DEX-AR: A Dynamic Explainability Method for Autoregressive Vision-Language Models

Walid Bousselham[1], Angie Boggust[2], Hendrik Strobelt[3,4], and Hilde Kuehne[1,4]

[1] University of Tuebingen & Tuebingen AI Center
[2] MIT CSAIL
[3] IBM Research
[4] MIT-IBM Watson AI Lab

Abstract. As Vision-Language Models (VLMs) become increasingly sophisticated and widely used, it becomes more and more crucial to understand their decision-making process. Traditional explainability methods, designed for classification tasks, struggle with modern autoregressive VLMs due to their complex token-by-token generation process and intricate interactions between visual and textual modalities. We present DEX-AR (Dynamic Explainability for AutoRegressive models), a novel explainability method designed to address these challenges by generating both per-token and sequence-level 2D heatmaps highlighting image regions crucial for the model's textual responses. The proposed method offers to interpret autoregressive VLMs, accounting for the varying importance of layers and generated tokens—by computing layer-wise gradients with respect to attention maps during the token-by-token generation process. DEX-AR introduces two key innovations: a dynamic head filtering mechanism that identifies attention heads focused on visual information, and a sequence-level filtering approach that aggregates per-token explanations while distinguishing between visually-grounded and purely linguistic tokens. Our evaluation on ImageNet, VQAv2, and PascalVOC, shows a consistent improvement in both perturbation-based metrics, using a novel normalized perplexity measure, as well as segmentation-based metrics. Code is available at https://walidbousselham.com/DEX-AR

1 Introduction

Vision-Language Models (VLMs) have emerged as a transformative force in AI, with remarkable capabilities in bridging visual understanding and natural language generation. Recent advances have produced increasingly better models like LLaVA [25, 26], PaliGemma [5], Gemini-1.5 [37] and GPT-4o [2], which can engage in complex visual reasoning tasks ranging from image captioning to open-ended dialogue about visual content. These models have found applications across diverse domains, from assisting visually impaired users [30,42] to enabling more natural human-AI interaction through multimodal interfaces [44]. However,

as these models grow in complexity and capability, understanding their decision-making process becomes increasingly challenging. Modern VLMs employ architectures that combine visual encoders with LLMs through multiple layers of attention mechanisms, making it difficult to trace how visual information influences generated text. Understanding this process is crucial, as interpretability studies [19, 38] have revealed critical failure modes in VLMs and shown that model explanations significantly improve human-AI collaboration. This understanding becomes particularly important as VLMs are deployed in high-stakes applications like autonomous systems [8].

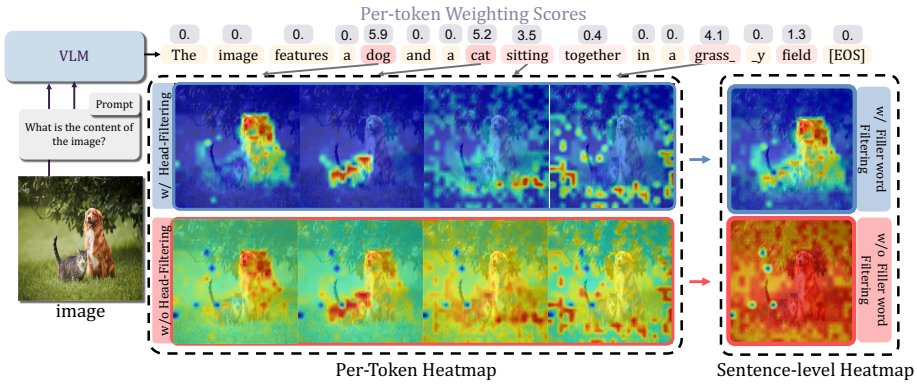

Fig. 1: Example of token-level and sentence-level attribution maps for Vision-Language Models (VLMs). Given an input image and prompt, DEX-AR produces per-token heatmaps highlighting relevant image regions for each generated word. These are then aggregated into a final sentence-level heatmap using token-specific weighting scores that reflect visual relevance.

While numerous explainability methods exist for computer vision models [9, 10, 32] and language models([43, 45]), these approaches are not designed to account for the dynamic nature of autoregressive generation. Traditional computer vision explainability methods typically focus on classification tasks with fixed outputs, failing to capture the dynamic nature of token-by-token generation and the complex interaction between visual and textual modalities in VLMs. Furthermore, while significant progress has been made in explaining contrastive models (e.g., CLIP) [1, 6, 9, 10, 15, 16], these approaches are designed to account for the dynamic nature of autoregressive generation. Although emerging methods like TAM [24] have begun to address this, accurately capturing the varying importance of different generated tokens remains challenging, as some serve primarily linguistic functions while others directly reference visual content. This gap suggests that directly applying traditional methods to autoregressive tasks may lead to incomplete interpretations, highlighting the need for specialized approaches that address both visual grounding and sequential dependencies This lack of suitable explainability methods could lead to potentially misleading in-

terpretations from traditional approaches, inhibiting the improvement of model reliability as well as the detection of potential failure modes.

We argue that the unique characteristics of autoregressive VLMs demand a specialized approach to explainability. Namely, as these models generate text tokens sequentially, with each token potentially attending to different parts of the image and previous textual context, understanding this process requires tracking the flow of visual information through the model's layers and identifying which image regions influence specific generated tokens, a challenge not addressed by current explainability methods. To fill this gap, we introduce DEX-AR (Dynamic Explainability for AutoRegressive models), a novel explainability method specifically designed for autoregressive VLMs. DEX-AR leverages layer-wise gradients with respect to attention maps to produce 2D heatmaps that highlight the image regions most influential for each generated token. On top of that, we apply a dynamic head filtering mechanism that identifies attention heads focused on visual information, and a token-level filtering approach that distinguishes between visually-grounded and purely linguistic tokens. The resulting framework provides per-token explanation maps that uncover the model's decision-making process at each generation step, enabling a layer-wise analysis to understand information flow throughout the network, while remaining model-agnostic by leveraging the attention's gradient, common to all transformer-based architectures.

We evaluate the proposed method on various downstream tasks and datasets, including perturbation on ImageNet [31] and VQAv2 [18], showing a consistent improvement over alternative explainability methods across different VLM architectures. To further quantitatively evaluate the proposed filtering, we introduce PascalVOC-QA, a specialized dataset that provides natural language question-answer pairs with segmentation information and explicit annotations distinguishing between tokens derived from visual content and linguistic filler tokens. It shows that the dual-filtering approach effectively distinguishes visually-relevant content, improving the Signal-to-Noise Ratio from 9.16 to 96.12 on Pascal-QA.

We summarize the contributions of this work as follows:

1) We propose a gradient-based explainability method specifically designed for autoregressive VLMs, handling the specific characteristics of token-by-token generation.

2) We extend the proposed method by a general dual-filtering mechanism that dynamically weighs attention heads and tokens based on their visual relevance.

3) We propose a new evaluation setup together with a set of metrics to assess the quality of the explainability methods for autoregressive VLMs.

2 Related Works

Explainability in Deep Learning Models Recent years have witnessed significant advances in explaining deep learning models' decisions, particularly in computer vision tasks. Traditional gradient-based methods such as Grad-CAM [32], Guided Backpropagation [35], and Integrated Gradients [36] compute gradients of target outputs with respect to input features or intermediate

activations to generate saliency maps. While effective for convolutional neural networks (CNNs) in classification tasks, these methods face limitations when applied to modern transformer-based architectures [39] and sequential generation tasks [21]. Attention mechanisms have emerged as an alternative approach to model interpretability [11, 14]. Methods like Attention Rollout [1] aggregate attention weights across layers to trace information flow. However, recent studies have shown that attention weights alone may not reliably indicate feature importance [33, 40], particularly in deep architectures where attention patterns are complex [7]. This limitation is especially pronounced in multimodal settings where visual and textual information interact [5, 22, 23, 25, 26, 41]. **Explainability in Vision-Language Models** VLMs present unique challenges for explainability due to their multimodal nature and complex architectures. A substantial body of work has focused on interpreting contrastive models like CLIP. Methods such as [6, 15] and adaptations of generic transformer explainability [9] have proven effective in visualizing how these models align image and text embeddings. However, these techniques are primarily designed for static alignment tasks and do not account for the dynamic state changes inherent to autoregressive generation. Existing methods have been adapted for VLMs, such as developing hybrid approaches that combine gradients with attention [3, 9, 10]. Furthermore, established techniques such as Integrated Gradients (IG) [36] and model-agnostic approaches like RISE [28] could also been utilized to interpret VLMs. However, several key challenges remain unaddressed in current approaches. **Sequential Generation:** Most existing methods are designed for fixed outputs and struggle with the token-by-token generation process in autoregressive models [2, 5, 25, 26, 34, 37]. **Token-Level vs Sequence-Level Attribution:** Current approaches either lack the granularity to attribute model outputs at the level of individual tokens, or fail to distinguish between content and filler words. Recent work has attempted to address these challenges through various approaches. Methods like [12] have explored visualizing information flow in neural machine translation, while others have focused on developing token-level attribution techniques [13]. Most recently, TAM [24] proposed an estimated causal inference method to mitigate context interference in autoregressive VLMs. However, TAM relies on static visual features and post-hoc statistical estimation. In contrast, DEX-AR utilizes layer-wise gradients to capture the dynamic attention mechanism at each specific generation step. DEX-AR, builds upon these foundations while extending those ideas by a novel layer-wise gradient computation approach combined with dynamic filtering mechanisms, enabling attribution of visual information throughout the autoregressive generation process.

3 Method

3.1 Autoregressive Vision-Language Models

Current VLMs such as LLaVA combine a visual encoder (typically a Vision Transformer) with a Large Language Model (LLM) to process multimodal content and generate text.

Let N denote the number of visual tokens produced by the visual encoder, T_c the number of context tokens in the input prompt, and T_a the number of answer tokens generated autoregressively by the model. The visual encoder processes an input image to generate a set of N visual tokens, which are concatenated with T_c context tokens that encapsulate instructions or queries, thereby forming an initial token sequence. This sequence, comprising $N + T_c$ tokens, is ingested by the LLM.

The LLM consists of L transformer layers, each equipped with h attention heads, and generates a textual response comprising T_a tokens. At each generation step $t \in \{1, \ldots, T_a\}$, the model processes a token sequence of length $T_t = N+T_c+t$, where t denotes the current step in the autoregressive process. Consequently, the total number of tokens processed by the LLM upon completion of generation is $T = N + T_c + T_a$.

Formally, for each layer $l \in \{1, \ldots, L\}$ and generation step t, the LLM produces hidden states denoted as $Z^{l,t} \in \mathbb{R}^{T_t \times d}$, where d is the embedding dimension. For clarity and without loss of generality, we assume that the first N tokens correspond to the visual tokens and the subsequent T_c tokens correspond to the context tokens, such that:

$$\begin{aligned} Z^{l,t} &= \{ \underbrace{z_1^l, \ldots, z_N^l}_{\text{visual tokens } Z_v^l}, \underbrace{z_{N+1}^l, \ldots, z_{N+T_c}^l}_{\text{context tokens } Z_c^l}, \underbrace{y_1^l, \ldots, y_t^l}_{\text{answer tokens } Y_a^l} \} \\ &= \{Z_v^l, Z_c^l, Y_a^{l,t}\}. \end{aligned} \tag{1}$$

Here, $Z_v^l \in \mathbb{R}^{N \times d}$ and $Z_c^l \in \mathbb{R}^{T_c \times d}$ denote the visual and context tokens, respectively, which remain invariant across generation steps due to the causal attention mechanism employed by the LLM.

At each step t, within each layer l, the model computes attention maps $A^{l,t} \in \mathbb{R}^{h \times T_t \times T_t}$, where $T_t = N+T_c+t$, capturing the interaction weights across the h attention heads. At every generation step t, the LLM outputs a logits vector $o^t \in \mathbb{R}^V$, where V is the vocabulary size, via a linear projection of the final hidden state:

$$o^t = \text{LM\_Head}(Z_{-1}^{L,t}) = \text{LM\_Head}(y_t^L) \tag{2}$$

Then based on these logits, typically by applying a softmax function followed by a selection mechanism such as argmax or sampling the predicted word is selected. We denote $\hat{o}^t \in R$ the logit corresponding to the selected word in the vocabulary. This autoregressive process generates the answer, with each token generation step influenced by both visual, context and already generated answer, as encoded in the hidden states and attention maps.

3.2 Explainability per Token

Based on the objective of computing explainability maps that highlight the influence of regions of the image on each generated token in the autoregressive process, we leverage gradients w.r.t. the attention maps in the transformer layers of the LLM. The method operates as follows:

Computing Intermediate Logits: At each generation step $t \in \{1, \ldots, T_a\}$, we aim to assess the influence of the visual tokens Z_v^l on the prediction of the next token. To isolate the contributions from each layer $l \in \{1, \ldots, L\}$, we compute the intermediate logits using the hidden states from layer l instead of the final output layer. Specifically, we obtain:

$$o^{l,t} = \text{LM\_Head}(Z_{-1}^{l,t}) = \text{LM\_Head}(y_t^l) \in \mathbb{R}^V \tag{3}$$

where $Z_{-1}^{l,t}$ is the hidden state corresponding to the last token (the one being predicted) at layer l. We denote $\hat{o}^{l,t} \in \mathbb{R}$ as the logit corresponding to the sampled word at step t. This operation follows the *Logit Lens* approach [4, 17, 27], which interprets intermediate hidden states by projecting them into the vocabulary space. The residual stream architecture means intermediate states are not out-of-distribution for the LM head, and our ablation (Supp. Tab. 7) confirms that using intermediate features consistently improves IoU across all four architectures. Conditioning on the last token index is dictated by the causal attention mechanism, where the hidden state at step t accumulates the context of all preceding tokens and visual embeddings required for prediction.

Gradient Computation: We compute the gradient of the logit $\hat{o}^{l,t}$ w.r.t. the attention map$A^{l,t} \in \mathbb{R}^{h \times T_t \times T_t}$ at layer l: $\nabla A^{l,t} = \frac{\partial \hat{o}^{l,t}}{\partial A^{l,t}} \in \mathbb{R}^{h \times T_t \times T_t}$ where h is the number of attention heads and $T_t = N + T_c + t$ is the total number of tokens processed up to step t.

Focusing on the Last Token and Visual Tokens: Since we are interested in the generation of the current token, we focus on the gradient w.r.t. the attention maps involving the last token in the sequence. We extract the gradients corresponding to the last row (see Figure 2): $\nabla A_{-1}^{l,t} = \nabla A^{l,t}[:, -1, :] \in \mathbb{R}^{h \times T_t}$. Next, we isolate the gradients of the visual tokens: $\nabla A_{-1,v}^{l,t} = \nabla A_{-1}^{l,t}[:, : N] \in \mathbb{R}^{h \times N}$ where N is the number of visual tokens.

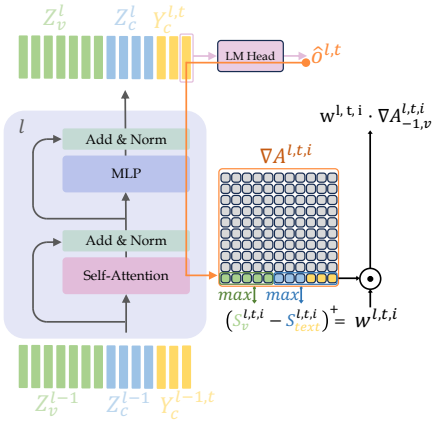

Fig. 2: Architecture overview of DEX-AR. At each layer l, head i and generation step t, gradients of attention maps are computed and weighted based on their relative focus on visual versus textual tokens to produce attribution maps.

Dynamic Head Filtering: In practice, not all heads/layers contribute equally to attending to the visual tokens during the generation process. Some attention heads may focus primarily on textual information or other aspects that are not directly relevant to the image content.

Including gradients from such heads can introduce noise into the explainability maps, reducing their interpretability. To address this, we introduce a filtering mechanism that dynamically weighs the contributions of different attention heads and layers based on their relative focus on visual tokens. For each

attention head $i \in \{1, \ldots, h\}$ at layer l and generation step t, we compute the maximum gradient magnitude with respect to the visual tokens and the text tokens (i.e., context and answer tokens):

$$S^{l,t,i}_{\text{img}} = \max_j \left(\nabla A^{l,t}_{-1,v}[i,j] \right) \in R, \quad S^{l,t,i}_{\text{text}} = \max_j \left(\nabla A^{l,t}_{-1,\text{text}}[i,j] \right) \in R, \tag{4}$$

where $\nabla A^{l,t}_{-1,v}[i,:]$ represents the gradients with respect to visual tokens and $\nabla A^{l,t}_{-1,\text{text}}[i,:]$ represents the gradients with respect to text tokens for head i (see Figure 2).
The weighting factor for each attention head is computed as:

$$w^{l,t,i} = \left(S^{l,t,i}_{\text{img}} - S^{l,t,i}_{\text{text}} \right)^+ \tag{5}$$

where $(\cdot)^+$ denotes the ReLU function, ensuring that only positive differences contribute. This weighting factor reflects the focus of the attention head on visual tokens relative to text tokens. A higher $w^{l,t,i}$ indicates that head i at layer l is more attentive to the image content than to the text. A maximum-based approach is particularly effective here because it captures the strongest visual attention signals regardless of object size. For instance, when localizing a small object like a "tennis ball" versus a large object like "sky", averaging-based approaches would bias towards larger objects due to their spatial extent. In contrast, a maximum-based approach identifies the most salient visual signals independent of their coverage, leading to more accurate localization across objects of varying sizes (see also Sec.4.5). The heatmap for the generated token t, noted $E^{(t)}$, is then a weighted sum of:

$$\bar{E}^{(t)} = \overbrace{\sum_{l=1}^{L} \underbrace{\sum_{i=1}^{h} w^{l,t,i} \cdot \nabla A^{l,t}_{-1,v}}_{\text{over heads}}}^{\text{over layers}} \in R^N, \tag{6}$$

$$E^{(t)} = \texttt{Norm}\left(\texttt{Reshape}(\bar{E}^{(t)}) \right) \in R^{W \times H},$$

where `Norm` is the min-max normalization, `Reshape` reshape the tokens to a 2D grid and H and W are the height and width dimensions of the visual token grid.

3.3 Sequence-level Explainability Map

When VLMs generate descriptions or answer questions about images, they produce sequences of tokens that vary in their reliance on visual information. Understanding the model's reasoning requires aggregating these token-level explanations while accounting for their varying visual relevance. For example, in the response "The young woman is wearing a floral dress", tokens like "woman", "floral", and "dress" directly reference visual content, while "the", "is", and "wearing" serve primarily grammatical functions. We ground this filtering in

the interpretation of gradients as a measure of prediction sensitivity. A high gradient magnitude implies that perturbing the attending features would significantly degrade model confidence for the current token. By comparing the maximum sensitivity to visual features (S^l_{img}) against textual history (S^l_{text}), δ^t quantifies the "visual necessity" of the generated token. This effectively filters out tokens governed by linguistic priors, where the model is robust to visual perturbation, while amplifying those where visual evidence is a necessary condition for prediction. To address this, we introduce an additional filtering mechanism that weighs the contribution of each generated token based on its reliance on visual information, see Fig. 1. For each token t, we compute a token-specific weight by comparing the maximum attention to visual versus textual content across all heads and layers: $\delta^t = \left(\max_{l,i} S^{l,t,i}_{\text{img}} - \max_{l,i} S^{l,t,i}_{\text{text}}\right)^+$, where $\max_{l,i}$ denotes the maximum over all layers l and heads i. This weighting scheme effectively suppresses the contribution of tokens that are primarily predicted based on linguistic context rather than visual information. The final explainability map for the whole sequence of T_a generated tokens is:

$$\bar{E} = \overbrace{\sum_{t=1}^{T_a} \delta^t \cdot E^{(t)}}^{over\ tokens} \in R^N, \quad E = \texttt{Norm}\left(\texttt{Reshape}(\bar{E})\right) \in R^{W \times H}. \tag{7}$$

This dual-filtering approach at both head/layer and token level ensure that the explainability maps focus on tokens and attention patterns that are genuinely informed by visual content, rather than those driven by linguistic context.

Batched Computation. Since DEX-AR operates *post-hoc*, the full generated sentence is available at explanation time. This enables replacing the T_a sequential forward passes with a single forward pass over the complete sequence (prompt concatenated with all T_a target tokens). Causal masking guarantees that the hidden state at each position is identical to the one obtained during autoregressive generation, such that no information leaks from future tokens.

For the gradient computation, we exploit a structural property of causal attention. At layer l, the intermediate logit at position n depends on the hidden state at that position, which in turn is a function of attention row n only. The gradient $\partial \hat{o}^{l,n} / \partial A^l$ is therefore non-zero exclusively at row n, meaning that gradients for different target tokens occupy disjoint rows of the attention matrix. We can thus sum the logits of all T_a target positions and perform a single backward pass per layer, recovering the exact per-token gradients without interference. This reduces the overall cost from T_a forward passes and $T_a \times L$ backward passes to 1 forward pass and L backward passes, while producing mathematically identical attribution maps (see Sec. 4.7).

4 Experiments

4.1 Datasets and Tasks

Perturbation: ♢**Task:** To quantitatively evaluate the proposed method, we employ a perturbation-based evaluation protocol. This approach is particularly valuable for assessing explainability methods as it directly measures how the removal of supposedly important image regions impacts model performance. By systematically perturbing regions identified as significant, it can be verify whether these regions truly contribute to the model's decision-making process. Specifically, for a range of percentages $p \in \{0\%, 10\%, ..., 90\%\}$, we replace the top-$p$ pixels according to the heatmap values with the dataset's mean pixel value. ♢**Metric:** While such evaluation protocols are well-established for classification tasks, they present unique challenges for generative models like VLMs. Traditional metrics such as classification accuracy are not directly applicable due to the open-ended nature of text generation and the complexity of comparing generated text sequences. Prior work often relies on additional LLMs or manual evaluation to assess the correctness of generated responses, which can introduce external biases and typically yields only binary success metrics. We propose a more natural approach leveraging the model's internal confidence through perplexity measurements. For a sequence of ground truth tokens $y = (y_1, ..., y_T)$, the perplexity is computed as: $\text{PPL}(y) = \exp\left(\frac{1}{T}\sum_{t=1}^{T} - \log P(y_t|y_{<t}, \mathcal{I})\right)$ where $\mathcal{I}$ represents the (potentially perturbed) input image. Higher perplexity scores indicate increased uncertainty in the model's predictions, making it an effective measure of how perturbations affect the model's confidence and accuracy. When important visual regions are removed, we expect to see a significant increase in perplexity, reflecting the model's reduced ability to generate accurate responses without crucial visual information. This metric provides a continuous score reflecting how confidently the model predicts the expected sequence of tokens. To account for varying baseline perplexities across different samples and model capacities, we normalize the perplexity at each perturbation level by the original (unperturbed) perplexity: $\text{PPL}^{\text{norm}}(p) = \frac{\exp(\text{PPL}(y|\mathcal{I}_p)}{\exp(\text{PPL}(y|\mathcal{I}_0))}$ where $\mathcal{I}_p$ denotes the image with $p\%$ of pixels perturbed. This normalization helps isolate the impact of the perturbation from the inherent difficulty of predicting certain sequences. The final evaluation metric is computed as the Area Under the Curve (AUC) of the normalized perplexity versus perturbation percentage: $\text{AUC} = \int_0^{0.9} \text{PPL}^{\text{norm}}(p)dp$ For positive perturbation, a higher AUC indicates that perturbing pixels identified as important by the heatmap lead to a larger degradation in model performance, suggesting a more accurate attribution map. We evaluate all methods using both positive perturbation (removing highest-valued pixels) and negative perturbation (removing lowest-valued pixels) to ensure a comprehensive assessment of the attribution quality. For a detailed discussion of perplexity as evaluation metric, we refer to the Supp. Matt., Sec 11. ♢**Datasets:** We compare the performance on two dataset: ImageNet [31] and VQAv2 [18]. From the ImageNet-val, we randomly sampled 5,000

images spanning 1,000 object categories.From VQAv2-val, we randomly sampled 1,000 image-question pairs spanning various question types (e.g., yes/no, counting, open-ended).This diverse selection ensures a comprehensive evaluation of all method across recognition tasks.

Segmentation-based Evaluation ♢**Task:** While perturbation analysis provides insights into the overall quality of the resulting attributions, we also evaluate how precisely explicitly referenced objects can be localized. This evaluation serves as a critical sanity check: when a VLM identifies specific objects in an image, the attribution map should highlight the corresponding regions. ♢**Metrics:** To quantitatively assess localization accuracy, we employ two complementary metrics. First, we compute the Intersection over Union (IoU) between the continuous attribution maps and ground truth masks. Given the continuous nature of attribution maps A, we determine the optimal threshold τ^* for each prediction by maximizing the IoU score across $k = 20$ thresholds: $\text{IoU}(A, M) = \max_{\tau \in \mathcal{T}} \frac{|\{A>\tau\} \cap M|}{|\{A>\tau\} \cup M|}$ where M represents the ground truth binary mask and $\mathcal{T}$ is a set of k equally spaced thresholds spanning the range of attribution values. We also introduce a soft version of IoU that directly operates on continuous-valued attribution maps without requiring thresholding: $\text{Soft-IoU}(A, M) = \frac{\sum_{i,j} A_{i,j} M_{i,j}}{\sum_{i,j} A_{i,j} + \sum_{i,j} M_{i,j} - \sum_{i,j} A_{i,j} M_{i,j}}$.

This formulation provides a smooth evaluation metric that does not depend on thresholding, thus avoids the potential loss of information from binary thresh. Additionally, we employ the Energy Pointing Game (EPG) metric, which evaluates how well the total attribution energy aligns with the target object without requiring threshold selection: $\text{EPG}(A, M) = \frac{\sum_{i,j} A_{i,j} M_{i,j}}{\sum_{i,j} A_{i,j}} \times 100$. This metric measures the percentage of the total attribution energy within the ground truth mask, providing a threshold-free assessment of localization accuracy. ♢**Dataset:** We evaluate these metrics on the Pascal VOC dataset, which provides object segmentation masks across 20 common object categories. For each image, we prompt the VLM with a simple classification query (e.g., `"Classify the image"`) and evaluate the attribution map against the ground truth segmentation masks of all objects present in the image. This setup ensures that the model must identify and localize multiple objects simultaneously, providing a more challenging and realistic evaluation scenario. We report IoU and EPG metrics as complementary metrics: while IoU assesses the spatial alignment of the attributions with ground truth segments, EPG measures how well the method concentrates attribution energy on relevant objects.High performance on both metrics indicates that a method can reliably identify regions important to the model's reasoning process.

Filler-words Filtering Evaluation To quantitatively evaluate the dual-filtering strategy, we construct PascalVOCQA, a specialized dataset derived from PascalVOC that enables quantitative assessment of filler word detection. For each image, a controlled natural language description is generated where objects are connected using predefined filler phrases (e.g., *"I see a {object 1} as well as a {object 2}"*). By constructing the expected answer in a systematic way, with predefined start phrases (e.g., "I see") and connecting phrases (e.g., "as well

Model	Method	ImageNet Pos(↑)	ImageNet Neg.(↓)	VQAv2 Pos(↑)	VQAv2 Neg.(↓)	(↓)sec. /img.
LlaVA-1.5	Attention	2.17	1.01	0.88	0.89	0.45
	GradCAM	1.43	1.1	0.91	**0.77**	0.66
	CheferCAM	2.06	1.03	**0.93**	0.78	6.20
	Attn×Grad	1.94	1.00	0.90	0.80	0.65
	Int.Grad	1.63	1.52	0.91	0.80	8.20
	RISE	1.24	0.99	0.92	0.78	11.8
	IIA	1.66	0.90	0.92	**0.77**	15.3
	DEX-AR	**2.31**	**0.96**	**0.93**	**0.77**	0.71
BakLlaVA-v1	Attention	11.47	4.36	0.98	0.86	0.45
	Rollout	6.13	3.12	0.95	0.90	0.51
	GradCAM	7.49	4.39	1.01	0.86	0.66
	CheferCAM	11.15	4.07	1.05	0.84	6.20
	Attn×Grad	12.6	3.74	1.06	0.86	0.65
	Int.Grad	13.50	12.77	1.03	0.84	8.20
	RISE	6.39	3.87	1.09	0.86	11.8
	IIA	11.08	3.77	1.02	0.86	15.3
	DEX-AR	**18.10**	**2.48**	**1.13**	**0.81**	0.71

Model	Method	ImageNet Pos(↑)	ImageNet Neg.(↓)	VQAv2 Pos(↑)	VQAv2 Neg.(↓)
Florence2	Attention	0.78	0.84	0.94	1.00
	Rollout	0.82	0.80	0.94	0.98
	CheferCAM	0.81	0.82	**0.97**	0.97
	Attn×Grad	0.80	0.81	0.95	0.99
	Int.Grad	0.84	0.80	0.76	**0.74**
	RISE	0.83	**0.78**	0.89	0.89
	IIA	0.79	0.80	0.95	0.99
	DEX-AR	**0.85**	**0.78**	**0.97**	0.95
PaliGemma	Attention	1.71	1.45	0.94	0.75
	Rollout	1.68	1.49	0.90	0.80
	GradCAM	1.74	**0.87**	0.85	0.87
	CheferCAM	1.71	1.44	0.94	**0.74**
	Attn×Grad	1.74	1.32	**0.96**	0.75
	Int.Grad	1.86	1.48	0.86	0.86
	RISE	1.63	1.04	0.95	0.79
	IIA	1.75	1.40	0.93	0.76
	DEX-AR	**2.71**	0.90	0.95	0.75

Table 1: Perturbation-based evaluation across different VLM architectures (Decoder-only, Encoder-Decoder, Prefix-Decoder) on ImageNet and VQAv2. Values represent Area Under the normalized perplexity Curve (AUC) for positive (↑) and negative (↓) perturbations. Runtime (sec/img) is averaged over 1k ImageNet images. LLaVA-1.5 and BakLLaVA share the same architecture and parameter count, hence identical runtimes. Rollout is omitted for LLaVA-1.5 as its attention implementation does not expose the required per-layer maps. TAM [24] results for LLaVA/BakLLaVA are in Supp. Sec. 17.

as"), we maintain token-level annotations of which parts of the response correspond to filler words versus content-bearing tokens. This automated construction provides ground truth annotations for the position of the "filler words" in the sequence, enabling quantitative evaluation of the filtering mechanism's ability to distinguish between tokens that convey visual content and those that serve linguistic functions.

4.2 Experimental Setup

Implementation Details. We evaluate the method on diverse state-of-the-art VLMs with varying architectural designs. We first consider decoder-only models: LLaVA-1.5 [25], which combines a CLIP ViT-L/14 [29] vision encoder with Vicuna, and BakLLaVA [34], which adopts a similar architecture but uses Mistral-7B [20] as its foundation. We also evaluate on PaliGemma [5], which employs a prefix language modeling approach allowing bidirectional attention between visual and textual tokens, and Florence-2 [41], which utilizes cross-attention mechanisms for multimodal integration. This diverse selection allows us to validate DEX-AR's effectiveness across different VLM architectures.

Baselines: We compare DEX-AR against a comprehensive set of explainability methods, including gradient-based (GradCAM [32], Attn×Grad [10], Integrated Gradient [36], CheferCAM [9] and IIA [3]), attention-based (Raw Attention, Attention Rollout [1]) and perturbation-based (RISE [28]), as detailed in Table 1. Implementation details are provided in the Supp. Matt. Sec. 9.

4.3 Perturbation Results

Table 1 compares DEX-AR against established explainability methods across different VLM architectures. Note that metric ranges vary across models due to base performance differences, making relative improvements more meaningful. ♢On ImageNet, DEX-AR shows superior performance across architectures, notably on BakLLaVA-v1 (AUC of 18.1 for positive perturbation, 5.5 points above Attn×Grad) while achieving better negative perturbation scores (2.48), indicating accurate identification of both relevant and irrelevant regions. On the more complex VQAv2 dataset, DEX-AR maintains higher positive perturbation scores (1.13 for BakLLaVA-v1, 0.95 for PaliGemma), though with more modest gains. On PaliGemma, while achieving the best positive scores (2.71), it performs similarly to baselines in negative perturbation (0.90 vs GradCAM's 0.87). Moreover, DEX-AR is significantly faster to traditional methods like Integrated Gradient, RISE, IIA or CheferCAM. DEX-AR's consistent performance across architectures (Decoder-only, Encoder-Decoder, and Prefix-Decoder) validates its approach for understanding VLMs' visual reasoning.

Table 2: Segmentation Performance on PascalVOC. Comparison of different methods on four VLM architectures using soft-IoU and IoU, and EPG (Energy Pointing Game).

Model	Method	soft-IoU(↑)	IoU(↑)	EPG(↑)
LlaVA-1.5	Attention	2.00	19.10	16.00
	Rollout	2.54	19.59	11.41
	GradCAM	<u>10.20</u>	<u>28.90</u>	19.30
	CheferCAM	1.60	21.01	17.10
	Attn×Grad	5.10	24.20	<u>26.60</u>
	DEX-AR	**17.70**	**36.34**	**27.75**
BakLlaVA-v1	Attention	2.04	19.56	16.84
	Rollout	3.30	20.14	14.67
	GradCAM	<u>8.19</u>	<u>24.36</u>	24.76
	CheferCAM	1.62	21.07	17.07
	Attn×Grad	5.09	24.23	<u>26.32</u>
	DEX-AR	**17.00**	**35.85**	**26.33**
Florence2	Attention	8.37	21.20	20.51
	Rollout	5.67	26.59	21.34
	CheferCAM	<u>9.68</u>	21.16	18.87
	Attn×Grad	9.35	<u>23.92</u>	<u>23.15</u>
	DEX-AR	**17.10**	**32.48**	**24.46**
PaliGemma	Attention	4.11	20.38	16.55
	GradCAM	<u>8.44</u>	22.55	**26.95**
	CheferCAM	4.61	21.05	18.02
	Attn×Grad	6.55	<u>23.32</u>	<u>23.52</u>
	DEX-AR	**15.26**	**23.55**	20.43

4.4 Segmentation Results

Table 2 compares the localization performance using the *aggregated sequence-level map E*. On LLaVA-1.5, DEX-AR achieves a soft-IoU of 17.70%, IoU of 36.34%, and EPG of 27.75%, representing relative improvements of 73.5%, 25.7%, and 4.3% respectively over the next best method. This advantage is consistently maintained across BakLLaVA and Florence-2, where DEX-AR outperforms all baselines by significant margins. The method's effectiveness is particularly evident in the soft-IoU metric, where it achieves scores approximately 2–3× higher than other approaches. On PaliGemma, DEX-AR leads in soft-IoU (15.26 vs 8.44) and IoU (23.55 vs 23.32) but trails GradCAM in EPG (20.43 vs 26.95). This is because GradCAM produces narrow, peaky maps that concentrate energy inside the object mask, scoring high on EPG, but lack spatial coverage. We therefore report three complementary metrics. Note that segmentation serves as a spatial sanity check. Our **primary** evaluation is perturbation-based faithfulness

(Tab. 1), which directly measures whether removing highlighted regions degrades the model's predictions, without assuming that attribution maps should resemble full segmentation masks.

4.5 Dynamic Filtering Ablations

Heads Filtering: We conduct an ablation study to evaluate the effectiveness of the attention head filtering mechanism, which aims to identify and prioritize attention heads that are more focused on visual information. We compare different filtering approaches: using only the maximum gradient value (max), selecting a percentage of top gradient values ($k_{\%}$), or averaging all gradients (avg). We evaluate these variants using two complementary metrics: Signal-to-Noise Ratio (SNR) and Mean Squared Error (MSE). SNR measures the ratio between the explanation intensity inside versus outside the ground truth mask: $\text{SNR} = 10 \cdot \log\left(\frac{\langle E, M\rangle / \|M\|_1}{\langle E, (1-M)\rangle / \|1-M\|_1}\right)$ where $E \in \mathbb{R}^{H\times W}$ is the explanation map, $M \in \{0,1\}^{H\times W}$ is the binary ground truth mask, $\langle\cdot,\cdot\rangle$ denotes the element-wise dot product, and $\|\cdot\|_1$ is the L1 norm. Higher SNR values indicate better localization as they reflect stronger activation inside the target region compared to the background. MSE quantifies the direct alignment between the explanation map and the ground truth mask, where lower values indicate better agreement.

Table 3: Head Filtering Ablation: Analysis of different filtering thresholds ($k_{\%}$) measuring Signal-to-Noise Ratio (SNR) and Mean Squared Error (MSE) on PascalVOC.

	LlaVA		BakLlaVA	
$k_{\%}$	SNR(↑)	MSE(↓)	SNR(↑)	MSE(↓)
-	1.64	0.33	5.27	0.15
max	**3.64**	**0.22**	**6.02**	**0.14**
0.05	3.35	0.24	4.13	0.17
0.1	3.09	0.25	4.13	0.19
0.2	2.45	0.28	3.18	0.22
0.3	1.98	0.31	2.36	0.25
0.4	1.66	0.33	1.92	0.26
0.5	1.45	0.34	1.62	0.27
0.6	1.29	0.35	1.42	0.28
0.7	1.20	0.35	1.30	0.29
0.8	1.16	0.36	1.20	0.29
0.9	1.12	0.36	1.14	0.29
avg	1.09	0.30	1.05	0.30

Results in Table 3 demonstrate that selective filtering of attention heads significantly improves explanation quality across both models. The max-based filtering achieves the best performance, improving SNR from 1.64 to 3.64 for LLaVA and from 5.27 to 6.02 for BakLLaVA, while also reducing MSE. Using more gradient values to compute the head scoring ($k_{\%}$) consistently degrades performance, suggesting that considering too many gradients introduces noise in the head selection process. This is further evidenced when using the average of all gradients (avg) to score the heads, which performs even worse than the baseline (no filtering), highlighting the importance of selective head filtering based on the most salient gradient signals.

Table 4: Filler words Filtering Ablation: Analysis of the impact of the different filtering strategies on the filtering of the filler words produced by the VLM on PascalVOC using LlaVA-1.5-7B.

Filtering Head	Filler	SNR(↑)	MSE(↓)	EPG(↑)
✗	✗	9.16	0.13	44.96
✓	✗	16.84	**0.10**	63.38
✗	✓	89.29	0.11	92.50
✓	✓	**96.12**	0.12	**95.04**

Filler words Filtering: Using PascalVOC-QA with LLaVA-1.5-7B, we assess the impact of both, head-level filtering, which weighs attention heads based on their visual focus, and filler word filtering, which identifies tokens primarily driven by linguistics.

We use three complementary metrics: Signal-to-Noise Ratio (SNR), measuring the ratio between attention given to non-filler words versus the attention given to filler words; Mean Squared Error (MSE) quantifying the pixel-wise deviation from ground truth masks; and Energy Point Game (EPG), evaluating the proportion of explanation energy that falls within the ground truth regions.

Results in Table 4 demonstrate the benefits of the dual-filtering approach. While head filtering alone improves performance across all metrics compared to the baseline (SNR: 9.16 → 16.84, EPG: 44.96% → 63.38%), the most substantial gains come from filler word filtering.

4.6 Qualitative Examples

Figure 4 compares DX-AR against to baselines across diverse queries, showing focused attributions that align with the objects being discussed, while comparable methods tend to generate more diffuse or scattered heatmaps. The "no Filtering" row particularly highlights the advantage of the dual-filtering approach, producing clean, object-centric attributions where other methods struggle to distinguish between relevant objects and backgrounds. Further qualitative results can be found in Supp. Matt. Section 15.

4.7 Runtime Analysis

Figure 3 reports wall-clock time as a function of the number of generated tokens T_a for DEX-AR and its batched variant on LLaVA-1.5-7B. Both variants scale linearly with T_a, consistent with the $O(T_a)$ complexity. Batched DEX-AR achieves up to **18**× speedup at T_a=100 while producing mathematically identical attribution maps. Even without batching, DEX-AR is already 8–21× faster than methods such as CheferCAM, RISE, IIA, and Integrated Gradients (Tab. 1), making it practical for typical VLM explainability tasks (short-answer VQA with 5–10 tokens, classification with 1–3 tokens, short captions with 15–20 tokens).

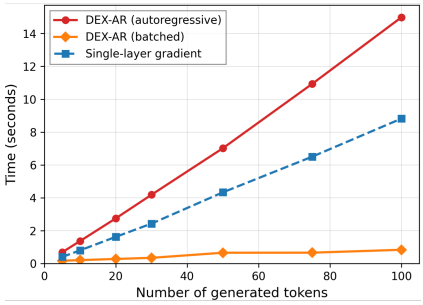

Fig. 3: Wall-clock time vs. number of generated tokens T_a on LLaVA-1.5-7B.

5 Conclusion

We presented DEX-AR, a novel explainability method for autoregressive VLMs that introduces dynamic head and token-level filtering mechanisms, demonstrat-

ing substantial improvements over existing methods across multiple architectures. The dual-filtering approach validates the importance of selective attention for accurate attribution, while providing a tool for better model understanding and facilitating the responsible deployment of AI systems.

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

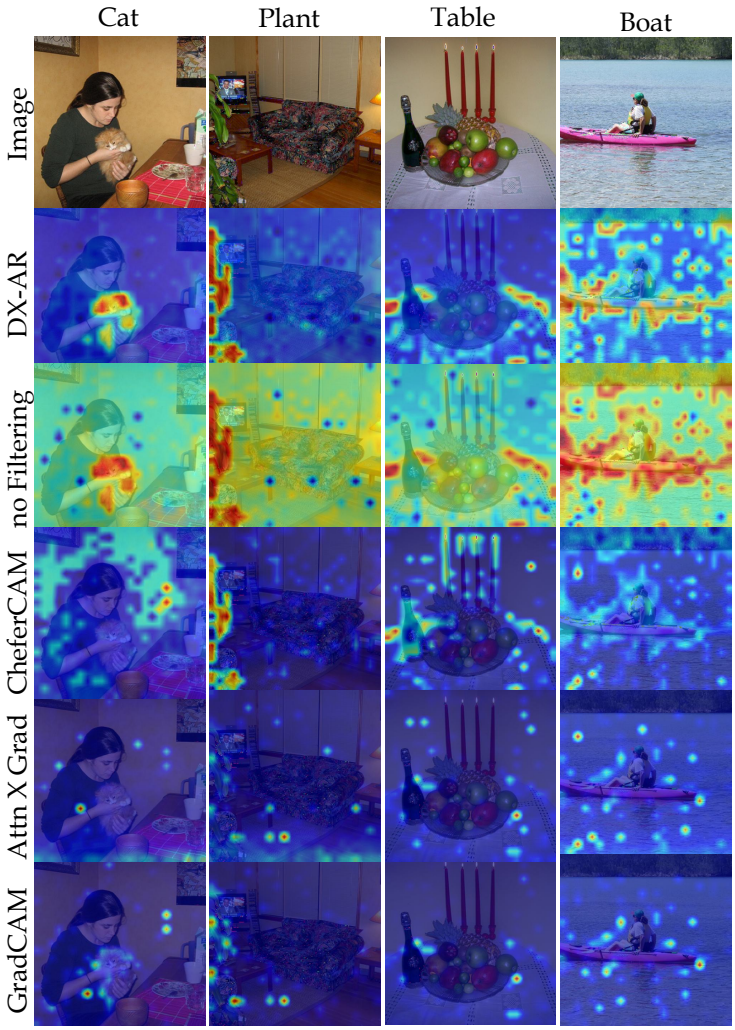

Fig. 4: Qualitative Comparison

References

1. Abnar, S., Zuidema, W.: Quantifying attention flow in transformers. ACL (2020)
2. Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F.L., Almeida, D., Altenschmidt, J., Altman, S., Anadkat, S., et al.: Gpt-4 technical report. arXiv preprint arXiv:2303.08774 (2023)
3. Barkan, O., Asher, Y., Eshel, A., Koenigstein, N., et al.: Visual explanations via iterated integrated attributions. In: ICCV (2023)
4. Belrose, N., Ostrovsky, I., McKinney, L., Furman, Z., Smith, L., Halawi, D., Biderman, S., Steinhardt, J.: Eliciting latent predictions from transformers with the tuned lens. arXiv preprint arXiv:2303.08112 (2023)
5. Beyer, L., Steiner, A., Pinto, A.S., Kolesnikov, A., Wang, X., Salz, D., Neumann, M., Alabdulmohsin, I., Tschannen, M., Bugliarello, E., et al.: Paligemma: A versatile 3b vlm for transfer. arXiv preprint arXiv:2407.07726 (2024)
6. Bousselham, W., Boggust, A., Chaybouti, S., Strobelt, H., Kuehne, H.: Legrad: An explainability method for vision transformers via feature formation sensitivity. In: ICCV (2025)
7. Brunner, G., Liu, Y., Pascual, D., Richter, O., Ciaramita, M., Wattenhofer, R.: On identifiability in transformers. ICLR (2020)
8. Buçinca, Z., Malaya, M.B., Gajos, K.Z.: To trust or to think: cognitive forcing functions can reduce overreliance on ai in ai-assisted decision-making. Proceedings of the ACM on Human-computer Interaction (2021)
9. Chefer, H., Gur, S., Wolf, L.: Generic attention-model explainability for interpreting bi-modal and encoder-decoder transformers. In: ICCV (2021)
10. Chefer, H., Gur, S., Wolf, L.: Transformer interpretability beyond attention visualization. In: CVPR (2021)
11. Clark, K., Khandelwal, U., Levy, O., Manning, C.D.: What does BERT look at? an analysis of BERT's attention. ACL (2019)
12. Ding, Y., Liu, Y., Luan, H., Sun, M.: Visualizing and understanding neural machine translation. ACL (2017)
13. Ferrando, J., Gállego, G.I., Alastruey, B., Escolano, C., Costa-jussà, M.R.: Towards opening the black box of neural machine translation: Source and target interpretations of the transformer. ACL (2022)
14. Galassi, A., Lippi, M., Torroni, P.: Attention, please! a critical review of neural attention models in natural language processing. ArXiv (2019)
15. Gandelsman, Y., Efros, A.A., Steinhardt, J.: Interpreting clip's image representation via text-based decomposition. In: ICLR (2024)
16. Gandelsman, Y., Efros, A.A., Steinhardt, J.: Interpreting the second-order effects of neurons in clip. In: ICLR (2025)
17. Geva, M., Caciularu, A., Wang, K., Goldberg, Y.: Transformer feed-forward layers build predictions by promoting concepts in the vocabulary space. In: Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing (2022)
18. Goyal, Y., Khot, T., Summers-Stay, D., Batra, D., Parikh, D.: Making the v in vqa matter: Elevating the role of image understanding in visual question answering. In: CVPR (2017)
19. Guan, T., Liu, F., Wu, X., Xian, R., Li, Z., Liu, X., Wang, X., Chen, L., Huang, F., Yacoob, Y., et al.: Hallusionbench: An advanced diagnostic suite for entangled language hallucination and visual illusion in large vision-language models. arXiv preprint arXiv:2310.14566 (2023)

20. Jiang, A.Q., Sablayrolles, A., Mensch, A., Bamford, C., Chaplot, D.S., Casas, D.d.l., Bressand, F., Lengyel, G., Lample, G., Saulnier, L., et al.: Mistral 7b. arXiv preprint arXiv:2310.06825 (2023)
21. Karpathy, A., Johnson, J., Fei-Fei, L.: Visualizing and understanding recurrent networks. ICLR (workshop) (2016)
22. Li, J., Li, D., Savarese, S., Hoi, S.: Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In: ICML. PMLR (2023)
23. Li, J., Li, D., Xiong, C., Hoi, S.: Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In: ICML. PMLR (2022)
24. Li, Y., Wang, H., Ding, X., Wang, H., Li, X.: Token activation map to visually explain multimodal llms. ICCV (2025)
25. Liu, H., Li, C., Li, Y., Lee, Y.J.: Improved baselines with visual instruction tuning. In: CVPR (2024)
26. Liu, H., Li, C., Wu, Q., Lee, Y.J.: Visual instruction tuning. Advances in neural information processing systems (2024)
27. nostalgebraist: Interpreting gpt: the logit lens. https://www.lesswrong.com/posts/AcKRB8wDpdaN6v6ru/interpreting-gpt-the-logit-lens (2020)
28. Petsiuk, V., Das, A., Saenko, K.: Rise: Randomized input sampling for explanation of black-box models. In: BMVC (2018)
29. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. In: ICML. PMLR (2021)
30. Roslyn, S.A., A, N., R, V.S.: Enhancing accessibility for visually impaired users: A blip2-powered image description system in tamil. 2024 International Conference on Advances in Data Engineering and Intelligent Computing Systems (ADICS) (2024)
31. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al.: Imagenet large scale visual recognition challenge. IJCV (2015)
32. Selvaraju, R.R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., Batra, D.: Grad-cam: visual explanations from deep networks via gradient-based localization. IJCV (2020)
33. Serrano, S., Smith, N.A.: Is attention interpretable? ACL (2019)
34. SkunkworksAI: Bakllava: Vision-language model. `https://github.com/SkunkworksAI/BakLLaVA` (2024)
35. Springenberg, J., Dosovitskiy, A., Brox, T., Riedmiller, M.: Striving for simplicity: The all convolutional net. In: ICLR (workshop track) (2015)
36. Sundararajan, M., Taly, A., Yan, Q.: Axiomatic attribution for deep networks. In: ICML. PMLR (2017)
37. Team, G.: Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. arXiv preprint arXiv:2403.05530 (2024)
38. Templeton, A., Conerly, T., Marcus, J., Lindsey, J., Bricken, T., Chen, B., Pearce, A., Citro, C., Ameisen, E., Jones, A., Cunningham, H., Turner, N.L., McDougall, C., MacDiarmid, M., Tamkin, A., Durmus, E., Hume, T., Mosconi, F., Freeman, C.D., Sumers, T.R., Rees, E., Batson, J., Jermyn, A., Carter, S., Olah, C., Henighan, T.: Scaling monosemanticity: Extracting interpretable features from claude 3 sonnet. Anthropic (2024)
39. Vaswani, A.: Attention is all you need. NIPS (2017)

40. Wiegreffe, S., Pinter, Y.: Attention is not not explanation. ACL (2019)
41. Xiao, B., Wu, H., Xu, W., Dai, X., Hu, H., Lu, Y., Zeng, M., Liu, C., Yuan, L.: Florence-2: Advancing a unified representation for a variety of vision tasks. In: CVPR (2024)
42. Yang, B., He, L., Liu, K., Yan, Z.: Viassist: Adapting multi-modal large language models for users with visual impairments. arXiv preprint arXiv:2404.02508 (2024)
43. Zhao, H., Chen, H., Yang, F., Liu, N., Deng, H., Cai, H., Wang, S., Yin, D., Du, M.: Explainability for large language models: A survey. ACM Transactions on Intelligent Systems and Technology (2024)
44. Zhao, Y., Zhang, Y., Xiang, R., Li, J., Li, H.: Vialm: A survey and benchmark of visually impaired assistance with large models. arXiv preprint arXiv:2402.01735 (2024)
45. Zini, J.E., Awad, M.: On the explainability of natural language processing deep models. ACM Computing Surveys (2022)