Attention-Guided Saliency Maps for Interpreting Visualization Literacy in VLMs

Maeve Hutchinson*†
City St George's,
University of London

Abderrahmane Wassim Mehdaoui*
City St George's,
University of London

Pranava Madhyastha†
City St George's, University of London;
The Alan Turing Institute

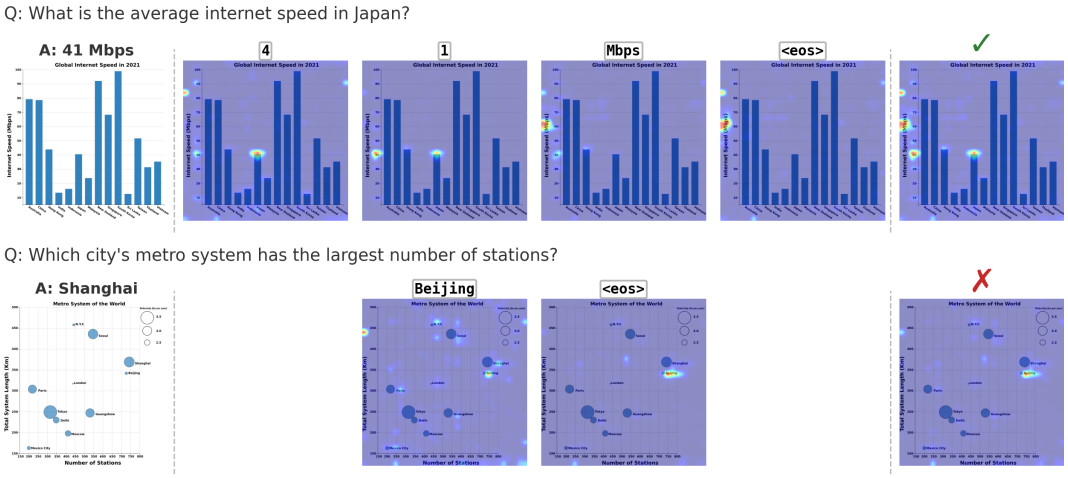

Figure 1: Our token-to-patch attention-guided saliency maps for two Mini-VLAT questions answered by ChartGemma. Each row shows, from left to right: the original chart image with the ground-truth answer (A), the saliency map for each generated token, and the normalized mean aggregation across all tokens.

Abstract

Understanding how vision-language models (VLMs) interpret data visualizations remains an open problem, and is increasingly important as these models are used for analytical tasks where reliable reasoning is essential. We introduce a lightweight, diagnostic saliency map method tailored for text generation over images using transformer models, the current state-of-the-art models in visualization interpretation. Our approach aggregates the language model's attention over the visual tokens across all heads and layers, then maps this attention back onto the vision encoder's patch grid to localise it over the image, producing a direct correspondence between each generated answer token and the image regions it attended to. This yields fast, gradient-free saliency maps that expose how VLMs allocate focus across visual elements during answer generation, enabling inspection of whether model attention aligns with semantically relevant components. We evaluate our approach using a deletion metrics which validates the causal faithfulness of our saliency maps to the model's behavior.

Index Terms: Vision-language models, Saliency, Explainable AI

1 Introduction

Vision-language models (VLMs) are increasingly being deployed as analytical engines for data visualization-related tasks, such as querying charts, summarising trends, and extracting values from complex figures. However, the fundamental question of whether these models actually focus on the correct visual elements when answering remains less rigorously addressed. A model that consistently returns the correct number whilst attending to the wrong bar can hardly be considered a reliable tool.

Current benchmarks for Chart Question Answering (CQA) [17, 19, 12] measure only textual correctness, leaving this blind spot unaddressed. Recent research reveals that state-of-the-art VLMs maintain surprisingly high accuracy even when chart images are corrupted or removed entirely [10]. This indicates that parametric memory and superficial textual biases often serve as shortcuts, underpinning many superficially correct answers [5]. Accuracy alone, therefore, is an insufficient proxy for visual reasoning.

In light of these limitations, we present a lightweight, gradient-free saliency method designed specifically for text generation over chart images in transformer-based VLMs. For each generated output token, our approach aggregates the language model's attention over the visual tokens across all heads and layers, and maps the resulting distribution onto the vision encoder's patch grid. This yields a per-token saliency map that reveals *which pixels the model attended to* whilst producing each word of its answer without any gradient computation, architectural modification, or additional inference pass beyond standard generation

In this paper we contribute a token-level, attention-guided saliency extraction method tailored to the ViT-LLM architectures prevalent in contemporary VLMs, producing spatially coherent maps over chart images during free-form answer generation and a deletion-based faithfulness evaluation on Mini-VLAT [20], demonstrating the causal validity of our maps by showing that the removal of top-ranked pixels induces a rapid collapse in model accuracy.

2 Related Work

Research into visualization literacy has historically focused on evaluating people's ability to read and interpret data visualizations. Instruments such as the VLAT [13] and CALVI [8] have been created to do this. Recent work has repurposed these datasets to evaluate

*Both authors contributed equally to this work.
†Correspondence: {maeve.hutchinson, pranava.madhyastha}@city.ac.uk

the state-of-the-art VLMs [2, 6, 10]. A parallel line of work has explored large-scale benchmarking datasets for evaluating VLMs on tasks such as chart question answering [17, 19, 12] and chart captioning [26]. These benchmarks have spurred the development of specialized architectures, including ChartGemma [18] and ChartLlama [9], which are fine-tuned specifically to handle the structural syntax of visualizations. Furthermore, recent work has also focused on identifying misleading visualizations [16]; or assessing the ability to explain or justify visual encoding choices [11]. However, these benchmarks only evaluate whether a model outputs the correct answer, without addressing whether it arrived there by using the correct visual evidence. Recent studies have shown that some VLMs maintain performance even when chart images are corrupted or removed [10]. This makes accuracy alone a weak indicator of visual reasoning, and motivates the use of explainability techniques for interpreting model behavior.

Explainability has long been a central research area in computer vision and deep learning, particularly saliency-based methods that reveal which input features in an image drive a model's decision. A large catalogue of saliency methods exist. Model-agnostic approaches, such as LIME [23], approximate local decision boundaries via perturbing the image, whereas gradient-dependent approaches, such as the Grad-CAM [25], trace calculate the importance of pixels in the image through the network's computational graph. Prior work has applied AG-CAM [14], a version of Grad-CAM adapted for the transformer architecture, to visualization literacy explainability [6], while others have explored aligning model-generated saliency maps with human gaze patterns [24].

Despite their popularity, existing saliency methods rely on assumptions that do not hold for modern VLMs. Techniques such as Grad-CAM [25] were developed for CNNs, where spatial locality and hierarchical feature maps provide a meaningful basis for gradient-based attribution. Transformer-based architectures do not share these structural priors, instead representing images as patch tokens and propagating information globally through attention. Consequently, applying canonical saliency methods to ViTs can yield misleading explanations [4]. Moreover, most saliency methods are designed for classification tasks on natural images, whereas visualization understanding in VLMs typically involves open-ended text generation. This means that explainability techniques for VLMs must reflect both the structure of the underlying model and the task being completed.

In this work, we specifically address this limitation. Our framework introduces an attention-guided approach that leverages the intrinsic token-to-token attention patterns in these models, providing saliency maps that align with the mechanisms through which VLMs process visual inputs and generate open-ended answers that are grounded in the task of chart question answering.

3 Preliminaries: Vision-Language Alignment

In this section, we cover the necessary background for the exposition of our proposed methodology. We focus on the prevalent paradigm used by VLMs popular within visualization research, such as LLaVA [15] and ChartGemma [18]. These architectures couple a pre-trained visual encoder with a language model via a trainable projection adapter. In LLaVA, both the visual encoder and the LLM are frozen during training and only the adapter is updated; in ChartGemma, the LLM is additionally fine-tuned on chart-specific data.

Visual Encoding and Discretization Let the input be an image $\mathbf{I} \in \mathbb{R}^{H_{\text{img}} \times W_{\text{img}} \times 3}$, where H_{img} and W_{img} are assumed to be exact multiples of the patch size P (achieved in practice via resizing or padding prior to encoding). The visual encoder transforms this signal into a sequence of patch embeddings. A Vision Transformer (ViT) [7, 22] serves as the visual backbone, treating the image as a flat sequence of tokens rather than imposing the local spatial inductive bias of CNNs. The image is partitioned into fixed-size patches $\mathbf{x}_p \in \mathbb{R}^{P \times P \times 3}$, which are flattened and linearly projected into the encoder's latent space. A CLS token $\mathbf{x}_{\text{cls}}$ is prepended to this sequence. The encoder then produces a sequence of embeddings:

$$\mathbf{Z}_v = f_{\text{enc}}(\mathbf{I}) \in \mathbb{R}^{(N_p+1) \times D_v}, \tag{1}$$

where $N_p = (H_{\text{img}} \cdot W_{\text{img}})/P^2$ is the number of patches, the additional token accounts for the CLS token, and D_v is the encoder's hidden dimension. In our saliency method we consider only the N_p patch tokens, excluding the CLS token, as these correspond directly to spatial regions of the image.

Cross-Modal Alignment via Projection The visual embeddings $\mathbf{Z}_v \in \mathbb{R}^{(N_p+1) \times D_v}$ and the LLM's token embeddings occupy disjoint spaces; in general $D_v \neq D_{\text{llm}}$, so direct concatenation is not possible. A trainable projection adapter ϕ maps the patch embeddings into the LLM's embedding space. In LLaVA, ϕ is a Multi-Layer Perceptron:

$$\mathbf{H}_v = \phi(\mathbf{Z}_v) \in \mathbb{R}^{N_p \times D_{\text{llm}}}. \tag{2}$$

The resulting visual tokens $\mathbf{H}_v$ act as a *visual prefix*, conditioning subsequent text generation on the image content.

Autoregressive Generation and Attention Dynamics The input to the LLM is a unified multimodal sequence $\mathbf{X}_{\text{in}}$, formed by concatenating the projected visual tokens $\mathbf{H}_v$ with the tokenised user query $\mathbf{X}_q \in \mathbb{R}^{N_q \times D_{\text{llm}}}$:

$$\mathbf{X}_{\text{in}} = [\mathbf{H}_v; \mathbf{X}_q] \in \mathbb{R}^{T \times D_{\text{llm}}}, \tag{3}$$

where $T = N_p + N_q$ is the total sequence length.

The LLM processes $\mathbf{X}_{\text{in}}$ through a stack of Transformer layers, each employing Multi-Head Self-Attention (MSA). For attention head h in layer l, the queries, keys, and values are obtained via learned linear projections of the input: $\mathbf{Q} = \mathbf{X}_{\text{in}}\mathbf{W}^Q$, $\mathbf{K} = \mathbf{X}_{\text{in}}\mathbf{W}^K$, $\mathbf{V} = \mathbf{X}_{\text{in}}\mathbf{W}^V$. The attention matrix is then:

$$\mathbf{A}^{(l,h)} = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d_k}}\right) \in \mathbb{R}^{T \times T}, \tag{4}$$

where d_k is the per-head key dimension. The element $A_{i,j}^{(l,h)}$ quantifies how much token i attends to token j in layer l, head h.

Answer generation is autoregressive: at each step t, the model produces the next token y_t conditioned on the projected visual tokens and all previously generated text,

$$p(y_t \mid \mathbf{H}_v, y_{<t}) = \text{LLM}(\mathbf{H}_v, y_{<t}). \tag{5}$$

It is within the attention weights $\mathbf{A}^{(l,h)}$, specifically the attention from the token at position t back to the visual prefix $\mathbf{H}_v$, that we seek to locate the visual evidence underlying the model's decisions. The following section describes our methodology.

4 Methodology: Attention-Guided Saliency Extraction

Our objective is to derive a spatial saliency map S_t that quantifies the influence of specific image regions (patches) on the generation of a target text token y_t. We rely on a perturbation-free approach based on attention aggregation [1], tailored for multimodal sequences.

Notation. We define the structural parameters of the model as follows: $N_l \in \mathbb{N}$ is the total number of transformer layers; $N_h \in \mathbb{N}$ is the number of attention heads per layer; $N_p \in \mathbb{N}$ is the number of vision tokens (patches); $N_q \in \mathbb{N}$ is the number of question (text prompt) tokens; $T = N_p + N_q \in \mathbb{N}$ is the length of the input sequence (the visual prefix followed by the question); and $G = \sqrt{N_p}$ is the grid dimension of the vision encoder, assuming a square $G \times G$ patch

layout. The input tokens occupy sequence positions $\{0,\dots,N_p-1\}$ for the visual prefix and $\{N_p,\dots,T-1\}$ for the question, with generated tokens following at positions $\geq T$. We write $y_1,\dots,y_M$ for the M generated answer tokens, and let $t \in \{1,\dots,M\}$ index the *generated* text token we wish to explain (equivalently, the decoding step that produces y_t).

Attention Aggregation. Transformers distribute information processing across N_h heads, which often specialise in distinct linguistic features. Relying on a single head or the final layer alone yields noisy and incomplete attribution. To obtain a robust signal of visual importance, we aggregate attention weights across the full depth and width of the (language related) network. Let $\mathbf{A}^{(\mathbf{t},\mathbf{l},\mathbf{k})}$ denote the raw attention matrix of head k in layer l at decoding step t. We first average over heads to neutralise head-specific biases,

$$\bar{\mathbf{A}}^{(t,l)} = \frac{1}{N_h}\sum_{k=1}^{N_h} \mathbf{A}^{(t,l,k)}, \tag{6}$$

and then over layers to obtain a single attention map for that step,

$$\mathbf{A}^{(t)}_{\text{global}} = \frac{1}{N_l}\sum_{l=1}^{N_l} \bar{\mathbf{A}}^{(t,l)}. \tag{7}$$

While more complex rollout strategies exist (consider [1] for a thorough exposition), we find that mean aggregation provides a stable and inexpensive signal of visual importance.

Vision Token Extraction. Rows of $\mathbf{A}^{(t)}_{\text{global}}$ are indexed by sequence position, not by generated-token index. Let r_t denote the query position whose hidden state predicts y_t: $r_t = T+t-1$. We extract this query's attention over the visual prefix, yielding the raw attention vector:

$$\mathbf{w}_t = \mathbf{A}^{(t)}_{\text{global}}\big[r_t,\, 0{:}N_p\big] \in \mathbb{R}^{N_p}. \tag{8}$$

Since $\mathbf{A}^{(t)}_{\text{global}}$ is row-normalised over the attended keys, $\mathbf{w}_t$ does not in general sum to 1 over the visual tokens alone. We therefore re-normalise to obtain a distribution over patches:

$$\tilde{\mathbf{w}}_t = \frac{\mathbf{w}_t}{\sum_{j=0}^{N_p-1} w_{t,j}},, \tag{9}$$

so that $\tilde{\mathbf{w}}_t$ represents the model's relative allocation of attention across image patches, conditional on attending to the visual prefix.

Topological Reconstruction. The vector $\tilde{\mathbf{w}}_t$ is defined over a flattened patch sequence. We recover the spatial structure by mapping each patch index i to its position (r,c) in the $G \times G$ grid:

$$r = \left\lfloor \frac{i}{G} \right\rfloor, \qquad c = i \bmod G, \tag{10}$$

and reshaping to form the coarse 2D attention map:

$$\mathbf{M}_t = \text{Reshape}(\tilde{\mathbf{w}}_t, (G,G)) \in \mathbb{R}^{G\times G}. \tag{11}$$

This map represents the raw attention per patch.

Normalisation and Spatial Upsampling. $\mathbf{M}_t$ is defined at patch resolution, which is substantially coarser than the input image. To produce a heatmap suitable for overlaying on the original visualization, we first apply max-normalisation to $\mathbf{M}_t$, rescaling values to $[0,1]$ relative to the most attended patch. This is purely a display-level operation (for rendering better visualizations) that preserves the relative importance of regions across tokens while removing dependence on absolute attention magnitude:

$$\hat{\mathbf{M}}_t = \frac{\mathbf{M}_t}{\max_{r,c} M_t[r,c]}. \tag{12}$$

We then upsample $\hat{\mathbf{M}}_t$ to the original image resolution $(H_{\text{img}}, W_{\text{img}})$ via bilinear interpolation, which estimates the saliency value at each pixel (x,y) as a weighted average of the four nearest patch values:

$$S_t(x,y) = \sum_{i=0}^{G-1}\sum_{j=0}^{G-1} \hat{M}_t[i,j]\, K\left(\frac{xG}{W_{\text{img}}} - j,\ \frac{yG}{H_{\text{img}}} - i\right), \tag{13}$$

where $K(a,b) = \max(0,1-|a|)\cdot\max(0,1-|b|)$ is the bilinear interpolation kernel. The resulting map S_t provides a pixel-wise approximation of the VLM's visual attention during generation of token y_t.

5 Evaluation

5.1 Experimental Setup and Baselines

To validate our method, we first reproduce some of the results Dong and Crisan [6], who evaluate the visualization literacy of VLMs using Mini-VLAT [20] and apply AG-CAM [14] to probe the model behavior over input tokens. Specifically, we reproduce the test results obtained by two models (ChartGemma and LLaVA) to confirm that our setup aligns with previous results and to provide a baseline of their visualization literacy, from which we can then validate the faithfulness of our maps. Below describes the experimental setup and results. We then extend their study to use the full VLAT [13].

Datasets We evaluate our attention-guided saliency extraction method using Mini-VLAT [20] and VLAT [13], but allowing the model to answer with open-ended text rather than multiple choices.

Models Two VLMs are used. LLaVA-1.5-7B is a general-purpose multimodal model using a CLIP-style ViT encoder fused with a LLaMA-based language model [15]. ChartGemma is a chart-focused VLM fine-tuned specifically for chart reasoning tasks, and uses a Gemma-family language model coupled with a ViT variant adapted for chart structure [18]. The models differ in both their visual and linguistic backbones so they provide a meaningful comparison between a general multimodal system and a specialised chart-oriented one. For each question–image pair, the model receives the chart and the question in a standard prompt. The model then generates free-form text autoregressively. During generation, we compute a saliency map for the final token(s) that constitute the model's predicted answer using the method detailed in section 4.

Evaluation Metrics Answer correctness is measured using relaxed accuracy, as is standard to evaluate results in chart question answering tasks [17, 19]. Rather than enforcing strict string matching, the VLM-generated answer is considered correct if the correct answer appears anywhere within the generated token sequence, and additionally for quantitative numerical answers, answers are considered correct if they are within 5% of the ground truth answer. We report the accuracy over all samples as a percentage.

Results On MiniVLAT, ChartGemma achieves 58.3% accuracy on this setup, which matches exactly the result reported by [6]. LLaVA-1.5 achieves 25%, which is slightly more than the 20% reported by [6], but within an acceptable range given that VLMs are not deterministic. These results confirm the validity of our generation setup. On VLAT, ChartGemma achieves 34.0% and LLaVA-1.5 22.6%. These values confirm that ChartGemma's finetuning increases performance on visualization literacy tests.

5.2 Deletion-Based Faithfulness Evaluation

To evaluate the faithfulness of our saliency maps, we employ a deletion test, a causal metric for evaluating saliency explanations [21]. This methodology systematically removes image regions identified as most salient by a given attribution method and measures the resulting degradation in model performance. This provides a quantitative assessment of whether the highlighted regions genuinely contribute to the model's performance, and thus whether the saliency maps themselves are faithful representations of the model's focus. We run these tests on both our maps and AG-CAM.

Dataset and Models We use the same setup as 5.1, using VLAT. However, given that deletion measures *causal* faithfulness, we only use samples which the model answered correctly in the vanilla setting. Additionally, we exclude questions which have a True/False answer, as the probability of the model "guessing" the right answer is high, which interferes with the meaningful signal from the deletion testing. Given this, we only perform deletion evaluations with ChartGemma, as LLaVA did not answer enough questions correctly to provide a representative sample. Thus, we run the following evaluations using 13 samples from VLAT and ChartGemma. Sample sizes of this order are consistent with prior deletion-based evaluations.[3, 21].

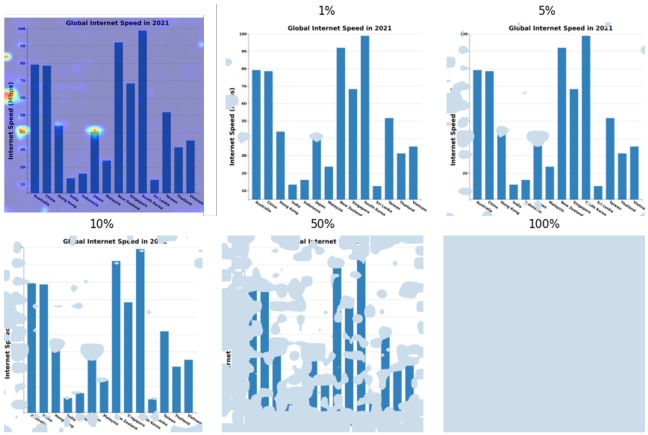

Figure 2: Progressive pixel deletion applied to a visualization from VLAT using our saliency map method. The top-left panel shows the aggregate saliency map; subsequent panels show the chart after removing the top $k\%$ of pixels ranked by saliency.

Deletion Methodology For each question–chart pair in the evaluation set, we first obtain an aggregate saliency map by running the model on the unmodified question and image.

For our approach, we do this by first extract the attention weights over image patch tokens at each decoding step, as described in 4. We then L1-normalize each token's attention distribution to ensure that tokens with diffuse attention patterns contribute equally to the aggregate map rather than being dominated by tokens with sharply peaked distributions. Finally, we average across all generated tokens to produce a single spatial heatmap representing the entire generated answer. The rightmost heatmaps in figure 1 show this.

For AG-CAM, we perform a forward pass followed by backpropagation from the maximum logit, compute the element-wise product of the ReLU-activated gradients and the attention weights at each layer, sum across attention heads, and accumulate across all transformer layers to obtain the final attribution map. It is worth noting that in contrast to our approach, AG-CAM operates over *input* tokens rather than *output* tokens generated by the model, and so the saliency maps do not represent focus during answering, but parts of the image that are salient to the question.

Given the aggregate saliency map, we rank all pixels by their attributed importance and progressively replace the top $k\%$ most salient pixels with the channel-wise mean color of the original image. We choose mean-color infill to avoid introducing out-of-distribution artifacts that solid black or gray patches might cause [21]. We do this for all values of $k \in \{0, 1, 2, ..., 100\}$. Figure 2 shows an example our saliency map and corresponding degraded images at different values of k. At each deletion level, we prompt the model with the original question and degraded image, and record whether the model's answer remains correct. We report accuracy as a percentage as a function of the percentage of pixels deleted, producing an accuracy-versus-deletion curve. As a control, we also repeat the procedure with randomly selected pixels.

Evaluation Metrics We evaluate each accuracy-versus-deletion curve using the area under the curve (AUC) [21]. A lower AUC for saliency-based deletion relative to the random baseline indicates that the saliency method successfully identifies pixels that are causally relevant to the model's output: removing them degrades performance more rapidly than removing an equivalent number of arbitrarily chosen pixels.

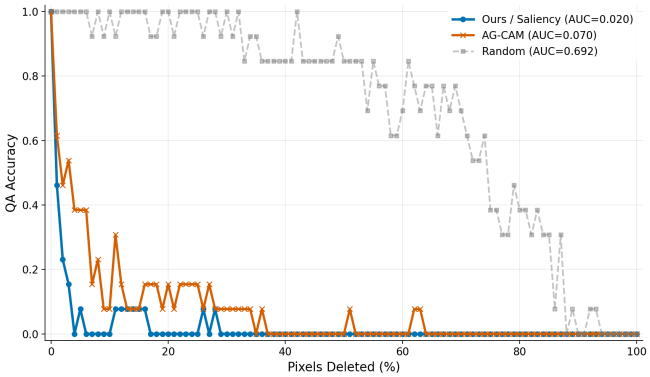

Figure 3: Deletion test results for ChartGemma on 13 VLAT samples. Accuracy is plotted against the percentage of pixels deleted for our saliency method, AG-CAM and random pixel deletion.

Results Figure 3 presents the results of the progressive deletion test for ChartGemma. Both our method and AG-CAM cause a rapid collapse in accuracy compared to random deletion, confirming that both methods identify genuinely informative regions rather than arbitrary pixels. Our method achieves an AUC of 0.020, outperforming AG-CAM at an AUC of 0.070, indicating that our approach more faithfully reflects the focus of the model. The steepness of both saliency curves also reveals that chart question answering is highly spatially concentrated: the model's ability to answer correctly depends on a very small fraction of the image, typically the specific axis label(s) or mark(s) that encode the queried value. Once these critical pixels are removed, performance collapses almost entirely, even though the majority of the visualization remains intact.

6 Conclusion

We have presented a lightweight, gradient-free saliency extraction method for interpreting how vision-language models process data visualizations during text generation. By aggregating attention weights across the language model's layers and mapping them back through the vision encoder's patch topology, our approach produces per-token and aggregate saliency maps that reveal which visual elements the model relies on when generating each part of its answer.

Our deletion-based evaluation confirms that these maps are causally faithful, achieving this with a lower AUC than AG-CAM, while requiring no gradient computation, no architectural modification, and no additional inference passes beyond standard generation, and importantly being caculated over *output* tokens rather than *input*. This makes it practical for deployment as a real-time diagnostic tool alongside model predictions.

More broadly, this work contributes to an emerging need within the visualization community: as VLMs are increasingly used for analytical tasks, we require interpretability tools that are tailored to both the architecture of these models and the structured nature of data visualizations. Saliency maps that faithfully reveal whether a model is attending to the correct regions of a visualization provide a necessary complement to accuracy-based evaluation, helping identify not just when a model fails, but why.

Supplementary Material

The dataset is code available at https://github.com/maevehutch/token-patch-saliency.

Acknowledgments

This work was supported, in part, by the Alan Turing Institute under Fundamental Research (Project No. PP00029).

References

[1] S. Abnar and W. Zuidema. Quantifying attention flow in transformers. In D. Jurafsky, J. Chai, N. Schluter, and J. Tetreault, eds., *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 4190–4197. Association for Computational Linguistics, Online, July 2020. doi: 10.18653/v1/2020.acl-main.385 2, 3

[2] A. Bendeck and J. Stasko. An Empirical Evaluation of the GPT-4 Multimodal Language Model on Visualization Literacy Tasks. *IEEE Transactions on Visualization and Computer Graphics*, pp. 1–11, 2024. Conference Name: IEEE Transactions on Visualization and Computer Graphics. doi: 10.1109/TVCG.2024.3456155 2

[3] A. Boggust, H. Suresh, H. Strobelt, J. Guttag, and A. Satyanarayan. Saliency Cards: A Framework to Characterize and Compare Saliency Methods. In *Proceedings of the 2023 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '23, pp. 285–296. Association for Computing Machinery, New York, NY, USA, June 2023. doi: 10.1145/3593013.3593997 4

[4] H. Chefer, S. Gur, and L. Wolf. Transformer interpretability beyond attention visualization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 782–791, 2021. doi: 10.1109/CVPR46437.2021.00084 2

[5] C. Dancette, R. Cadène, D. Teney, and M. Cord. Beyond question-based biases: Assessing multimodal shortcut learning in visual question answering. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 1554–1563, 2021. doi: 10.1109/ICCV48922.2021.00160 1

[6] L. Dong and A. Crisan. Probing the visualization literacy of vision language models: The good, the bad, and the ugly, 2026. doi: 10.1109/TVCG.2025.3634791 2, 3

[7] A. Dosovitskiy. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 2

[8] L. W. Ge, Y. Cui, and M. Kay. CALVI: Critical Thinking Assessment for Literacy in Visualizations. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, CHI '23, pp. 1–18. Association for Computing Machinery, New York, NY, USA, Apr. 2023. doi: 10.1145/3544548.3581406 1

[9] Y. Han, C. Zhang, X. Chen, X. Yang, Z. Wang, G. Yu, B. Fu, and H. Zhang. Chartllama: A multimodal llm for chart understanding and generation. *arXiv preprint arXiv:2311.16483*, 2023. 2

[10] J. Hong, C. Seto, A. Fan, and R. Maciejewski. Do llms have visualization literacy? an evaluation on modified visualizations to test generalization in data interpretation. *IEEE Transactions on Visualization and Computer Graphics*, 2025. 1, 2

[11] M. Hutchinson, R. Jianu, A. Slingsby, J. Wood, and P. Madhyastha. Capturing visualization design rationale. In *2025 IEEE Visualization and Visual Analytics (VIS)*, pp. 231–235, 2025. doi: 10.1109/VIS60296.2025.00052 2

[12] M. Hutchinson, R. Jianu, A. Slingsby, J. Wood, and P. Madhyastha. Chart Question Answering from Real-World Analytical Narratives. In J. Zhao, M. Wang, and Z. Liu, eds., *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 4: Student Research Workshop)*, pp. 760–773. Association for Computational Linguistics, Vienna, Austria, July 2025. doi: 10.18653/v1/2025.acl-srw.50 1, 2

[13] S. Lee, S.-H. Kim, and B. C. Kwon. VLAT: Development of a Visualization Literacy Assessment Test. *IEEE Transactions on Visualization and Computer Graphics*, 23(1):551–560, Jan. 2017. Conference Name: IEEE Transactions on Visualization and Computer Graphics. doi: 10.1109/TVCG.2016.2598920 1, 3

[14] S. Leem and H. Seo. Attention Guided CAM: Visual Explanations of Vision Transformer Guided by Self-Attention. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(4):2956–2964, Mar. 2024. doi: 10.1609/aaai.v38i4.28077 2, 3

[15] H. Liu, C. Li, Y. Li, and Y. J. Lee. Improved Baselines with Visual Instruction Tuning. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 26286–26296, June 2024. ISSN: 2575-7075. doi: 10.1109/CVPR52733.2024.02484 2, 3

[16] L. Y.-H. Lo and H. Qu. How Good (Or Bad) Are LLMs at Detecting Misleading Visualizations? *IEEE Trans. Visual. Comput. Graphics*, 31(1):1116–1125, Jan. 2025. doi: 10.1109/TVCG.2024.3456333 2

[17] A. Masry, D. X. Long, J. Q. Tan, S. Joty, and E. Hoque. ChartQA: A Benchmark for Question Answering about Charts with Visual and Logical Reasoning. In S. Muresan, P. Nakov, and A. Villavicencio, eds., *Findings of the Association for Computational Linguistics: ACL 2022*, pp. 2263–2279. Association for Computational Linguistics, Dublin, Ireland, May 2022. doi: 10.18653/v1/2022.findings-acl.177 1, 2, 3

[18] A. Masry, M. Thakkar, A. Bajaj, A. Kartha, E. Hoque, and S. Joty. ChartGemma: Visual Instruction-tuning for Chart Reasoning in the Wild. In O. Rambow, L. Wanner, M. Apidianaki, H. Al-Khalifa, B. D. Eugenio, S. Schockaert, K. Darwish, and A. Agarwal, eds., *Proceedings of the 31st International Conference on Computational Linguistics: Industry Track*, pp. 625–643. Association for Computational Linguistics, Abu Dhabi, UAE, Jan. 2025. 2, 3

[19] N. Methani, P. Ganguly, M. M. Khapra, and P. Kumar. PlotQA: Reasoning over Scientific Plots. In *2020 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 1516–1525, Mar. 2020. ISSN: 2642-9381. doi: 10.1109/WACV45572.2020.9093523 1, 2, 3

[20] S. Pandey and A. Ottley. Mini-VLAT: A Short and Effective Measure of Visualization Literacy. *Computer Graphics Forum*, 2023. Publisher: The Eurographics Association and John Wiley & Sons Ltd. 1, 3

[21] V. Petsiuk, A. Das, and K. Saenko. RISE: Randomized Input Sampling for Explanation of Black-box Models. In *British Machine Vision Conference 2018, BMVC 2018, Newcastle, UK, September 3-6, 2018*, p. 151. BMVA Press, 2018. 3, 4

[22] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PmLR, 2021. 2

[23] M. Ribeiro, S. Singh, and C. Guestrin. "Why Should I Trust You?": Explaining the Predictions of Any Classifier. In J. DeNero, M. Finlayson, and S. Reddy, eds., *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*, pp. 97–101. Association for Computational Linguistics, San Diego, California, June 2016. doi: 10.18653/v1/N16-3020 2

[24] A. Salamatian, A. Abaskohi, W.-C. Fan, M. R. I. Hossain, L. Sigal, and G. Carenini. ChartGaze: Enhancing Chart Understanding in LVLMs with Eye-Tracking Guided Attention Refinement. In C. Christodoulopoulos, T. Chakraborty, C. Rose, and V. Peng, eds., *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 12104–12124. Association for Computational Linguistics, Suzhou, China, Nov. 2025. doi: 10.18653/v1/2025.emnlp-main.607 2

[25] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 618–626, 2017. doi: 10.1109/ICCV.2017.74 2

[26] B. Tang, A. Boggust, and A. Satyanarayan. VisText: A benchmark for semantically rich chart captioning. In A. Rogers, J. Boyd-Graber, and N. Okazaki, eds., *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 7268–7298. Association for Computational Linguistics, Toronto, Canada, July 2023. doi: 10.18653/v1/2023.acl-long.401 2