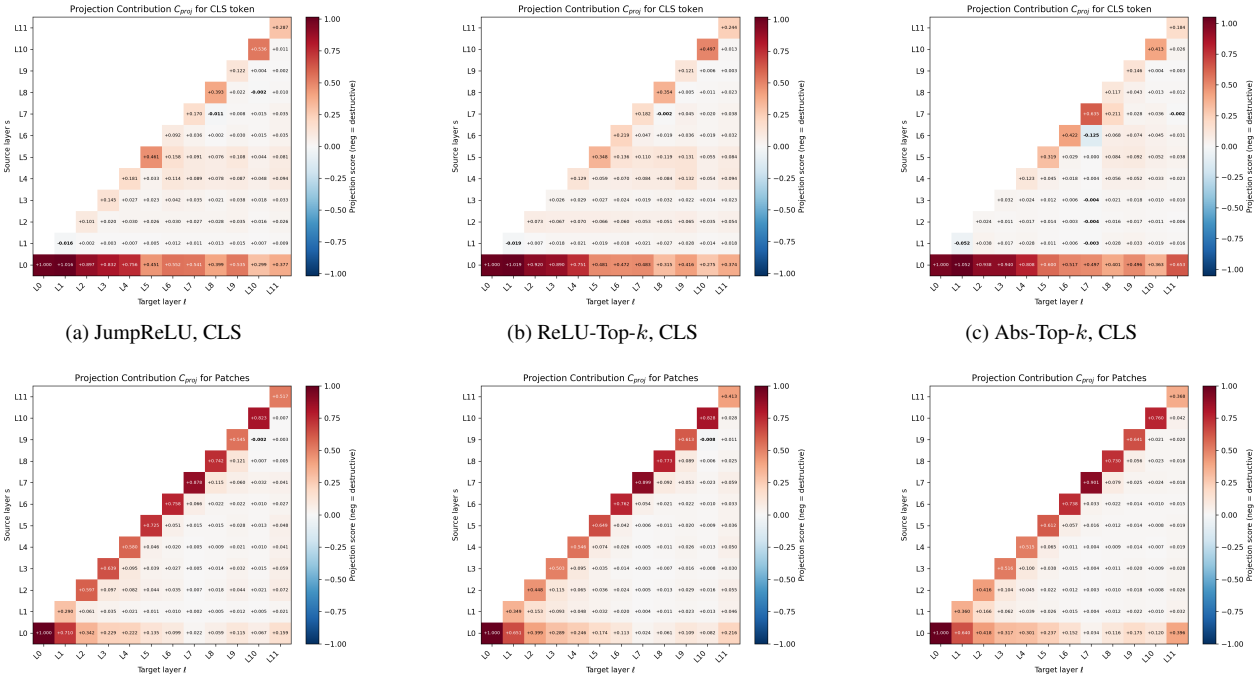
Figure 14. Cross-layer contribution scores $C_{s\to\ell}$ on COCO with ViT-B/16. Columns vary the sparsifier (JumpReLU, ReLU-Top-k, Abs-Top-k) and rows show CLS (top) and patch tokens (bottom). Each heatmap visualizes the proportional contribution of source layer s to the reconstructed activation at target layer ℓ, averaged over the validation set.



(a) JumpReLU, CLS (b) ReLU-Top-k, CLS (c) Abs-Top-k, CLS

(d) JumpReLU, patches (e) ReLU-Top-k, patches (f) Abs-Top-k, patches

Figure 15. Cross-layer contribution scores $C_{s\to\ell}$ on ImageNet-100 with ViT-B/32. Columns vary the sparsifier (JumpReLU, ReLU-Top-k, Abs-Top-k) and rows show CLS (top) and patch tokens (bottom). Each heatmap visualizes the proportional contribution of source layer s to the reconstructed activation at target layer ℓ, averaged over the validation set.



(a) JumpReLU, CLS (b) ReLU-Top-k, CLS (c) Abs-Top-k, CLS

(d) JumpReLU, patches (e) ReLU-Top-k, patches (f) Abs-Top-k, patches

Figure 16. Cross-layer contribution scores $C_{s\to\ell}$ on ImageNet-100 with ViT-B/16. Columns vary the sparsifier (JumpReLU, ReLU-Top-k, Abs-Top-k) and rows show CLS (top) and patch tokens (bottom). Each heatmap visualizes the proportional contribution of source layer s to the reconstructed activation at target layer ℓ, averaged over the validation set.

Table 13. Top-1 classification accuracy (%) on COCO for ViT-B/32 with Patches tokens, across layer ranges $s \rightarrow 11$. JR = JumpReLU, RTK = ReLU-Top-k, ATK = Abs-Top-k. Baseline top-1 accuracies (in %) are ViT-B/32: 43.12 and ViT-B/16: 43.56.

Range	JR	RTK	ATK
0→11	39.04	38.94	**39.68**
1→11	39.38	39.44	**40.02**
2→11	39.54	40.24	**40.42**
3→11	40.58	40.92	**41.48**
4→11	41.38	41.88	**42.10**
5→11	42.06	**42.28**	**42.28**
6→11	42.48	42.58	**42.60**
7→11	42.98	**43.14**	42.80
8→11	**43.14**	43.02	42.92
9→11	**43.14**	42.92	42.92
10→11	43.14	**43.18**	43.06
11→11	**43.12**	**43.12**	**43.12**

Table 14. Top-1 classification accuracy (%) on COCO for ViT-B/32 with All tokens, across layer ranges $s \rightarrow 11$. JR = JumpReLU, RTK = ReLU-Top-k, ATK = Abs-Top-k. Baseline top-1 accuracies (in %) are ViT-B/32: 43.12 and ViT-B/16: 43.56.

Range	JR	RTK	ATK
0→11	38.60	39.12	**40.06**
1→11	39.02	39.62	**40.54**
2→11	39.98	40.10	**40.44**
3→11	41.00	41.34	**41.46**
4→11	41.62	41.68	**41.92**
5→11	41.88	**42.28**	42.26
6→11	**42.88**	42.60	42.82
7→11	**43.14**	43.10	42.76
8→11	**43.26**	**43.26**	42.86
9→11	**43.28**	43.20	42.74
10→11	42.78	**43.10**	42.92
11→11	43.24	**43.26**	43.00

Table 15. Top-1 classification accuracy (%) on COCO for ViT-B/16 with CLS tokens, across layer ranges $s \rightarrow 11$. JR = JumpReLU, RTK = ReLU-Top-k, ATK = Abs-Top-k. Baseline top-1 accuracies (in %) are ViT-B/32: 43.12 and ViT-B/16: 43.56.

Range	JR	RTK	ATK
0→11	**43.62**	**43.62**	42.76
1→11	43.56	**43.62**	42.40
2→11	43.60	**43.64**	42.34
3→11	**43.68**	43.64	42.64
4→11	43.66	**43.72**	42.64
5→11	43.58	**43.80**	42.98
6→11	43.56	**43.66**	43.50
7→11	43.52	**43.64**	43.34
8→11	**43.72**	43.66	43.68
9→11	43.82	**43.84**	43.24
10→11	43.52	**43.88**	43.38
11→11	43.50	**43.62**	43.28

Table 16. Top-1 classification accuracy (%) on COCO for ViT-B/16 with Patches tokens, across layer ranges $s \rightarrow 11$. JR = JumpReLU, RTK = ReLU-Top-k, ATK = Abs-Top-k. Baseline top-1 accuracies (in %) are ViT-B/32: 43.12 and ViT-B/16: 43.56.

Range	JR	RTK	ATK
0→11	**42.72**	42.00	35.46
1→11	**42.84**	41.98	37.28
2→11	**42.68**	42.20	38.36
3→11	**43.00**	42.34	39.46
4→11	**43.00**	42.96	40.86
5→11	42.98	**43.00**	42.58
6→11	**43.38**	43.16	42.98
7→11	43.26	**43.36**	43.30
8→11	43.44	43.56	**43.86**
9→11	43.62	43.70	**43.76**
10→11	43.62	43.64	**43.72**
11→11	**43.50**	**43.50**	**43.50**

Table 17. Top-1 classification accuracy (%) on COCO for ViT-B/16 with All tokens, across layer ranges $s \rightarrow 11$. JR = JumpReLU, RTK = ReLU-Top-k, ATK = Abs-Top-k. Baseline top-1 accuracies (in %) are ViT-B/32: 43.12 and ViT-B/16: 43.56.

Range	JR	RTK	ATK
0→11	**43.00**	42.08	34.16
1→11	**42.92**	42.08	36.06
2→11	**42.74**	42.20	36.82
3→11	**42.78**	42.66	38.18
4→11	42.84	**43.10**	39.80
5→11	43.04	**43.12**	42.22
6→11	43.44	**43.56**	43.04
7→11	**43.62**	43.58	43.48
8→11	**43.72**	43.70	43.66
9→11	43.66	**43.96**	43.56
10→11	43.42	**43.92**	43.60
11→11	43.50	**43.62**	43.28

Table 18. Top-1 classification accuracy (%) on ImageNet-100 for ViT-B/32 with CLS tokens, across layer ranges $s \rightarrow 11$. JR = JumpReLU, RTK = ReLU-Top-k, ATK = Abs-Top-k. Baseline top-1 accuracies (in %) are ViT-B/32: 80.42 and ViT-B/16: 84.34.

Range	JR	RTK	ATK
0→11	**80.92**	80.86	80.26
1→11	**80.84**	80.82	80.32
2→11	80.82	**80.86**	80.24
3→11	**80.86**	80.78	80.16
4→11	80.80	**80.86**	80.24
5→11	80.74	**80.86**	80.44
6→11	80.68	**80.84**	79.92
7→11	80.64	**80.72**	80.16
8→11	80.66	**80.68**	80.06
9→11	80.46	**80.60**	80.10
10→11	80.56	**80.72**	80.18
11→11	**80.54**	**80.54**	80.36

Table 19. Top-1 classification accuracy (%) on ImageNet-100 for ViT-B/32 with Patches tokens, across layer ranges $s \rightarrow 11$. JR = JumpReLU, RTK = ReLU-Top-k, ATK = Abs-Top-k. Baseline top-1 accuracies (in %) are ViT-B/32: 80.42 and ViT-B/16: 84.34.

Range	JR	RTK	ATK
0→11	**71.96**	68.74	60.26
1→11	**72.34**	70.10	64.80
2→11	**73.10**	72.24	67.54
3→11	**75.58**	75.26	72.46
4→11	**77.86**	77.18	75.88
5→11	78.92	**79.10**	77.54
6→11	**79.80**	79.78	79.56
7→11	80.34	80.18	**80.64**
8→11	80.60	80.38	**80.86**
9→11	80.62	80.50	**80.76**
10→11	**80.52**	80.46	80.46
11→11	**80.42**	**80.42**	**80.42**

Table 20. Top-1 classification accuracy (%) on ImageNet-100 for ViT-B/32 with All tokens, across layer ranges $s \rightarrow 11$. JR = JumpReLU, RTK = ReLU-Top-k, ATK = Abs-Top-k. Baseline top-1 accuracies (in %) are ViT-B/32: 80.42 and ViT-B/16: 84.34.

Range	JR	RTK	ATK
0→11	**71.60**	68.90	60.26
1→11	**71.92**	70.10	63.82
2→11	**72.88**	72.32	66.72
3→11	75.12	**75.36**	71.44
4→11	**77.56**	77.10	74.68
5→11	78.74	**78.84**	77.30
6→11	79.50	**79.86**	78.82
7→11	80.18	**80.44**	79.38
8→11	**80.52**	80.24	80.32
9→11	**80.72**	80.48	80.46
10→11	80.38	**80.58**	80.18
11→11	**80.54**	**80.54**	80.36

Table 21. Top-1 classification accuracy (%) on ImageNet-100 for ViT-B/16 with CLS tokens, across layer ranges $s \rightarrow 11$. JR = JumpReLU, RTK = ReLU-Top-k, ATK = Abs-Top-k. Baseline top-1 accuracies (in %) are ViT-B/32: 80.42 and ViT-B/16: 84.34.

Range	JR	RTK	ATK
0→11	**84.54**	84.02	83.28
1→11	**84.50**	84.04	83.52
2→11	**84.44**	83.98	83.50
3→11	**84.56**	84.08	83.74
4→11	**84.44**	84.04	83.82
5→11	**84.42**	84.10	84.20
6→11	**84.52**	84.04	84.24
7→11	**84.66**	84.16	84.40
8→11	**84.60**	84.30	84.46
9→11	**84.62**	84.30	84.42
10→11	**84.56**	84.38	84.34
11→11	**84.46**	84.36	84.36

Table 22. Top-1 classification accuracy (%) on ImageNet-100 for ViT-B/16 with Patches tokens, across layer ranges $s \rightarrow 11$. JR = JumpReLU, RTK = ReLU-Top-k, ATK = Abs-Top-k. Baseline top-1 accuracies (in %) are ViT-B/32: 80.42 and ViT-B/16: 84.34.

Range	JR	RTK	ATK
0→11	**83.04**	81.40	73.06
1→11	**83.00**	81.44	76.74
2→11	**83.18**	81.88	77.54
3→11	**83.16**	82.76	79.36
4→11	**83.56**	83.34	80.98
5→11	83.78	**83.88**	82.50
6→11	**84.20**	84.12	83.26
7→11	83.94	**84.24**	83.84
8→11	84.02	84.24	**84.26**
9→11	84.20	84.22	**84.72**
10→11	84.20	84.22	**84.58**
11→11	**84.34**	**84.34**	**84.34**

Table 23. Top-1 classification accuracy (%) on ImageNet-100 for ViT-B/16 with All tokens, across layer ranges $s \rightarrow 11$. JR = JumpReLU, RTK = ReLU-Top-k, ATK = Abs-Top-k. Baseline top-1 accuracies (in %) are ViT-B/32: 80.42 and ViT-B/16: 84.34.

Range	JR	RTK	ATK
0→11	**83.12**	80.94	70.78
1→11	**83.08**	81.58	74.70
2→11	**83.28**	81.90	76.30
3→11	**83.00**	82.64	78.08
4→11	**83.62**	83.08	80.12
5→11	**84.16**	83.86	81.76
6→11	**84.46**	83.92	82.56
7→11	84.28	**84.32**	83.34
8→11	**84.38**	84.10	84.08
9→11	84.20	84.22	**84.74**
10→11	84.40	84.36	**84.74**
11→11	**84.46**	84.36	84.36