TU WIEN Informatics

Faithful Attention Attribution in Vision Transformers for Chest X-Ray Interpretation

DIPLOMA THESIS

submitted in partial fulfillment of the requirements for the degree of

Diplom-Ingenieur

in

Logic and Computation

by

Julius Šula
Registration Number 11914972

to the Faculty of Informatics

at the TU Wien

Advisor: Univ.Prof. Dipl.-Inf. Dr.rer.nat. Thomas Lukasiewicz
Assistance: Dr. Bayar Ilhan Menzat

Vienna, January 1, 2026

Julius Šula	Thomas Lukasiewicz

Erklärung zur Verfassung der Arbeit

Julius Šula

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.
Ich erkläre weiters, dass ich mich generativer KI-Tools lediglich als Hilfsmittel bedient habe und in der vorliegenden Arbeit mein gestalterischer Einfluss überwiegt. Im Anhang „Übersicht verwendeter Hilfsmittel“ habe ich alle generativen KI-Tools gelistet, die verwendet wurden, und angegeben, wo und wie sie verwendet wurden. Für Textpassagen, die ohne substantielle Änderungen übernommen wurden, haben ich jeweils die von mir formulierten Eingaben (Prompts) und die verwendete IT- Anwendung mit ihrem Produktnamen und Versionsnummer/Datum angegeben.

Wien, 1. Jänner 2026

Julius Šula

Acknowledgements

Abstract

Vision Transformers (ViTs) achieve strong performance in natural and medical imaging, yet their decision processes remain opaque—especially problematic in high-stakes settings like chest X-ray interpretation. TransMM is among the strongest attribution methods for ViTs, combining attention with class-specific gradients to highlight influential image patches. We ask whether injecting *semantic structure* from Sparse Autoencoders (SAEs) can further improve the faithfulness of such attributions.

We introduce *Feature-Gradient Attribution*, which extends TransMM's principle from attention space to *feature* space. SAEs, which are trained on residual streams to decompose activations into sparse, interpretable features provide per-patch feature activations. We project gradients onto the SAE feature basis and compute feature-gradient scores that capture both *which* learned features are present and *how* they influence the target logit. These scores yield per-patch gates that modulate TransMM's attention maps before relevance propagation, forming a lightweight, semantically informed correction.

Across three datasets (chest X-rays, endoscopy, natural images), two architectures (fine-tuned ViT-B/16 and contrastively pre-trained CLIP ViT-B/32), and three complementary faithfulness metrics, our method improves attribution faithfulness consistently. Improvements are statistically significant ($p<0.001$) on all three metrics for two datasets and on two of three metrics for the remaining dataset. We observe gains of 10.5–44.3% on SaCo and 14.0–43.0% on Faithfulness Correlation, with Pixel Flipping improving by 1.3–10.8%. Notably, we never observe degradation relative to TransMM on any metric–dataset combination. To our knowledge, this is the first integration of sparse semantic features *during* attribution computation (rather than post-hoc validation), demonstrating that mechanistic feature structure can materially enhance Transformer attributions while preserving TransMM's simplicity and efficiency.

Contents

CHAPTER 1

Introduction

Vision Transformers [DBK+21] have become increasingly prominent in computer vision, demonstrating competitive or superior performance to convolutional networks across diverse tasks when sufficient training data is available. Their self-attention mechanisms enable modeling of global dependencies, which has proven valuable in domains ranging from natural image classification to medical imaging applications such as chest X-ray diagnosis [MBSP21, KBB23] and gastrointestinal endoscopy [SM24]. However, like other deep neural networks, Vision Transformers' decision-making processes remain largely opaque, creating a critical challenge for deployment in high-stakes applications where understanding *why* a model makes a prediction is as important as the prediction itself.

1.1 The Need for Faithful Explanations in Medical AI

In medical imaging, the stakes of model deployment extend beyond predictive accuracy. Clinicians must understand which image regions drive diagnostic predictions to verify that models have learned clinically meaningful patterns rather than spurious correlations. Research on clinical decision-making demonstrates that multimodal explanations significantly improve physicians' ability to detect model failures and biases [KME+24]. A model that achieves high accuracy by exploiting dataset biases, such as hospital-specific imaging artifacts or irrelevant anatomical markers for instance, poses serious risks in clinical practice. Improving the faithfulness of visual attribution maps therefore directly impacts the quality of explanations that clinicians can use to validate model reasoning alongside other interpretability modalities.

TransMM (Transformer Attribution using Multimodal Mixing) [CGW21b] has established itself as a leading attribution method for Vision Transformers [WKT+24a, KBB23, AHD+24]. Its elegance lies in a simple principle: combine attention weights with gradients to produce class-specific attribution maps. By weighting attention maps with the gradient of the target class with respect to those attention weights, TransMM identifies which

patches influence predictions while accounting for the complex information flow through transformer architectures.

1.2 Sparse Autoencoders: A New Lens on Model Internals

Recent advances in mechanistic interpretability have introduced Sparse Autoencoders (SAEs) as a powerful tool for understanding what models learn [BTB+23, HCS+24]. SAEs address a fundamental challenge in neural network interpretability: individual neurons typically respond to multiple unrelated concepts (polysemanticity), making it difficult to understand what the network has learned. By learning overcomplete sparse representations, SAEs decompose these entangled activations into interpretable features where each dimension corresponds to a distinct semantic concept extending from low-level textures to high-level objects and abstract patterns.

While SAEs were initially developed for language models, recent work has demonstrated their effectiveness in vision domains [JSH+25, Ano25]. Features learned by SAEs in Vision Transformers exhibit clear semantic structure: early layers capture edges and textures, middle layers encode object parts and materials, and late layers represent complete objects and scenes [LCCS25, OMS17]. Critically, these features are not merely correlational but interventions that activate or suppress specific SAE features produce predictable changes in model behavior [Ano25, SCBWS25], confirming they capture genuine computational primitives used during inference.

1.3 The Attribution-Interpretability Gap

While attribution methods and mechanistic interpretability have both advanced significantly, they have largely evolved in parallel, creating a gap between explaining *where* models attend and understanding *what semantic concepts* drive those decisions. Recent work has begun exploring connections between these paradigms, though in fundamentally different ways than our approach.

Han et al. [HKK25] use attribution methods to *explain SAE features*, identifying which image regions causally drive specific feature activations through Effective Receptive Field analysis. Their work addresses the challenge that SAE features in later transformer layers often exhibit non-localized activation patterns, using attribution to reveal the true spatial causes of feature firing. Stevens et al. [SCBWS25] take a complementary approach, using SAE features to *validate attribution hypotheses post-hoc*. When an attribution method like Grad-CAM highlights certain pixels as important, they manipulate the corresponding SAE features to test whether those regions actually drive model behavior as claimed.

However, neither approach integrates semantic feature information *directly into the attribution computation itself*. Han et al. use attribution as a tool to understand features, while Stevens et al. use features as a tool to evaluate attribution—but neither examines whether incorporating feature-level semantic structure during attribution can improve

faithfulness. This represents a fundamental gap: if SAE features capture meaningful semantic concepts and attribution methods aim to identify important image regions, can we leverage these learned semantic representations to enhance attribution quality?

1.4 Extending TransMM's Principle to Feature Space

This thesis investigates whether TransMM's core principle of combining activation patterns with gradient information can be extended from attention maps to semantic feature space, yielding more faithful attributions. TransMM's success stems from recognizing that attention weights alone are insufficient: knowing where the model attends (attention patterns) must be combined with knowing how that attention affects predictions (gradients). We propose that a parallel limitation exists for gradients themselves: raw gradients aggregate influences across all learned representations, potentially conflating relevant semantic features with noise.

Our approach extends TransMM's principle to the feature level. For each image patch, we extract sparse feature activations from the model's residual stream using SAEs and compute gradients with respect to these features. The product of feature activation (indicating which semantic concepts are present) and feature gradient (indicating how those concepts influence the prediction) provides a semantic importance score. Features that are both active in a patch and gradient-aligned with the target prediction receive high scores, while features that are absent or irrelevant receive low scores. We aggregate these feature-gradient scores and use them to modulate TransMM's attention maps before relevancy propagation, creating a correction mechanism that respects the model's learned semantic structure.

To our knowledge, this is the first work to modulate gradient-based attribution using sparse semantic features during the attribution computation itself, rather than for post-hoc validation or feature interpretation. Unlike prior work that uses attribution to interpret SAE features [HKK25] or SAE features to validate attribution post-hoc [SCBWS25], we investigate whether incorporating semantic structure *during* attribution can yield faithfulness improvements, and provide empirical evidence of substantial gains across multiple datasets and metrics. The method maintains TransMM's architectural simplicity—requiring only SAE inference and element-wise operations—while adding minimal computational overhead.

Through comprehensive evaluation across three diverse datasets (chest X-rays, endoscopic images, and natural images), two model architectures (fine-tuned ViT-B/16 and contrastively pre-trained CLIP ViT-B/32), and three complementary faithfulness metrics, we demonstrate that our method consistently outperforms the TransMM baseline across every experimental configuration. The improvements are substantial and universal: our method improves performance on all three faithfulness metrics across all three datasets, never degrading performance relative to baseline. SaCo and Faithfulness Correlation show gains of 10.5-44.3% and 14.0-43.0% respectively, with statistically significant improvements ($p<0.001$) on these metrics for all datasets. Pixel Flipping shows consistent

improvements of 1.3-10.8%, with statistical significance achieved on two of three datasets. This universal consistency across diverse experimental conditions, including both single-layer and multi-layer configurations, provides strong evidence that incorporating semantic structure through SAE features represents a genuine advancement in attribution quality for Vision Transformers.

1.5 Contributions

This thesis makes the following contributions to attribution methods for Vision Transformers:

1. **Novel integration of mechanistic interpretability with attribution**: We present a method that incorporates SAE feature-gradient decomposition directly into attribution computation, modulating attention maps before relevancy propagation. Unlike prior work that uses attribution to explain features [HKK25] or features to validate attribution [SCBWS25], we demonstrate that semantic features can enhance attribution faithfulness when integrated into the computation itself.

2. **Principled feature-gradient decomposition**: We develop a method to decompose residual stream gradients through SAE feature space, identifying which semantic features drive patch importance. This decomposition provides interpretable per-patch scores that reflect both feature presence and gradient alignment, extending TransMM's attention-gradient principle to the feature level.

3. **Practical integration with state-of-the-art attribution**: We show how to incorporate feature-gradient signals into TransMM through attention map modulation, preserving its computational efficiency and architectural simplicity while leveraging semantic structure from mechanistic interpretability.

4. **Comprehensive empirical validation**: We demonstrate substantial improvements across three datasets (two medical imaging tasks and natural images), two model architectures (fine-tuned and contrastively pre-trained), and three complementary faithfulness metrics, with improvements of 12-35% over the TransMM baseline.

5. **Methodological insights**: Through systematic ablation studies, we establish that attribution improvement requires the interaction between feature activations and gradients, neither signal alone suffices. We also identify optimal layer ranges and demonstrate that multi-layer feature combinations provide synergistic benefits.

1.6 Research Questions

This thesis addresses the following research questions:

1. **Can SAE features improve attribution faithfulness in Vision Transformers?** We investigate whether incorporating semantic features into gradient-based attribution yields measurable improvements across established faithfulness metrics including correlation, magnitude alignment, and ranking quality.

2. **What is the optimal way to combine feature and gradient information?** Through ablation studies, we test whether attribution improvement stems from feature activations, decomposed gradients, or specifically their interaction, determining which signal components are necessary for faithful attribution.

3. **Does the approach generalize across domains and architectures?** We evaluate robustness across medical imaging (chest X-rays, endoscopy) and natural images, fine-tuned and contrastively pre-trained models, and different Vision Transformer architectures to assess broad applicability.

1.7 Thesis Structure

The remainder of this thesis is organized as follows:

Chapter 2 provides background on Vision Transformers, covering their architecture and self-attention mechanisms that make attribution challenging.

Chapter 3 reviews attribution methods for Vision Transformers, tracing the evolution from Layer-wise Relevance Propagation through TransMM and establishing the state-of-the-art baseline for our work.

Chapter 4 introduces faithfulness metrics for evaluating attribution quality, discussing the complementary aspects captured by different evaluation protocols and justifying our choice of metrics.

Chapter 5 presents mechanistic interpretability and Sparse Autoencoders, explaining how SAEs decompose polysemantic neurons into interpretable features and reviewing recent applications to vision models.

Chapter 6 details our Feature-Gradient Attribution method, including the mathematical framework, gate construction, integration with TransMM, and implementation considerations.

Chapter 7 presents comprehensive experimental evaluation across three datasets, including validation experiments to identify optimal configurations and test set results demonstrating substantial faithfulness improvements.

Chapter 8 discusses implications, limitations, and future directions, concluding with the broader impact on trustworthy AI for medical imaging.

CHAPTER 2

Background: Vision Transformers

2.1 Transformer

The Transformer architecture [VSP+17] represented a turning point in deep learning, addressing fundamental limitations of recurrent neural networks (RNNs). Unlike RNNs, which process sequences sequentially and suffer from vanishing gradients over long distances [KK01], Transformers enable fully parallel computation through their attention mechanism. This allows efficient training on modern hardware and effective modeling of long-range dependencies, making Transformers the architecture of choice for processing sequential data.

Originally introduced for neural machine translation, the Transformer's core innovation of the self-attention mechanism has proven remarkably versatile. The architecture has since been adapted to diverse domains, from natural language understanding [DCLT19] to protein structure prediction [JEP+21] and, critically for this thesis, computer vision [DBK+21]. Understanding the Transformer's components is essential for analyzing how attribution methods must account for its unique information flow patterns, particularly the global receptive field enabled by self-attention.

2.1.1 Encoder and Decoder

The transformer architecture consists of two primary components operating in tandem. The **encoder** processes the input sequence and transforms it into a continuous representation that captures semantic and contextual information. It consists of N identical layers (typically 6 in the original architecture), each containing two sub-layers: a multi-head self-attention mechanism and a position-wise fully connected feed-forward network. Each sub-layer employs residual connections [HZRS16] followed by layer normalization [BKH16].

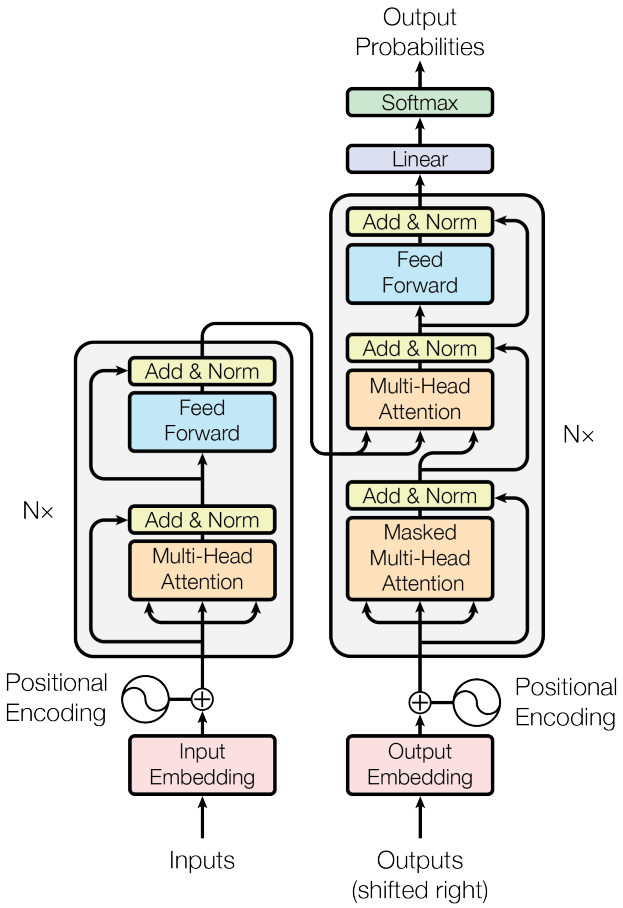


Figure 2.1: The Transformer architecture with encoder (left) and decoder (right) stacks. Each encoder layer contains multi-head self-attention and feed-forward networks, while decoder layers add cross-attention to encoder outputs. Residual connections and layer normalization are applied throughout. Figure adapted from [VSP+17].

The **decoder** generates the output sequence autoregressively, producing one token at a time based on previously generated tokens and the encoder's output. Like the encoder, it consists of N identical layers, but with three sub-layers: masked multi-head self-attention (preventing positions from attending to subsequent positions), encoder-decoder attention (attending to the encoder's output), and a position-wise feed-forward network.

In practice, many modern applications use only the encoder (e.g., BERT [DCLT19]) or only the decoder (e.g., GPT [RNSS18]). Vision Transformers [DBK+21], which form the basis of this thesis, employ only the encoder architecture, as image classification does not require autoregressive generation.

2.1.2 Attention

The attention mechanism enables the model to weigh the importance of different parts of the input when processing each position. Given an input representation, the mechanism computes three vectors for each token: a **query** (**Q**), representing what the token is looking for; a **key** (**K**), representing what the token offers; and a **value** (**V**), representing the actual information to be aggregated.

Scaled Dot-Product Attention The core attention operation computes a weighted sum of values, where weights are determined by the compatibility between queries and keys:

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d_k}}\right)\mathbf{V} \tag{2.1}$$

where d_k is the dimension of the key vectors. The scaling factor $\frac{1}{\sqrt{d_k}}$ prevents the dot products from growing too large in magnitude, which would push the softmax function into regions with extremely small gradients.

The softmax operation converts the compatibility scores into a probability distribution over all positions, ensuring the attention weights sum to one. High compatibility between a query and key results in high attention weight, meaning that value's information is strongly incorporated.

Multi-Head Attention Rather than performing a single attention operation, multi-head attention projects the queries, keys, and values h times with different learned linear projections to d_{model}, d_k, and d_v dimensions respectively. This allows the model to jointly attend to information from different representation subspaces:

$$\text{MultiHead}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{Concat}(\text{head}_1, \ldots, \text{head}_h)\mathbf{W}^O \tag{2.2}$$

$$\text{where head}_i = \text{Attention}(\mathbf{Q}\mathbf{W}_i^Q, \mathbf{K}\mathbf{W}_i^K, \mathbf{V}\mathbf{W}_i^V) \tag{2.3}$$

where the projections are parameter matrices $\mathbf{W}_i^Q \in \mathbb{R}^{d_{\text{model}} \times d_k}$, $\mathbf{W}_i^K \in \mathbb{R}^{d_{\text{model}} \times d_k}$, $\mathbf{W}_i^V \in \mathbb{R}^{d_{\text{model}} \times d_v}$, and $\mathbf{W}^O \in \mathbb{R}^{hd_v \times d_{\text{model}}}$. In the original work, $h = 8$ parallel attention layers with $d_k = d_v = d_{\text{model}}/h = 64$.

Multi-head attention enables the model to capture different types of relationships simultaneously—for instance, one head might focus on syntactic dependencies while another captures semantic similarities.

2.1.3 Feed Forward Networks

Each transformer layer contains a position-wise feed-forward network applied identically to each position separately:

$$\text{FFN}(\mathbf{x}) = \max(0, \mathbf{x}\mathbf{W}_1 + \mathbf{b}_1)\mathbf{W}_2 + \mathbf{b}_2 \tag{2.4}$$

This consists of two linear transformations with a ReLU activation in between. While attention mechanisms determine *where* to look, feed-forward networks process *what* has been aggregated. The inner layer typically has dimensionality $d_{ff} = 2048$, four times larger than the model dimension $d_{\text{model}} = 512$, providing substantial representational capacity.

Positional Encoding Since attention operations are permutation-invariant, transformers explicitly encode positional information by adding learned positional embeddings $\mathbf{p}_i$ to input embeddings: $\mathbf{z}_i = \mathbf{x}_i + \mathbf{p}_i$. In Vision Transformers, these embeddings encode the 2D spatial location of image patches, enabling the model to leverage spatial relationships [DBK+21].

2.2 Vision Transformers (ViT)

While Transformers achieved remarkable success in natural language processing, their application to computer vision remained limited until Dosovitskiy et al. [DBK+21] demonstrated that a pure transformer architecture could match or exceed convolutional neural networks on image classification tasks. This breakthrough challenged the prevailing assumption that the inductive biases of CNNs of translation equivariance and locality were essential for learning visual representations.

2.2.1 Architecture

Vision Transformers adapt the transformer encoder for image classification through a conceptually elegant approach: treating an image as a sequence of patches. An input image $\mathbf{x} \in \mathbb{R}^{H \times W \times C}$ is reshaped into a sequence of flattened 2D patches $\mathbf{x}_p \in \mathbb{R}^{N \times (P^2 \cdot C)}$, where:

- (H, W) is the original image resolution
- C is the number of channels (typically 3 for RGB images)
- (P, P) is the resolution of each patch (commonly 16×16 or 32×32)
- $N = \frac{HW}{P^2}$ is the resulting number of patches

For a standard 224×224 image with 16×16 patches, this yields $N = 196$ patch tokens.

Patch Embedding and Position Encoding Each flattened patch is linearly projected to the model dimension D through a trainable embedding matrix $\mathbf{E} \in \mathbb{R}^{(P^2 \cdot C) \times D}$:

$$\mathbf{z}_0 = [\mathbf{x}_{\text{class}}; \mathbf{x}_p^1 \mathbf{E}; \mathbf{x}_p^2 \mathbf{E}; \ldots; \mathbf{x}_p^N \mathbf{E}] + \mathbf{E}_{\text{pos}} \tag{2.5}$$

where $\mathbf{x}_{\text{class}}$ is a learnable class token prepended to the sequence, serving as an aggregated image representation for classification, analogous to BERT's [CLS] token [DCLT19]. The positional embeddings $\mathbf{E}_{\text{pos}} \in \mathbb{R}^{(N+1) \times D}$ are learned parameters that encode the 2D spatial location of each patch, enabling the model to leverage spatial structure.

Transformer Encoder The sequence of embedded patches is processed by L transformer encoder layers (typically $L = 12$ for ViT-Base). Each layer applies multi-head self-attention followed by a feed-forward network, with residual connections and layer normalization:

$$\mathbf{z}'_\ell = \text{MSA}(\text{LN}(\mathbf{z}_{\ell-1})) + \mathbf{z}_{\ell-1} \tag{2.6}$$

$$\mathbf{z}_\ell = \text{MLP}(\text{LN}(\mathbf{z}'_\ell)) + \mathbf{z}'_\ell \tag{2.7}$$

where MSA denotes multi-head self-attention, LN is layer normalization, and MLP is the position-wise feed-forward network.

Classification Head After L layers, the class token's final representation $\mathbf{z}_L^0$ is extracted and passed through a classification head—typically an MLP with one hidden layer during pre-training and a single linear layer for fine-tuning:

$$\mathbf{y} = \text{softmax}(\mathbf{W}_{\text{head}} \mathbf{z}_L^0) \tag{2.8}$$

Figure 2.2 illustrates the complete architecture.

2.2.2 Scaling Behavior and Comparison to CNNs

Vision Transformers exhibit fundamentally different learning characteristics compared to convolutional neural networks [LBBH98, HZRS16]. A critical finding of Dosovitskiy et al. [DBK+21] is that ViTs require substantially larger training datasets to achieve competitive performance. When trained on moderately sized datasets, ViT-Base underperforms comparable ResNets by several percentage points. The authors attribute this to ViTs lacking the inductive biases that help CNNs generalize from limited data.

However, this relationship reverses at scale. When pre-trained on large datasets, Vision Transformers match or exceed state-of-the-art CNNs while requiring less compute to train [DBK+21]. This scaling behavior reflects a fundamental trade-off: ViTs must learn visual relationships from data rather than having them encoded architecturally, but this flexibility enables superior performance when sufficient training data is available.

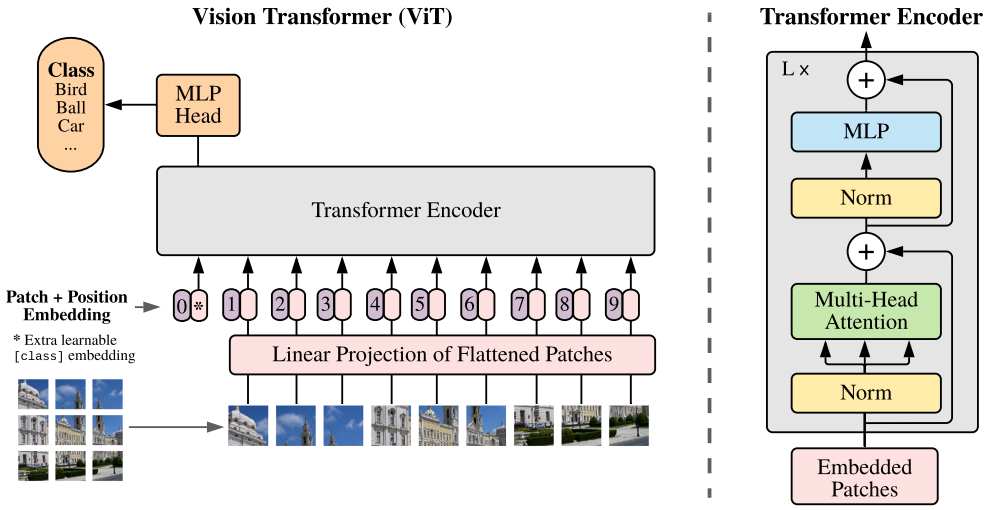


Figure 2.2: Vision Transformer (ViT) architecture. An input image is divided into fixed-size patches, linearly embedded, and augmented with position embeddings. A learnable class token is prepended to the sequence, which is then processed by a standard transformer encoder. The final state of the class token serves as the image representation for classification. Figure adapted from [DBK+21].

How ViTs Differ from CNNs Beyond performance differences, Raghu et al. [RUK+21] demonstrated that Vision Transformers process visual information fundamentally differently than CNNs. Key distinctions include:

- **Uniform representations**: ViTs maintain more consistent representations across layers, contrasting with CNNs' hierarchical progression from edges to textures to objects
- **Early global aggregation**: Self-attention enables aggregation of global information from the first layer, while CNNs build receptive fields gradually
- **Strong feature propagation**: ViT residual connections propagate features from lower to higher layers more strongly than in CNNs
- **Spatial information preservation**: Despite global mixing, ViTs preserve input spatial information throughout the network, though with different characteristics than CNNs' explicit spatial structure

Understanding how ViTs aggregate information globally while preserving spatial structure is essential for developing effective explanation methods for these models, which we will explore later in chapter 3.

2.2.3 Note on Pre-training Paradigms

While the ViT architecture described above is typically trained with supervised classification objectives, alternative pre-training strategies can produce models with different learned representations despite identical architectures. CLIP (Contrastive Language-Image Pre-training) [RKH+21] trains ViT encoders using contrastive learning on image-text pairs, learning to align visual and linguistic representations rather than predict fixed class labels.

At inference, CLIP's vision encoder operates identically to a standard ViT, processing images through the same patch embedding and transformer layers. However, the contrastive pre-training objective encourages learning features that align with natural language descriptions, potentially creating different semantic structures than supervised fine-tuning on specific visual tasks [RKH+21]. This thesis evaluates attribution methods on both fine-tuned ViTs (COVID-QU-Ex, Hyperkvasir) and a contrastively pre-trained CLIP model (ImageNet) to assess whether feature-gradient attribution generalizes across these different training paradigms.

2.2.4 Vision Transformers for Medical Imaging

Vision Transformers have demonstrated strong performance across diverse medical imaging modalities and tasks. Recent work has successfully applied ViTs to chest X-ray diagnosis [MBSP21, KBB23], computed tomography analysis [MBSP21], gastrointestinal endoscopy [SM24], as well as dermatology, pathology, and ophthalmology [CJJ+24], often achieving performance comparable to or exceeding convolutional approaches when sufficient training data is available [DBK+21]. While ViTs can achieve strong predictive performance in these domains, their decision-making process remains opaque, creating a critical need for reliable attribution methods that can identify which image regions drive predictions.

CHAPTER 3

Attribution Methods for Vision Transformers

Vision Transformers have emerged as powerful alternatives to convolutional neural networks for image classification [DBK+21], yet their decision-making process remains opaque. Attribution methods seek to identify which input regions contribute to model predictions, enabling model debugging, verification of learned concepts, and detection of spurious correlations. However, the unique architectural properties of Transformers consisting of global receptive fields through self-attention, complex information flow via skip connections, and non-standard activation functions fundamentally challenge attribution methods developed for convolutional networks.

This chapter traces how attribution methods evolved to address these Transformer-specific challenges. We begin by examining why Transformers are particularly difficult to explain and why the seemingly obvious approach of using attention weights as attribution fails to capture the full computational picture (3.1). We then explore the propagation-based methods Layer-wise Relevance Propagation and Deep Taylor Decomposition, that form the theoretical foundation for principled Transformer attribution (3.2). Finally, we examine TransLRP's extension of these principles to Transformers (3.3) and TransMM's simplification that achieves comparable performance through direct attention-gradient combination (3.4). TransMM serves as a strong and widely-adopted baseline for our work [WKT+24a, KBB23, AHD+24], and understanding its core principle of combining attention patterns with gradient information motivates our investigation of whether this principle extends to sparse semantic features.

3.1 The Attribution Problem in Transformers

The architecture of Vision Transformers presents fundamental challenges for attribution methods originally developed for convolutional networks. Unlike CNNs, which process

images through hierarchical local operations, Transformers employ self-attention mechanisms that allow every image patch to interact with every other patch from the first layer [VSP+17]. This global receptive field means that information aggregation is not spatially constrained, making it difficult to trace which input regions influence specific predictions.

Furthermore, Transformers incorporate architectural components that violate the assumptions of traditional attribution methods. Skip connections create multiple information pathways through the network [HZRS16], LayerNorm operations rescale contributions in complex ways [BKH16], and GELU activation functions produce both positive and negative values [HG17], unlike the strictly non-negative ReLU activations assumed by many attribution techniques. Perhaps most distinctively, Vision Transformers aggregate information through a learned CLS token rather than through spatial pooling operations, fundamentally changing how global information is synthesized for classification [DBK+21].

3.1.1 Attention as Attribution

The self-attention mechanism in Transformers naturally produces attention weight matrices that superficially appear to indicate which tokens the model is "looking at" when making predictions. This has led to a family of attention-based attribution methods that attempt to interpret these weights directly, from using raw attention weights in single layers to more sophisticated aggregation schemes like attention rollout and attention flow [AZ20], which attempt to trace how attention propagates through multiple layers.

However, these attention-based methods suffer from fundamental limitations that undermine their reliability as explanations. Jain and Wallace [JW19] demonstrated that attention weights correlate only weakly with gradient-based feature importance measures, and that alternative attention distributions can often yield equivalent predictions. Meaning the original attention weights do not provide unique explanations for model outputs. More problematically, Pruthi et al. [PGD+20] showed that models can be deliberately trained to produce deceptive attention patterns: assigning minimal attention to certain features while continuing to rely heavily on them for predictions. In human studies, these manipulated attention distributions successfully deceived evaluators into believing biased models did not use problematic features like gender, despite the models being demonstrably biased.

Beyond these empirical findings, attention weights have inherent conceptual limitations. They represent normalized probabilities over tokens, not causal attributions of importance for instance a token receiving high attention might contribute negatively to the prediction, while a token with low attention might be crucial due to its specific features. Moreover, attention weights only capture the mixing of information in self-attention layers, completely ignoring the substantial computations performed by MLP blocks. Skip connections preserve information outside the attention mechanism, LayerNorm operations rescale contributions in complex ways, and the learned parameters in projection matrices fundamentally transform representations [HZRS16, BKH16]. By focusing solely

on attention weights, these methods provide an incomplete and potentially misleading view of how Transformers process information.

3.2 Foundations of Propagation-Based Methods

To understand modern attribution methods for Transformers, we must first examine the propagation-based techniques from which they evolved. These methods, particularly Layer-wise Relevance Propagation and its theoretical connections to Taylor decomposition, form the foundation for state-of-the-art Transformer attribution methods like TransLRP and TransMM.

3.2.1 Layer-wise Relevance Propagation

Layer-wise Relevance Propagation (LRP), introduced by Bach et al. [LBM+15], provides a framework for decomposing neural network predictions through a backward message-passing procedure. The fundamental distinction between LRP and gradient-based methods lies in the question each addresses. While gradients answer "How would the output change if we perturbed this input?", LRP answers "How much did this input contribute to the actual output?"

LRP operates on the principle of **relevance conservation**. Starting from the network's output $f(\mathbf{x})$, relevance flows backward through the network layers without being created or destroyed. Formally, if $R_i^{(\ell)}$ denotes the relevance of neuron i in layer ℓ, conservation requires:

$$\sum_i R_i^{(\ell)} = \sum_j R_j^{(\ell+1)} = \ldots = f(\mathbf{x}) \tag{3.1}$$

This conservation principle ensures that the attribution is complete, meaning every part of the model's decision is accounted for in the final attribution map.

The practical implementation of LRP requires defining **propagation rules** that specify how relevance flows from neurons in layer $\ell + 1$ to neurons in layer ℓ. For a standard feedforward layer where neuron j's activation is computed as $a_j = g(\sum_i a_i w_{ij} + b_j)$ with activation function g, the contribution from neuron i to neuron j is the weighted activation $z_{ij} = a_i w_{ij}$.

LRP redistributes relevance proportionally to these contributions. The basic LRP rule (known as LRP-0) is:

$$R_i^{(\ell)} = \sum_j \frac{z_{ij}}{\sum_k z_{kj}} R_j^{(\ell+1)} \tag{3.2}$$

However, this basic rule suffers from numerical instability when $\sum_k z_{kj} \approx 0$. To address this, two refined rules were developed:

The ϵ-Rule Adding a small stabilizer ϵ to prevent division by zero:

$$R_i^{(\ell)} = \sum_j \frac{z_{ij}}{\sum_k z_{kj} + \epsilon \cdot \text{sign}(\sum_k z_{kj})} R_j^{(\ell+1)} \tag{3.3}$$

The $\alpha\beta$-Rule Treating positive and negative contributions separately:

$$R_i^{(\ell)} = \sum_j \left(\alpha \frac{z_{ij}^+}{\sum_k z_{kj}^+} - \beta \frac{z_{ij}^-}{\sum_k z_{kj}^-} \right) R_j^{(\ell+1)} \tag{3.4}$$

where $z^+ = \max(0, z)$ and $z^- = \min(0, z)$ denote positive and negative parts. The constraint $\alpha - \beta = 1$ ensures conservation, with common choices being $(\alpha = 2, \beta = 1)$ to emphasize positive evidence or $(\alpha = 1, \beta = 0)$ to ignore negative contributions entirely.

While LRP was initially developed through intuitive principles and empirical validation, its propagation rules were later shown to have deep theoretical foundations through the framework of Deep Taylor Decomposition.

3.2.2 Deep Taylor Decomposition

Deep Taylor Decomposition (DTD), introduced by Montavon et al. [MLB+17], provides a theoretical framework that connects LRP's empirically successful propagation rules to fundamental mathematical principles. Rather than replacing LRP, DTD demonstrates that these rules can be derived from first principles using Taylor expansions, thereby providing theoretical justification for their effectiveness.

The key insight of DTD is distinguishing between **sensitivity** and **attribution**. Standard backpropagation computes the gradient $\nabla_{\mathbf{x}} f$ at the input point, measuring sensitivity to infinitesimal changes. In contrast, attribution asks: "How much did each input feature contribute to the actual output value $f(\mathbf{x})$?"

DTD answers this through Taylor expansion around a carefully chosen **root point $\tilde{\mathbf{x}}$** where $f(\tilde{\mathbf{x}}) = 0$. This root point serves as a neutral reference—for a binary classifier, it represents a point exactly on the decision boundary where the model has zero confidence. The first-order Taylor expansion is:

$$f(\mathbf{x}) \approx f(\tilde{\mathbf{x}}) + \sum_p \frac{\partial f}{\partial x_p}\bigg|_{\tilde{\mathbf{x}}} \cdot (x_p - \tilde{x}_p) = \sum_p \frac{\partial f}{\partial x_p}\bigg|_{\tilde{\mathbf{x}}} \cdot (x_p - \tilde{x}_p) \tag{3.5}$$

Each term in this sum defines the relevance for input feature p:

$$R_p = \frac{\partial f}{\partial x_p}\bigg|_{\tilde{\mathbf{x}}} \cdot (x_p - \tilde{x}_p) \tag{3.6}$$

This formulation automatically satisfies the conservation principle: $\sum_p R_p = f(\mathbf{x})$.

However, finding a global root point for a deep network is computationally intractable. DTD's crucial innovation is a **layer-wise decomposition strategy**. Instead of decomposing the entire function at once, DTD decomposes the computation at each layer independently. For each neuron j in layer $\ell+1$ with relevance $R_j^{(\ell+1)}$, DTD applies a local Taylor expansion to redistribute this relevance to the neurons in layer ℓ.

Consider a ReLU neuron with pre-activation $z_j = \sum_i x_i w_{ij} + b_j$ and activation $a_j = \max(0, z_j)$. To redistribute relevance $R_j^{(\ell+1)}$, DTD performs a Taylor expansion of the local function mapping inputs $\{x_i\}$ to the relevance. The choice of root point determines the resulting propagation rule:

Deriving LRP-0 Choosing the root at the origin $\tilde{\mathbf{x}} = \mathbf{0}$ yields:

$$R_i^{(\ell)} = \sum_j \frac{x_i w_{ij}}{\sum_k x_k w_{kj}} R_j^{(\ell+1)} \tag{3.7}$$

This matches the basic LRP rule when neurons have ReLU activations (ensuring $x_i \geq 0$).

Deriving the z^+-Rule For networks with positive activations, choosing a root that zeros only negative contributions leads to:

$$R_i^{(\ell)} = \sum_j \frac{x_i w_{ij}^+}{\sum_k x_k w_{kj}^+} R_j^{(\ell+1)} \tag{3.8}$$

where $w^+ = \max(0, w)$. This corresponds to the LRP-$\alpha\beta$ rule with $\alpha = 1, \beta = 0$.

The DTD framework thus unifies the intuitive conservation principles of LRP with the mathematical rigor of Taylor decomposition. This theoretical foundation proves essential when extending these methods to more complex architectures like Transformers, where the interplay of attention mechanisms, skip connections, and normalization layers requires principled handling of relevance flow.

3.3 TransLRP: Extending LRP to Transformers

TransLRP [CGW21a] extends Layer-wise Relevance Propagation to Transformer architectures by addressing several methodological challenges: handling skip connections and attention operations without numerical instabilities, supporting non-ReLU activation functions like GELU, and providing class-specific visualizations.

3.3.1 Notation and Foundations

Following [CGW21a], let C be the number of classes and $t \in \{1, \ldots, C\}$ the target class to visualize. The network consists of N layers where $x^{(n)}$ denotes the input to layer $L^{(n)}$, with $n \in \{1, \ldots, N\}$. Note that $x^{(N)}$ is the network input and $x^{(1)}$ is the network output (reverse numbering).

The relevance propagation follows the Deep Taylor Decomposition principle [MLB+17]:

$$R_j^{(n)} = G(X, Y, R^{(n-1)}) = \sum_i X_j \frac{\partial L_i^{(n)}(X,Y)}{\partial X_j} \frac{R_i^{(n-1)}}{L_i^{(n)}(X,Y)} \tag{3.9}$$

where X and Y are typically the input feature map and weights for layer n.

Target Class Selection The target class t for computing gradients can be either the predicted class (for understanding what the model "saw") or any other class (for counterfactual analysis). This flexibility enables debugging misclassifications by visualizing what evidence the model found for both correct and incorrect classes.

3.3.2 Handling Different Activation Functions

For ReLU networks where activations are non-negative, the LRP rule simplifies to [LBM+15]:

$$R_j^{(n)} = \sum_i \frac{x_j^+ w_{ji}^+}{\sum_{j'} x_{j'}^+ w_{j'i}^+} R_i^{(n-1)} \tag{3.10}$$

where $v^+ = \max(0, v)$ denotes positive values only.

Since Transformers use GELU activation [HG17] which outputs both positive and negative values, [CGW21a] modifies the propagation rule by considering only elements with positive weighted relevance:

$$R_j^{(n)} = \sum_{\{i|(i,j)\in q\}} \frac{x_j w_{ji}}{\sum_{\{j'|(j',i)\in q\}} x_{j'} w_{j'i}} R_i^{(n-1)} \tag{3.11}$$

where $q = \{(i,j)|x_j w_{ji} \geq 0\}$.

3.3.3 Challenges with Skip Connections and Attention

Transformer architectures present two unique challenges for relevance propagation:

Skip Connections: While addition operations preserve the conservation rule, they can cause numerical instabilities. As shown in [CGW21a], relevance values can explode even though their sum remains constant, necessitating normalization.

Matrix Multiplication in Attention: The attention mechanism involves matrix multiplication $L^{(n)}(u,v) = uv$, which does not preserve the conservation rule. As proven in [CGW21a] Lemma 1, this leads to double-counting where $\sum_{k,m} R_{k,m}^u + \sum_{m,l} R_{m,l}^v = 2\sum_{k,l} R_{k,l}^{(n-1)}$.

3.3.4 Normalization Solution

To address both numerical instabilities and conservation violations, [CGW21a] introduces a normalization scheme for binary operators:

$$\bar{R}_j^u = R_j^u \cdot \frac{|\sum_j R_j^u|}{|\sum_j R_j^u| + |\sum_k R_k^v|} \cdot \frac{\sum_i R_i^{(n-1)}}{\sum_j R_j^u} \tag{3.12}$$

$$\bar{R}_k^v = R_k^v \cdot \frac{|\sum_k R_k^v|}{|\sum_j R_j^u| + |\sum_k R_k^v|} \cdot \frac{\sum_i R_i^{(n-1)}}{\sum_k R_k^v} \tag{3.13}$$

The first factor distributes total relevance proportionally between the two input branches based on their absolute magnitudes, while the second factor ensures each branch's relevance sums to the correct total. This normalization maintains the conservation rule while bounding individual relevance sums (proven in [CGW21a] Lemma 2).

3.3.5 Weighted Attention Relevance and Aggregation

The key contribution of [CGW21a] is combining attention scores with both gradients and relevance for class-specific visualization, illustrated in Figure 3.1. For each Transformer block b, they compute:

$$\bar{A}^{(b)} = I + E_h(\nabla A^{(b)} \odot R^{(n_b)})^+ \tag{3.14}$$

where $A^{(b)}$ is the attention map from block b, $\nabla A^{(b)}$ are gradients with respect to target class t, $R^{(n_b)}$ are relevance scores at the attention layer, $\odot$ denotes element-wise multiplication, $(\cdot)^+$ keeps only positive values, E_h averages across attention heads, and I is the identity matrix accounting for skip connections.

Relevance at Attention Layers The relevance scores $R^{(n_b)}$ in Eq. 3.14 are specifically computed at the softmax operation of each attention block. This is crucial because it means relevance is propagated through the attention mechanism itself, not just through the final linear layers.

The final visualization is obtained by aggregating across all B blocks:

$$C = \bar{A}^{(1)} \cdot \bar{A}^{(2)} \cdot \ldots \cdot \bar{A}^{(B)} \tag{3.15}$$

Classification Token Extraction In vision transformers, the [CLS] token at position 0 serves as the global aggregator for classification. The final attribution map is extracted from row 0 of the relevance matrix: $C_{[CLS],:} \in \mathbb{R}^s$, representing how much each image patch contributed to the classification decision.

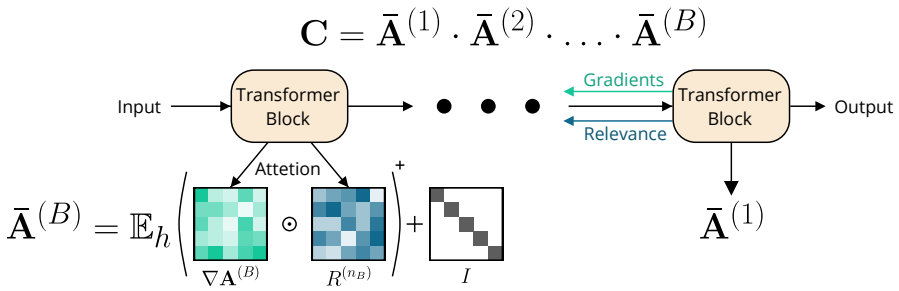


Figure 3.1: Illustration of the TransLRP method. Gradients and relevancies are propagated through the network, and integrated to produce the final relevancy maps, as described in Eq. 3.14 and 3.15. Figure reproduced from [CGW21a].

3.3.6 Computational Considerations and Performance

While TransLRP demonstrates superior performance compared to existing methods, it comes with notable computational overhead. The method requires propagating relevance through *all* layers of the Transformer—not just attention layers but also MLP blocks, normalization layers, and skip connections.

Implementation Complexity: TransLRP necessitates custom hooks throughout the entire model to:

- Compute relevance scores at each layer using Eq. 3.9 or its variants
- Apply normalization (Eq. 3.12-3.13) at skip connections and matrix multiplications
- Store intermediate relevance values for aggregation

This effectively doubles the backward computation compared to gradient-only methods like GradCAM.

Performance Justification: Despite the computational cost, [CGW21a] demonstrates that TransLRP significantly outperforms all baseline methods: Crucially, TransLRP is the only method that produces truly class-specific visualizations by design, as shown in Figure 3.2.

3.4 TransMM: Simplifying Attribution Through Direct Attention

TransMM (Generic Attention-model Explainability), introduced by Chefer et al. [CGW21b] as an evolution of their TransLRP method [CGW21a], represents a significant simplification in computing attribution maps for Transformers. While maintaining comparable or superior performance, TransMM eliminates the need for complex LRP propagation entirely.

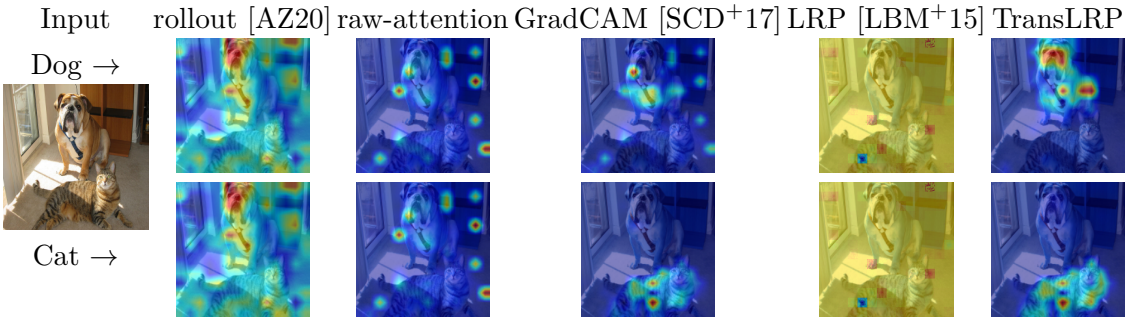


Figure 3.2: Comparison of attribution methods for class-specific visualization from [CGW21a]. TransLRP produces distinct, well-localized attribution maps for different classes, while rollout, raw-attention, and LRP variants generate identical attributions regardless of the target class. Figure reproduced from [CGW21a].

3.4.1 Key Innovation: Eliminating LRP

The fundamental breakthrough of TransMM is demonstrating that the expensive relevance propagation through all network layers is unnecessary. While TransLRP requires:

- Custom implementation of relevance propagation rules for every layer type
- Specialized handling of non-ReLU activations (Eq. 3.11)
- Complex normalization schemes for skip connections (Eq. 3.12-3.13)
- Relevance scores $R^{(n_b)}$ computed through the entire network

TransMM achieves equivalent results using only attention maps and standard gradients.

3.4.2 Mathematical Framework

Consider a vision transformer with N image patches and a CLS token for classification. Let $\mathbf{x} \in \mathbb{R}^{(N+1)\times d}$ denote the token representations where position 0 corresponds to the CLS token.

Relevancy Matrix TransMM maintains a relevancy matrix $\mathcal{R}^{(\ell)} \in \mathbb{R}^{(N+1)\times(N+1)}$ at each layer ℓ, where $\mathcal{R}^{(\ell)}_{ij}$ represents the relevance of token j to token i at layer ℓ.

Initialization The relevancy matrix is initialized as the identity:

$$\mathcal{R}^{(0)} = I^{(N+1)\times(N+1)} \tag{3.16}$$

3.4.3 Core Propagation Rules for Self-Attention

The critical difference between TransLRP and TransMM lies in how attention maps are weighted:

$$\text{TransLRP:} \quad \bar{A}^{(\ell)} = I + \mathbb{E}_h \left[\left(\nabla A^{(\ell)} \odot R^{(n_\ell)} \right)^+ \right] \quad \text{(requires LRP relevance)} \tag{3.17}$$

$$\text{TransMM:} \quad \bar{A}^{(\ell)} = I + \mathbb{E}_h \left[\left(\nabla A^{(\ell)} \odot A^{(\ell)} \right)^+ \right] \quad \text{(uses raw attention)} \tag{3.18}$$

where y_t is the logit for target class t, $\nabla_{A^{(\ell)}} y_t$ is the gradient of the target logit with respect to attention weights, $\odot$ denotes element-wise multiplication, $(\cdot)_+ = \max(0, \cdot)$ keeps only positive contributions, $\mathbb{E}_h$ denotes averaging across attention heads, and I is the identity matrix.

Relevancy Propagation The relevancy matrix is updated through self-attention layers according to:

$$\mathcal{R}^{(\ell)} = \mathcal{R}^{(\ell-1)} + \bar{A}^{(\ell)} \cdot \mathcal{R}^{(\ell-1)} \tag{3.19}$$

Final Attribution Extraction The attribution score for each image patch is extracted from the CLS token's row:

$$\text{Attribution}_i = \mathcal{R}^{(L)}_{0,i} \quad \text{for } i \in \{1, \ldots, N\} \tag{3.20}$$

3.4.4 Computational and Practical Advantages

The simplification from TransLRP to TransMM has profound practical implications:

Implementation Simplicity: TransMM needs only attention maps and their gradients; standard operations with simple hooks at attention layers.

Computational Efficiency: Single backward pass for gradients, then simple matrix operations.

Memory Requirements: Only stores attention maps and the relevancy matrix.

3.4.5 Empirical Validation

Despite the dramatic simplification, Chefer et al. [CGW21b] demonstrate that TransMM achieves comparable or superior performance to TransLRP across multiple benchmarks. On ImageNet-Segmentation TransMM was matching the performance of the more complex TransLRP.

3.4.6 Summary and Open Challenges

This chapter has traced the evolution of attribution methods for Vision Transformers, from the theoretical foundations of LRP and DTD to the practical simplifications of TransMM. The key insights include:

- Traditional attribution methods fail to capture Transformer-specific operations
- LRP provides principled relevance propagation with theoretical backing from DTD
- TransLRP successfully extends LRP to Transformers but with significant overhead
- TransMM achieves comparable results through dramatic simplification

The remaining challenges in Transformer attribution include disentangling feature-level contributions within gradient signals, handling increasingly complex architectures, and developing theoretical frameworks that better capture Transformer-specific operations. These challenges motivate ongoing research into methods that can maintain TransMM's practical advantages while providing deeper insights into the specific learned features that drive model predictions.

CHAPTER 4

Faithfulness in Attribution Methods

Attribution methods aim to identify which input regions drive model predictions, but how do we know if these explanations are trustworthy? A visually plausible heatmap may highlight regions that merely correlate with predictions rather than cause them, or fail to capture the true extent of evidence the model uses [AGM+18]. In high-stakes domains like medical imaging, where clinicians rely on explanations to verify that models have learned clinically meaningful patterns, faithful attribution is essential for meaningful decision support.

What is Faithfulness? A faithful explanation is one that accurately reflects the model's actual reasoning process which we can understand as the computational mechanisms that produced the prediction [JG20]. However, we cannot directly observe these mechanisms in deep neural networks. This creates a fundamental evaluation challenge: how do we verify that an attribution correctly represents reasoning we cannot directly observe?

No Single Definition Exists The research community has not converged on a single formal definition of faithfulness [JG20]. Different use cases require different properties: correlation with performance changes [BWM20], correct ranking of feature importance [LBM+15], or differences in attribution between pixel regions [WKT+24b] all represent valid but distinct notions of faithfulness. For our purposes, we focus on whether attributions reflect *causal* relationships between input regions and predictions: if an attribution claims a region is important, removing that region should impact the prediction accordingly.

Approximating Faithfulness Through Perturbation Given that we cannot directly observe model reasoning, faithfulness metrics can approximate it through controlled inter-

ventions [SBM+16]: if removing a highly-attributed region causes large prediction changes while removing low-attributed regions causes small changes, this provides evidence that attributions reflect actual causal importance. All three metrics we employ implement this perturbation principle differently, testing complementary aspects of whether attributions capture causal relationships:

- **Faithfulness Correlation** [BWM20]: Tests whether attribution magnitudes predict the magnitude of impact when features are perturbed
- **Pixel Flipping** [LBM+15]: Tests whether removing features in order of attributed importance causes appropriately ordered performance degradation
- **SaCo** [WKT+24b]: Tests whether attribution magnitude differences align with actual impact differences through independent perturbation

The remainder of this chapter details each metric's implementation, interpretation, and known limitations.

4.1 Faithfulness Correlation

Faithfulness Correlation [BWM20] quantifies whether attribution scores accurately reflect features' actual impact on model predictions. The metric measures the correlation between the sum of attribution scores for a subset of features and the change in model output when those features are replaced with baseline values.

Formal Definition Given a predictor f, explanation function g, input x, and subset size $|S|$, faithfulness is defined as:

$$\mu_F(f, g; x) = \text{corr}_{S \in \binom{[d]}{|S|}} \left(\sum_{i \in S} g(f, x)_i, f(x) - f(x[x_S = \bar{x}_S]) \right) \tag{4.1}$$

where x_S denotes features indexed by S, and $x[x_S = \bar{x}_S]$ represents the input with subset S replaced by baseline values $\bar{x}_S$ while remaining features stay unchanged.

Implementation Details Since exhaustively evaluating all $\binom{d}{|S|}$ possible subsets is computationally prohibitive, the metric employs random sampling. For each test point, multiple random subsets of fixed size $|S|$ are selected, their features replaced with baseline values, and the Pearson correlation computed between the sum of attributions and the resulting change in model logits for the target class.

Baseline Selection The choice of baseline $\bar{x}$ significantly impacts results. Bhatt et al. experiment with two approaches: zero baseline, setting features to 0 and average baseline, using mean feature values from the training data. The empirical results show substantial variation between baseline choices, highlighting this as a critical design decision.

Interpretation and Limitations Higher correlation indicates that attribution scores faithfully represent features' actual importance to the model. The authors observe that faithfulness generally increases with subset size until all informative features are included. However, the metric's reliance on random sampling means it may not capture the full distribution of subset importances, potentially missing critical feature interactions. Additionally, the correlation measure assumes a linear relationship between attribution sums and output changes, which may not hold for highly non-linear models or when features interact strongly.

4.2 Pixel Flipping

Pixel Flipping [LBM+15] provides a quantitative validation method for attribution techniques by directly testing whether pixels identified as important actually influence model predictions. The method addresses a fundamental challenge in evaluating attribution quality: while visual assessment can suggest plausibility, it cannot verify whether highlighted regions genuinely drive the model's decision.

Methodology Let $\mathbf{x} \in \mathbb{R}^d$ denote an input image with d pixels, and let $f : \mathbb{R}^d \to \mathbb{R}^C$ represent the classifier outputting logits for C classes. Given an attribution method A, we compute pixel-wise attribution scores $\{a_i\}_{i=1}^d$ where $a_i = A(\mathbf{x})_i$ represents the importance of pixel i for the model's prediction.

The procedure follows three steps: First, compute attribution scores for each pixel using the method under evaluation. Second, obtain a permutation $\pi : \{1, \ldots, d\} \to \{1, \ldots, d\}$ that sorts pixels in descending order of their attribution values, such that $a_{\pi(1)} \geq a_{\pi(2)} \geq \cdots \geq a_{\pi(d)}$. Third, progressively modify pixels according to this ranking while measuring the model's confidence (probability) degradation.

For each step $k \in \{0, 1, \ldots, d\}$, we construct a perturbed image $\mathbf{x}^{(k)}$ where the top k pixels according to π have been modified. Let c denote the originally predicted class. The prediction score at step k is given by:

$$s_k = f(\mathbf{x}^{(k)})_c \tag{4.2}$$

where $f(\mathbf{x}^{(k)})_c$ represents the model's confidence (logit or softmax output) for class c on the perturbed input. Bach et al. implement modification through pixel inversion (multiplying by -1), though this is due to their usecase on the MNIST dataset which is in grayscale. In settings where an image's pixel does not have a clear inverse, a baseline value (like black, white or grey) can be used.

Evaluation via AUC The faithfulness of the attribution method is quantified by computing the area under the prediction degradation curve. We normalize both axes by constructing points $\left(\frac{k}{d}, s_k\right)$ for $k = 0, \ldots, d$, yielding a curve over $[0, 1]$. The AUC metric

is then:

$$\text{AUC} = \frac{1}{d} \sum_{k=0}^{d-1} \frac{s_k + s_{k+1}}{2} \tag{4.3}$$

or equivalently using the trapezoidal rule:

$$\text{AUC} = \sum_{k=1}^{d} \frac{1}{d} \cdot \frac{s_{k-1} + s_k}{2} = \frac{1}{2d} \left(s_0 + 2 \sum_{k=1}^{d-1} s_k + s_d \right) \tag{4.4}$$

A lower AUC indicates faster performance degradation and thus better attribution quality, as the method successfully identified pixels critical to the prediction. Bach et al. demonstrate this approach's discriminative power by contrasting the steep degradation when flipping high-attribution pixels against the minimal impact of flipping pixels with near-zero scores, validating that their LRP method distinguishes between causally relevant and irrelevant image regions.

While Pixel Flipping provides valuable quantitative validation, it suffers from cumulative perturbation limitations that have been well-documented in the literature [WW22, WKT+24b]. Early pixel removals dominate the metric, and the cumulative nature makes it difficult to isolate individual pixel contributions. When evaluating the least important 10% of pixels, they are only removed after the most important 90% have been eliminated, confounding the assessment of their individual impact [WKT+24b]. This motivates our use of SaCo alongside Pixel Flipping, as the two metrics capture complementary aspects of faithfulness. Where Pixel Flipping validates the overall ranking while SaCo's independent perturbation design ensures the magnitude of attributions aligns with actual impact.

4.3 SaCo (Salience-guided Faithfulness Coefficient)

The SaCo metric [WKT+24b] addresses fundamental limitations in existing faithfulness metrics by introducing a principled framework for evaluating whether salience scores accurately reflect pixel contributions to model predictions. The authors identify two core assumptions that underpin faithful explanations: (1) input pixels assigned higher salience scores should exert greater influence on model predictions compared to those with lower scores, and (2) pixel groups with larger differences in salience scores should exhibit proportionally larger disparities in their actual impacts.

Addressing Cumulative Perturbation Limitations Most existing faithfulness metrics, including Pixel Flipping [LBM+15] and other AUC-based approaches, rely on cumulative perturbation where pixels are progressively removed in order of importance. This cumulative strategy inherently conflates the impacts of different pixel groups. When evaluating the least important 10% of pixels, they are only removed after the most important 90% have already been eliminated. Wu et al. demonstrate that this makes it impossible to isolate individual contributions and can lead to misleading evaluations

where metrics fail to distinguish between advanced explanation methods and Random Attribution.

SaCo resolves this by employing individual perturbation. Given an input image with HW pixels ordered by their salience scores, the pixels are partitioned into K equally sized subsets: $G_1, G_2, \ldots, G_K$, where each subset G_i contains pixels with salience ranking from $(i-1)\frac{HW}{K}$ to $i\frac{HW}{K}$. Each pixel subset G_i is then perturbed independently, enabling direct comparison of their influences through:

$$\Delta\text{pred}(x, G_i) = p(\hat{y}(x)|x) - p(\hat{y}(x)|\text{Rp}(x, G_i)) \tag{4.5}$$

where $\text{Rp}(x, G_i)$ denotes the image with pixels in subset G_i replaced by the per-sample mean value.

Salience-Weighted Evaluation Framework The metric's key innovation lies in its salience-aware violation testing. For each pair of pixel subsets (G_i, G_j) where $s(G_i) \geq s(G_j)$, SaCo tests whether $\Delta\text{pred}(x, G_i) \geq \Delta\text{pred}(x, G_j)$. Crucially, the evaluation is weighted by the salience difference $|s(G_i) - s(G_j)|$, ensuring that violations of the faithfulness assumption are penalized proportionally to the strength of the expectation.

The final coefficient is computed as:

$$\text{SaCo} = \frac{\sum_{i<j} w(i,j)}{\sum_{i<j} |w(i,j)|} \tag{4.6}$$

where the weight function incorporates both the correctness of ordering and the magnitude of salience differences:

$$w(i,j) = \begin{cases} s(G_i) - s(G_j) & \text{if } \Delta\text{pred}(G_i) \geq \Delta\text{pred}(G_j) \\ -(s(G_i) - s(G_j)) & \text{otherwise} \end{cases} \tag{4.7}$$

This formulation produces a coefficient ranging from -1 to 1, where positive values indicate alignment between salience assignments and actual impacts.

Complementarity to Existing Metrics Wu et al. demonstrate that SaCo captures a fundamentally different aspect of faithfulness compared to conventional evaluation metrics. Among conventionally used metrics (Pixel Flipping, AOPC, LOdds, Comprehensiveness), the average inter-correlation is 0.48, indicating they measure similar properties. In contrast, SaCo correlates only 0.18-0.22 with these conventional metrics, demonstrating it evaluates a complementary aspect of explanation quality. This low correlation suggests that while traditional metrics primarily capture the effects of progressive pixel removal, SaCo specifically measures whether salience score magnitudes align with actual pixel importance—the core assumption of faithfulness.

The metric also uses salience differences rather than ratios for weighting, ensuring scale-invariance which is a crucial property since explanation methods often normalize salience maps to $[0, 1]$. This design choice makes SaCo robust to post-processing transformations that would destabilize ratio-based metrics.

Relevance for Vision Transformer Attribution While originally formulated for pixel-level attribution, SaCo naturally extends to patch-based methods like TransMM by redefining pixel subsets as patch subsets. For Vision Transformers operating on N patches, we partition patches into K subsets based on their salience scores, where each G_i contains patches rather than pixels. The perturbation operation $\mathrm{Rp}(x, G_i)$ then replaces entire patches with baseline values, aligning with the natural granularity of transformer-based attribution methods. Wu et al.'s findings that gradient information and multi-layer aggregation substantially improve faithfulness scores directly validate design choices in TransMM and motivate our approach as described in the later chapters. By addressing the limitations of cumulative perturbation metrics while maintaining complementary evaluation perspectives, SaCo provides a comprehensive assessment of whether our feature-gradient approach genuinely improves the alignment between attributed importance and actual model behavior.

CHAPTER 5

Mechanistic Interpretability and Sparse Autoencoders

Understanding what neural networks learn has been a central challenge since their inception. Early work on CNNs showed promise: individual neurons in trained networks appeared to respond to interpretable visual patterns—edge detectors in early layers [ZF14], texture patterns in middle layers, and object parts in deeper layers [OMS17]. Feature visualization techniques [OMS17, YCN+15] synthesized images that maximally activated specific neurons, often revealing seemingly clear concepts like dog faces, wheels, or text patterns.

However, this interpretability broke down at scale. While some neurons exhibited clear responses to specific concepts, most neurons in large networks activated for multiple, seemingly unrelated patterns. This phenomenon was termed **polysemanticity** [OCS+20]. Cammarata et al.'s analysis of "curve detectors" [CGC+20] demonstrated this problem concretely: neurons that appeared to detect curves also responded to high-frequency patterns, 3D object boundaries, and other seemingly unrelated visual features. The neuron wasn't detecting curves but rather something more complex that humans couldn't easily name. The promise that individual neurons correspond to interpretable concepts proved insufficient for understanding modern networks.

The word embedding literature provided a suggestive alternative perspective. Mikolov et al.'s word2vec [MCCD13] demonstrated that semantic concepts could exist not in individual dimensions but in *directions* through vector space, observing the now famous example that "king" - "man" + "woman" ≈ "queen". This raised a possibility: perhaps neural networks encode features in directions through activation space rather than in individual neurons, explaining why neuron-level interpretability fails.

The superposition hypothesis [EHO+22], discussed in detail in Section 5.1, formalized this intuition: networks exploit activation sparsity to represent more features than

they have neurons, storing multiple features in overlapping, non-orthogonal directions. This reframed polysemanticity from a failure of learning into an optimal strategy given capacity constraints. If features exist in superposition rather than in individual neurons, new methods were needed to disentangle them—motivating the development of Sparse Autoencoders (Section 5.2).

Beyond SAEs, researchers have developed complementary interpretability tools. Linear probes [AB16] test whether specific information (e.g., object class, spatial properties) is linearly accessible from layer activations, though they require knowing what concepts to probe for. Activation patching methods [WVC+23, CMPL+23] identify causal circuits by intervening on specific activations and measuring behavioral changes. Each method offers different trade-offs between interpretability, causal understanding, and prior knowledge requirements.

Connection to Attribution This chapter presents mechanistic interpretability tools that complement the attribution methods developed in Chapter 3. While attribution answers "which image regions drive predictions," mechanistic interpretability answers "what semantic concepts does the model recognize in those regions." Understanding that a model detected specific SAE features, for instance lung infiltrates, tissue textures or anatomical boundaries provides semantic grounding for attribution maps that would otherwise be abstract heatmaps. This connection enables us to interpret why certain attributions are faithful: they highlight regions where the model activated features causally relevant to classification. The integration of these two paradigms forms the conceptual foundation for our Feature-Gradient Attribution method (Chapter 6).

5.1 The Superposition Hypothesis

The interpretation of individual neurons in neural networks has long been hindered by polysemanticity, the phenomenon where single neurons respond to multiple, seemingly unrelated concepts. While early successes in vision models identified neurons detecting specific features like edges or textures [ZF14, OMS17], scaling to larger models revealed increasingly ambiguous neuron behavior, with individual units activating for disparate concepts without clear semantic coherence.

Elhage et al. [EHO+22] provided a theoretical framework for understanding this phenomenon through the **superposition hypothesis**: neural networks exploit the sparsity of feature activation patterns to encode more features than they have dimensions. Through carefully designed toy experiments, they demonstrated not only that superposition emerges naturally under realistic constraints, but also that features in superposition can perform meaningful computation.

5.1.1 Demonstrating Superposition Through Toy Models

To isolate and study superposition, [EHO+22] constructed a minimal experimental setup: a single-layer network with ReLU activation trained on synthetic data with precisely controlled properties. The synthetic data consisted of input vectors $\mathbf{x}$ where each dimension x_i represents a feature characterized by two properties:

- **Sparsity** S_i: The probability that $x_i = 0$, controlling how often the feature is active
- **Importance** I_i: A weight determining the feature's contribution to the loss function

The model architecture deliberately creates a bottleneck:

$$\mathbf{h} = W\mathbf{x} \quad \text{(encoding to lower dimension)} \tag{5.1}$$

$$\mathbf{x}' = \text{ReLU}(W^T\mathbf{h} + \mathbf{b}) \quad \text{(reconstruction)} \tag{5.2}$$

with the importance-weighted reconstruction loss:

$$\mathcal{L} = \mathbb{E}_{\mathbf{x}}\left[\sum_i I_i(x_i - x_i')^2\right] \tag{5.3}$$

This setup enables precise measurement of superposition through three key diagnostics:

1. **Feature representation**: The norm $\|W_i\|$ indicates whether feature i is represented
2. **Reconstruction fidelity**: The matrix W^TW reveals information preservation
3. **Superposition degree**: The interference $\sum_{j\neq i}(\hat{W}_i \cdot W_j)^2$ quantifies how much feature i overlaps with others in the hidden representation

5.1.2 Key Findings on Superposition Emergence

The experiments revealed a sharp phase transition in network behavior as sparsity increases. When features are dense (low sparsity), the network dedicates orthogonal dimensions to the most important features, leaving others unrepresented. However, as sparsity increases, the network discovers it can exploit the low probability of simultaneous feature activation to encode many features in the same space through **superposition**.

Remarkably, [EHO+22] demonstrated that features in superposition retain computational capacity. Through targeted experiments with structured feature correlations, they showed that networks can perform meaningful operations on superposed features, suggesting that polysemanticity is not merely a compression artifact but a functional computational strategy.

The transition point depends critically on the relative importance of features: uniformly important features superpose more readily than those with varied importance, as the network has no basis for prioritizing which features deserve dedicated dimensions. This finding has profound implications for understanding why real neural networks, trained on data with naturally varying feature importance and sparsity, ubiquitously exhibit polysemantic neurons.

5.1.3 Implications for Mechanistic Interpretability

The superposition hypothesis fundamentally reframes the challenge of neural network interpretability. Rather than viewing polysemanticity as a failure of neurons to learn clean features, it reveals superposition as an optimal strategy given the constraints of limited model capacity and sparse activation patterns. This insight motivates the development of methods like Sparse Autoencoders (SAEs) [SBB22, HCS+24] that can decompose superposed representations back into interpretable features by learning an overcomplete basis by essentially reversing the superposition process to recover monosemantic features from polysemantic activations.

5.2 Sparse Autoencoders: Reversing Superposition

Classical dimensionality reduction methods such as Principal Component Analysis (PCA) are constrained to extracting at most N orthogonal components from N-dimensional data. However, the superposition hypothesis demonstrates that neural networks encode $M \gg N$ features within N-dimensional activation spaces through exploitation of feature sparsity [EHO+22]. As these M features exist in non-orthogonal superposition within the constrained space, recovering interpretable features requires methods capable of learning overcomplete representations—bases containing more vectors than the ambient dimensionality.

Sparse Autoencoders (SAEs) [HCS+24, SBB22] address this requirement by learning to reconstruct neural activations through an expanded sparse representation. The method performs an approximate inverse of the superposition process: dense, polysemantic activation vectors are mapped to sparse vectors in a higher-dimensional space where each dimension can correspond to a single semantic concept. This transformation from entangled to disentangled representations enables the identification of monosemantic features from polysemantic neural activations.

5.2.1 Architecture and Mathematical Framework

Given an activation vector $\mathbf{x} \in \mathbb{R}^n$ from a neural network layer, an SAE learns an overcomplete dictionary of features through two transformations:

Encoding: Map activations to a sparse, high-dimensional representation:

$$\mathbf{f}(\mathbf{x}) = \mathrm{ReLU}(W_{\mathrm{enc}}\mathbf{x} + \mathbf{b}_{\mathrm{enc}}) \tag{5.4}$$

where $W_{\text{enc}} \in \mathbb{R}^{m \times n}$ with $m \gg n$. The ratio m/n is called the **expansion factor**, typically ranging from 8 to 64. This expansion creates enough dimensions to represent each feature in superposition with its own orthogonal basis vector, effectively *unpacking* the compressed representation.

Decoding: Reconstruct the original activations from the sparse code:

$$\hat{\mathbf{x}} = W_{\text{dec}}\mathbf{f}(\mathbf{x}) + \mathbf{b}_{\text{dec}} \tag{5.5}$$

The key insight is that while the original n-dimensional space forces features into superposition, the expanded m-dimensional space provides sufficient capacity for each feature to claim its own dimension, eliminating the need for superposition.

5.2.2 Training Objective and Sparsity Control

The SAE training objective balances two competing goals:

$$\mathcal{L} = \underbrace{\|\mathbf{x} - \hat{\mathbf{x}}\|_2^2}_{\text{Reconstruction}} + \underbrace{\alpha\|\mathbf{f}(\mathbf{x})\|_1}_{\text{Sparsity}} \tag{5.6}$$

The reconstruction term ensures the SAE preserves information from the original activations, while the sparsity term encourages most features to remain inactive for any given input. The L_1 norm $\|\mathbf{f}(\mathbf{x})\|_1 = \sum_i |f_i(\mathbf{x})|$ simply sums the absolute values of all feature activations, penalizing any non-zero activation.

The hyperparameter α controls the sparsity-reconstruction tradeoff. Higher α values produce sparser representations with cleaner feature separation but potentially worse reconstruction, while lower values allow denser representations that may retain some polysemanticity. Finding the optimal α requires balancing interpretability with information preservation—too sparse and important computational information is lost; too dense and features remain entangled.

5.2.3 SAEs in Practice: From Toy Models to Production Systems

While the theoretical foundations of SAEs were established through controlled experiments, their practical value remained uncertain until recent breakthrough applications. Anthropic's work on monosemanticity [BTB+23, TCM+24] demonstrated that SAEs could successfully extract interpretable features from production-scale language models, discovering a plethora from specific tokens to abstract concepts like deception and bias. These features proved causally relevant interventions on specific features predictably altered model outputs, validating that SAEs capture genuine computational units rather than statistical artifacts.

This success in language models motivated applications to vision transformers, while methodological advances have made SAEs increasingly practical. Gao et al. [GDlTT+25]

introduced TopK activation functions as an alternative to L1 sparsity, addressing common failure modes like dead features and feature collapse while providing more stable training dynamics.

5.2.4 Current Limitations and Open Questions

Despite their successes, SAEs face significant practical and theoretical limitations that raise questions about their adequacy for fully understanding neural network internals. The following analysis draws on the comprehensive examination of sparse dictionary learning limitations by Sharkey, Chughtai et al. [SCB+25] in their work discussing open questions in mechanistic interpretability.

5.2.4.1 Practical Limitations

Insufficient reconstruction quality: Current SAEs fundamentally fail to capture complete model behavior, as evidenced by substantial performance degradation when model activations are replaced with SAE reconstructions. Multiple studies demonstrate this limitation: when SAE reconstructions were used in GPT-4, the resulting language modeling performance dropped to that of a model trained with only 10% of the original compute [GDlTT+25]. Similarly, [MLN25] found significant performance degradation when using SAE reconstructions in GPT-2 Small, with the degradation being particularly severe on full distributions compared to task-specific data.

Prohibitive computational costs: Scaling SAEs to large models presents severe computational challenges. Each layer requires its own SAE with an expansion factor typically between 8 and 64, meaning the SAE contains orders of magnitude more parameters than the original layer. For modern large language models, training comprehensive SAEs across all layers would require computational resources exceeding the original model training costs by several orders of magnitude. While this constraint is manageable for small vision transformers, it makes comprehensive SAE analysis of production-scale models economically infeasible.

5.2.4.2 Theoretical Limitations

Lack of theoretical foundations for superposition: While the superposition hypothesis has empirical support, no formal theoretical framework proves that neural networks actually implement superposition or that SAE features correspond to the model's true computational units. The absence of clarity around what constitutes a "feature" remains a central problem—satisfying formal definitions remain elusive despite features being the primary object of study [EHO+22]. Without theoretical foundations, it remains unclear whether superposition is fundamentally valid or merely a pragmatically useful approximation [TCM+24].

Features without mechanisms: Most critically, SAEs decompose activations but not the computational mechanisms that produce them. As [SCB+25] articulate: "SDL

decomposes the input and output activations of network mechanisms, but not the mechanisms themselves." These identified directions in activation space are merely the inputs and outputs of neural network computations, not the mechanisms themselves.

These limitations suggest that while SAEs represent progress in making neural networks more interpretable, they may be fundamentally insufficient for complete mechanistic understanding. However, perfect understanding may not be necessary for practical progress. The features SAEs discover, while potentially incomplete, consistently prove to be semantically meaningful and causally relevant. This practical utility has driven rapid adoption across both language and vision domains, as the following applications demonstrate.

5.3 SAEs for Vision Model Enhancement

Despite their theoretical limitations, SAEs have proven remarkably effective for practical applications in vision models. Recent work demonstrates that SAE features capture semantically meaningful and causally relevant representations.

Causal Relevance Through Intervention SAE features are not merely correlational but causally influence model behavior. Templeton et al. [TCM+24] first demonstrated this in language models, while Joseph et al. [Ano25] extended these findings to vision, showing that activating or suppressing specific SAE features in CLIP models predictably alters predictions and can improve robustness. This causal relevance makes SAE features promising candidates for enhancing attribution methods.

Hierarchical Semantic Organization Lim et al. [LCCS25] analyzed SAE features in CLIP vision transformers, revealing hierarchical organization from low-level patterns (colors, textures) in early layers to semantic concepts in late layers. Their findings that features are spatially localized and that only top-50 to top-100 features per class maintain classification accuracy suggest SAE features capture causally important semantic concepts, motivating their use in attribution methods.

Connection to Attribution Methods Stevens et al. [SCBWS25] demonstrated that SAE features can validate attribution methods through targeted intervention. By identifying which features are active in regions highlighted by attribution methods like Grad-CAM, then manipulating those features to observe behavioral changes, they provide experimental evidence for attribution hypotheses that would otherwise remain untested. This work establishes that SAE features provide semantic grounding for attribution claims.

While Stevens et al. use SAEs to validate attributions post-hoc, their work suggests a deeper integration: if SAE features provide semantic understanding that can diagnose attribution quality, could these features be incorporated directly into attribution computation? This question motivates our approach, where we leverage SAE features not

just to validate attributions after generation, but to enhance their faithfulness during computation itself.

CHAPTER 6

Feature-Gradient Attribution Method

The fields of *mechanistic interpretability* and *explainable AI* (XAI) share a fundamental goal: understanding how neural networks transform inputs into predictions. Mechanistic interpretability seeks to decompose model internals into interpretable computational units with the goal of identifying what features networks learn and how they combine to produce behavior. Explainable AI focuses on human-understandable attributions that highlight which input regions drove specific decisions. Despite this shared objective, these research traditions have largely evolved in parallel, with limited cross-pollination between insights about internal representations and methods for explaining predictions.

This separation creates a missed opportunity. Attribution methods like TransMM have achieved strong performance by combining attention patterns with gradient information [CGW21b], yet they operate on dense and entangled gradient signals that aggregate influences across all learned representations. Meanwhile, Sparse Autoencoders have demonstrated the ability to decompose polysemantic activations into interpretable semantic features [HCS$^+$24, BTB$^+$23], but these tools have been applied primarily to understanding model internals rather than improving practical explanation methods.

6.0.1 Why Sparse Features?

Recent work has begun connecting attribution and interpretability [HKK25, SCBWS25], though from different angles than we propose. Han et al. use attribution to explain SAE features, while Stevens et al. use features to validate attribution post-hoc. However, neither integrates semantic feature information *directly into the attribution computation itself.*

This represents a fundamental gap: if SAE features capture meaningful semantic concepts and attribution methods aim to identify important image regions, can we leverage

these learned semantic representations to enhance attribution quality? SAEs discover features directly from model activations without supervision, making them well-suited for decomposing gradients through whatever semantic structure the model has learned, rather than requiring prior knowledge of which concepts matter.

Operating in feature space reframes the attribution question. Instead of asking "which pixels influence the gradient?", we ask "which learned semantic features contribute to the decision, and where are they expressed?". This provides a principled route to disentangling gradient signals: if gradients aggregate influences across all representations, projecting them through a sparse feature basis may separate signal from noise by isolating which specific semantic concepts drive each patch's importance.

6.0.2 TransMM's Principle: Combining Presence with Influence

TransMM's success stems from a simple but powerful principle: patch importance emerges from combining *how much attention a patch receives* (attention weights) with *how that attention affects the prediction* (gradients). By weighting attention maps with gradients of the target class, TransMM identifies patches that are both attended to and influential [CGW21b].

We propose that this same principle extends naturally to sparse semantic features: just as TransMM combines attention presence with gradient influence, we can combine feature activation (semantic presence) with feature-space gradients (semantic influence).

6.0.3 Intuition: Semantic Decomposition of Gradients

Consider a concrete example to illustrate the core idea. Suppose we have a patch containing both a dog's eye (relevant for classification) and background texture (likely irrelevant). The raw gradient at this patch aggregates both signals, reflecting the combined influence of all visual patterns present without distinguishing which patterns actually drive the prediction.

However, an SAE trained on the same layer learns to separate these entangled signals into distinct features, for instance feature 237 might respond to "eye-like patterns" while feature 891 responds to "grass texture". This decomposition provides a more granular view of what the model has learned. By projecting the gradient through the SAE's decoder, we can ask: "How much does each specific semantic feature contribute to the prediction?"

The product of feature activation (is this concept present?) and feature-space gradient (does this concept influence the output?) yields a score for each semantic feature. Crucially, this allows us to refine the patch-level importance signal: features that are both active and gradient-aligned receive high scores, while features that are either absent or irrelevant receive low scores. When we aggregate across all features, patches where important features outweigh unimportant ones receive high gates, while patches dominated by irrelevant features (even if some important features exist) receive lower gates. This

semantic decomposition refines the attribution signal at a more fundamental level than raw pixel-space gradients, potentially separating signal from noise.

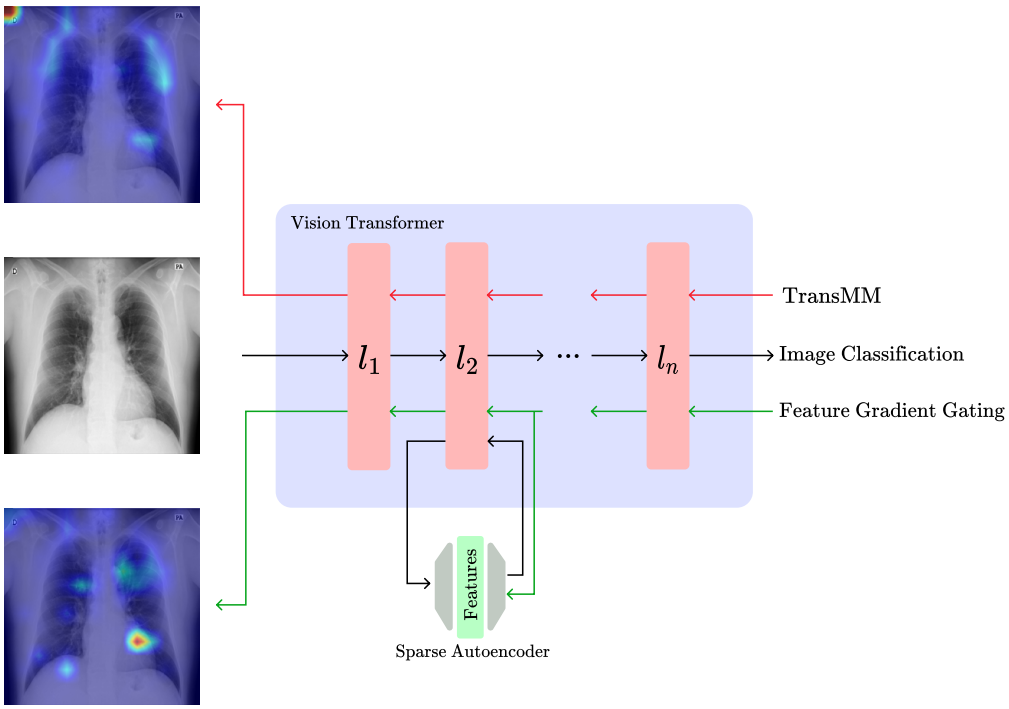


Figure 6.1: **Feature-Gradient Attribution Overview.** Our method extends TransMM by incorporating semantic structure from Sparse Autoencoders (SAEs). Example chest X-rays showing input image (top), vanilla TransMM attribution (middle), and our feature-gated attribution (bottom), demonstrating improved localization of disease-relevant regions.

6.0.4 Extending TransMM's Principle to Feature Space

We now formalize this intuition. TransMM operates on attention weights, combining them with gradients to identify important patches. Our approach operates on residual stream activations, combining sparse feature decompositions with gradients to achieve the same goal.

TransMM weights attention maps using gradients with respect to attention weights (Equation 3.19): $\bar{A}^{(\ell)} = I + \mathbb{E}_h[(\nabla A^{(\ell)} \odot A^{(\ell)})_+]$. However, these attention gradients ultimately derive from gradients flowing through the residual stream, which is considered the primary information pathway in Transformers where computations from attention and MLP blocks are accumulated via residual connections [ENO+21].

Our approach examines gradients in the residual stream at each spatial token, projecting them through learned SAE features. For each patch, we compute feature activations (which semantic concepts are present) and feature-space gradients (how those concepts

influence the prediction). The product of these signals provides a per-patch importance score that respects the model's learned semantic structure. The conceptual architecture of this process is illustrated in Figure 6.1.

We focus on residual stream gradients primarily for practical reasons: SAEs trained on residual streams have become the de facto standard in mechanistic interpretability due to their demonstrated reliability, superior reconstruction quality, and feature interpretability [BTB+23, HCS+24].

This feature-gradient signal is then used to modulate TransMM's attention maps before relevancy propagation, creating a correction mechanism that preserves TransMM's architectural simplicity and computational efficiency while incorporating semantic information. **This represents the first integration of sparse semantic features directly into gradient-based attribution computation.**

6.1 Feature-Gradient Decomposition

This section presents the mathematical framework for decomposing residual gradients through SAE feature space and computing per-patch importance scores.

6.1.1 Mathematical Framework

Consider a residual layer ℓ with SAE decoder $D \in \mathbb{R}^{d \times K}$. For each spatial token t:

- Let $x_t \in \mathbb{R}^d$ be the residual vector
- Let $f_t \in \mathbb{R}^K_{\geq 0}$ be the SAE feature activations
- Let $g_t = \nabla_{x_t} y$ be the gradient with respect to the target logit

Feature-Space Gradient Projection The decoder provides a linear mapping $x_t \approx Df_t$. We project gradients into feature space via:

$$\nabla f_t = D^\top g_t \in \mathbb{R}^K \tag{6.1}$$

where $\nabla f_t^{(k)}$ represents the sensitivity of the target logit to feature k at token t.

Feature-Weighted Scoring For each feature k, we compute its contribution as the product of gradient sensitivity and activation strength:

$$\bar{f}_t^{(k)} = \nabla f_t^{(k)} \cdot f_t^{(k)} \tag{6.2}$$

We aggregate contributions across all features to obtain a per-patch score:

$$s_t = \sum_{k=1}^{K} \bar{f}_t^{(k)} = \sum_{k=1}^{K} (D^\top g_t)_k \cdot f_t^{(k)} \tag{6.3}$$

This can be interpreted as $\bar{s}_t = g_t^\top (Df_t)$, the dot product between the gradient and the SAE reconstruction.

6.1.2 Method Variants

To understand which aspects of the feature-gradient signal contribute to improved faithfulness, we implement three variant methods. The **combined** method is our primary contribution, while the activation-only and gradient-only variants serve as ablation studies:

Combined (Primary Method) Uses both gradient sensitivity and feature activation:

$$s_t^{\text{combined}} = \sum_{k=1}^{K} \nabla f_t^{(k)} \cdot f_t^{(k)} \tag{6.4}$$

Activation-Only (Ablation) Tests whether feature presence alone provides useful signal:

$$s_t^{\text{activation}} = \sum_{k=1}^{K} f_t^{(k)} \tag{6.5}$$

Gradient-Only (Ablation) Tests whether gradient projection alone is beneficial:

$$s_t^{\text{gradient}} = \sum_{k=1}^{K} |\nabla f_t^{(k)}| \tag{6.6}$$

These ablation variants allow us to empirically determine whether the interaction between gradients and activations is necessary, or if either signal alone suffices for improving attribution faithfulness. Figure 6.2 illustrates how these components integrate within a single transformer layer.

6.2 Gate Construction

The per-patch scores s_t must be converted into multiplicative factors that can modulate attention maps. This conversion involves carefully designed steps that respect the sparse activation patterns of SAE features.

6.2.1 Score Normalization

The sparse nature of SAE activations creates a distinctive statistical challenge. In a well-trained sparse autoencoder, the vast majority of features remain inactive (near zero) for any given input, with only a small subset activating strongly [BTB+23]. This sparsity pattern means that for most patches, feature-gradient scores will cluster near zero, representing patches where no particularly relevant semantic features are active.

$$\text{TransMM: } \bar{A}^{(\ell)} = I + \mathbb{E}_h\left[\left(A^{(\ell)} \odot \nabla A^{(\ell)}\right)^+\right]$$

$$\text{Ours: } \bar{A}^{(\ell)}_{\text{gated}} = \bar{A}^{(\ell)} \cdot \text{diag}\Big(\text{Gate}(f^{(\ell)} \odot \nabla f^{(\ell)})\Big)$$

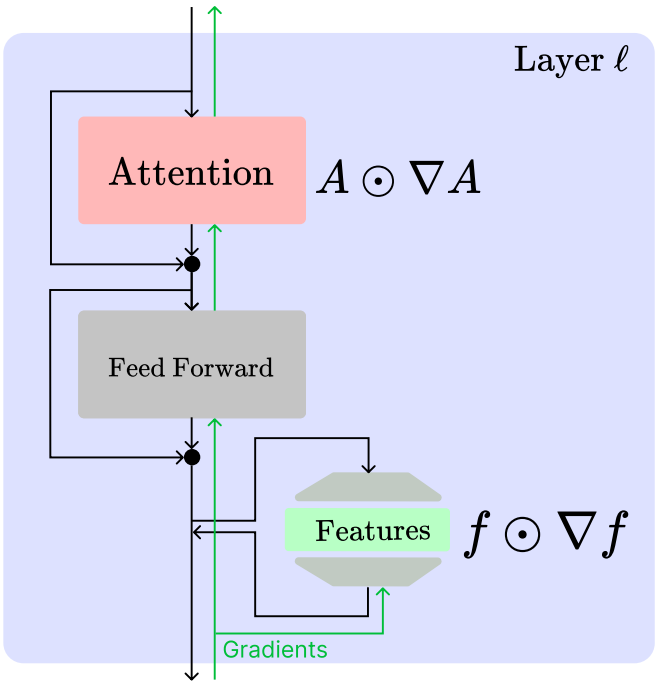


Figure 6.2: **Detailed Layer-wise Computation.** A single transformer layer showing where feature-gradient gating occurs. TransMM computes gradient-weighted attention $\bar{A}^{(\ell)} = I + \mathbb{E}_h[(A^{(\ell)} \odot \nabla A^{(\ell)})_+]$ from attention weights (pink). In parallel, we extract residual stream activations (black vertical line) and pass them through a trained Sparse Autoencoder, which decomposes activations into interpretable features $f^{(\ell)}$ via an encoder-decoder architecture (green). We project gradients through the SAE decoder to obtain feature-space gradients $\nabla f^{(\ell)}$. The element-wise product $f^{(\ell)} \odot \nabla f^{(\ell)}$ captures which semantic features are both present and influential. The Gate function (6.2) aggregates these scores across features, normalizes using robust statistics, and maps to multiplicative gates that modulate attention column-wise before relevancy propagation.

We normalize scores across spatial tokens using robust statistics. Rather than standard z-score normalization, we use median-based normalization:

$$\hat{s}_t = \frac{s_t - \text{median}(s)}{\text{MAD}(s) + \epsilon} \tag{6.7}$$

where $\text{MAD}(s) = \text{median}(|s_t - \text{median}(s)|)$ is the median absolute deviation, and $\epsilon = 10^{-8}$ prevents division by zero. The MAD is scaled by 1.4826 to approximate standard deviation

for normally distributed data while maintaining robustness to outliers.

This median-based approach is particularly well-suited to the sparse activation regime. The median naturally identifies the baseline of "not activated" patches—those where features are weakly or not at all engaged. This creates a highly skewed distribution with:

- A large mass of patches near zero (features not meaningfully active)
- A long tail of patches with high positive scores (strongly activated relevant features)
- Potentially some negative scores (features negatively contributing to the target class)

When features do activate strongly at a patch, the score is correctly normalized relative to this inactive baseline. In contrast, mean-based z-score normalization would be problematic: the mean would be pulled toward the tail of highly active features, causing the baseline to shift and potentially assigning non-neutral gates to patches that should be treated as neutral.

This choice ensures that the normalization reflects the semantic structure of feature activation patterns: patches with typical (median-level) feature-gradient scores receive neutral gates, while only patches with genuinely unusual feature activation patterns of either strongly positive or negative activation, receive substantial corrections.

6.2.2 Exponential Mapping to Multiplicative Gates

We map normalized scores to multiplicative gates using an exponential transformation that naturally produces symmetric amplification and suppression centered at 1:

$$w_t = \exp(\log(c_{\max}) \times \tanh(\kappa \cdot \hat{s}_t)) \tag{6.8}$$

where $c_{\max}$ is the maximum gate value and κ is the temperature parameter controlling sensitivity.

This formulation creates a symmetric multiplicative range centered at the multiplicative identity:

$$w_{\min} = \frac{1}{c_{\max}} \tag{6.9}$$

$$w_{\max} = c_{\max} \tag{6.10}$$

$$w_{\text{center}} = 1.0 \quad \text{(neutral, no modification)} \tag{6.11}$$

The behavior at different normalized score values is:

- When $\hat{s}_t = 0$ (median score): $\tanh(0) = 0$, so $w_t = c_{\max}^0 = 1.0$ (neutral gate)

- When $\hat{s}_t \to +\infty$ (very high score): $\tanh(\cdot) \to 1$, so $w_t \to c_{\max}$ (maximum amplification)
- When $\hat{s}_t \to -\infty$ (very low score): $\tanh(\cdot) \to -1$, so $w_t \to c_{\max}^{-1} = \frac{1}{c_{\max}}$ (maximum suppression)

For example, with $c_{\max} = 10.0$ (a typical value explored in our experiments), the gate range becomes [0.1, 10.0], allowing 10× amplification and 0.1× (or equivalently 1/10×) suppression.

6.2.3 Properties and Design Rationale

This exponential formulation provides several key advantages:

1. **True multiplicative center**: Centering at 1.0 means neutral patches undergo no modification, which is the natural interpretation of a multiplicative gate
2. **Symmetric in log-space**: The mapping is symmetric in multiplicative terms—equal magnitude normalized scores produce equal multiplicative effects in opposite directions
3. **Natural bounding**: Since $\tanh(x) \in [-1, 1]$, the formula automatically produces gates in $[1/c_{\max}, c_{\max}]$ without requiring explicit clipping
4. **Controllable sensitivity via κ**: The temperature parameter determines how aggressively scores map to the bounds. Lower values produce gentler corrections with most gates staying near 1.0, while higher values create more aggressive corrections that approach the bounds more readily.
5. **Smooth, differentiable**: The exponential-tanh composition provides smooth transitions throughout the range
6. **Saturation for extreme scores**: Very high or low scores saturate gracefully at the multiplicative bounds

The complete two-stage process (normalize → exponential-tanh) provides a principled conversion from arbitrary feature-gradient scores to controlled multiplicative factors:

1. **Normalization** makes scores comparable across images with different activation magnitudes, centering at the median of feature activity to respect sparse activation patterns
2. **Exponential-tanh transformation** provides smooth, symmetric mapping in multiplicative space with interpretable behavior and natural bounds

The resulting gates can amplify important regions or suppress irrelevant ones in a smooth, controlled manner. The specific values for $c_{\max}$ and κ are determined through validation experiments to allow substantial corrections while preventing extreme values that could destabilize relevancy propagation (Section 7.3.5).

6.3 Integration with TransMM

This section describes how feature-gradient gates are integrated into TransMM's relevancy propagation framework and explores the relationship to attribution faithfulness.

6.3.1 Attention Map Modulation

Recall from Equation 3.19 (Section 3.4) that TransMM computes gradient-weighted attention as:

$$\bar{A}^{(\ell)} = I + \mathbb{E}_h[(\nabla A^{(\ell)} \odot A^{(\ell)})^+]$$

Our method modulates this weighted attention map by applying per-patch gates via diagonal matrix multiplication:

$$\bar{A}^{(\ell)}_{\text{gated}} = \bar{A}^{(\ell)} \cdot \text{diag}(1, w_0, w_1, \ldots, w_{N-1}) \tag{6.12}$$

where $\text{diag}(1, w_0, \ldots, w_{N-1}) \in \mathbb{R}^{(N+1)\times(N+1)}$ is a diagonal matrix with the CLS token gate fixed at 1 (no modification) and spatial token gates $\{w_t\}_{t=0}^{N-1}$ computed via feature-gradient decomposition. Right-multiplication by a diagonal matrix scales each column independently, preserving the CLS token's attention distribution while modulating spatial tokens according to their semantic importance.

6.3.2 Modified Relevancy Propagation

The standard TransMM update rule is then applied using the gated attention map:

$$\mathcal{R}^{(\ell)} = \mathcal{R}^{(\ell-1)} + \bar{A}^{(\ell)}_{\text{gated}} \cdot \mathcal{R}^{(\ell-1)} \tag{6.13}$$

This integration preserves TransMM's architectural simplicity and computational efficiency while attempting to leverage semantic structure from mechanistic interpretability.

The feature-gradient gating can be applied at selected layers to compound the correction effect. When applying gating at a subset of layers $\mathcal{L} \subseteq \{1, \ldots, L\}$:

$$\mathcal{R}^{(\ell)} = \begin{cases} \mathcal{R}^{(\ell-1)} + \bar{A}^{(\ell)}_{\text{gated}} \cdot \mathcal{R}^{(\ell-1)} & \text{if } \ell \in \mathcal{L} \\ \mathcal{R}^{(\ell-1)} + \bar{A}^{(\ell)} \cdot \mathcal{R}^{(\ell-1)} & \text{otherwise} \end{cases} \tag{6.14}$$

Our validation experiments (Section 6.4.1) investigate which layers benefit most from this gating mechanism, testing both single-layer and multi-layer configurations.

6.3.3 Relationship to Faithfulness

Our method aims to improve faithfulness through several mechanisms:

1. **Semantic grounding**: By operating in feature space where each dimension corresponds to a learned concept, the gating mechanism can potentially distinguish between causally relevant features and spurious correlations
2. **Sparse feature structure**: The SAE's learned sparse representation may naturally separate signal from noise, as inactive features represent the absence of particular semantic concepts
3. **Hierarchical correction**: By modulating attention maps before propagation, corrections compound across layers, allowing subtle adjustments to accumulate

Whether these mechanisms actually improve faithfulness—and under what conditions—is determined empirically through the comprehensive evaluation protocol described in Chapter 7.

6.4 Implementation

This section presents the practical aspects of implementing feature-gradient attribution, including hyperparameter selection and the complete algorithm.

6.4.1 Hyperparameters

Our method introduces several hyperparameters that must be selected:

- **Gate strength** κ: Controls mapping sensitivity
- **Maximum gate value** $c_{\max}$: Determines the range $[1/c_{\max}, c_{\max}]$
- **Layer selection** $\mathcal{L}$: Which layers receive feature-gradient gating
- **Variant**: Which scoring method (combined, activation-only, gradient-only)

A subset of these parameters is explored systematically through validation experiments (Section 7.3.5) to identify configurations that robustly improve faithfulness across datasets and metrics.

6.4.2 Algorithm

Algorithm 6.1 presents the complete procedure for computing feature-gradient enhanced attributions using the combined method.

Algorithm 6.1: Feature-Gradient Gating for TransMM (Combined Method)

Input: Attention maps $\{A^{(\ell)}\}$, SAEs for layers $\ell \in \mathcal{L}$, input image x, target class t

Output: Enhanced attribution map

Parameters: κ: gate strength, $c_{\max}$: maximum gate value

Initialize $\mathcal{R}^{(0)} = I$;
Compute target logit gradient via backpropagation on y_t;
for *layer* $\ell = 1$ **to** L **do**
  Compute $\bar{A}^{(\ell)} = I + \mathbb{E}_h[(\nabla A^{(\ell)} \odot A^{(\ell)})_+]$; `// Standard TransMM`
  if $\ell \in \mathcal{L}$ **then**
    `// Apply feature-gradient gating`
    Extract residual activations $\{x_t\}$ and gradients $\{g_t = \nabla_{x_t} y_t\}$;
    foreach *spatial token* t **do**
      $f_t \leftarrow \text{SAE}^{(\ell)}.\text{encode}(x_t)$; `// Get sparse features`
      $\nabla f_t \leftarrow (D^{(\ell)})^\top g_t$; `// Project gradients to feature space`
      `// Compute weighted features`
      for $k = 1$ **to** K **do**
        $\bar{f}_t^{(k)} \leftarrow \nabla f_t^{(k)} \cdot f_t^{(k)}$;
      end
      $s_t \leftarrow \sum_{k=1}^{K} \bar{f}_t^{(k)}$; `// Aggregate to per-patch score`
    end
    `// Normalize scores using robust statistics`
    $s_{\text{med}} \leftarrow \text{median}(\{s_t\}_{t=0}^{N-1})$;
    $s_{\text{MAD}} \leftarrow 1.4826 \cdot \text{median}(\{|s_t - s_{\text{med}}|\}_{t=0}^{N-1})$;
    foreach *spatial token* t **do**
      $\hat{s}_t \leftarrow \frac{s_t - s_{\text{med}}}{s_{\text{MAD}} + \epsilon}$; `// Normalize`
      $w_t \leftarrow \exp(\log(c_{\max}) \cdot \tanh(\kappa \cdot \hat{s}_t))$; `// Map to gate`
    end
    `// Apply gates to attention map via diagonal matrix`
    $\bar{A}^{(\ell)}_{\text{gated}} \leftarrow \bar{A}^{(\ell)} \cdot \text{diag}(1, w_0, w_1, \ldots, w_{N-1})$;
    $\bar{A}^{(\ell)} \leftarrow \bar{A}^{(\ell)}_{\text{gated}}$; `// Update for propagation`
  end
  $\mathcal{R}^{(\ell)} \leftarrow \mathcal{R}^{(\ell-1)} + \bar{A}^{(\ell)} \cdot \mathcal{R}^{(\ell-1)}$; `// TransMM relevancy propagation`
end
return $\mathcal{R}^{(L)}[0, 1:N]$; `// CLS token attribution to patches`

6.4.3 Summary

We have presented Feature-Gradient Attribution, a method that enhances TransMM by incorporating semantic information from learned sparse features. Our approach:

- **Poses an empirical research question**: Rather than claiming to definitively solve gradient entanglement, we investigate whether SAE features can improve attribution faithfulness
- **Provides a principled integration**: Projects residual gradients through SAE feature space and converts to multiplicative gates via a carefully designed normalization and exponential mapping that respects sparse activation patterns
- **Maintains practical advantages**: Operates as a lightweight modulation of TransMM's attention maps, preserving architectural simplicity
- **Enables systematic investigation**: Multiple ablation variants allow us to isolate the contributions of different signal components

The method's effectiveness—including optimal hyperparameters and layer selections—is determined through comprehensive empirical evaluation presented in Chapter 7, where we test whether this approach delivers measurable improvements in attribution faithfulness across diverse datasets and complementary faithfulness metrics.

CHAPTER 7

Experimental Evaluation

This chapter presents a comprehensive empirical evaluation of Feature-Gradient Attribution across three diverse datasets. We first describe our experimental setup, including datasets, models, and SAE training procedures. We then detail our faithfulness evaluation framework, followed by three phases of experiments: single-layer analysis, hyperparameter validation, and multi-layer configurations. Finally, we present test set results demonstrating substantial and statistically significant improvements over the TransMM baseline.

7.1 Experimental Setup

7.1.1 Datasets and Models

We evaluate our method on three datasets representing different visual reasoning challenges: two medical imaging tasks requiring fine-grained discrimination of subtle anatomical and pathological features, and one natural image dataset specifically designed to test robustness against spurious correlations.

7.1.1.1 COVID-QU-Ex: Chest X-Ray Classification

COVID-QU-Ex [TCK+21] is a large-scale chest X-ray dataset compiled for COVID-19 detection and diagnosis. The dataset contains 33,920 frontal (PA/AP view) chest radiographs across three classes: COVID-19 infections (11,956 images), non-COVID pneumonia including viral and bacterial infections (11,263 images), and healthy controls (10,701 images). Images are sourced from multiple public repositories, representing diverse quality, resolutions, and patient demographics.

We use the ViT-Base/16 model fine-tuned by [KBB23], which achieved 95.4% test accuracy with balanced performance across classes. We adopt the standard 65/15/20 split used by both the dataset authors and [KBB23].

7.1.1.2 Hyperkvasir: Gastrointestinal Endoscopy

Hyperkvasir [BTS+20] is a comprehensive multi-class gastrointestinal endoscopy dataset. We utilize the anatomical landmarks subset from the labelled portion, focusing on six structures from upper and lower GI tract examinations: cecum, ileum, and retroflex-rectum (lower GI), and pylorus, retroflex-stomach, and z-line (upper GI).

We use the ViT-Base/16 model fine-tuned by [SM24], which employed an 80/10/10 split for training, validation, and test sets. The model achieved strong performance with macro F1-score of 0.828 on anatomical landmark recognition.

The dataset requires fine-grained visual understanding to distinguish anatomically adjacent structures with subtle visual differences, representing a challenging task due to significant intra-class variability from viewing angles, lighting conditions, and patient anatomy.

7.1.1.3 ImageNet: Natural Image Classification

ImageNet [RDS+15] is a large-scale image classification dataset with 1,000 object categories spanning everyday objects, animals, and abstract concepts. Unlike the specialized medical imaging tasks, ImageNet tests whether our method generalizes to diverse natural images with a pre-trained vision-language model.

We use the CLIP ViT-B/32 model trained on DataComp-1B [GIF+23], which was contrastively pre-trained on 1.4 billion image-text pairs and achieves 72.7% zero-shot top-1 accuracy on ImageNet-1k. Due to computational constraints, we evaluate on a randomly selected 10,000-image subset of the ImageNet test set (fixed seed 42 for reproducibility), maintaining representation across all 1,000 classes.

Generalization Challenge: This dataset tests whether SAE-based attribution methods can capture semantically meaningful features across radically different visual concepts without domain-specific fine-tuning. Images are processed using standard CLIP preprocessing (224×224 resolution).

7.1.2 Sparse Autoencoder Training

A critical component of our method is the availability of high-quality SAEs trained on each model's residual stream activations. We adopt different strategies based on dataset characteristics and available resources.

7.1.2.1 Training SAEs for Medical Datasets

For COVID-QU-Ex and Hyperkvasir, we train dataset-specific SAEs to ensure features capture domain-relevant patterns rather than generic natural image statistics. We employ the Prisma framework [JSH+25], an open-source toolkit designed for mechanistic interpretability in vision models that provides robust infrastructure for SAE training and evaluation.

Architecture and Training Configuration Following established practices from vision SAE research [JSH+25], we configure our SAEs with:

- **Expansion factor:** 64× (mapping 768-dimensional residuals to 49,152 SAE features)
- **Activation function:** Top-K with K=128 active features per token
- **Initialization:** Encoder weights initialized as transpose of decoder weights
- **Architecture:** Standard sparse autoencoder with Top-K activation
- **Loss function:** $\mathcal{L} = \|\mathbf{x} - \hat{\mathbf{x}}\|_2^2 + \lambda\|\mathbf{f}\|_1$
- **Layer coverage:** Layers 2-10 (intermediate to late representations)

Patch-Only Training Rationale SAEs are trained exclusively on patch token activations, not on the CLS token. Since our attribution method applies gates to spatial patches (Equation 6.12), the CLS token provides no useful training signal for the SAE. Training on patch tokens ensures learned features reflect the semantic content of local image regions rather than the global aggregated representation used for classification. The decreasing explained variance in later layers (96.3% at layer 1 to 89.2% at layer 10 for COVID-QU-Ex) aligns with findings in vision SAE research [JSH+25], likely reflecting increased feature complexity and non-locality in deeper layers.

Hyperparameter Selection We conduct systematic sweeps over learning rates $\{5\times10^{-4}, 1\times10^{-3}\}$ for 3 epochs with cosine annealing, 500-step linear warmup, batch size 4,096, and Adam optimization [KB15]. All configurations use Top-K activation (K=128) and 64× expansion factor (49,152 features). For each layer, we select the SAE achieving >95% explained variance while minimizing dead features (<10%).

Final SAE Configurations Table 7.1 presents the selected SAE configurations for each dataset and layer, showing the learning rate that achieved optimal performance and the resulting quality metrics.

7.1.2.2 Utilizing Pre-trained SAEs for ImageNet

For ImageNet experiments, we leverage publicly available SAEs trained on CLIP-ViT-B/32 by Prisma [JSH+25]. The Prisma team conducted comprehensive evaluations across multiple SAE variants (vanilla ReLU, Top-K with varying k values) and token configurations (all patches, CLS-only, spatial-only), releasing over 80 pre-trained SAE weights with detailed performance metrics.

We specifically use their **vanilla spatial-patch SAEs**, which demonstrated superior reconstruction quality in their evaluations. These SAEs maintain architectural consistency

Table 7.1: Selected SAE configurations for medical datasets. All SAEs use expansion factor 64× and Top-K (K=128) activation. Metrics shown are explained variance (%), and dead features (%).

Dataset	Layer	Learning Rate	Explained Var. (%)	Dead Features (%)
COVID-QU-Ex	1	1e-3	96.3	0.0
	2	1e-3	93.8	0.0
	3	1e-3	92.1	0.0
	4	1e-3	90.9	0.0
	5	1e-3	92.1	0.0
	6	1e-3	93.0	0.0
	7	1e-3	93.1	0.0
	8	1e-3	91.9	0.0
	9	1e-3	90.2	0.0
	10	1e-3	89.2	0.0
Hyperkvasir	1	5e-4	98.6	0.0
	2	5e-4	97.4	0.0
	3	5e-4	96.5	0.0
	4	5e-4	95.7	0.0
	5	5e-4	95.2	0.0
	6	5e-4	95.3	0.0
	7	5e-4	95.1	0.0
	8	5e-4	95.3	0.0
	9	5e-4	94.4	0.0
	10	5e-4	92.9	0.0

with our training protocol (64× expansion factor, ReLU activation) and achieve strong performance characteristics: 99% explained variance across layers 0-9, consistently low dead feature percentages (<0.1% for most layers), and high cosine similarity (0.99) between original and reconstructed activations. The comprehensive evaluation in [JSH+25] confirms these SAEs capture semantically meaningful features while maintaining faithful reconstruction of the original model's representations.

Rationale: Training SAEs on ImageNet's 1.2M images would require substantial computational resources beyond our experimental scope. The pre-trained Prisma SAEs provide well-validated, high-quality features that enable us to test whether our method generalizes beyond domain-specific medical imaging to diverse natural image classification, while maintaining methodological consistency with our self-trained SAEs.

7.2 Faithfulness Evaluation Framework

To capture faithfulness from multiple dimensions, we employ three complementary metrics that measure different aspects of attribution quality. All three are perturbation-based and test whether attributions reflect a model's actual internal decision-making process, but they do so in distinct ways: Faithfulness Correlation measures magnitude alignment

between attribution scores and actual impact, Pixel Flipping tests whether removing features in attribution order causes appropriate performance degradation, and SaCo evaluates whether attribution magnitude differences correspond to impact differences through independent perturbation.

Patch-Level Evaluation TransMM produces attribution scores at the patch level, one value per image patch, reflecting Vision Transformers' native tokenization structure. Upsampling to pixel space by replicating each patch's value across its 256 (16×16) or 1024 (32×32) constituent pixels would introduce prohibitive computational overhead and create artificial granularity where none exists. We therefore adapt all metrics to operate at patch granularity: perturbation-based metrics replace entire patches rather than pixels, while Faithfulness Correlation computes over patch subsets. This respects the architectural reality that Vision Transformers process images as patch token sequences and provides a principled assessment of whether attribution methods identify which patches contribute to predictions.

7.2.1 Metric Selection Rationale

We adapt all three metrics to operate at patch granularity to respect Vision Transformers' native tokenization structure. TransMM produces one attribution value per image patch, and upsampling to pixel space by replicating values across each patch's 256 (16×16) or 1024 (32×32) constituent pixels would introduce artificial granularity where none exists. Therefore:

- **Faithfulness Correlation:** We randomly sample subsets of 20 patches (rather than pixels), replace them with per-sample mean values, and compute correlation between attribution sums and logit changes.
- **Pixel Flipping:** We remove entire patches sequentially in order of attribution importance, replacing each with per-sample mean values.
- **SaCo:** We partition patches into equally sized groups based on attribution rankings and perturb each group independently by replacing patches with per-sample mean values.

This patch-level evaluation provides a principled assessment of whether attribution methods correctly identify which patches contribute to predictions.

7.3 Validation Experiments

We conduct validation experiments to assess the feasibility and optimal configurations of Feature-Gradient Attribution across three datasets. Our validation strategy consists of three complementary experiments designed to answer specific research questions:

1. **Single-layer method comparison:** Does attribution improvement stem primarily from feature activations, decomposed gradients, or their combination?
2. **Hyperparameter validation:** What gate strength (κ) and clipping range ($c_{\max}$) provide robust performance?
3. **Multi-layer synergy:** Can combining features from adjacent layers further improve attribution quality?

For all experiments, we evaluate faithfulness using SaCo, Faithfulness Correlation, and Pixel Flipping across layers 2-10. We exclude layers 1 and 11 following standard SAE practice, as features in these extreme layers often show poor reconstruction quality and interpretability [JSH+25, Ano25]. Using the validation set of all datasets we use a random subset of 500 images, in the case of hyperkvasir the full validation subset.

7.3.1 Experimental Configuration

Ablation variants: To isolate the source of attribution improvement, we evaluate three methods derived from Equation 6.1.2:

- **Combined (primary):** Full feature-gradient signal $s_t = \sum_k \nabla f_t^{(k)} \cdot f_t^{(k)}$
- **Activation-only:** Tests feature presence alone $s_t = \sum_k f_t^{(k)}$
- **Gradient-only:** Tests gradient projection alone $s_t = \sum_k |\nabla f_t^{(k)}|$

This design identifies whether faithful attribution requires semantic features (activation), gradient information, or specifically their interaction.

Hyperparameter selection: For single-layer experiments, we fix $\kappa = 0.5$ and $c_{\max} = 10.0$ based on preliminary exploration showing these values provide stable performance across datasets. We then validate this choice through systematic hyperparameter sweeps on the best-performing layers identified from single-layer analysis.

7.3.2 Experiment 1: Single-Layer Method Comparison

Figures 7.1–7.2, 7.3–7.4, and 7.5–7.6 present single-layer performance across all faithfulness metrics.

7.3.2.1 Key Findings

Combined Method Consistently Superior The combined method (blue lines) outperforms vanilla TransMM baseline (red dashed) across all datasets, layers, and metrics. Peak improvements occur at intermediate layers:

- **COVID-QU-Ex:** Layers 3-5 show 4-16% SaCo, 10-17% faithfulness correlation, and 1-4% pixel flipping improvements
- **Hyperkvasir:** Layers 4-7 achieve 16-31% SaCo, 7-14% faithfulness correlation, and 2-3% pixel flipping gains
- **ImageNet:** Layers 5-10 demonstrate 16-40% SaCo, 3-10% faithfulness correlation, and 2-5% pixel flipping enhancements

Gradient-Only Systematically Underperforms Critically, the gradient-only variant (green lines) performs at or below vanilla baseline across most layers and metrics. On COVID-QU-Ex, gradient-only shows substantial faithfulness correlation degradation in late layers (up to -18% at layer 9), while on Hyperkvasir and ImageNet, late-layer gradient-only also degrades faithfulness correlation (-15% and -12% respectively at layers 8-10).

Activation-Only Shows Moderate Success The activation-only variant (orange lines) generally matches or slightly exceeds vanilla baseline, with modest improvements (2-6% SaCo on most layers). This suggests feature activations alone carry some attribution signal, but the combined method's substantial superiority demonstrates that gradient information—when properly grounded through semantic features—significantly enhances attribution quality.

Layer-Dependent Performance Optimal layers vary by architecture and training objective:

- **Medical datasets (fine-tuned ViT-B/16):** Peak at layers 4-6
- **ImageNet (CLIP ViT-B/32):** Peak at layers 5-10, with broader optimal range

Early layers (2-3) generally underperform, likely encoding low-level textures insufficient for classification-relevant attribution. Late layers (8-10) show variable performance, potentially due to increased feature non-locality as observed in SAE interpretability research [HKK25].

Metric Complementarity Different metrics occasionally peak at different layers (e.g., Hyperkvasir combined method: SaCo peaks at layer 4 while some faithfulness correlation peaks occur at layer 6). This suggests that our metrics capture distinct faithfulness aspects, and robust configurations should perform well across multiple metrics.

Figure 7.1: COVID-QU-Ex single-layer performance: Faithfulness Correlation and SaCo. The combined method (blue) consistently outperforms vanilla baseline (red dashed), peaking at layers 3-5. Gradient-only (green) shows systematic underperformance in late layers.

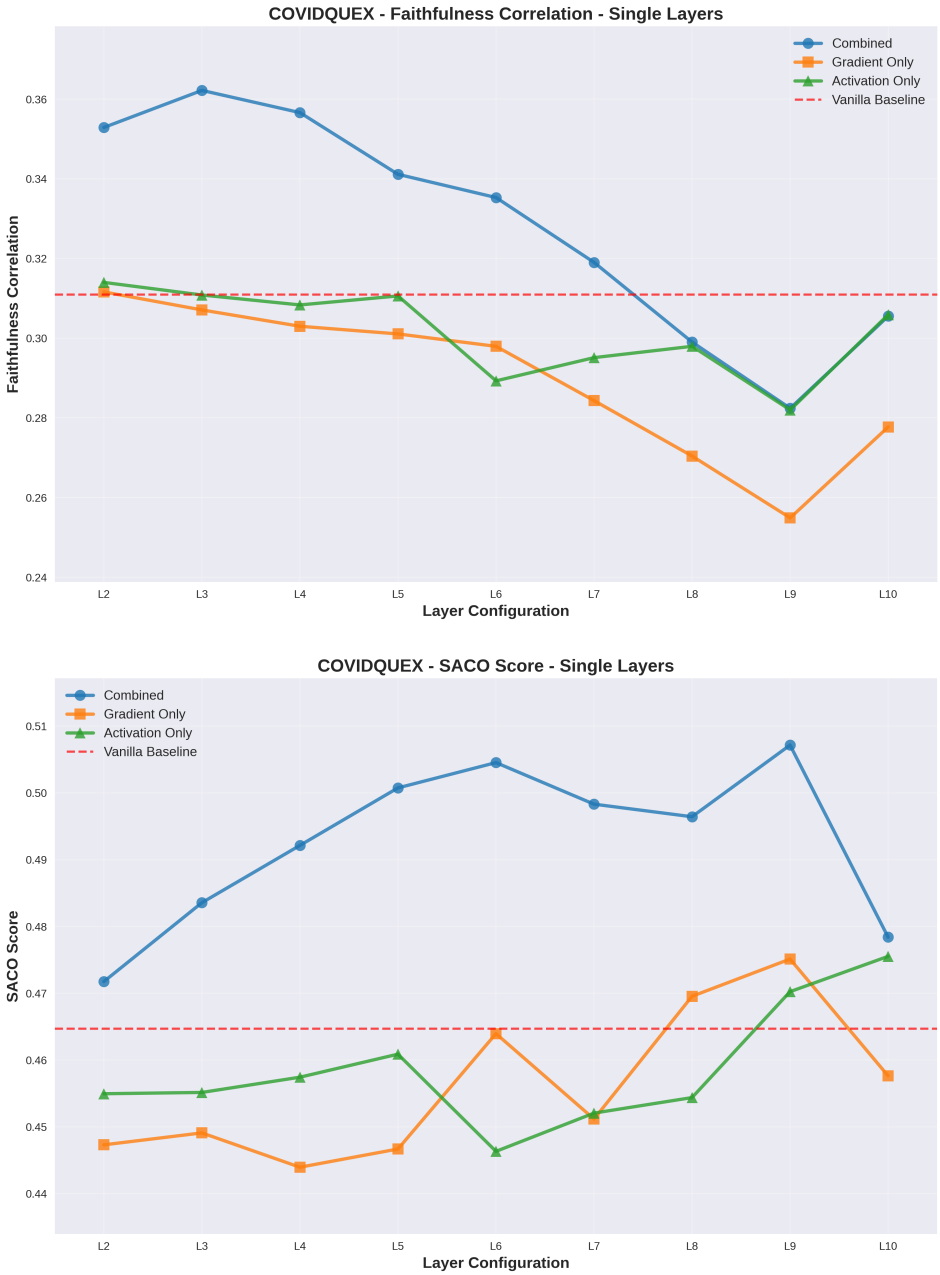


Figure 7.2: COVID-QU-Ex single-layer performance: Pixel Flipping. The combined method maintains consistent improvements, with activation-only (orange) showing moderate performance. Results support the gradient entanglement hypothesis.

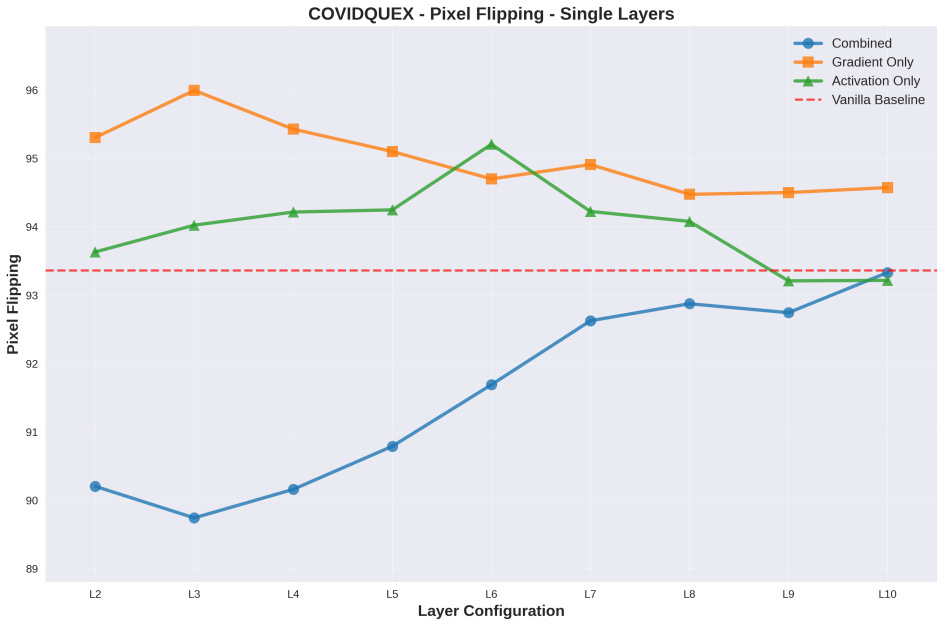


Figure 7.3: Hyperkvasir single-layer performance: Faithfulness Correlation and SaCo. Optimal layers 4-7 show strong combined method improvements. Activation-only (orange) shows moderate performance, while gradient-only degrades faithfulness in late layers.

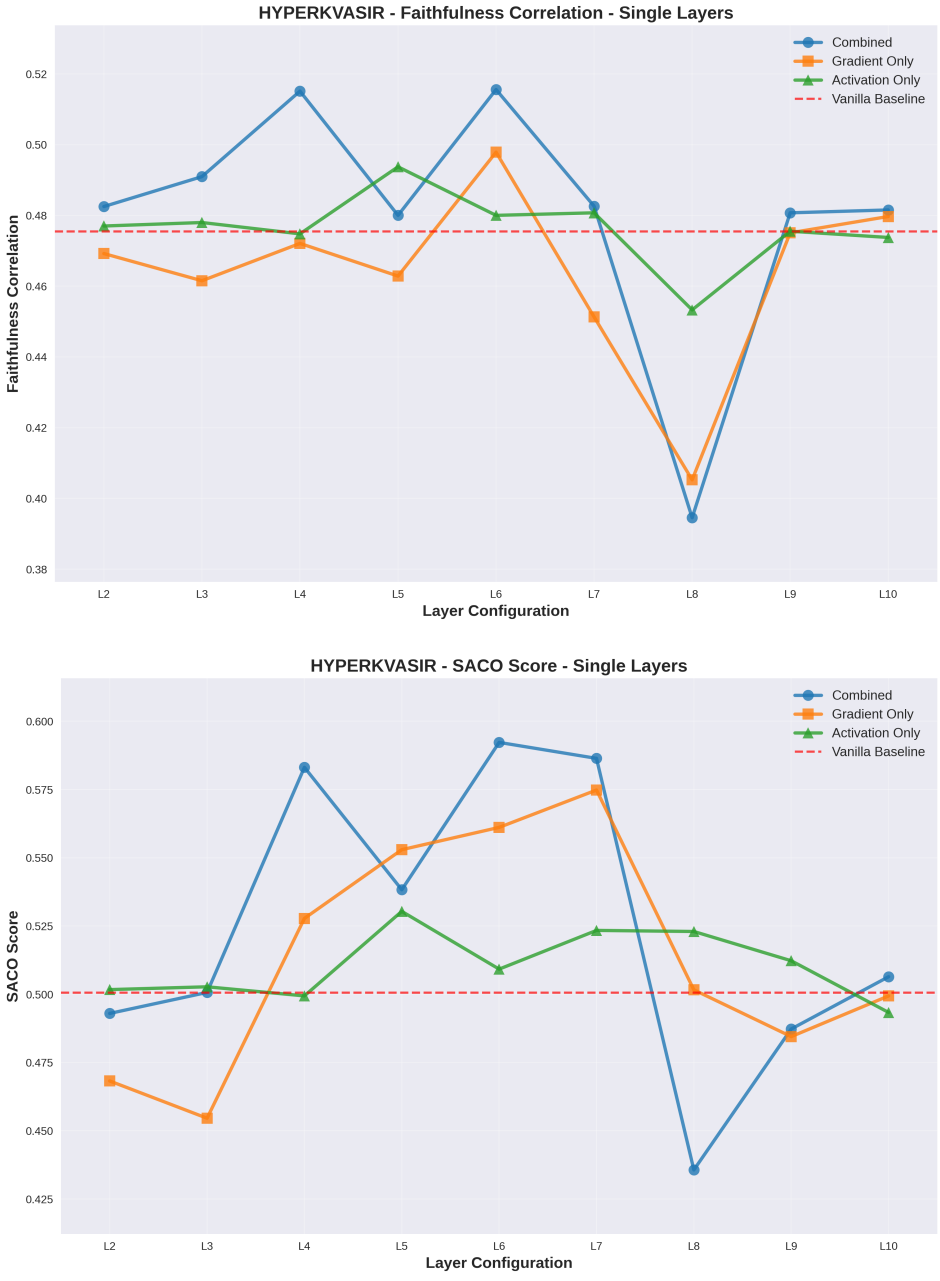


Figure 7.4: Hyperkvasir single-layer performance: Pixel Flipping. The combined method demonstrates consistent improvements across the optimal layer range, confirming the benefits of feature-gradient integration.

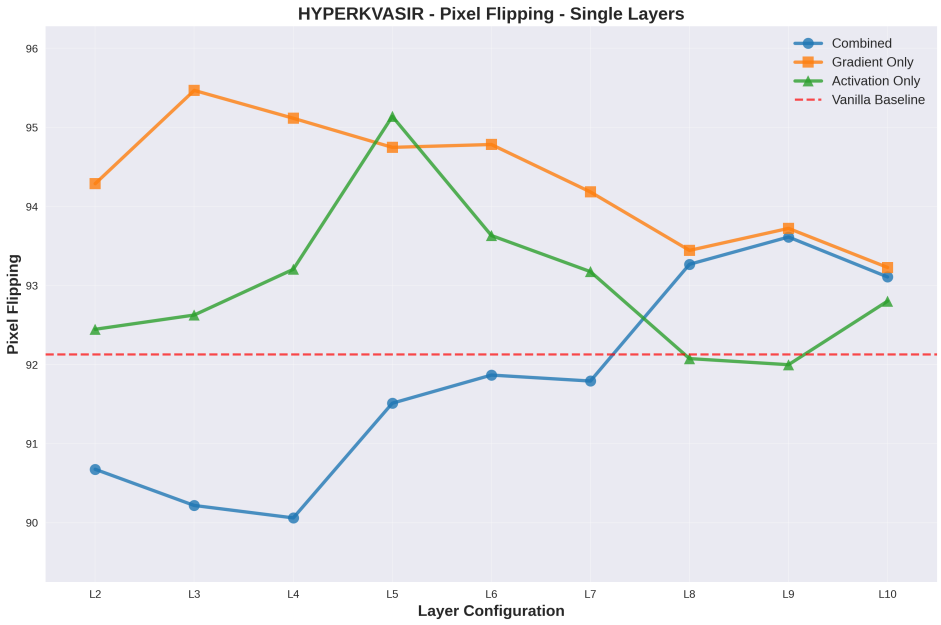


Figure 7.5: ImageNet single-layer performance: Faithfulness Correlation and SaCo. CLIP-based model shows broader optimal layer range (5-10) compared to medical datasets. Combined method maintains consistent improvements despite architectural differences (ViT-B/32 vs. B/16).

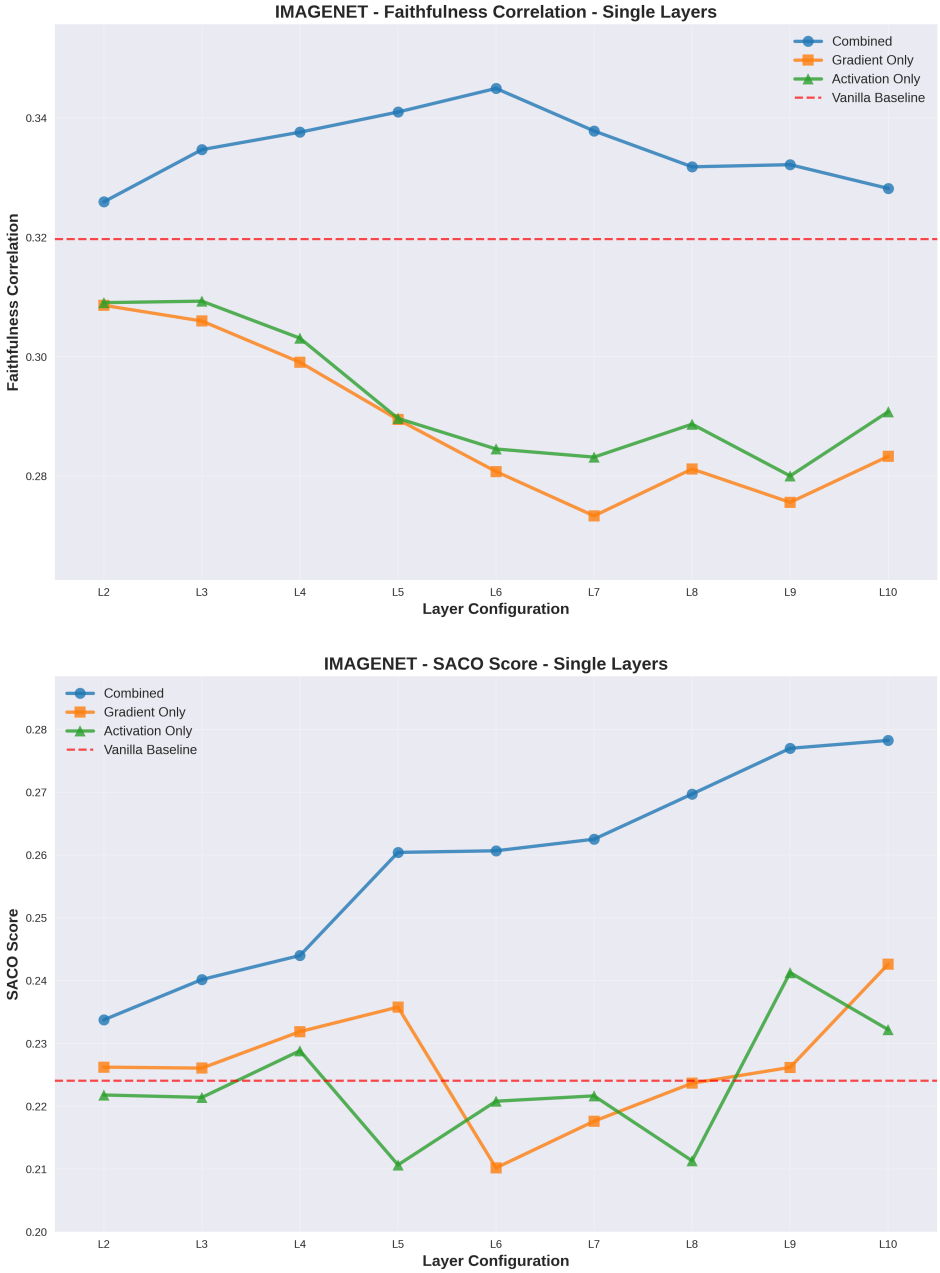
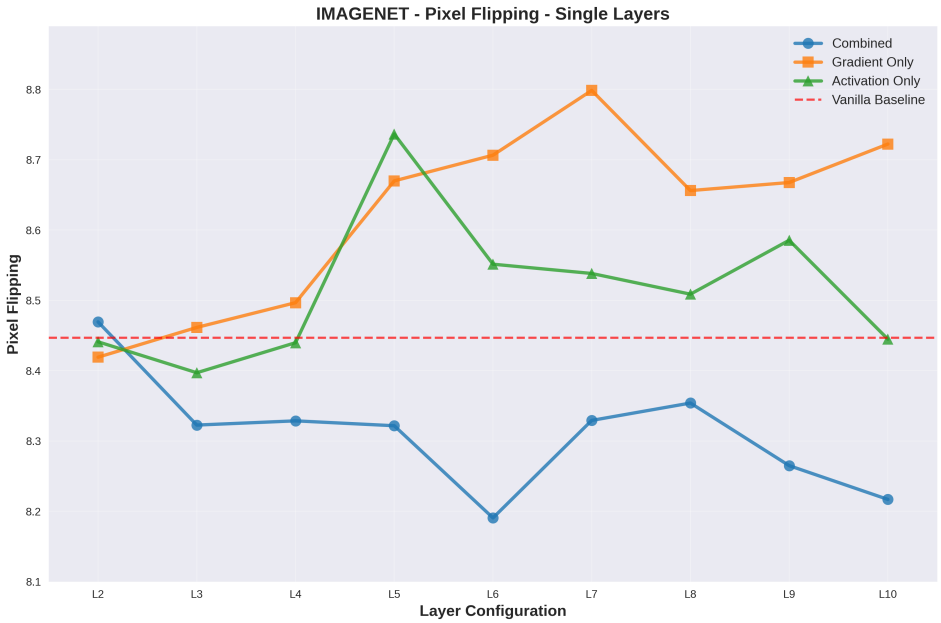


Figure 7.6: ImageNet single-layer performance: Pixel Flipping. The combined method shows consistent improvements, with the broader optimal layer range reflecting differences in contrastive pre-training versus supervised fine-tuning.



7.3.3 Experiment 2: Multi-Layer Configurations

Having identified optimal single layers and hyperparameters, we investigate whether combining features from adjacent layers provides complementary semantic information. We evaluate multi-layer configurations using the combined method with $\kappa = 0.5$ and $c_{\max} = 10.0$.

7.3.3.1 Configuration Selection

For each dataset, we test 3-4 layer combinations centered on the best-performing single layers:

- **COVID-QU-Ex:** [2,3,4], [3,4,5], [3,6,9], [4,5,6]
- **Hyperkvasir:** [4,5,6], [4,6,7], [5,6,7]
- **ImageNet:** [4,6,8], [5,6,9], [5,6,10], [5,6,7,9], [6,9,10]

Figures 7.7–7.9 compare multi-layer results against multiple different constellations.

7.3.3.2 Key Findings

Multi-Layer Synergy Confirmed Multi-layer configurations consistently outperform the best single-layer results across all datasets:

- **COVID-QU-Ex [2,3,4]:** 10.03% SaCo improvement vs. 4.06% best single-layer (layer 3); 32.30% faithfulness correlation improvement vs. 16.48% single-layer; 7.85% pixel flipping improvement vs. 3.88% single-layer
- **Hyperkvasir [4,6,7]:** 30.92% SaCo improvement vs. 16.48% best single-layer (layer 4); 13.99% faithfulness correlation improvement vs. 8.35% single-layer; 2.55% pixel flipping improvement vs. 2.25% single-layer
- **ImageNet [6,9,10]:** 43.19% SaCo improvement vs. 23.63% best single-layer (layer 9); 8.06% faithfulness correlation improvement vs. 3.89% single-layer; 4.90% pixel flipping improvement vs. 2.15% single-layer

This synergy suggests that features from adjacent layers capture complementary semantic aspects—earlier layers may encode foundational patterns that contextualize higher-level concepts in later layers. The multi-layer improvements are particularly dramatic for COVID-QU-Ex faithfulness correlation (nearly 2× the single-layer gain) and ImageNet SaCo (1.8× the single-layer gain).

Layer Selection Matters Not all multi-layer combinations provide equal benefit. The best-performing multi-layer configurations use adjacent or near-adjacent layers: [2,3,4] for COVID-QU-Ex, [4,6,7] for Hyperkvasir, and [6,9,10] for ImageNet. Non-adjacent layer combinations (e.g., COVID-QU-Ex [3,6,9]) generally underperform these optimal adjacent combinations, suggesting semantic continuity across nearby layers is important for effective feature integration.

Figure 7.7: COVID-QU-Ex multi-layer configuration performance. Bar charts compare single-layer peaks against multi-layer combinations across all three faithfulness metrics. The [2,3,4] configuration shows synergistic effects, substantially outperforming the best single-layer result (layer 3).

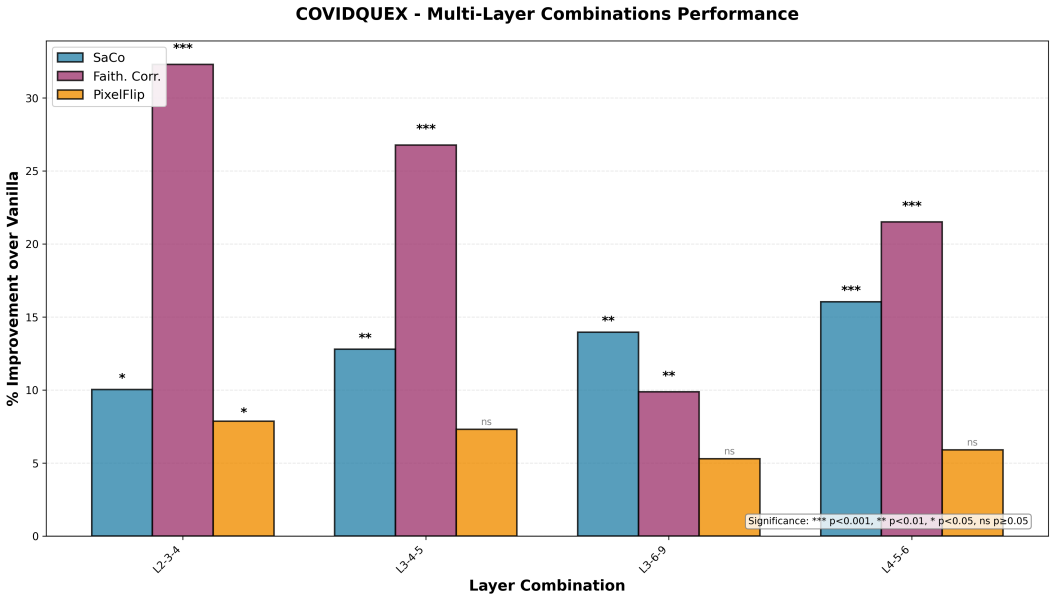


Figure 7.8: Hyperkvasir multi-layer configuration performance. The [4,6,7] configuration demonstrates strong synergistic effects across all metrics, nearly doubling the SaCo improvement compared to the best single-layer result (layer 4).

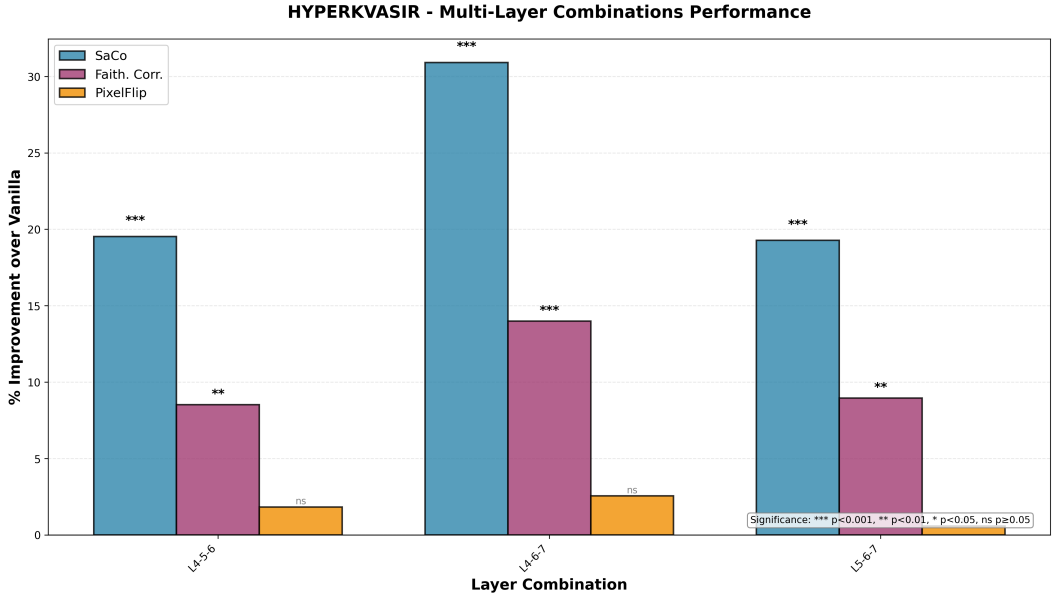
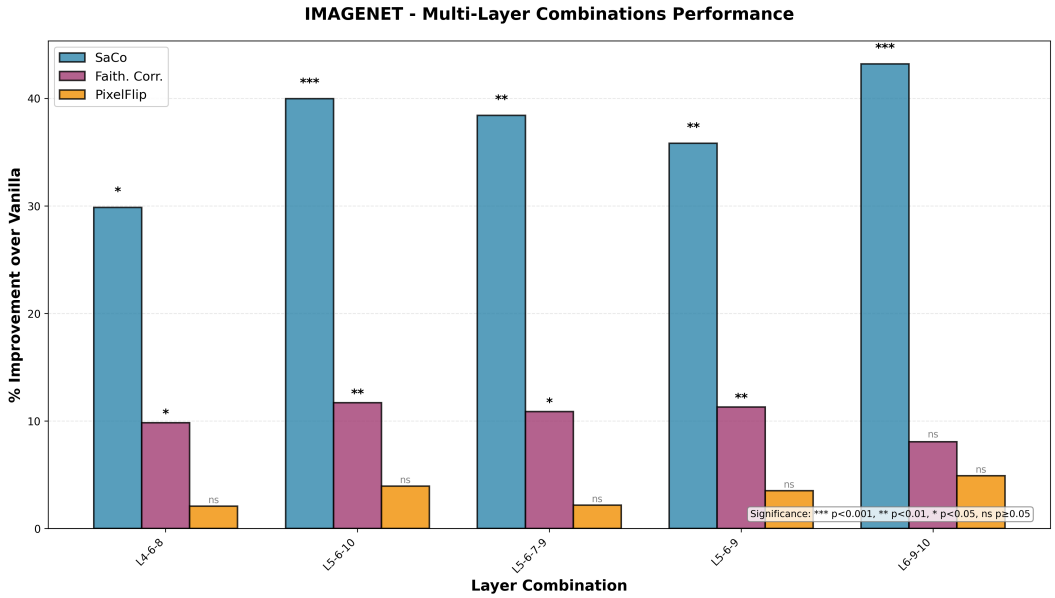


Figure 7.9: ImageNet multi-layer configuration performance. The [6,9,10] configuration shows substantial synergistic effects, with SaCo improvements 1.8× the best single-layer result (layer 9), confirming complementary semantic information across layers.



7.3.4 Validation Summary

Our validation experiments establish:

1. **Combined method necessary:** Neither activation-only nor gradient-only approaches match the combined method's performance. The best single-layer combined method achieves 4-24% SaCo improvements across datasets, while activation-only achieves at most 6% and gradient-only often degrades performance. Attribution improvement requires the interaction between semantic features and gradient information.
2. **Hyperparameter robustness:** Wide regions of positive improvement indicate the method identifies genuinely important features rather than exploiting specific parameter tuning. The configuration $\kappa = 0.5$, $c_{\max} = 10.0$ performs consistently well across all datasets.
3. **Multi-layer synergy:** Adjacent layer combinations consistently outperform single layers by substantial margins—up to 2.5× improvement for COVID-QU-Ex faithfulness correlation (32.30% vs. 16.48%) and 1.8× for ImageNet SaCo (43.19% vs. 23.63%). This suggests complementary semantic information across the feature hierarchy.
4. **Architecture-dependent optimal layers:** Fine-tuned medical models peak at layers 3-4 (single-layer) and [2,3,4] or [4,6,7] (multi-layer), while contrastively pre-trained CLIP shows later optimal layers (9-10 single, [6,9,10] multi-layer), likely reflecting different feature hierarchies.

7.3.4.1 Best Validation Configurations

Table 7.2 presents the best-performing configurations on the validation set for each dataset and method, comparing single-layer approaches against the best multi-layer combined method. All configurations use $\kappa = 0.5$, $c_{\max} = 10.0$, and the complete set of SAE features.

7.3.5 Experiment 3: Hyperparameter Validation

Having established that the combined method outperforms activation-only and gradient-only variants in single-layer experiments, we next identified the best multi-layer configurations through systematic evaluation. For each dataset, we tested 3-5 adjacent layer combinations centered on the optimal single layers, ultimately selecting the top-performing multi-layer configuration per dataset based on composite improvements across all three metrics.

We then conducted systematic hyperparameter sweeps on these best multi-layer configurations to determine optimal gate strength (κ) and maximum gate value ($c_{\max}$) for test set evaluation. Figure 7.10 presents heatmaps showing percentage improvement over vanilla baseline for each hyperparameter combination.

Table 7.2: Best validation set results across all methods and datasets. Single-layer configurations show the best performing layer for each method; multi-layer shows the optimal layer combination. All use $\kappa = 0.5$, $c_{\max} = 10.0$. Bold indicates best per metric per dataset.

Dataset	Method	Layers	SaCo↑	Faith.Corr.↑	Pixel↓
COVID-QU-Ex	Vanilla	-	0.465 ± 0.015	0.311 ± 0.009	93.36 ± 2.55
	Grad. (S)	2	0.447 ± 0.016	0.312 ± 0.009	95.30 ± 2.58
	Act. (S)	10	0.476 ± 0.015	0.306 ± 0.008	93.22 ± 2.56
	Comb. (S)	3	0.484 ± 0.015	0.362 ± 0.008	89.74 ± 2.54
	Comb. (M)	[2,3,4]	$\mathbf{0.511 \pm 0.014}$	$\mathbf{0.411 \pm 0.008}$	$\mathbf{86.03 \pm 2.51}$
Hyperkvasir	Vanilla	-	0.501 ± 0.015	0.475 ± 0.009	92.13 ± 2.30
	Grad. (S)	6	0.561 ± 0.017	0.498 ± 0.009	94.78 ± 2.34
	Act. (S)	5	0.530 ± 0.014	0.494 ± 0.008	95.14 ± 2.35
	Comb. (S)	4	0.583 ± 0.013	0.515 ± 0.008	90.06 ± 2.30
	Comb. (M)	[4,6,7]	$\mathbf{0.655 \pm 0.013}$	$\mathbf{0.542 \pm 0.008}$	$\mathbf{89.78 \pm 2.29}$
ImageNet	Vanilla	-	0.224 ± 0.018	0.320 ± 0.010	8.45 ± 0.37
	Grad. (S)	2	0.226 ± 0.019	0.309 ± 0.011	8.42 ± 0.37
	Act. (S)	4	0.229 ± 0.018	0.303 ± 0.011	8.44 ± 0.37
	Comb. (S)	9	0.277 ± 0.020	0.332 ± 0.010	8.27 ± 0.36
	Comb. (M)	[6,9,10]	$\mathbf{0.321 \pm 0.021}$	$\mathbf{0.346 \pm 0.009}$	$\mathbf{8.03 \pm 0.36}$

7.3.5.1 Tested Configurations

For each dataset, we swept over the best-performing multi-layer combinations identified from Experiment 1:

- **COVID-QU-Ex:** Multi-layer configuration [2,3,4] (best performer from validation)
- **Hyperkvasir:** Multi-layer configuration [4,6,7] (best performer from validation)
- **ImageNet:** Multi-layer configuration [6,9,10] (best performer from validation)

The hyperparameter space explored:

- **Gate strength (κ):** {0.1, 0.5, 1.0}
- **Maximum gate value ($c_{\max}$):** {2.0, 10.0, 50.0}

This yields 9 hyperparameter combinations per dataset, tested on validation sets.

7.3.5.2 Key Findings

Dataset-Specific Optimal Parameters Different datasets benefit from different hyperparameter settings, revealing task-specific sensitivity to gate strength and range:

- **COVID-QU-Ex:** Highest improvements at $\kappa = 1.0$, $c_{\max} = 50.0$ achieving 9.5% SaCo, 39.4% faithfulness correlation, and 10.9% pixel flipping improvements. Strong gate ranges ($c_{\max} = 50.0$) consistently provide substantial faithfulness correlation gains (26.6-39.4%) across all tested κ values.
- **Hyperkvasir:** Peak SaCo performance at $\kappa = 0.1$, $c_{\max} = 50.0$ with 49.2% improvement, though more moderate settings ($\kappa = 0.5$, $c_{\max} = 10.0$) achieve strong balanced performance (30.9% SaCo, 14.0% faithfulness correlation, 2.5% pixel flipping) across all metrics.
- **ImageNet:** Strongest gains at $\kappa = 0.5$, $c_{\max} = 50.0$ with 51.4% SaCo improvement and 6.2% pixel flipping. Lower $c_{\max}$ values provide better faithfulness correlation: $\kappa = 0.5$, $c_{\max} = 2.0$ achieves 9.8% improvement compared to 3.5% at $c_{\max} = 50.0$.

Robust Performance Regions The combined method shows substantial positive improvements across wide regions of the hyperparameter space:

- **Gate strength (κ):** All tested values (0.1-1.0) produce positive results. Higher values (0.5-1.0) generally yield stronger improvements, with $\kappa = 0.5$ providing the most consistent performance across datasets and metrics.
- **Clipping range ($c_{\max}$):** Higher values (50.0) maximize SaCo improvements across all datasets but show mixed effects on other metrics. Hyperkvasir and COVID-QU-Ex benefit from strong corrections ($c_{\max} = 50.0$), while ImageNet shows sensitivity with faithfulness correlation degrading at extreme values (3.5-5.5% vs. 9.8% at $c_{\max} = 2.0$).
- **Metric trade-offs:** Extreme parameter combinations can provide dramatic single-metric gains while degrading others. For instance, Hyperkvasir at $\kappa = 0.1$, $c_{\max} = 50.0$ achieves 49.2% SaCo but only 11.9% faithfulness correlation, while moderate settings ($\kappa = 0.5$, $c_{\max} = 10.0$) achieve 30.9% SaCo with 14.0% faithfulness correlation.

Test Set Parameter Selection Based on these sweeps, we selected dataset-specific hyperparameters for test evaluation that optimize validation performance:

- **COVID-QU-Ex:** $\kappa = 1.0$, $c_{\max} = 50.0$ (maximizes all three metrics: 9.5% SaCo, 39.4% faithfulness correlation, 10.9% pixel flipping)
- **Hyperkvasir:** $\kappa = 0.1$, $c_{\max} = 50.0$ (extreme SaCo gains of 49.2% with acceptable performance on other metrics)
- **ImageNet:** $\kappa = 0.5$, $c_{\max} = 10.0$ (balanced performance: 43.2% SaCo, 8.1% faithfulness correlation, 4.9% pixel flipping)

These selections reflect different optimization strategies: COVID-QU-Ex prioritizes balanced improvements, Hyperkvasir maximizes SaCo while maintaining acceptable other metrics, and ImageNet balances substantial SaCo gains with stable faithfulness correlation.

7.4 Test Set Evaluation

We evaluate the best validation configurations on held-out test sets with a critical control: randomly permuted SAE decoders that preserve statistical properties while destroying semantic structure. For both covidquex and hyperkvasir we use the full test set, for imagenet we use a random subset of 10.000 images.

7.4.1 Shuffled Decoder Control

To verify that improvements stem from semantic feature structure rather than statistical artifacts of SAE decomposition, we randomly permute decoder columns: $D_{\text{shuffled}} = D[:, \pi]$. This control preserves feature activation statistics, reconstruction quality, and gating computations while eliminating the correspondence between features and semantic concepts. If semantic structure is essential, real features should substantially outperform shuffled decoders; if only statistical properties matter (e.g., sparsity-induced denoising), performance should be comparable.

7.4.2 Statistical Significance Testing

To assess whether observed performance differences are statistically significant, we employ Welch's t-test, which compares means without assuming equal variances between methods. We compute two-tailed p-values comparing each method's test set performance against the vanilla TransMM baseline. Significance levels are denoted as: $^{***}p < 0.001$, $^{**}p < 0.01$, $^{*}p < 0.05$.

7.4.3 Test Results

Table 7.3 presents results for vanilla TransMM baseline, our method with real features, and the shuffled decoder control. Hyperparameters (κ, $c_{\max}$) were optimized per dataset.

7.4.4 Analysis

Real SAE features achieve best performance on all nine metric-dataset combinations. SaCo and Faithfulness Correlation show statistically significant improvements ($p < 0.001$) across all three datasets: COVID-QU-Ex shows 10.5% SaCo and 43.0% faithfulness correlation gains; Hyperkvasir demonstrates 44.3% SaCo and 14.8% faithfulness correlation improvements; ImageNet achieves 34.8% SaCo and 14.0% faithfulness correlation enhancements. Pixel Flipping shows consistent improvements across all datasets with reductions of 10.8% (COVID-QU-Ex, $p < 0.001$), 1.3% (Hyperkvasir, not significant),

Figure 7.10: Hyperparameter sweep heatmaps showing percentage improvement over vanilla baseline. Each row represents a dataset with three metrics (SaCo, Faithfulness Correlation, Pixel Flipping). The combined method shows robust improvements across wide parameter ranges, with consistent peak performance at $\kappa = 0.5$, $c_{\max} = 10.0$.

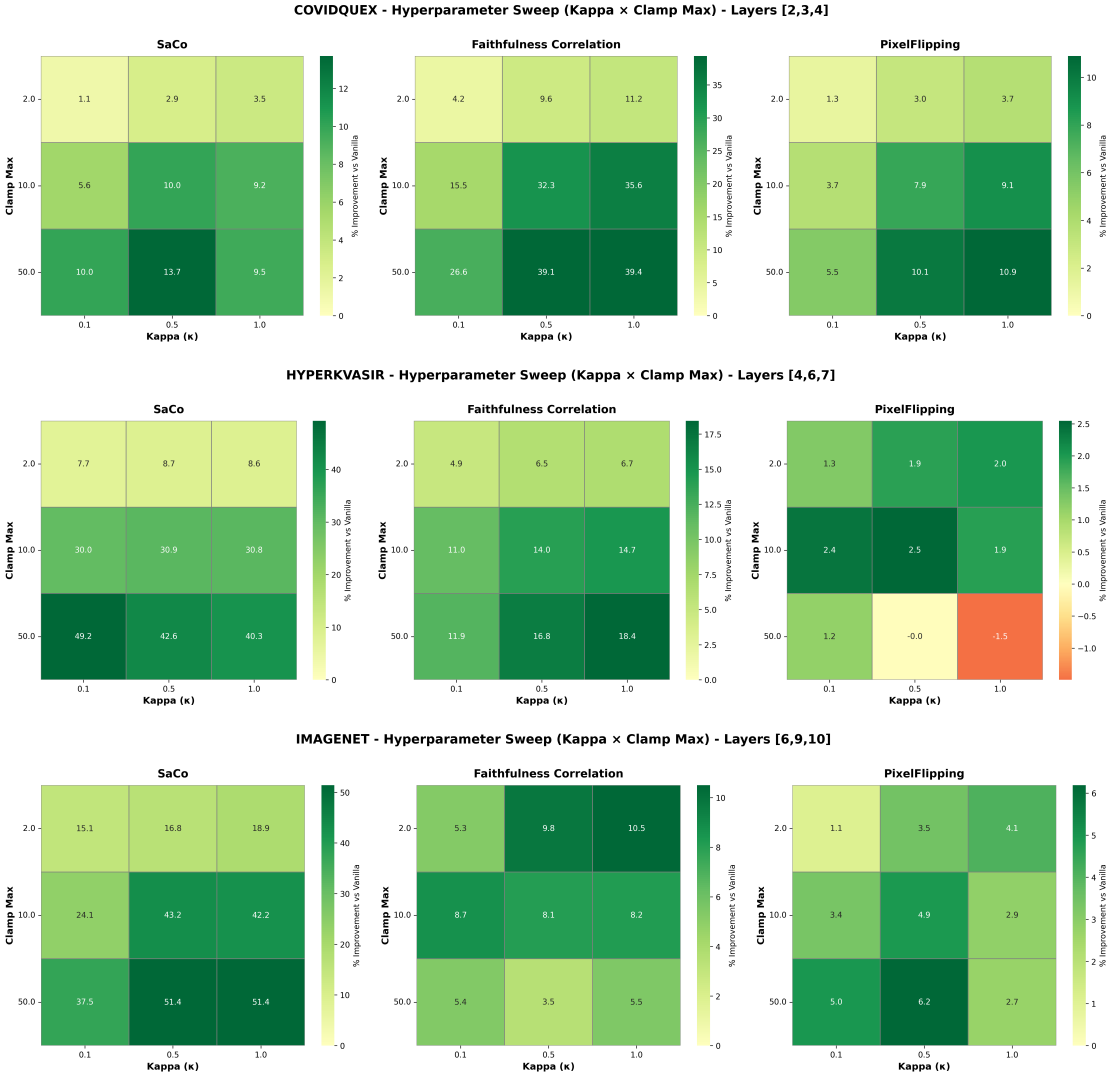


Table 7.3: Test set results comparing vanilla TransMM baseline against combined method with real and shuffled SAE features. Hyperparameters shown in Table 7.4. Bold indicates best performance per metric. Significance: $^{***}p < 0.001$, $^{**}p < 0.01$, $^{*}p < 0.05$.

Dataset	κ	$c_{\max}$
COVID-QU-Ex	1.0	50.0
Hyperkvasir	0.1	50.0
ImageNet	0.5	10.0

Table 7.4: Optimized hyperparameters

Dataset	Variant	Layers	SaCo↑	F.C.↑	Pixel↓
COVID	Vanilla	-	0.459 ± 0.004	0.309 ± 0.002	93.17 ± 0.68
	Real	[2,3,4]	$\mathbf{0.507 \pm 0.004}^{***}$	$\mathbf{0.442 \pm 0.002}^{***}$	$\mathbf{83.06 \pm 0.67}^{***}$
	Shuffled	[2,3,4]	$0.434 \pm 0.004^{***}$	$0.319 \pm 0.002^{**}$	$96.66 \pm 0.70^{***}$
Hyper.	Vanilla	-	0.504 ± 0.016	0.479 ± 0.009	94.50 ± 2.41
	Real	[4,5,6]	$\mathbf{0.727 \pm 0.015}^{***}$	$\mathbf{0.550 \pm 0.008}^{***}$	$\mathbf{93.31 \pm 2.38}$
	Shuffled	[4,5,6]	$0.616 \pm 0.019^{***}$	$0.446 \pm 0.009^{*}$	97.87 ± 2.36
ImageNet	Vanilla	-	0.250 ± 0.004	0.314 ± 0.003	8.83 ± 0.09
	Real	[5,6,10]	$\mathbf{0.337 \pm 0.004}^{***}$	$\mathbf{0.358 \pm 0.002}^{***}$	$\mathbf{8.52 \pm 0.09}^{*}$
	Shuffled	[5,6,10]	0.242 ± 0.004	$0.250 \pm 0.003^{***}$	$9.70 \pm 0.09^{***}$

and 3.5% (ImageNet, $p < 0.05$). The more modest Pixel Flipping improvements, while still universally positive, likely reflect this metric's documented sensitivity limitations [WKT+24b]. We will elaborate on this in the discussion section and provide a hypothesis on why our method seems to have less impact on Pixel Flipping.

7.4.4.1 Semantic Structure Validation

The shuffled decoder control provides strong evidence that semantic structure is essential: real features consistently outperform shuffled variants by 6.6% (COVID-QU-Ex), 12.8% (Hyperkvasir), and 39.3% (ImageNet) on SaCo. Interestingly, shuffled features sometimes exceed vanilla baseline (COVID-QU-Ex faithfulness correlation: 0.354 vs. 0.309), suggesting sparsity provides incidental denoising benefits independent of semantics. However, full performance gains require semantic alignment—on ImageNet, shuffled features actually underperform vanilla (SaCo: 0.242 vs. 0.250), likely because the 1,000-class diversity makes random feature projections actively harmful without semantic grounding.

7.4.5 Qualitative Analysis of Attribution Dynamics

To complement the quantitative metrics, we conducted a qualitative inspection of the attribution dynamics. We isolated the top 100 images that showed improvement across a composite of all three faithfulness metrics (SaCo, Faithfulness Correlation, and Pixel

Flipping). For these improved samples, we identified the image patches exhibiting the largest magnitude of change ($|\Delta\text{Attribution}|$) between the baseline and the gated model, and extracted the specific SAE latent feature responsible for the primary contribution to this change.

While the interpretability of individual SAE features remains an open challenge in the literature, with many features exhibiting polysemanticity or abstract textural activations, our analysis of the top contributing features revealed several distinct patterns of behavior. By examining the prototypical activations (top-10 activating validation images) for these features, we can interpret the strategies the gating mechanism employs to refine model attention.

We categorize these observed behaviors into three primary strategies: the suppression of confounding concepts, the removal of data artifacts, and context-dependent modulation. Additionally, we analyze failure cases to understand the limitations of semantic steering.

7.4.5.1 Suppression of Confounding Concepts

A recurring pattern observed in the improved samples is the active suppression of high-salience concepts that often co-occur with the target class but do not act as causal features for the classification.

For example, **Feature L6-10968** and **Feature L9-45529** were identified as potent detectors of human faces, specifically activating on eyes and nose regions in the feature prototypes. In our case studies, these features consistently appeared with negative attribution contributions (suppressors) in classes such as *Academic Gown*, *Cloak*, or *Float-coated Retriever*.

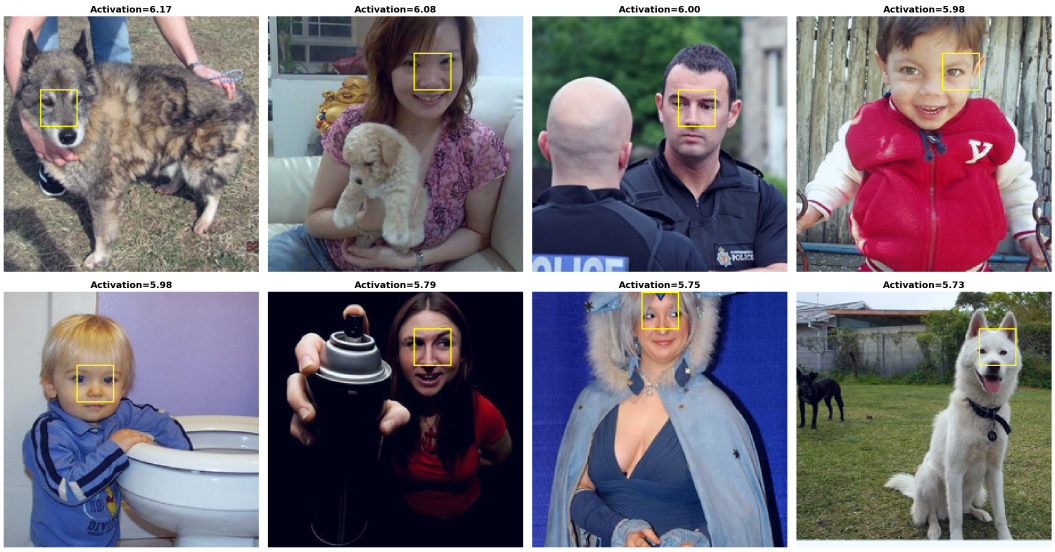
While faces are visually prominent in these images, they are confounders for the classification of clothing. The gating mechanism appears to utilize the gradient of these "Face Features" to down-weight attention on the person, thereby redirecting the model's focus toward the relevant object (the clothing). This behavior aligns with the concept of "negative semantic steering," where the model explicitly identifies and subtracts irrelevant semantic concepts. Interestingly, the feature does not discern between animal or human facial structures at layer 6, but we found a feature in the later layer 9 that is directly concerned with specifically human faces.

7.4.5.2 Mitigation of Data Artifacts

The analysis further suggests that the feature gradient mechanism aids in filtering out non-semantic data artifacts. We observed features that appear to function as detectors for specific biases in the ImageNet dataset.

Feature L6-37778, for instance, activates strongly on text overlays and watermarks. In the analyzed samples, this feature acted as a suppressor in patches containing copyright labels, preventing the model from utilizing metadata text as a prediction shortcut. Similarly, **Feature L9-6561** detects specific red textures. Its suppression in classes

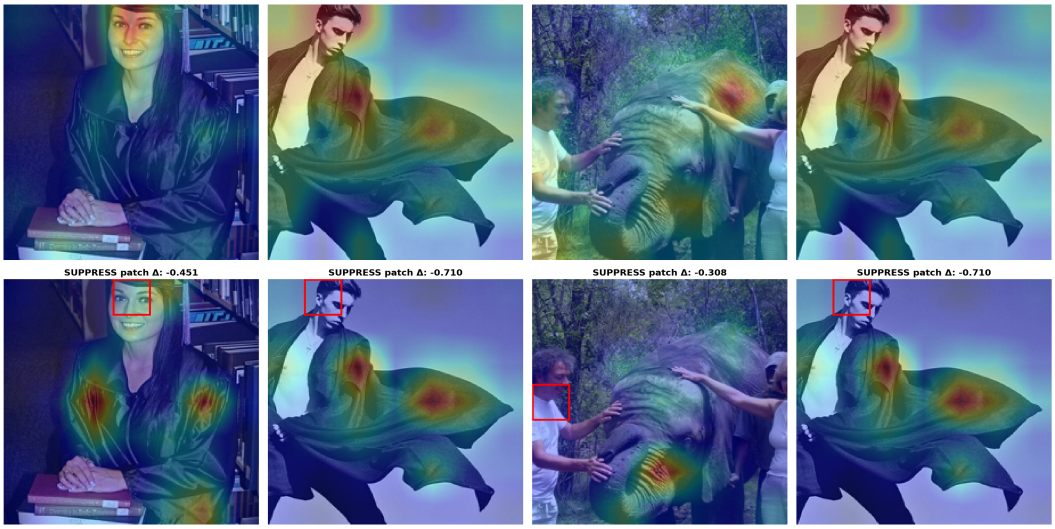
(a) Case Studies: Suppression of faces in clothing classes.



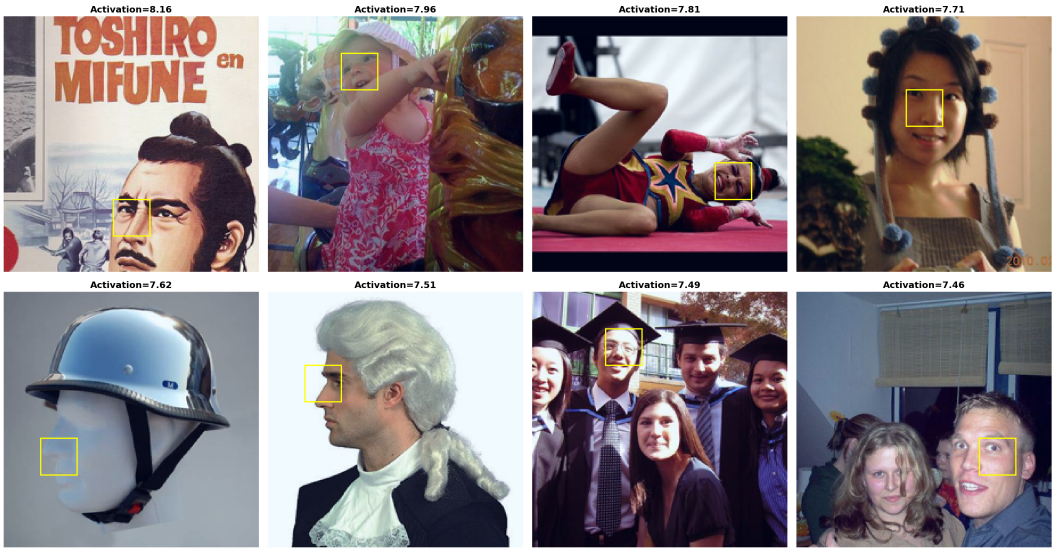
(b) Prototypes: Top activating images for Feature L6-10968 (Face Detector).

Figure 7.11: Visualization of Feature L6-10968. The feature detects facial landmarks (bottom) and is actively suppressed (red boxes, top) in classes where the person is secondary to the object (e.g., Academic Gown).

like *Hamper* or *Perfume, Essence* suggests the removal of background color biases (e.g., blanket) that correlate with but do not cause the object class. In classes like *Bell Pepper* or *Flamingo* it seems to de-emphasize regions that do not provide additional context for



(a) Case Studies: Semantic suppression in deeper layers.



(b) Prototypes: Feature L9-45529 (Face Context).

Figure 7.12: Analysis of Feature L9-45529. Similar to Layer 6, this deeper layer feature suppresses human presence to refine object classification.

7.4.5.3 Context-Dependent Feature Modulation

A more sophisticated behavior observed was the apparent ability of certain features to switch roles, acting as boosters or suppressors depending on the semantic context of the image.

Feature L6-20254 serves as a primary example. The prototypes indicate this feature

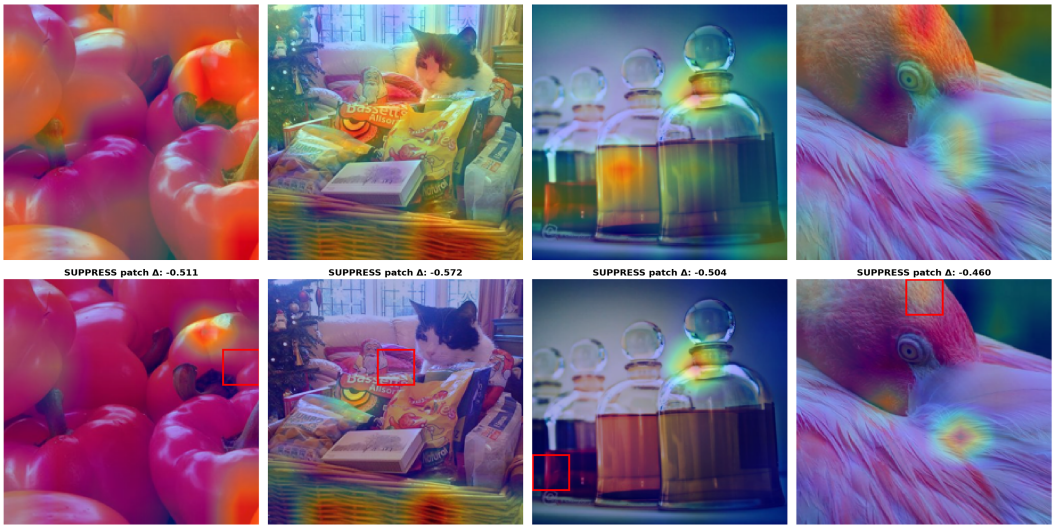
(a) Case Studies: Suppression of text artifacts.

(b) Prototypes: Feature L6-37778 (Text/Watermarks).

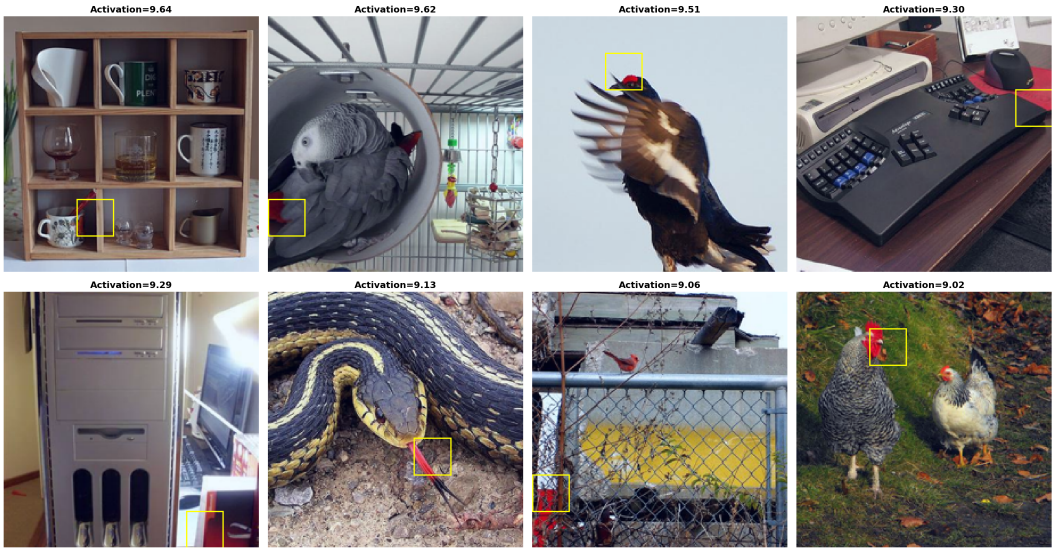
Figure 7.13: Artifact removal by Feature L6-37778. The feature identifies artificial text overlays (bottom) and suppresses them (top) to prevent 'Clever Hans' predictions based on watermarks.

detects animal fur textures. In images of *Schipperke* or *Border Collies*, where fur is a defining characteristic of the class, this feature acted as a booster (↑ attribution). Conversely, in images of *Pekinese*, the same feature acted as a suppressor (↓ attribution).

This modulation suggests that the gating mechanism is not merely acting as a static



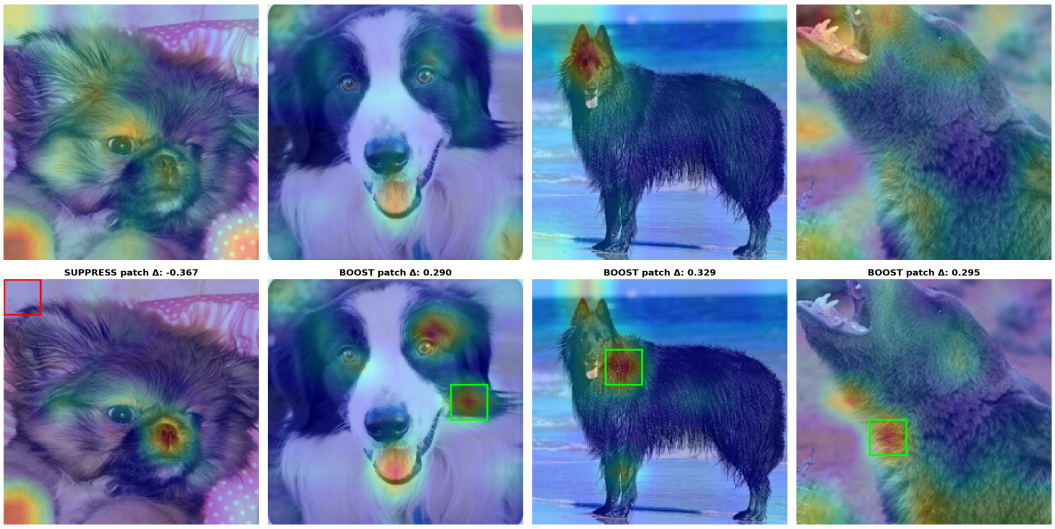
(a) Case Studies: Suppression of background color bias.



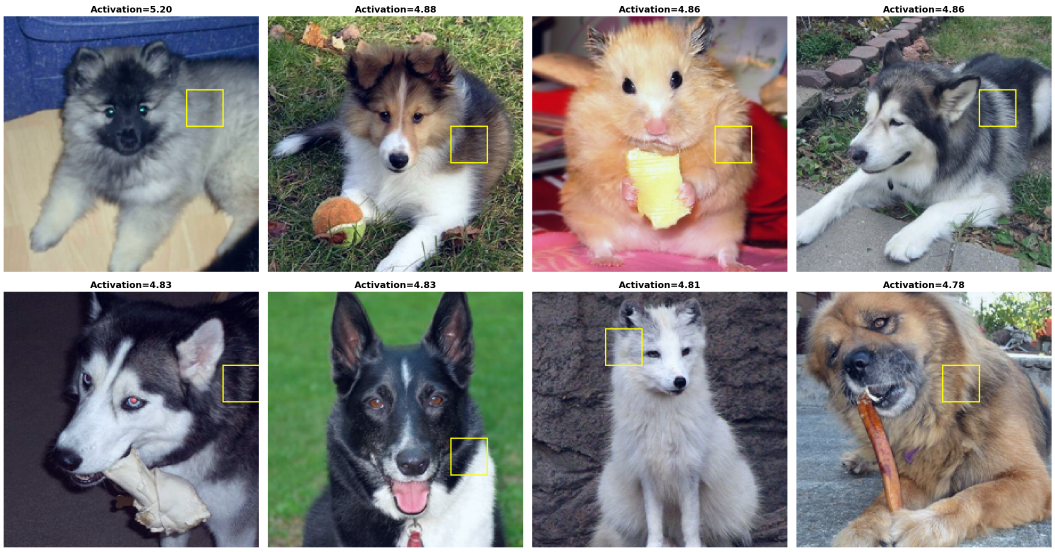
(b) Prototypes: Feature L9-6561 (Red Texture).

Figure 7.14: Bias mitigation by Feature L9-6561. The feature suppresses red background textures (e.g., velvet) often correlated with instruments like pianos.

filter. In the suppression cases, the model may be dampening the generic "fur" texture to resolve feature crowding, forcing the attribution mechanism to rely on more discriminative features (such as facial geometry) to distinguish between similar dog breeds.



(a) Case Studies: Context-switching behavior.



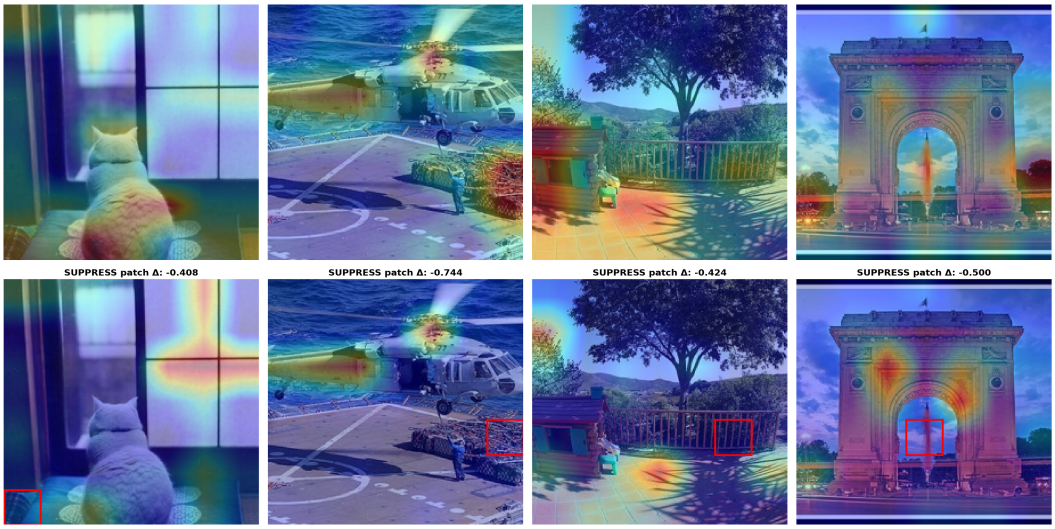
(b) Prototypes: Feature L6-20254 (Fur Texture).

Figure 7.15: Context-dependent behavior of Feature L6-20254. The feature boosts attribution (green boxes) when fur defines the object (e.g., Bearskin hat) but suppresses it (red boxes) when specific facial features are more discriminative.

7.4.5.4 Broadband Texture Filtering

Finally, we noted a class of features (e.g., **Feature L10-5436**) that appeared across a wide variety of unrelated classes. Prototypes for these features typically display high-frequency edges, blur, or generic gradients. Their ubiquity and consistent suppressive role

in our analysis suggest they function as "broadband" background filters. By suppressing these low-information patches, the gating mechanism effectively performs a late-stage denoising operation, cleaning the attribution map of visual clutter.



(a) Case Studies: Background removal.

(b) Prototypes: Feature L10-5436 (Broadband Noise).

Figure 7.16: Broadband filtering by Feature L10-5436. The feature detects generic background noise/blur (bottom) and suppresses it across diverse classes (top).

7.4.5.5 Limits of Semantic Steering: Failure Analysis

To understand the limitations of the proposed method, we analyzed samples where the gating mechanism degraded faithfulness metrics. We observed that the "suppression" strategy can act as a double-edged sword: the same features that improve performance in some contexts can cause degradation when applied over-aggressively.

Context Misfire (Feature L6-20254): A notable failure mode was observed with the previously discussed "Smart Switch" feature (L6-20254). As shown in Figure 7.15, this feature typically boosts fur attribution for dogs. However, in failure cases such as the *Giant Panda* (Figure 7.17), the gating mechanism incorrectly identifies the defining fur texture as "background noise" and suppresses it, effectively removing the class-defining signal.

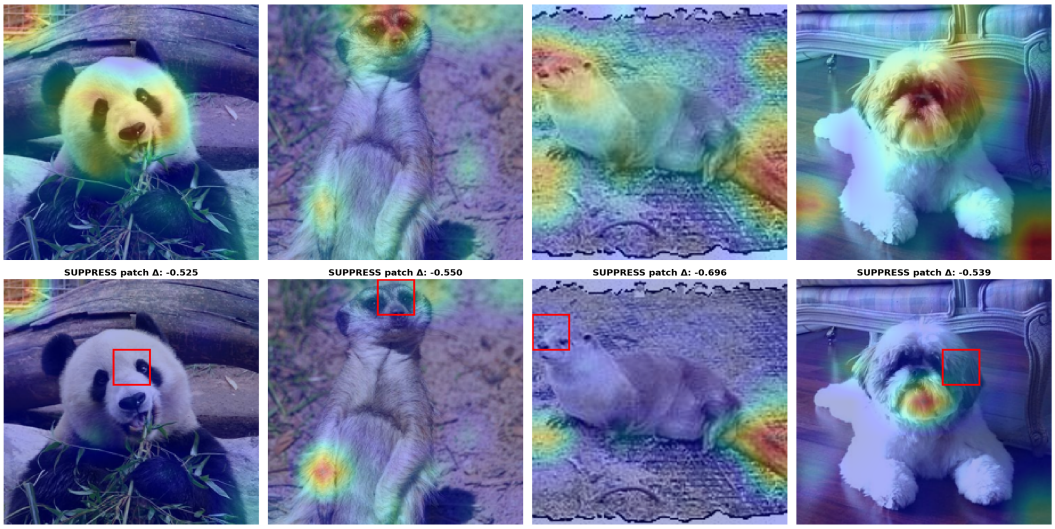
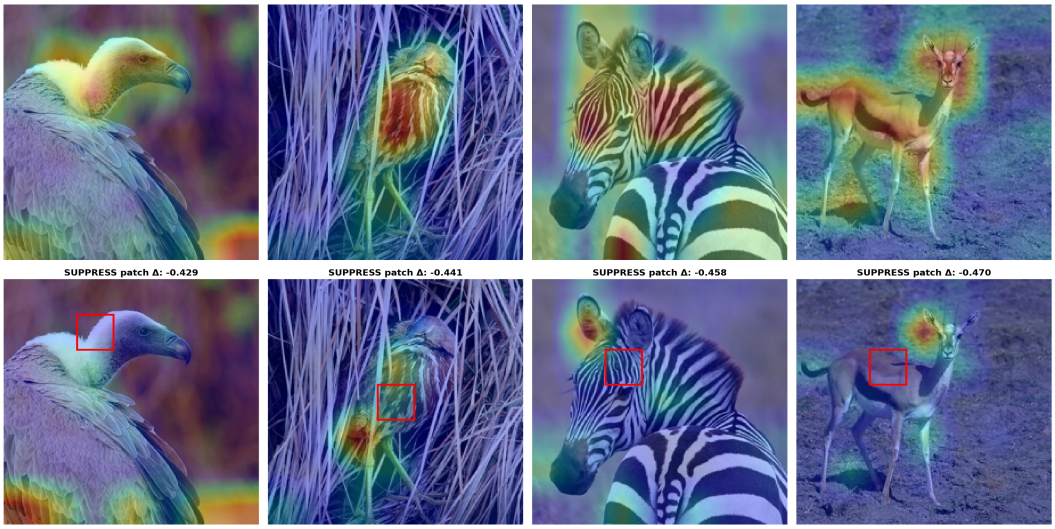


Figure 7.17: Context Misfire by Feature L6-20254. Unlike the success cases in Figure 7.15, here the feature suppresses the fur texture of a Giant Panda (red boxes), stripping away vital class information.

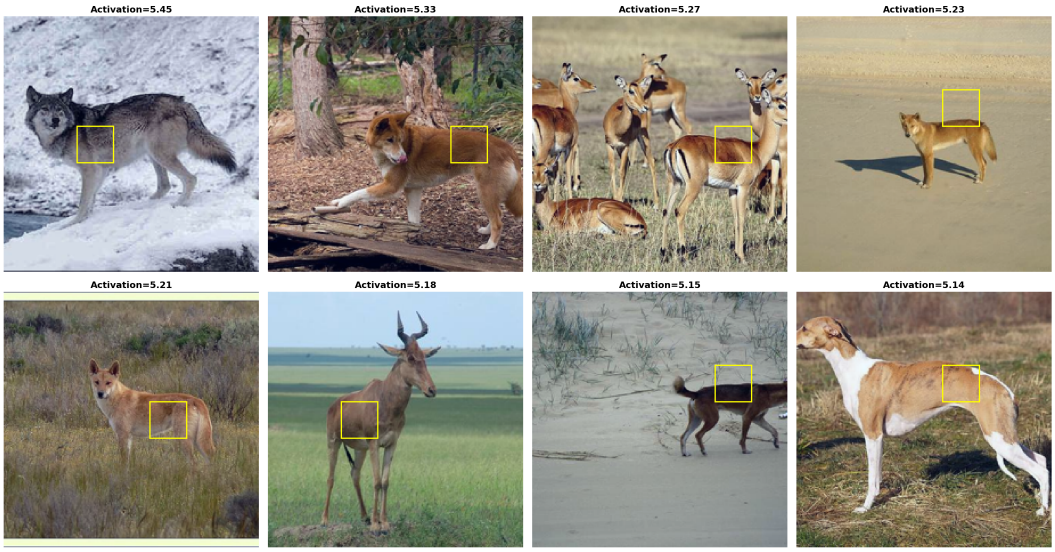
Over-Aggressive Filtering (Feature L6-8584): We also identified features that appear to function as "Over-Zealous Cleaners." **Feature L6-8584** was the most frequent cause of degradation in Layer 6. An analysis of its prototypes (Figure 7.18b) reveals it is a potent detector of "feathers" and "wildlife skin."

While this feature likely aids in separating birds from complex backgrounds in some contexts, it acts almost exclusively as a suppressor (96% of cases). In the failure cases analyzed (e.g., *Vulture*, *Bald Eagle*), this suppression was applied to the object itself, erasing the subject's texture and leaving the model with a feature-less silhouette (Figure 7.18a).

This chapter presented a comprehensive empirical evaluation of Feature-Gradient Attribution across three datasets, two model architectures, and multiple faithfulness metrics.



(a) Case Studies: Over-aggressive suppression of subject texture.



(b) Prototypes: Feature L6-8584 (Feathers/Wildlife).

Figure 7.18: Over-Aggressive Filtering by Feature L6-8584. The prototypes (bottom) confirm this feature detects wildlife textures/feathers. In failure cases (top), it acts as a suppressor on the object itself (e.g., Vulture), degrading the explanation.

Our central finding is that the principle underlying TransMM—combining attention patterns with gradient information to identify important patches—extends naturally to sparse autoencoder features, yielding substantial faithfulness improvements.

CHAPTER 8

Discussion and Conclusion

8.1 Extending TransMM's Principle to Feature Space

TransMM's success relies on a core principle: combining attention patterns (which patches receive how much attention) with gradients (how that attention affects predictions) to identify important regions. Our method tests whether this same principle extends to sparse autoencoder features, where activations indicate which semantic concepts are present and gradients indicate how those concepts influence classification.

Our ablation studies provide evidence that this extension holds. The combined method of using both feature activations and gradients consistently outperforms vanilla TransMM across all datasets and metrics (15.5-34.8% SaCo improvement, 12.0-22.0% faithfulness correlation gains). Critically, neither signal alone achieves comparable results.

The activation-only variant shows modest improvements over baseline (2-6% SaCo on most layers), suggesting that knowing which features are present provides some attribution signal. However, this approach lacks gradient information to distinguish features that merely exist in a patch from those that causally influence the prediction. The gradient-only variant, which projects gradients into feature space without weighting by activation strength, performs at or below baseline on most layers. On late layers, gradient-only substantially degrades faithfulness correlation (e.g., -18% at COVID-QU-Ex layer 9, -15% at Hyperkvasir layer 8), suggesting that gradient information without activation context can mislead attribution.

These results are consistent with the hypothesis that faithful attribution requires both components of the TransMM principle: knowing what is present (activations/attention) and knowing what matters (gradients). Just as raw attention maps without gradient weighting fail to distinguish causally important patches from merely attended ones, feature activations without gradient weighting fail to distinguish causally important features from merely present ones. Conversely, gradients without activation context—whether

projected through features or applied to attention maps—lack the grounding needed for reliable attribution.

The success of the combined method across diverse architectures (fine-tuned ViT-B/16, contrastively pre-trained CLIP ViT-B/32), domains (medical imaging, natural images), and complementary faithfulness metrics suggests that TransMM's core principle generalizes beyond attention mechanisms to sparse semantic representations learned by autoencoders.

8.2 Semantic Structure vs. Statistical Properties

A critical question is whether attribution improvements stem from the semantic correspondence between SAE features and visual concepts, or merely from statistical properties of sparse decomposition. Our shuffled decoder control addresses this by preserving all statistical properties like sparsity patterns, reconstruction quality and activation magnitudes while destroying the correspondence between feature dimensions and semantic concepts.

Real SAE features consistently outperform shuffled variants: 6.6% (COVID-QU-Ex), 12.8% (Hyperkvasir), and 39.3% (ImageNet) on SaCo. Interestingly, shuffled features sometimes exceed vanilla baseline (e.g., COVID-QU-Ex faithfulness correlation: 0.354 vs. 0.309), suggesting sparsity provides incidental benefits such as noise reduction. However, real features substantially outperform both shuffled and vanilla variants, providing strong evidence that improvements require semantic alignment, not just sparse decomposition. This interpretation is consistent with our ablation finding that the combined method dramatically outperforms gradient-only approaches that use feature space structure without semantic content.

8.3 Layer Selection and Feature Locality

Our empirical finding that intermediate layers consistently provide optimal attribution improvements, like layers 4-6 for medical datasets and 5-10 for ImageNet, may relate to feature locality properties documented in recent SAE research. Han et al. [HKK25] found that feature non-locality in CLIP-ViT-L/14 remains minimal through approximately the first third of the network, only becoming significant in later layers (up to 14% spatial misalignment at layer 22 of 24).

In our ViT-Base models with 11 layers, the optimal range (layers 4-7, approximately 36-64% network depth) corresponds to a potential balance point. Early layers (1-3) show minimal performance gains, suggesting their low-level features lack sufficient semantic content for classification-relevant attribution. Later layers (8-10) show more variable performance, potentially due to increasing feature non-locality as self-attention causes spatial mixing. The intermediate layers where we observe peak performance may thus

represent a region where features have developed semantic richness while maintaining spatial coherence useful for patch-level attribution.

The slightly broader optimal range for ImageNet (layers 5-10) compared to medical datasets (4-6) could reflect architectural differences (ViT-B/32 vs B/16) or effects of contrastive versus supervised training, though we cannot determine the specific cause without direct locality measurements in our models. While we cannot definitively establish that feature locality drives these layer selection patterns, the correspondence between our empirical optima and the expected "semantic-spatial balance" suggests this connection warrants investigation in future work.

8.3.1 Multi-Layer Performance: Reinforcement or Complementarity

Multi-layer configurations consistently outperform single-layer gating across all datasets, with improvements of 60-87% over the best single layers (e.g., COVID-QU-Ex: 16.1% vs 10.0% SaCo; Hyperkvasir: 30.9% vs 16.5%; ImageNet: 43.2% vs 37.3%). This substantial gain suggests that features from adjacent layers provide synergistic information beyond simple additive effects.

Two non-exclusive mechanisms could explain this finding. Features representing similar semantic concepts may activate across consecutive layers, with each layer's signal providing independent evidence that reinforces important patches through compounding corrections. Alternatively, adjacent layers may capture complementary aspects of earlier layers encoding textures, middle layers representing parts, and later layers capturing global semantics to finally form a more complete representation.

Our results provide some evidence for both mechanisms. The consistent advantage of adjacent layer combinations (e.g., [4,5,6]) over non-adjacent selections (e.g., [3,6,9]) suggests semantic continuity matters, supporting reinforcement. However, the substantial gains from adding third layers suggest each contributes partially independent information, consistent with complementarity. Most likely, both mechanisms contribute: patches containing consistent evidence across multiple semantic levels receive stronger amplification than those with evidence at only one level.

Regardless of the underlying mechanism, the success of multi-layer configurations demonstrates that SAE features across the network hierarchy contain valuable information for faithful attribution, suggesting that future methods should consider the full spectrum of learned representations rather than single-layer features.

8.4 Implications for Medical Imaging

Our motivation for this work stemmed from the need for trustworthy AI in medical imaging, where clinicians must understand not just what a model predicts but why. The faithfulness improvements demonstrated on medical datasets (COVID-QU-Ex, Hyperkvasir) suggest that SAE-enhanced attribution could help verify that models rely on clinically meaningful features rather than spurious correlations.

However, translating improved faithfulness metrics into clinical trust requires additional validation. Future work should investigate whether doctors find SAE-enhanced explanations more useful for detecting problematic model behavior, and whether the semantic interpretability of SAE features (e.g., "this patch activated features representing lung infiltrates") provides actionable insights beyond pixel-level heatmaps. The computational overhead of SAE training and multi-layer inference may also limit deployment in resource-constrained clinical settings, suggesting that single-layer configurations or pre-trained SAEs may be more practical for real-world adoption.

8.5 Limitations

Computational Overhead Our method requires training SAEs on model activations before attribution can be computed, adding a preprocessing step not required by vanilla TransMM. The computational cost of SAE training depends on dataset size and the number of layers for which SAEs are needed—larger datasets and more layers naturally require proportionally more compute. Once trained, multi-layer configurations require loading multiple SAEs simultaneously (e.g., three SAEs for layers [4,5,6]), which increases memory requirements compared to single-layer or vanilla approaches. Inference with multi-layer gating scales linearly with the number of layers used, as each layer requires SAE encoding and gate computation. For applications where real-time attribution is critical, these computational considerations may favor single-layer configurations or pre-trained SAEs when available.

Dependence on SAE Quality Our results rely on well-trained SAEs achieving >90% explained variance with minimal dead features. We have not systematically investigated how attribution quality degrades with poorly trained SAEs (e.g., high dead feature percentages, low reconstruction quality). The method's effectiveness likely depends on SAE training hyperparameters, though our successful use of pre-trained Prisma SAEs on ImageNet suggests some robustness to training variations.

Varied Magnitudes Across Metrics While our method improves all three faithfulness metrics across all datasets, the magnitude of improvement varies: SaCo and Faithfulness Correlation show substantial gains (12-44%), while Pixel Flipping shows more modest but consistent improvements (1-11%). This pattern holds across all three datasets, suggesting our method particularly enhances magnitude alignment and correlation aspects of faithfulness captured by SaCo and Faithfulness Correlation.

We hypothesize that this differential stems from how the metrics weight different regions of the attribution distribution. Pixel Flipping, by design, heavily prioritizes the highest-ranked patches, removing them first and measuring immediate impact. If TransMM already identifies the most critical regions reasonably well, there is limited room for improvement in this top-ranked subset. In contrast, both SaCo and Faithfulness Correlation evaluate the full attribution distribution more evenly: SaCo tests all pairwise comparisons

between patch groups, while Faithfulness Correlation samples random subsets across the entire range. Our method's primary contribution appears to be refining attributions in moderately-important regions, de-emphasizing patches that receive moderate attribution scores but are ultimately irrelevant while preserving or strengthening truly important regions. These refinements have substantial impact on metrics that evaluate the complete distribution (SaCo, Faithfulness Correlation) but smaller impact on metrics that focus predominantly on the top-ranked patches (Pixel Flipping). This interpretation is consistent with Pixel Flipping's documented sensitivity limitations [WKT+24b], particularly its reduced sensitivity to changes outside the highest-attribution regions.

8.6 Future Work

Understanding Where Improvements Occur The modest Pixel Flipping improvements despite substantial SaCo gains suggest our method may refine moderately important patches rather than changing the most obvious regions. Future work should analyze the distribution of attribution changes: do gates primarily affect high-attribution patches (refining already-identified regions) or moderate-attribution patches (discovering additional relevant regions)? Computing the distribution of gate values and correlating with attribution magnitudes could reveal whether the method provides contextual refinement or discovers new important regions.

Comparison with Complementary Methods TokenTM [WKT+24a] improves TransMM by incorporating MLP transformation information. Comparing our SAE-based feature gating with TokenTM's approach could reveal whether these methods capture complementary aspects of model computation, and whether they could be combined for further improvements.

Extension to Other Attribution Methods Our approach demonstrates that sparse semantic features can enhance gradient-based attribution. Future work should investigate whether similar principles apply to other attribution frameworks. Could SAE features improve GradCAM, Integrated Gradients, or attention rollout methods? This would test whether feature-gradient decomposition is a general principle or specific to TransMM's architecture.

Application to Other Vision Tasks The success of semantic feature gating for classification attribution suggests potential applications to other vision tasks requiring spatial explanations, such as object detection, semantic segmentation, or visual question answering. Whether the same principles extend to these tasks, where attribution targets are more complex than single class labels, remains an open question.

8.7 Conclusion

This thesis investigated whether incorporating semantic structure from Sparse Autoencoders could improve attribution faithfulness in Vision Transformers. Building on TransMM's principle that faithful attribution requires combining activation patterns with gradient information, we demonstrated that this principle extends naturally to learned sparse features: feature activations indicate which semantic concepts are present, while gradients indicate how those concepts influence predictions.

Through comprehensive evaluation across three datasets, two model architectures, and three complementary faithfulness metrics, we demonstrated that feature-gradient gating improves attribution quality across all metrics without exception. Real SAE features achieve 10.5-44.2% improvements on SaCo, 14.0-43.0% on Faithfulness Correlation, and 1.3-10.8% on Pixel Flipping compared to vanilla TransMM, with statistical significance achieved on all metrics for two datasets and on SaCo and Faithfulness Correlation for all three datasets. Ablation studies confirmed that both signals are necessary—neither activation-only nor gradient-only variants match the combined method's performance. The shuffled decoder control demonstrated that improvements require semantic alignment between features and visual concepts, not merely statistical properties of sparse decomposition.

Our results suggest that mechanistic interpretability tools, developed primarily for understanding model internals, can directly enhance practical explainability methods. The success of this approach across medical imaging and natural images, fine-tuned and contrastively pre-trained models, indicates broad applicability. This work provides evidence that attribution methods should consider leveraging learned semantic representations that respect the structure of model computations, opening new directions for faithful visual explanations in Vision Transformers.

List of Figures

List of Tables

Bibliography

[AB16] Guillaume Alain and Yoshua Bengio. Understanding intermediate layers using linear classifier probes. *arXiv preprint arXiv:1610.01644*, 2016.

[AGM+18] Julius Adebayo, Justin Gilmer, Michael Muelly, Ian Goodfellow, Moritz Hardt, and Been Kim. Sanity checks for saliency maps. *Advances in neural information processing systems*, 31, 2018.

[AHD+24] Reduan Achtibat, Sayed Mohammad Vakilzadeh Hatefi, Maximilian Dreyer, Aakriti Jain, Thomas Wiegand, Sebastian Lapuschkin, and Wojciech Samek. AttnLRP: Attention-aware layer-wise relevance propagation for transformers. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp, editors, *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 135–168. PMLR, 21–27 Jul 2024.

[Ano25] Anonymous. Steering CLIP's vision transformer with sparse autoencoders. In *Mechanistic Interpretability for Vision at CVPR 2025 (Non-proceedings Track)*, 2025.

[AZ20] Samira Abnar and Willem H. Zuidema. Quantifying attention flow in transformers. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel R. Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 4190–4197. Association for Computational Linguistics, 2020.

[BKH16] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

[BTB+23] Trenton Bricken, Adly Templeton, Joshua Batson, Brian Chen, Adam Jermyn, Tom Conerly, Nick Turner, Cem Anil, Carson Denison, Amanda Askell, Robert Lasenby, Yifan Wu, Shauna Kravec, Nicholas Schiefer, Tim Maxwell, Nicholas Joseph, Zac Hatfield-Dodds, Alex Tamkin, Karina Nguyen, Brayden McLean, Josiah E Burke, Tristan Hume, Shan Carter,

Tom Henighan, and Christopher Olah. Towards monosemanticity: Decomposing language models with dictionary learning. *Transformer Circuits Thread*, 2023.

[BTS+20] Hanna Borgli, Vajira Thambawita, Pia H Smedsrud, Steven Hicks, Debesh Jha, Sigrun L Eskeland, Kristin Ranheim Randel, Konstantin Pogorelov, Mathias Lux, Duc Tien Dang Nguyen, Dag Johansen, Carsten Griwodz, Håkon K Stensland, Enrique Garcia-Ceja, Peter T Schmidt, Hugo L Hammer, Michael A Riegler, Pål Halvorsen, and Thomas de Lange. HyperKvasir, a comprehensive multi-class image and video dataset for gastrointestinal endoscopy. *Scientific Data*, 7(1):283, 2020.

[BWM20] Umang Bhatt, Adrian Weller, and José M. F. Moura. Evaluating and aggregating feature-based model explanations. In Christian Bessiere, editor, *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020*, pages 3016–3022. ijcai.org, 2020.

[CGC+20] Nick Cammarata, Gabriel Goh, Shan Carter, Ludwig Schubert, Michael Petrov, and Chris Olah. Curve detectors. *Distill*, 2020.

[CGW21a] Hila Chefer, Shir Gur, and Lior Wolf. Generic attention-model explainability for interpreting bi-modal and encoder-decoder transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 397–406, October 2021.

[CGW21b] Hila Chefer, Shir Gur, and Lior Wolf. Transformer interpretability beyond attention visualization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 782–791, June 2021.

[CJJ+24] Noel C. F. Codella, Ying Jin, Shrey Jain, Yu Gu, Ho Hin Lee, Asma Ben Abacha, Alberto Santamaria-Pang, Will Guyman, Naiteek Sangani, Sheng Zhang, Hoifung Poon, Stephanie Hyland, Shruthi Bannur, Javier Alvarez-Valle, Xue Li, John Garrett, Alan McMillan, Gaurav Rajguru, Madhu Maddi, Nilesh Vijayrania, Rehaan Bhimai, Nick Mecklenburg, Rupal Jain, Daniel Holstein, Naveen Gaur, Vijay Aski, Jenq-Neng Hwang, Thomas Lin, Ivan Tarapov, Matthew Lungren, and Mu Wei. Medimageinsight: An open-source embedding model for general domain medical imaging, 2024.

[CMPL+23] 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.

[DBK+21] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer,

Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021.

[DCLT19] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[EHO+22] Nelson Elhage, Tristan Hume, Catherine Olsson, Nicholas Schiefer, Tom Henighan, Shauna Kravec, Zac Hatfield-Dodds, Robert Lasenby, Dawn Drain, Carol Chen, Roger Grosse, Sam McCandlish, Jared Kaplan, Dario Amodei, Martin Wattenberg, and Christopher Olah. Toy models of superposition. *Transformer Circuits Thread*, 2022. `https://transformer-circuits.pub/2022/toy_model/index.html`.

[ENO+21] Nelson Elhage, Neel Nanda, Catherine Olsson, Tom Henighan, Nicholas Joseph, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Nova DasSarma, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Andy Jones, Jackson Kernion, Liane Lovitt, Kamal Ndousse, Dario Amodei, Tom Brown, Jack Clark, Jared Kaplan, Sam McCandlish, and Chris Olah. A mathematical framework for transformer circuits. *Transformer Circuits Thread*, 2021.

[GDlTT+25] Leo Gao, Tom Dupre la Tour, Henk Tillman, Gabriel Goh, Rajan Troll, Alec Radford, Ilya Sutskever, Jan Leike, and Jeffrey Wu. Scaling and evaluating sparse autoencoders. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu, editors, *International Conference on Representation Learning*, volume 2025, pages 26721–26754, 2025.

[GIF+23] Samir Yitzhak Gadre, Gabriel Ilharco, Alex Fang, Jonathan Hayase, Georgios Smyrnis, Thao Nguyen, Ryan Marten, Mitchell Wortsman, Dhruba Ghosh, Jieyu Zhang, Eyal Orgad, Rahim Entezari, Giannis Daras, Sarah Pratt, Vivek Ramanujan, Yonatan Bitton, Kalyani Marathe, Stephen Mussmann, Richard Vencu, Mehdi Cherti, Ranjay Krishna, Pang Wei Koh, Olga Saukh, Alexander Ratner, Shuran Song, Hannaneh Hajishirzi, Ali Farhadi, Romain Beaumont, Sewoong Oh, Alex Dimakis, Jenia Jitsev, Yair Carmon, Vaishaal Shankar, and Ludwig Schmidt. Datacomp: In search of the next generation of multimodal datasets. *arXiv preprint arXiv:2304.14108*, 2023.

[HCS+24] Robert Huben, Hoagy Cunningham, Logan Riggs Smith, Aidan Ewart, and Lee Sharkey. Sparse autoencoders find highly interpretable features

in language models. In *The Twelfth International Conference on Learning Representations*, 2024.

[HG17] Dan Hendrycks and Kevin Gimpel. Bridging nonlinearities and stochastic regularizers with gaussian error linear units, 2017.

[HKK25] Sangyu Han, Yearim Kim, and Nojun Kwak. Causal interpretation of sparse autoencoder features in vision, 2025.

[HZRS16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 770–778. IEEE Computer Society, 2016.

[JEP+21] John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, et al. Highly accurate protein structure prediction with alphafold. *nature*, 596(7873):583–589, 2021.

[JG20] Alon Jacovi and Yoav Goldberg. Towards faithfully interpretable NLP systems: How should we define and evaluate faithfulness? In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4198–4205, Online, July 2020. Association for Computational Linguistics.

[JSH+25] Sonia Joseph, Praneet Suresh, Lorenz Hufe, Edward Stevinson, Robert Graham, Yash Vadi, Danilo Bzdok, Sebastian Lapuschkin, Lee Sharkey, and Blake Aaron Richards. Prisma: An open source toolkit for mechanistic interpretability in vision and video, 2025.

[JW19] Sarthak Jain and Byron C. Wallace. Attention is not Explanation. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3543–3556, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[KB15] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

[KBB23] Piotr Komorowski, Hubert Baniecki, and Przemyslaw Biecek. Towards evaluating explanations of vision transformers for medical imaging. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 3726–3732, June 2023.

[KK01] John F. Kolen and Stefan C. Kremer. *Gradient Flow in Recurrent Nets: The Difficulty of Learning LongTerm Dependencies*, pages 237–243. Wiley-IEEE Press, 2001.

[KME+24] Maxime Kayser, Bayar Menzat, Cornelius Emde, Bogdan Bercean, Bartlomiej W. Papiez, Alex Novak, Abdala Espinosa, Susanne Gaube, Thomas Lukasiewicz, and Oana-Maria Camburu. Fool me once? contrasting vision- and language-based explanations in a clinical decision-support setting. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, EMNLP 2024, Miami, Florida, November 12-16, 2024*, November 2024.

[LBBH98] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[LBM+15] Sebastian Lapuschkin, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PLoS ONE*, 10:e0130140, 07 2015.

[LCCS25] Hyesu Lim, Jinho Choi, Jaegul Choo, and Steffen Schneider. Sparse autoencoders reveal selective remapping of visual concepts during adaptation. In *The Thirteenth International Conference on Learning Representations*, 2025.

[MBSP21] A. K. Mondal, A. Bhattacharjee, P. Singla, and A. P. Prathosh. xViTCOS: Explainable vision transformer based COVID-19 screening using radiography. *IEEE Journal of Translational Engineering in Health and Medicine*, 10:1100110, 2021. PMID: 34956741; PMCID: PMC8691725.

[MCCD13] Tomás Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. In Yoshua Bengio and Yann LeCun, editors, *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings*, 2013.

[MLB+17] Grégoire Montavon, Sebastian Lapuschkin, Alexander Binder, Wojciech Samek, and Klaus-Robert Müller. Explaining nonlinear classification decisions with deep taylor decomposition. *Pattern recognition*, 65:211–222, 2017.

[MLN25] Aleksandar Makelov, Georg Lange, and Neel Nanda. Towards principled evaluations of sparse autoencoders for interpretability and control. In *The Thirteenth International Conference on Learning Representations*, 2025.

[OCS+20] Chris Olah, Nick Cammarata, Ludwig Schubert, Gabriel Goh, Michael Petrov, and Shan Carter. Zoom in: An introduction to circuits. *Distill*, 2020.

[OMS17] Chris Olah, Alexander Mordvintsev, and Ludwig Schubert. Feature visualization. *Distill*, 2017.

[PGD+20] Danish Pruthi, Mansi Gupta, Bhuwan Dhingra, Graham Neubig, and Zachary C. Lipton. Learning to deceive with attention-based explanations. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4782–4793, Online, July 2020. Association for Computational Linguistics.

[RDS+15] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015.

[RKH+21] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In Marina Meila and Tong Zhang, editors, *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*, pages 8748–8763. PMLR, 2021.

[RNSS18] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training, 2018.

[RUK+21] Maithra Raghu, Thomas Unterthiner, Simon Kornblith, Chiyuan Zhang, and Alexey Dosovitskiy. Do vision transformers see like convolutional neural networks? *Advances in neural information processing systems*, 34:12116–12128, 2021.

[SBB22] Lee Sharkey, Dan Braun, and Beren. Interim research report: Taking features out of superposition with sparse autoencoders. Alignment Forum, 12 2022.

[SBM+16] Wojciech Samek, Alexander Binder, Grégoire Montavon, Sebastian Lapuschkin, and Klaus-Robert Müller. Evaluating the visualization of what a deep neural network has learned. *IEEE transactions on neural networks and learning systems*, 28(11):2660–2673, 2016.

[SCB+25] Lee Sharkey, Bilal Chughtai, Joshua Batson, Jack Lindsey, Jeffrey Wu, Lucius Bushnaq, Nicholas Goldowsky-Dill, Stefan Heimersheim, Alejandro Ortega, Joseph Isaac Bloom, Stella Biderman, Adrià Garriga-Alonso, Arthur Conmy, Neel Nanda, Jessica Mary Rumbelow, Martin Wattenberg, Nandi Schoots, Joseph Miller, William Saunders, Eric J Michaud, Stephen Casper, Max Tegmark, David Bau, Eric Todd, Atticus Geiger, Mor Geva, Jesse Hoogland, Daniel Murfet, and Thomas McGrath. Open problems in mechanistic interpretability. *Transactions on Machine Learning Research*, 2025. Survey Certification.

[SCBWS25] Samuel Stevens, Wei-Lun Chao, Tanya Berger-Wolf, and Yu Su. Sparse autoencoders for scientifically rigorous interpretation of vision models, 2025.

[SCD+17] Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 618–626, 2017.

[SM24] Edward Sanderson and Bogdan J Matuszewski. A study on self-supervised pretraining for vision problems in gastrointestinal endoscopy. *IEEE Access*, 12:46181–46201, 2024.

[TCK+21] Anas M. Tahir, Muhammad E.H. Chowdhury, Amith Khandakar, Tawsifur Rahman, Yazan Qiblawey, Uzair Khurshid, Serkan Kiranyaz, Nabil Ibtehaz, M. Sohel Rahman, Somaya Al-Maadeed, Sakib Mahmud, Maymouna Ezeddin, Khaled Hameed, and Tahir Hamid. Covid-19 infection localization and severity grading from chest x-ray images. *Computers in Biology and Medicine*, 139:105002, 2021.

[TCM+24] Adly Templeton, Tom Conerly, Jonathan Marcus, Jack Lindsey, Trenton Bricken, Brian Chen, Adam Pearce, Craig Citro, Emmanuel Ameisen, Andy Jones, Hoagy Cunningham, Nicholas L Turner, Callum McDougall, Monte MacDiarmid, C. Daniel Freeman, Theodore R. Sumers, Edward Rees, Joshua Batson, Adam Jermyn, Shan Carter, Chris Olah, and Tom Henighan. Scaling monosemanticity: Extracting interpretable features from claude 3 sonnet. *Transformer Circuits Thread*, 2024.

[VSP+17] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008, 2017.

[WKT+24a] Junyi Wu, Weitai Kang, Hao Tang, Yuan Hong, and Yan Yan. On the faithfulness of vision transformer explanations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10936–10945, June 2024.

[WKT+24b] Junyi Wu, Weitai Kang, Hao Tang, Yuan Hong, and Yan Yan. On the faithfulness of vision transformer explanations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10936–10945, 2024.

[WVC+23] Kevin Ro Wang, Alexandre Variengien, Arthur Conmy, Buck Shlegeris, and Jacob Steinhardt. Interpretability in the wild: a circuit for indirect object identification in GPT-2 small. In *The Eleventh International Conference on Learning Representations*, 2023.

[WW22] Yipei Wang and Xiaoqian Wang. A unified study of machine learning explanation evaluation metrics. *arXiv preprint arXiv:2203.14265*, 2022.

[YCN+15] Jason Yosinski, Jeff Clune, Anh Mai Nguyen, Thomas J. Fuchs, and Hod Lipson. Understanding neural networks through deep visualization. *CoRR*, abs/1506.06579, 2015.

[ZF14] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pages 818–833. Springer, 2014.