LeGrad: An Explainability Method for Vision Transformers via Feature Formation Sensitivity

Supplementary Material

6. Overview

In this supplementary material, we first provide an extended theoretical discussion in section 7 that grounds LeGrad's design choices and motivates our gradient-based explainability approach. Section 8 details the implementation aspects, including pre-trained weights, baseline methodologies, and necessary adaptations for applying explainability methods to Vision Transformers (ViT) with different architectures. An expanded evaluation of LeGrad's performance across various models and datasets is documented in section 9. Section 9.3 offers a visual representation of the perturbation benchmarks utilized to assess the efficacy of various explainability approaches. Section 10 further explores the gradient distribution across layers for different models and datasets. Furthermore, section 11 conducts a sanity check using the FunnyBirds Co-12 setup to evaluate the robustness of our explainability method. Lastly, section 12 includes a disclaimer on the use of personal and human subject data within our research.

7. Extended Theoretical Discussion

This appendix provides a comprehensive theoretical analysis that grounds LeGrad's design choices and motivates our gradient-based explainability approach. We include a detailed derivation of the gradient signals, an analysis of register tokens, a discussion of intermediate representations, a formal treatment of layer-wise integration, a comparison with Layer-wise Relevance Propagation (LRP), and additional supporting proofs and empirical validations.

7.1. Mathematical Derivation of Gradient Signals

Let s denote the activation corresponding to a particular class (or prompt) computed from the decision module of the Vision Transformer, and let A be the attention map derived at a given layer. Considering a small perturbation δA in the attention map, a first-order Taylor expansion yields:

$$s(A + \delta A) \approx s(A) + \langle \frac{\partial s}{\partial A}, \delta A \rangle. \tag{8}$$

Here, the gradient $\frac{\partial s}{\partial A}$ quantifies the local sensitivity of the score s with respect to changes in A. This first-order approximation supports the claim that the magnitude of $\frac{\partial s}{\partial A}$ serves as a proxy for the influence of each component of the attention map on the final decision. In our method, gradients are computed via automatic differentiation during back-propagation, providing both a mathematically rigorous and computationally efficient estimate of feature importance. Positive values in $\frac{\partial s}{\partial A}$ indicate that an increase in the corresponding attention weight would lead to a stronger activation of the target class, making them natural candidates for attribution.

7.2. In-Depth Analysis of Register Tokens

Vision Transformer architectures routinely employ tokens that function as "registers" to stabilize internal computations. Although these tokens frequently acquire high raw attention weights, our analysis shows that their corresponding gradients, $\frac{\partial s}{\partial A}$, remain negligible. Theoretically, this behavior can be attributed to the role of register tokens in maintaining consistent internal representations rather than directly affecting the semantic decision. Formally, if A_{reg} represents the subset of attention weights corresponding to register tokens, then under typical conditions:

$$\| \frac{\partial s}{\partial A_{\text{reg}}} s(A) \| \approx 0. \tag{9}$$

Empirical observations confirm that, in contrast to semantically informative patch tokens, register tokens exhibit near-zero positive gradients. This insight justifies our choice to discard negative gradient values using a ReLU operation and to rely on the positive gradient signal for filtering out spurious contributions.

7.3. Extended Discussion on Intermediate Representations

Transformers produce hierarchical representations by progressively refining the input through multiple layers. Instead of solely considering the final output, LeGrad computes "proto-decision" signals s^l at each layer l. Early layers capture coarse, low-level features, while later layers encode semantically rich information. Let Z^l represent the intermediate tokens at layer l, and define $s^l = f(Z^l)$ as the activation computed from these tokens. By evaluating the gradient $\nabla_{A^l} s^l$ at each layer, we obtain layer-specific insights that reflect the evolution of feature importance along the network's depth. This decoupling permits the isolation of contributions from each layer before they are affected by downstream transformations, thereby mitigating the risk of confounding effects that might occur if attribution were computed solely at the final output.

7.4. Comprehensive Layer-Wise Integration

The aggregation of sensitivity measures across layers is designed to capture the cumulative influence of each token over the entire transformation pathway from input to prediction. Let L denote the total number of layers, and consider the per-layer gradient $\nabla_{A^l} s^l$ at each stage l. We define the aggregated importance as:

$$E_{\text{agg}} = \frac{1}{L} \sum_{l=1}^{L} \nabla_{A^l} s^l. \tag{10}$$

This summation is interpreted as a discrete approximation to integrating the continuous evolution of sensitivity over the model's computational path. Such an integration offers two key advantages. First, it preserves the contributions from tokens that may exert influence at early layers yet become attenuated in later stages. Second, by averaging local sensitivities across all layers, it provides a robust, holistic measure of token importance. Sensitivity analysis confirms that this approach yields heatmaps that accurately reflect the internal decision dynamics of the network.

7.5. Detailed Comparison with Layer-wise Relevance Propagation

Layer-wise Relevance Propagation (LRP) attributes a model's output to its input features via a recursive backward pass that conserves total relevance. Our gradient-based approach shares several theoretical tenets with LRP:

- **Computational efficiency:** A single backward pass computes the required gradients without the need for iterative, layer-specific propagation rules.
- **Adaptability:** The gradient approach naturally extends to transformer architectures, where attention mechanisms are the primary vehicles for information flow.
- **Theoretical consistency:** By focusing on first-order sensitivities, our method aligns with the core principles of LRP while avoiding some of its implementation complexities.

While LRP employs bespoke propagation rules for each layer type, our method leverages automatic differentiation to compute $\nabla_A s$ directly. Moreover, the conservation of relevance inherent in LRP is implicitly addressed when aggregating contributions from all layers, mirroring LRP's goal of complete attribution through a more straightforward computational pipeline.

7.6. Additional Proofs and Empirical Validations

To further substantiate our theoretical insights, we present additional proofs and empirical validations. First, under mild regularity conditions, one can formally prove that $\nabla_A s$ serves as an unbiased estimator for the local sensitivity in the vicinity of the given attention map. Second, empirical analyses on benchmark datasets demonstrate that the proposed gradient-based heatmaps correlate strongly with human-interpretable regions, as measured by quantitative metrics such as localization accuracy and correlation with ground-truth segmentation masks. Finally, ablation studies reveal that the exclusion of register tokens and the aggregation of gradients from intermediate representations consistently enhance the fidelity of the explanations.

7.7. Conclusion

The analysis presented in this appendix reinforces the design choices underlying LeGrad. By basing our approach on a first-order sensitivity analysis, addressing the role of register tokens, leveraging intermediate representations, and integrating contributions across layers, we propose a robust and computationally efficient framework for transformer explainability. The connection to established methods like LRP further situates our approach within the broader context of deep learning interpretability research—ensuring both theoretical rigor and practical utility.

This extended discussion complements the brief theoretical overview provided in the main paper and underpins the empirical and practical advances embodied by LeGrad.

8. Implementation details

8.1. LeGrad for Attentional Pooler

In place of the more conventional use of the [CLS] token, some ViT uses an attentional pooler. Attention Pooling, as e.g. used in SigLIP [42] employs an multi-head attention layer [25, 40] with a learnable query token $q_{pool} \in \mathbb{R}^d$. This token interacts with the final layer patch tokens to produce the pooled representation $\bar{z}_{AttnPool}$:

$$\bar{z}_{AttnPool} = \text{softmax}\left(\frac{q_{pool} \cdot (W_K Z^L)^T}{\sqrt{d}}\right)(W_V Z^L), \tag{11}$$

where $W_K, W_V \in \mathbb{R}^{d \times d}$ are learnable projection matrices.

LeGrad can easily be adapted to such models. Indeed, for ViTs using an attentional pooler (*e.g.* SigLIP [42]), a slight modification is made to compute the activation s^l at each layer. We apply the attentional pooler module $Attn_{pool}$ to each intermediate representation Z^l to obtain a pooled query $q^l \in \mathbb{R}^d$. The activation s^l with respect to the desired class c is then computed as $s^l = q^l \cdot \mathcal{C}_{:,c} \in \mathbb{R}$. Instead of considering the self-attention map, we use the attention map of the attentional pooler, denoted $\mathbf{A}_{pool} \in \mathbb{R}^{h \times 1 \times n}$. Thus, for every layer l, $\nabla A^l = \frac{\partial s^l}{\partial A^l_{pool}}$.

8.2. Pretrained weights

The experiments conducted in our study leverage a suite of models with varying capacities, including ViT-B/16, ViT-L/14, ViT-H/14, and ViT-bigG/14. These models are initialized with pretrained weights from the OpenCLIP library

respectively identified by: `"laion2b_s34b_b88k"`, `"laion2b_s32b_b82k"`, `"laion2b_s32b_b79k"`, and `"laion2b_s39b_b160k"`. For the SigLIP method, we utilize the ViT-B/16 model equipped with the `"webl"` weights. For the "gradient distribution over layers" graphs, Figure 6, we also used the pretrained weights from OpenAI [30] and MetaCLIP [39].

8.3. Detailed Description of Baselines

In this section, we provide an overview of the baseline methods against which our proposed approach is compared.

GradCAM: While originally designed for convolutional neural networks (CNNs), GradCAM can be adapted for Vision Transformers (ViTs) by treating the tokens as activations. To compute the GradCAM explainability map for a given activation s, we calculate the gradient of s with respect to the token dimensions. The gradients are aggregated across all tokens and serve as weights to quantify the contribution of each token dimension. Formally, for intermediate tokens $Z^l = \{z_0^l, z_1^l, \dots, z_n^l\} \in \mathbb{R}^{(n+1)\times d}$, the GradCAM map $E_{GradCAM}$ is defined as:

$$\begin{aligned} w &= \frac{1}{n}\sum_{i=0}^{n}\frac{\partial s}{\partial z_i^l} \quad \in \mathbb{R}^{d\times 1\times 1} \\ \hat{E}_{GradCAM} &= \left(\frac{1}{d}\sum_{k=1}^{d} w_d * Z^l_{1:,d}\right)^+ \quad \in \mathbb{R}^{n} \\ E_{GradCAM} &= \text{norm}(\text{resize}(\hat{E}_{GradCAM})) \quad \in \mathbb{R}^{W\times H}, \end{aligned} \tag{12}$$

with d representing the token dimension, $*$ denoting element-wise multiplication, and the superscript $+$ the ReLU operation. We empirically determined that applying GradCAM to layer 8 of ViT-B/16 yields optimal results.

AttentionCAM: This method extends the principles of GradCAM to ViTs by utilizing the attention mechanism within the transformer's architecture. AttentionCAM leverages the gradient signal to weight the attention maps in the self-attention layers. Specifically, for the last block's self-attention maps $\mathbf{A}^L$, the AttentionCAM map $E_{AttnCAM}$ is computed as:

$$\begin{aligned} \nabla\mathbf{A}^L &= \frac{\partial s}{\partial \mathbf{A}^L} \in \mathbb{R}^{h\times(n+1)\times(n+1)} \\ w &= \frac{1}{n^2}\sum_{i,j}\nabla\mathbf{A}^L_{:,i,j} \in \mathbb{R}^{h} \\ \hat{E}_{AttnCAM} &= \sum_{p}^{h}\left(w_p * A^L_{p,:,:}\right) \in \mathbb{R}^{(n+1)\times(n+1)} \\ E_{AttnCAM} &= \text{norm}(\text{resize}(\hat{E}_{AttnCAM})_{0,1:}) \end{aligned}, \tag{13}$$

where h denotes the number of heads in the self-attention mechanism.

Raw Attention: This baseline considers the attention maps from the last layer, focusing on the weights associated with the [CLS] token. The attention heads are averaged and the resulting explainability map is normalized. The Raw Attention map E_{Attn} is formalized as:

$$\begin{aligned} \hat{E}_{Attn} &= \mathbf{A}^L_{:,0,1:} \in \mathbb{R}^{h\times 1\times n} \\ E_{Attn} &= \text{norm}(\text{resize}(\frac{1}{h}\sum_{k=1}^{h}(\hat{E}_{Attn})_k)) \quad \in \mathbb{R}^{W\times H} \end{aligned} \tag{14}$$

These baselines provide a comprehensive set of comparative measures to evaluate the efficacy of our proposed method in the context of explainability for ViTs.

8.4. Details on adapting GradCAM to ViT

As GradCAM was designed for CNNs without [CLS] tokens, we tried both alternatives, *i.e.* including/excluding [CLS] token, Tab. 6 showcases a comparison of including/excluding the [CLS] token in the gradient computation on ViT-B. We observe that including the [CLS] token result in a marginal improvement. Overall, we would consider both options valid. GradCAM was originally designed for CNNs, therefore needs some adaptation to work for ViT. In an effort to consolidate the baselines used in this paper, we tried different configurations. One of which is whether or not include the [CLS] token in the gradient computation or not. We note that both alternatives are aligned with the original design of the GradCAM and that this choice is a matter of implementation.

We found that including the [CLS] token was producing better numbers, we therefore used that choice. Indeed, Table 6 shows the results on all the benchmark for GradCAM w/ and w/o the [CLS] token included in the gradient computation and shows that not including it translate in a slight decrease. Moreover, including the [CLS] token in the gradient computation is the option that makes more sense, as it is the [CLS] that is used to compute the similarity with the text query. We also tried to use the layer aggregation used in LeGrad for GradCAM and provide an evaluation in Table 6. We apply the same layer aggregation as in LeGrad (see 7) showing a slight improvement on the V7 benchmark. We attribute the only modest improvement to the fact that layers in ViT have different activation value ranges. LeGrad uses only gradients to produce layer-wise explainability maps, thereby avoiding this issue.

8.5. Adaptation of Baseline Methods to Attentional Pooler

In the main manuscript, we introduced our novel method, LeGrad, and its application to Vision Transformers (ViTs) with attentional poolers. Here, we provide supplementary

	mIoU(V7)	Neg ↑(INet)	Pos ↓(INet)
GradCAM w/ [CLS]	8.72	45.26	22.86
GradCAM w/o [CLS]	8.18	41.29	24.20
Multi-layer w/ [CLS]	9.51	45.31	22.54
LeGrad	*48.38*	*52.27*	*13.97*

Table 6. Evaluation of GradCAM with and without [CLS] token as well as with layer aggregation as proposed in eqaution 7 on ViT-B/16 for open-vocabulary detection(V7) and perturbation (ImageNet).

details on how LeGrad and other baseline methods were adapted ViTs employing attentional poolers:

CheferCAM: Following the original paper [10] that introduces CheferCAM, we considered the attentional pooler as an instance of a *"decoder transformer"* and applied the relevancy update rules described in equation (10) of that paper [10], (following the example of DETR). Since the attentional pooler has no skip connection we adapted the relevancy update rule not to consider the skip connection.

AttentionCAM: For AttentionCAM, instead of using the attention maps of the last layer, we use the attention maps of the attentional pooler. We found this variant to work better.

Raw Attention: Similarly, the Raw Attention baseline was adjusted by substituting the attention maps from the last self-attention layer with those from the attentional pooler.

Other Baselines: For the remaining baseline methods, no alterations were necessary. These methods were inherently compatible with the attentional pooler, and thus could be applied directly without any further adaptation.

The adaptations described above ensure that each baseline method is appropriately tailored to the ViTs with attentional poolers, allowing for a fair comparison with our proposed LeGrad method.

8.6. Mitigation of sensitivity to irrelevant regions:

We observe that for all evaluated explainability methods, SigLIP displays high activations in image regions corresponding to the background. These activations appeared to be invariant to the input, regardless of the gradient computation's basis, these regions were consistently highlighted. To address this issue, we computed the explainability map for a non-informative prompt, specifically `"a photo of"`. We then leveraged this map to suppress the irrelevant activations.

Namely, for an activation s under examination, we nullify any location where the activation exceeds a predefined threshold (set at 0.8) in the map generated for the dummy prompt. Formally, let E^s denote the explainability map for activation s, and E^{empty} represent the map for the prompt `"a photo of"`. The correction procedure is defined as follows:

$$E^s_{E^{empty} > th} = 0, \tag{15}$$

where $th = 0.8$. This method effectively addresses the issue without resorting to external data on the image content.

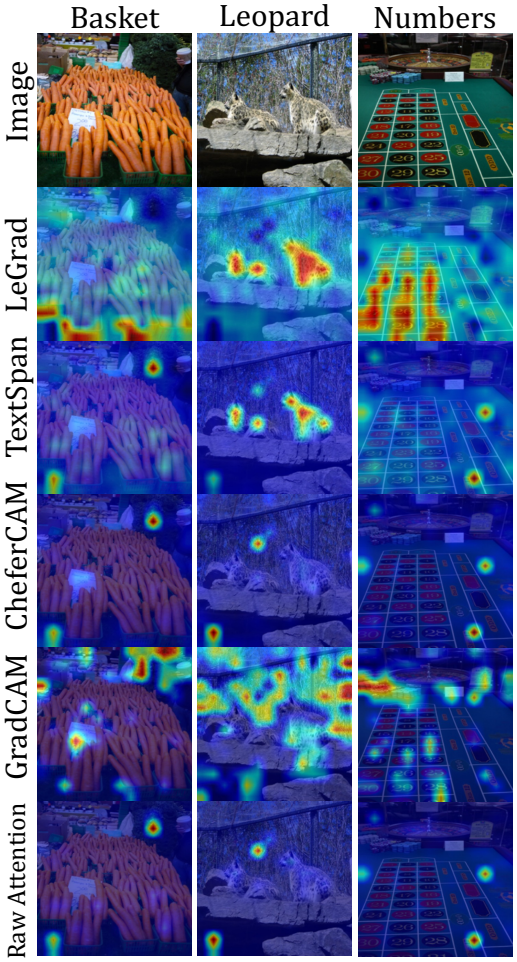

Figure 7. **SOTA Qualitative Comparison:** visual comparison of different explainability methods on images from OpenImagesV7.

8.7. Qualitative Comparison to SOTA

Here, we present a qualitative analysis of the explainability maps generated by LeGrad in comparison to other SOTA methods. The results are depicted in Figures 7 & 11, which includes a diverse set of explainability approaches such as gradient-based methods (e.g., CheferCAM, GradCAM), attention-based methods (e.g., Raw Attention weights visualization, Rollout), and methods that integrate intermediate visual representations with text prompts (e.g., TextSpan). Our observations indicate that raw attention visualizations

Method	ImageNet Negative Predicted ↑	ImageNet Negative Target ↑	ImageNet Positive Predicted ↓	ImageNet Positive Target ↓	OV7 p-mIoU ↑
rollout	47.81	47.81	25.74	25.74	0.07
Raw attention	44.42	44.42	25.85	25.85	0.09
GradCAM	41.25	44.42	35.10	33.50	6.97
AttentionCAM	45.62	45.71	45.01	44.92	0.19
CheferCAM	47.12	49.13	22.35	21.15	1.94
LeGrad	**50.08**	**51.67**	**18.48**	**17.55**	**25.40**

Table 7. **SOTA comparison on SigLIP-B/16:** Comparison of explainability methods on perturbation-based tasks on ImageNet-val and open-vocabulary localization on OpenImagesV7 (OV7).

ReLU	All Layers	L/14 Negative↑	L/14 Positive↓	H/14 Negative↑	H/14 Positive↓
✗	✗	47.81	20.80	57.57	21.73
✓	✗	49.32	19.95	59.55	19.50
✗	✓	52.01	16.80	60.28	18.26
✓	✓	**54.48**	**15.23**	**61.72**	**18.26**

Table 8. **Ablation study:** *"ReLU"* corresponds to whether or not negative gradients are set to 0. *"All layers"* corresponds to whether or not the intermediate tokens are used to compute the gradient for every layer or if only the features from the last layer are used. Numbers are the AUC score for the perturbation base benchmark using the target class to compute the explainability map.

tend to highlight a few specific pixels with high intensity, often associated with the background rather than the object of interest. This pattern, consistent with findings in the literature [6, 12], suggests that certain tokens disproportionately capture attention weights. Consequently, methods that rely on raw attention weights to construct explainability maps, such as CheferCAM, exhibit similar artifacts. For instance, in the localization of "Basket" (Figure 7, row 1), the basket is marginally accentuated amidst a predominance of noisy. In contrast, for LeGrad, the presence of uniform noisy activations across different prompts results in minimal gradients for these regions, effectively filtering them out from the final heatmaps. This characteristic enables LeGrad to produce more focused and relevant visual explanations.

9. Additional results

9.1. Performance on SigLIP

To assess LeGrad's adaptability to non-standard ViT architectures, we evaluate its performance on SigLIP-B/16, a vision-language model with an attentional pooler, with results presented in Table 7 The results underscore the methods performance across both negative and positive perturbation-based benchmarks. Notably, in the open-vocabulary benchmark on OpenImagesV7, LeGrad achieved a p-mIoU of 25.4, significantly surpassing GradCAM's 7.0 p-mIoU, the next best method. These findings affirm the versatility of LeGrad, demonstrating its robust applicability to various pooling mechanisms within Vision Transformers. Further details on the methodological adaptations of LeGrad and other evaluated methods for compatibility with SigLIP are provided in the annex.

9.2. Image Classification

In this section, we extend our evaluation of the proposed LeGrad method to Vision Transformers (ViTs) that have been trained on the ImageNet dataset for the task of image classification. The results of this evaluation are presented in Table 9, also providing a comparison with other state-of-the-art explainability methods.

It shows that LeGrad achieves superior performance on the perturbation-based benchmark, particularly in scenarios involving positive perturbations.

Another observation is that even elementary explainability approaches, such as examining the raw attention maps from the final attention layer of the ViT, demonstrate competitive results. In fact, these basic methods surpass more complex ones like GradCAM (achieving an AUC of 53.1 versus 43.0 for negative perturbations).

9.3. Pertubation example

Figure 8 illustrates the perturbation-based benchmark of Section 4.1 in the main paper. Given the explainability map generated by the explainability method, for the negative (respectively positive) we progressively remove the most im-

Method	Negative		Positive	
	Predicted ↑	Target ↑	Predicted ↓	Target ↓
rollout [1]	53.10	53.10	20.06	20.06
Raw attention	45.55	45.55	24.01	24.01
GradCAM [33]	43.17	42.97	26.89	26.99
AttentionCAM [10]	41.53	42.03	33.54	34.05
Trans. Attrib. [9]	54.19	55.09	17.01	16.36
Partial-LRP [38]	50.28	50.29	19.82	19.80
CheferCAM [10]	54.68	55.70	17.30	16.75
LeGrad	**54.72**	**56.43**	**15.20**	**14.13**

Table 9. **SOTA comparison on ViT-B/16** on perturbation-based tasks on ImageNet-val for a ViT trained on ImageNet.

Neg. Slope	Neg AUC (↑)	Pos AUC (↓)
0 (ReLU)	54.48	15.23
0.1	54.33	15.39
0.5	53.25	15.84
0.7	52.96	16.33
1 (no ReLU)	52.01	16.80

Table 10. Ablation of smoothly transitioning from ReLU (slope=0) to no ReLU (slope=1) by replacing the ReLU operation in LeGrad by a LeakyReLU with varying slope factor.

portant (respectively the least important) part of the image. We then look at the decline in the model accuracy.

9.3.1. Ablations of ReLU and Layer Aggregation

We finally scrutinize the design choices underpinning LeGrad in Table 8. Specifically, we investigate the effect of discarding negative gradients before aggregating layer-specific explainability maps vai ReLU, as well as the implications of leveraging intermediate feature tokens Z^l to compute gradients for each respective layer. We use the framework of the perturbation benchmark explained in Section 4.1 and both ViT-L/14 and ViT-H/14 models. The results indicate that the omission of either component induces decline in performance, thereby affirming the role these elements play in the architecture of the method. Additionally, in Table 10, we conducted a new experiment with LeakyReLU with varying slope factor, hence smoothly transitioning from ReLU (slope=0) to no ReLU (slope=1). These results show a performance decline as more negative signal is incorporated, supporting ReLU's use in our aggregation scheme.

10. Gradient Distribution over Layers

Figures 9 & 10 extend the "gradient distribution over Layers" analysis conducted Section 4.6 to more backbone size and more set of weights .

In Figures 12 & 13 & 14 & 15 & 16 & 17 & 18 provide the gradient distribution over layer independetly for each class in PascalVOC. We observe that for most models and classes, the layers contributing the most were typically located towards the end of the ViT. An interesting exception to this trend was observed for the 'person' class, which exhibited a higher sensitivity to the middle layers across all model sizes and weight sets. We hypothesize that this is due to the high frequency of the 'person' class in the training data, enabling the ViT to identify the object early in the layer sequence, thereby triggering the activation in the middle layers.

Furthermore, we note that the most activated layer varied significantly depending on the class and the model. This variability was observed even between two models of the same size, such as ViT-L(openai) and ViT-L(metaclip). This observation underscores the rationale behind the LeGrad method's approach of utilizing multiple layers, hence alleviating the need to select a specific layer, as the optimal choice would differ from model to model.

Method	CSDC	PC	DC	D	SD	TS
GradCAM	0.75	0.67	0.68	0.91	0.7	0.48
Rollout	0.86	0.8	0.82	0.8	0.76	0.
Chefer LRP	**0.91**	**0.92**	0.89	0.9	0.74	0.95
LeGrad	0.90	0.83	**0.92**	**1.**	**0.77**	**0.97**

Table 11. Evaluation of ViT-B/16 on the FunnyBirds Co-12 setup

11. Sanity Check

The Co-12 recipe[28] is a set of 12 properties for evaluating the explanation quality of explainability method for machine learing models. These properties provide a comprehensive framework for assessing how well one can explain the decision-making process of a model. In [23] proposed a dataset, called FunnyBirds, to evaluate explainability methods for visual models.

As a sanity check for the proposed LeGrad method, we follow the authors guidelines and evaluate on the provided ViT-B/16[1] using LeGrad, showing improvement over gradient-based methods while being on par with LRP.

[1] https://github.com/visinf/funnybirds-framework

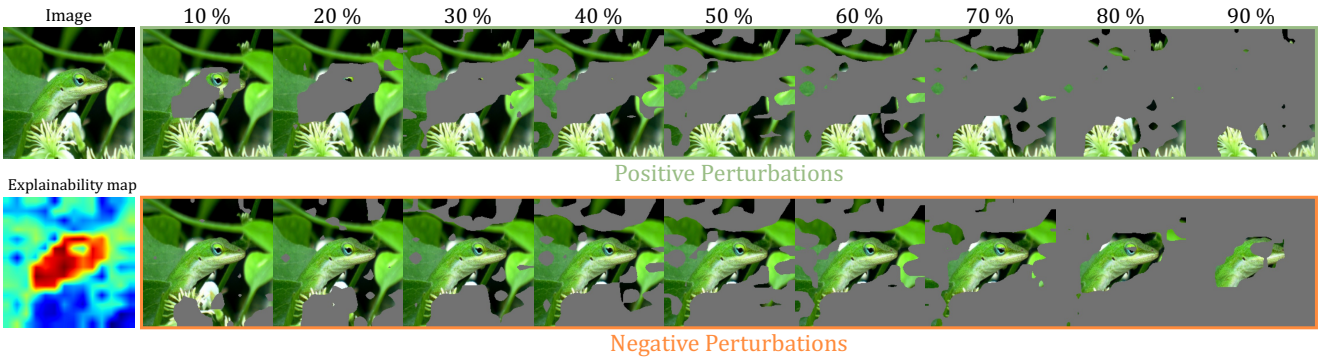

Figure 8. **Example of positive/negative perturbations:** illustration of positive and negative perturbations used in the perturbation-based benchmark. *(Top row)*: positive perturbation. *Bottom row*: negative perturbations

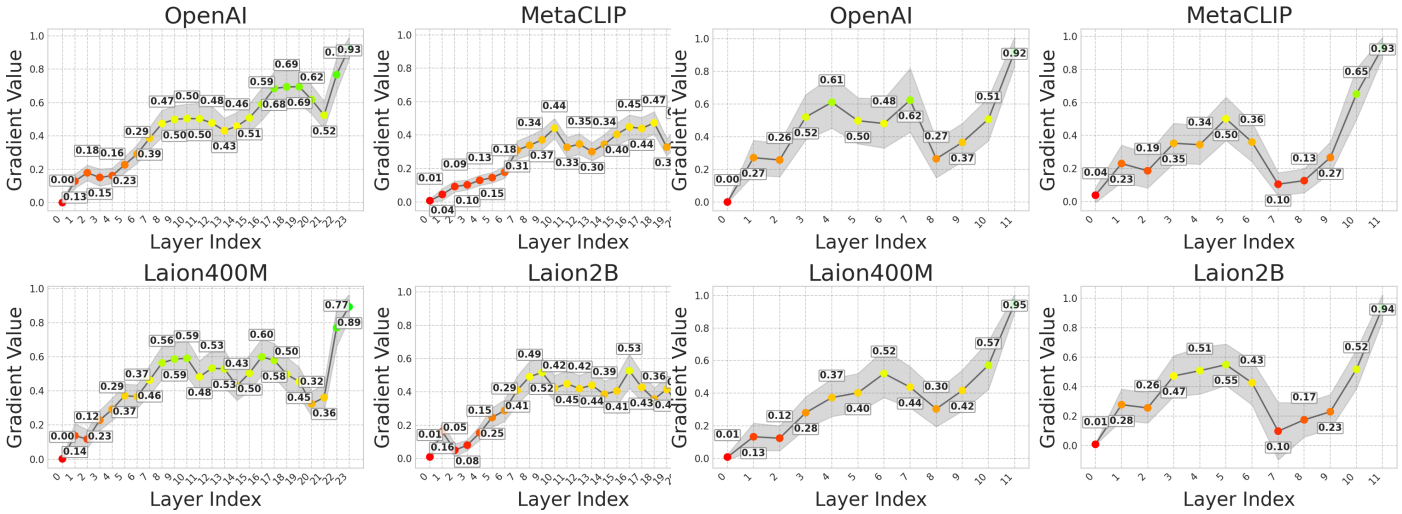

Figure 9. Gradient distribution over layers on PascalVOC for different pretrained ViT-L/14.

Figure 10. Gradient distribution over layers on PascalVOC for different pretrained ViT-B/16.

12. Personal and human subjects data

We acknowledge the use of datasets such as ImageNet and OpenImagesV7, which contain images sourced from the internet, potentially without the consent of the individuals depicted. We recognize that the VL models used in this study were trained on the LAION-2B dataset, which may include sensitive content. We emphasize the importance of ethical considerations in the use of such data.

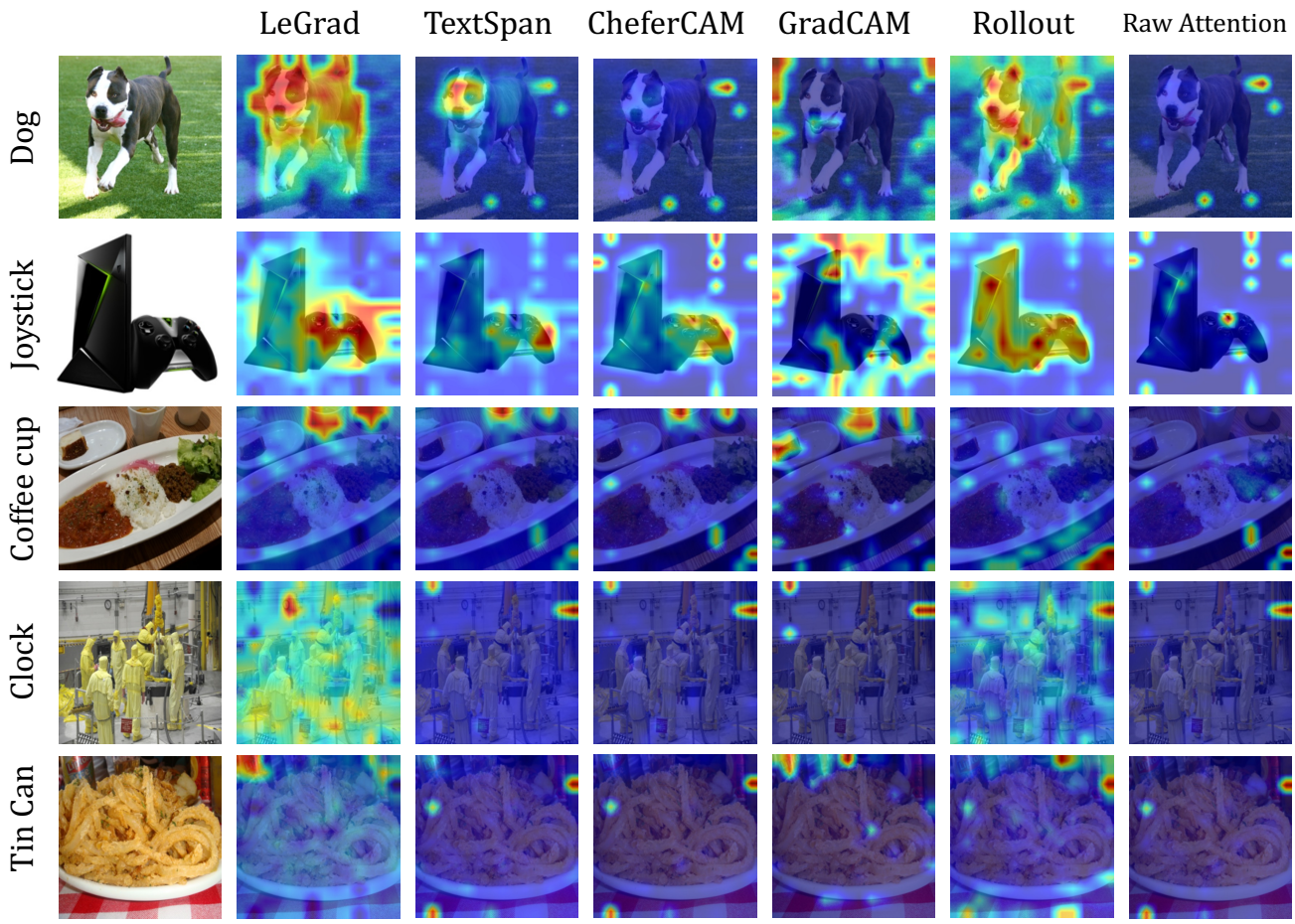

Figure 11. **Qualitative Comparison to SOTA:** visual comparison of different explainability methods on images from OpenImagesV7

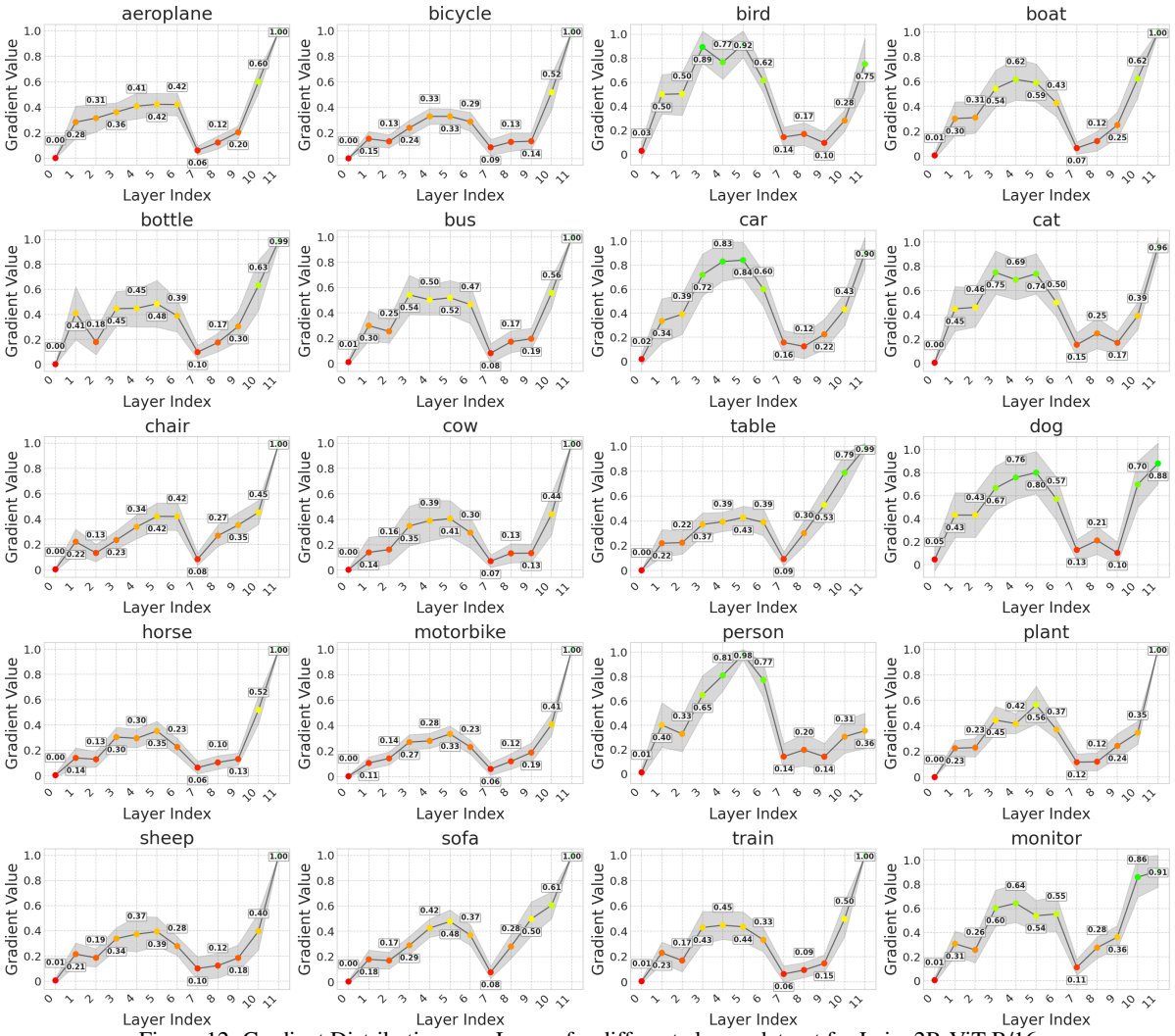

Figure 12. Gradient Distribution over Layers for different classes dataset for Laion2B-ViT-B/16.

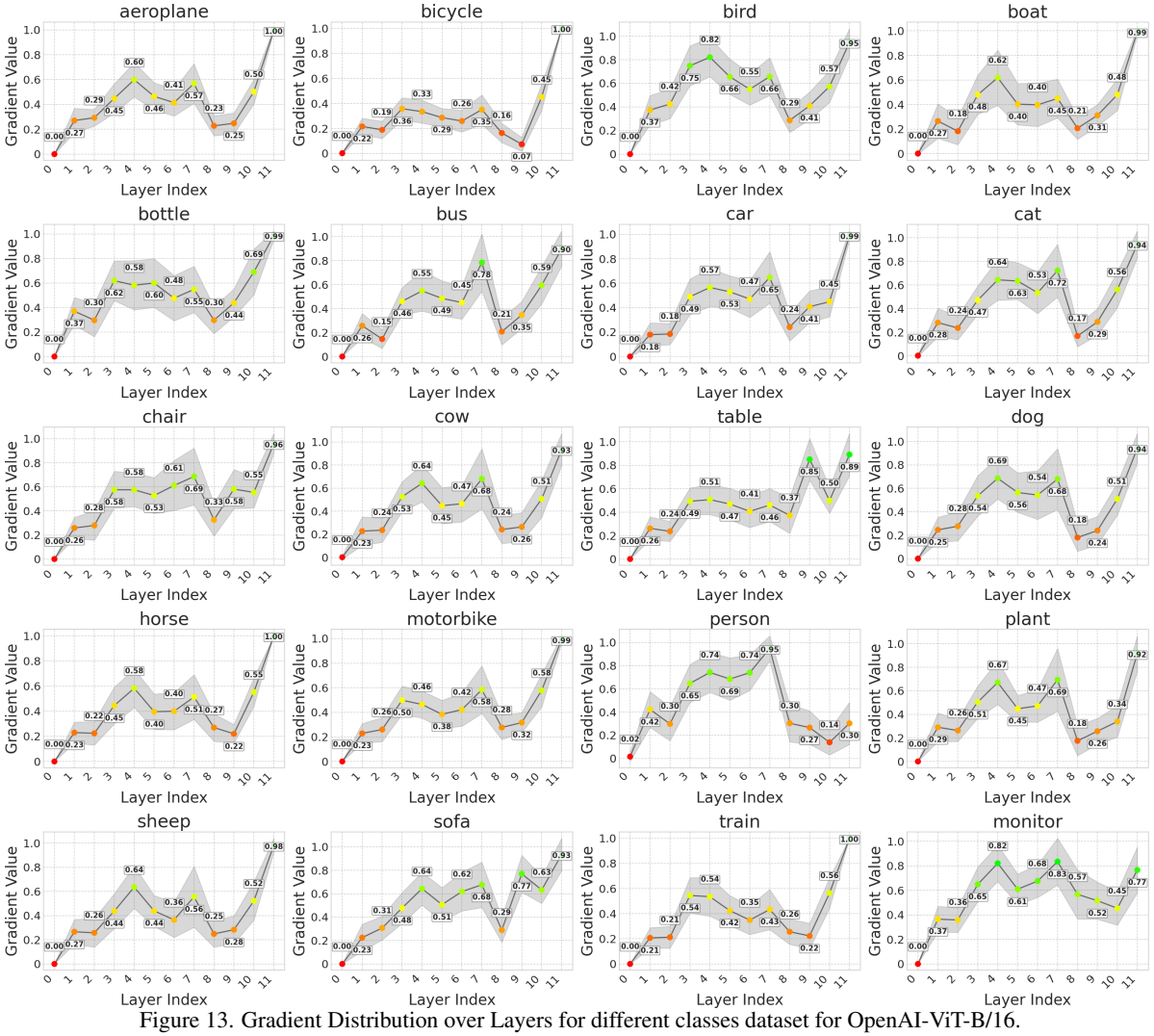

Figure 13. Gradient Distribution over Layers for different classes dataset for OpenAI-ViT-B/16.

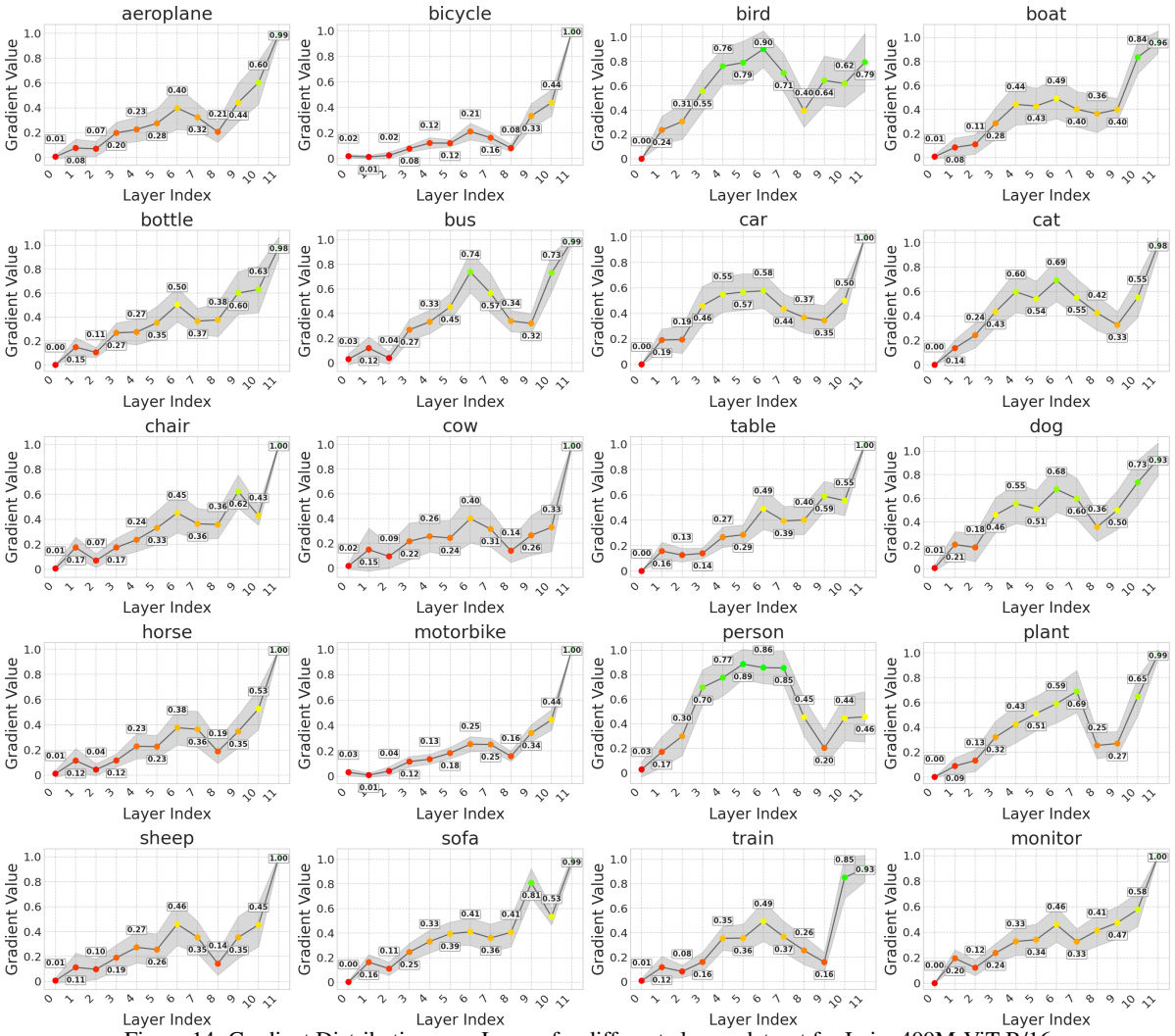

Figure 14. Gradient Distribution over Layers for different classes dataset for Laion400M-ViT-B/16.

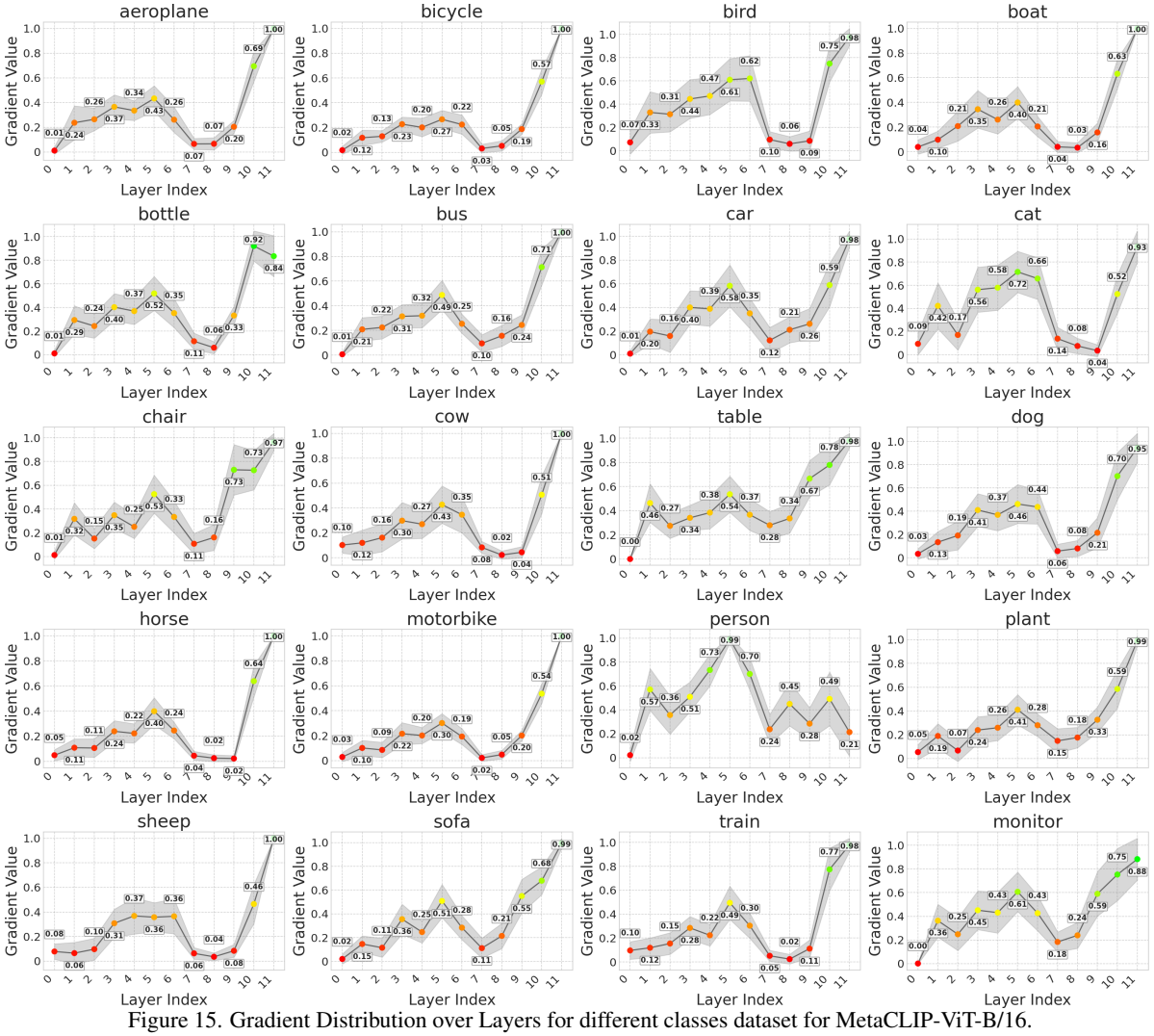

Figure 15. Gradient Distribution over Layers for different classes dataset for MetaCLIP-ViT-B/16.

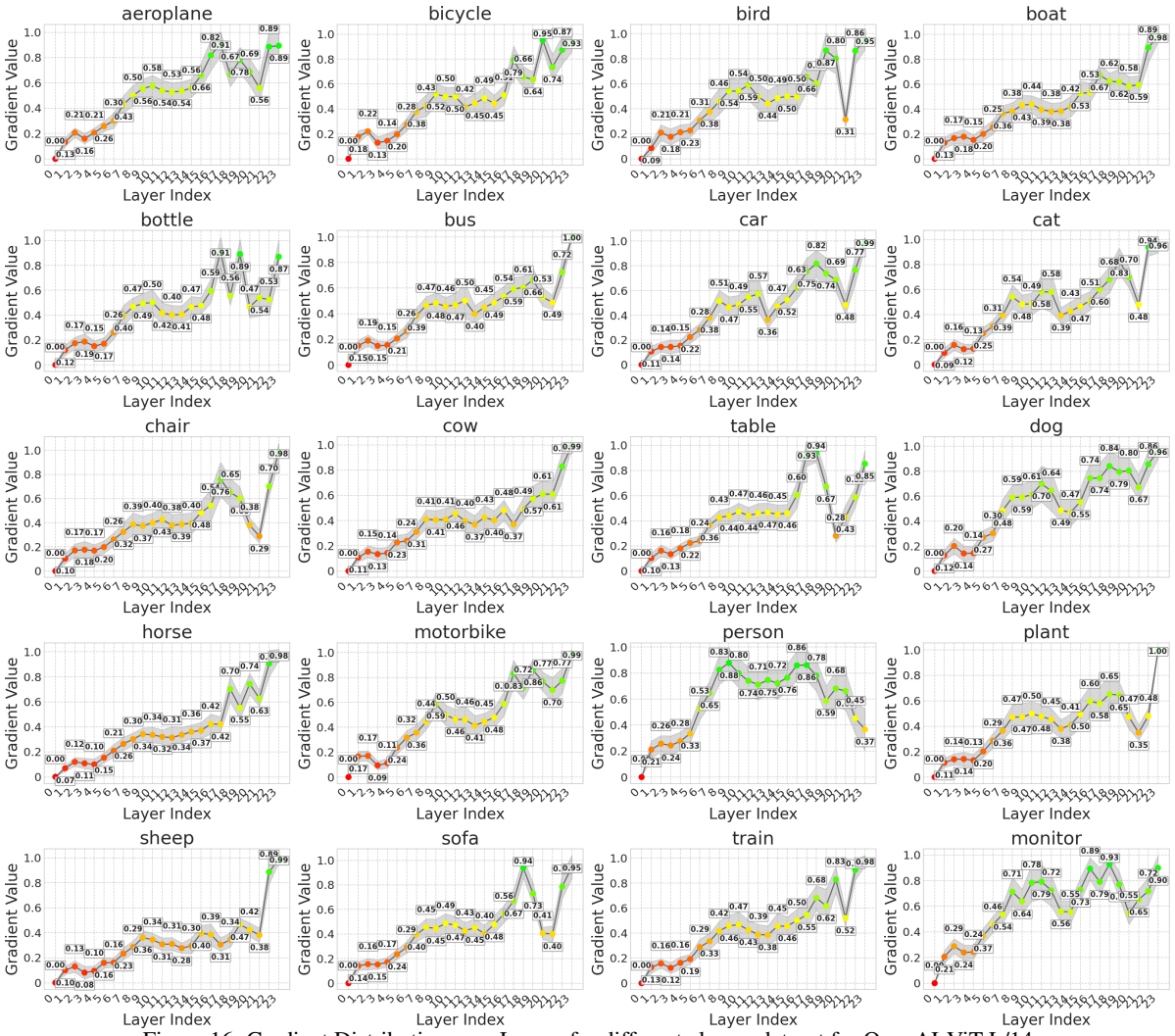

Figure 16. Gradient Distribution over Layers for different classes dataset for OpenAI-ViT-L/14.

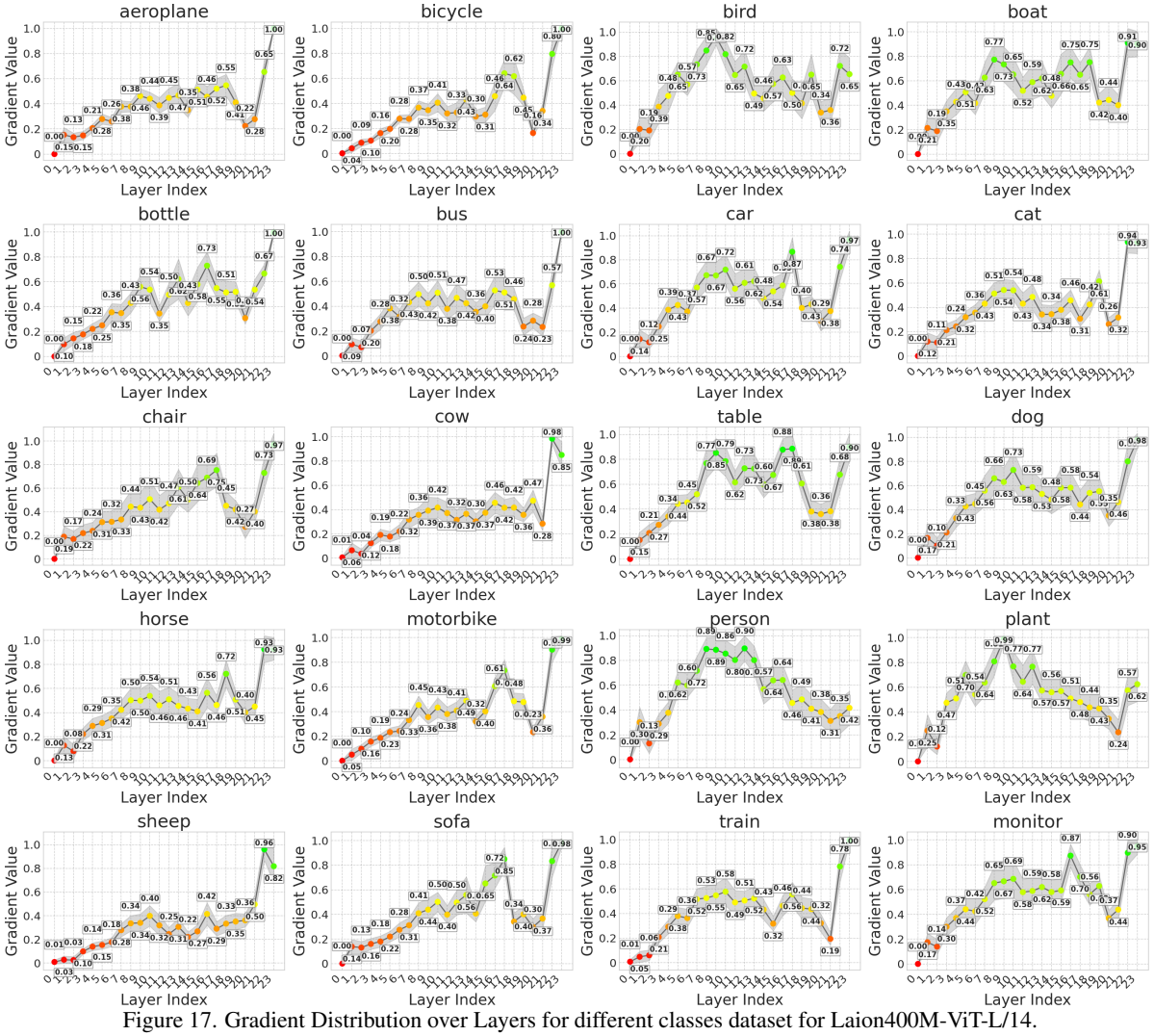

Figure 17. Gradient Distribution over Layers for different classes dataset for Laion400M-ViT-L/14.

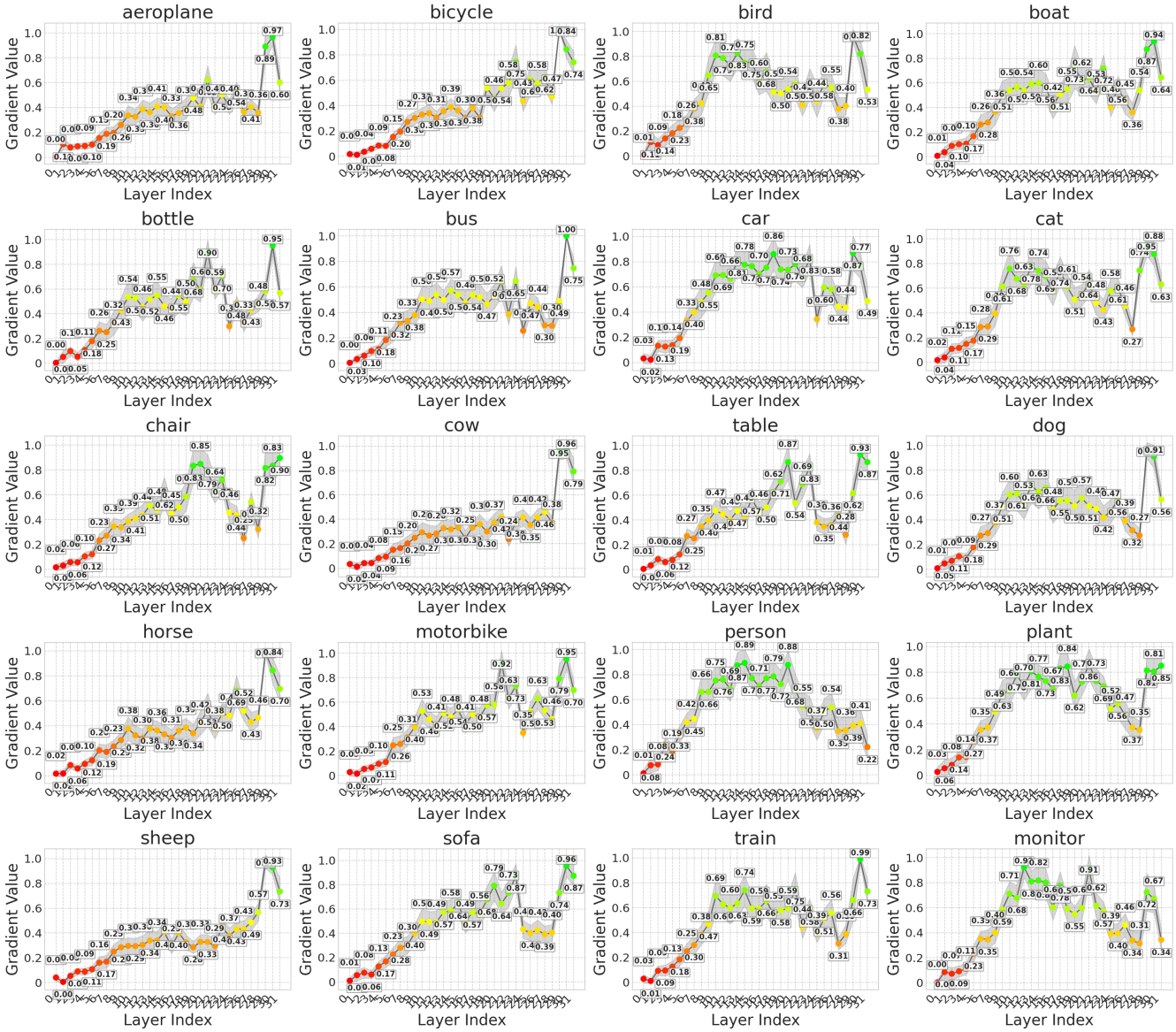

Figure 18. Gradient Distribution over Layers for different classes dataset for Laion2B-ViT-H/14.

Acknowledgments

Walid Bousselham is supported by the German Federal Ministry of Education and Research (BMBF) project STCL-01IS22067.

References

[1] Samira Abnar and Willem Zuidema. Quantifying attention flow in transformers. *ACL*, 2020. 1, 2, 3, 6

[2] Sebastian Bach, 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*, 2015. 1, 2

[3] Oren Barkan, Yuval Asher, Amit Eshel, Noam Koenigstein, et al. Visual explanations via iterated integrated attributions. In *ICCV*, 2023. 2

[4] Rodrigo Benenson and Vittorio Ferrari. From colouring-in to pointillism: revisiting semantic segmentation supervision. *arXiv preprint arXiv:2210.14142*, 2022. 2, 6

[5] Angie Boggust, Benjamin Hoover, Arvind Satyanarayan, and Hendrik Strobelt. Shared interest: Measuring human-ai alignment to identify recurring patterns in model behavior. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*, 2022. 2

[6] Walid Bousselham, Felix Petersen, Vittorio Ferrari, and Hilde Kuehne. Grounding everything: Emerging localization properties in vision-language transformers. *CVPR*, 2024. 6, 5

[7] Brandon Carter, Jonas Mueller, Siddhartha Jain, and David K. Gifford. What made you do this? Understanding black-box decisions with sufficient input subsets. In *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS)*. PMLR, 2019. 2

[8] Brandon Carter, Siddhartha Jain, Jonas Mueller, and David Gifford. Overinterpretation reveals image classification model pathologies. In *NeurIPS*, 2021. 2

[9] Hila Chefer, Shir Gur, and Lior Wolf. Transformer interpretability beyond attention visualization. In *CVPR*, 2021. 1, 2, 5, 6

[10] Hila Chefer, Shir Gur, and Lior Wolf. Generic attention-model explainability for interpreting bi-modal and encoder-decoder transformers. In *ICCV*, 2021. 1, 2, 3, 5, 6, 4

[11] Mehdi Cherti, Romain Beaumont, Ross Wightman, Mitchell Wortsman, Gabriel Ilharco, Cade Gordon, Christoph Schuhmann, Ludwig Schmidt, and Jenia Jitsev. Reproducible scaling laws for contrastive language-image learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023. 2

[12] Timothée Darcet, Maxime Oquab, Julien Mairal, and Piotr Bojanowski. Vision transformers need registers. *ICLR*, 2024. 1, 3, 5

[13] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2020. 1

[14] Dumitru Erhan, Y. Bengio, Aaron Courville, and Pascal Vincent. Visualizing higher-layer features of a deep network. *Technical Report, Univeristé de Montréal*, 2009. 2

[15] Yossi Gandelsman, Alexei A Efros, and Jacob Steinhardt. Interpreting clip's image representation via text-based decomposition. In *ICLR*, 2023. 3, 6, 7, 8

[16] Shanghua Gao, Zhong-Yu Li, Ming-Hsuan Yang, Ming-Ming Cheng, Junwei Han, and Philip Torr. Large-scale unsupervised semantic segmentation. *IEEE TPAMI*, 2022. 2, 6

[17] Rohit Girdhar, Alaaeldin El-Nouby, Zhuang Liu, Mannat Singh, Kalyan Vasudev Alwala, Armand Joulin, and Ishan Misra. Imagebind one embedding space to bind them all. In *CVPR*, 2023. 6, 7

[18] Gabriel Goh, Nick Cammarata †, Chelsea Voss †, Shan Carter, Michael Petrov, Ludwig Schubert, Alec Radford, and Chris Olah. Multimodal neurons in artificial neural networks. *Distill*, 2021. 1, 3

[19] Yuan Gong, Andrew Rouditchenko, Alexander H Liu, David Harwath, Leonid Karlinsky, Hilde Kuehne, and James R Glass. Contrastive audio-visual masked autoencoder. In *ICLR*, 2023. 7

[20] Mark Hamilton, Andrew Zisserman, John R Hershey, and William T Freeman. Separating the" chirp" from the" chat": Self-supervised visual grounding of sound and language. In *CVPR*, 2024. 2, 7

[21] David Harwath, Adria Recasens, Dídac Surís, Galen Chuang, Antonio Torralba, and James Glass. Jointly discovering visual objects and spoken words from raw sensory input. In *ECCV*, 2018. 7

[22] Evan Hernandez, Sarah Schwettmann, David Bau, Teona Bagashvili, Antonio Torralba, and Jacob Andreas. Natural language descriptions of deep visual features. In *ICLR*, 2021. 1, 3

[23] Robin Hesse, Simone Schaub-Meyer, and Stefan Roth. Funnybirds: A synthetic vision dataset for a part-based analysis of explainable ai methods. In *ICCV*, 2023. 6

[24] Andrei Kapishnikov, Tolga Bolukbasi, Fernanda B. Viégas, and Michael Terry. XRAI: Better attributions through regions. In *ICCV*. IEEE, 2019. 2

[25] Juho Lee, Yoonho Lee, Jungtaek Kim, Adam Kosiorek, Seungjin Choi, and Yee Whye Teh. Set transformer: A framework for attention-based permutation-invariant neural networks. In *ICML*. PMLR, 2019. 2

[26] Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. *NeurIPS*, 2017. 2

[27] Christoph Molnar. *Interpretable Machine Learning*. 2019. 2

[28] Meike Nauta, Jan Trienes, Shreyasi Pathak, Elisa Nguyen, Michelle Peters, Yasmin Schmitt, Jörg Schlötterer, Maurice Van Keulen, and Christin Seifert. From anecdotal evidence to quantitative evaluation methods: A systematic review on evaluating explainable ai. *ACM Computing Surveys*, 2023. 6

[29] Vitali Petsiuk, Abir Das, and Kate Saenko. RISE: Randomized input sampling for explanation of black-box models. In *BMVC*. BMVA, 2018. 2

[30] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry,

Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*. PMLR, 2021. 3

[31] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "Why should i trust you?": Explaining the predictions of any classifier. In *Proceedings of the International Conference on Knowledge Discovery and Data Mining (KDD)*. ACM, 2016. 2

[32] 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. *IJCV*, 2015. 2

[33] 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 *ICCV*, 2017. 1, 2, 6

[34] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. In *ICLR, (Workshop Track)*, 2014.

[35] Daniel Smilkov, Nikhil Thorat, Been Kim, Fernanda Viégas, and Martin Wattenberg. Smoothgrad: removing noise by adding noise. *arXiv preprint arXiv:1706.03825*, 2017.

[36] Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox, and Martin A. Riedmiller. Striving for simplicity: The all convolutional net. In *ICLR*, 2015.

[37] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *ICML*. JMLR. org, 2017. 2

[38] Elena Voita, David Talbot, Fedor Moiseev, Rico Sennrich, and Ivan Titov. Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned. In *ACL*, 2019. 1, 2, 6

[39] Hu Xu, Saining Xie, Xiaoqing Tan, Po-Yao Huang, Russell Howes, Vasu Sharma, Shang-Wen Li, Gargi Ghosh, Luke Zettlemoyer, and Christoph Feichtenhofer. Demystifying clip data. In *ICLR*, 2023. 3

[40] Jiahui Yu, Zirui Wang, Vijay Vasudevan, Legg Yeung, Mojtaba Seyedhosseini, and Yonghui Wu. Coca: Contrastive captioners are image-text foundation models. *arXiv preprint arXiv:2205.01917*, 2022. 2

[41] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *ECCV*. Springer, 2014. 2

[42] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. *ICCV*, 2023. 2