LOOK WHERE YOU SAY YOU'RE LOOKING: SELF-GROUNDED ATTENTION FOR VISUAL REASONING

Anonymous authors
Paper under double-blind review

ABSTRACT

We introduce SELF-SALIENCY, a method for training Vision-Language Models (VLMs) to increase the alignment between their visual attention and the image regions mentioned in their reasoning. SELF-SALIENCY uses a grounding model to localize the objects mentioned in each reasoning step and treats the resulting areas as supervision for the model's visual attention. Previous work on steering visual attention determines target image regions based solely on the image and question. In contrast, we show that conditioning the target regions on the model's generated reasoning improves downstream performance. For proper evaluation, we build a unified, broad suite of 25 visual reasoning benchmarks, where we reproduce the results of previous methods. We find that SELF-SALIENCY significantly outperforms both prior attention-steering methods and baselines that ground image-level text, achieving both a better average rank, and better mean score. Post-training analysis shows that the model primarily adapts its reasoning text to existing attention patterns, producing shorter steps that refer to larger regions. Nevertheless, when controlling for generated text, attention to grounded regions increases significantly across the relevant layer. Finally, we identify a consistent geometric bias in VLM visual attention toward the image border. However, our ablations show that SELF-SALIENCY's gains cannot be explained by simply aligning attention with the center of the image, highlighting the importance of aligning visual attention with the regions mentioned in the model's reasoning. We will publish our code and model weights upon acceptance.

1 INTRODUCTION

When answering questions about images, humans shift their gaze across different image regions. Vision-language models (VLMs) exhibit an analogous mechanism during visual reasoning: generated text tokens incorporate visual information by attending to image tokens. Surprisingly, however, the regions emphasized by VLMs differ substantially from those attended to by humans. Indeed, a growing body of work shows that visual attention often fails to focus on image regions that humans consider important for answering the question (Gong et al., 2024; Darcet et al., 2024; Zhang et al., 2025c). Numerous methods attempted to address this mismatch by post-training steering (Gong et al., 2026; Hu et al., 2026). All these methods determine the target salient regions solely from the input image and question. By prescribing where the model should attend, they also constrain the reasoning paths available to the model. However, there is no reason to assume that these prescribed paths are optimal, or that other reasoning paths could not also lead to the correct answer. This motivates an unexplored alternative: deriving the target regions from the model's own reasoning chain.

To this end, we propose a post-training method, SELF-SALIENCY (Figure 1). During training, SELF-SALIENCY uses a grounding model to localize the image regions that are referenced in each reasoning step. The trained model is rewarded for greater alignment between its visual attention and these grounded regions. At inference time, the trained model doesn't utilize external models.

Inspecting recent literature, we find no broad consensus on evaluation benchmarks for visual reasoning. A large number of benchmarks have been introduced in recent years, with individual papers typically reporting results on only a subset of them. To provide a more comprehensive evaluation, we identify the benchmarks used in recent work and construct a suite of 25 benchmarks.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

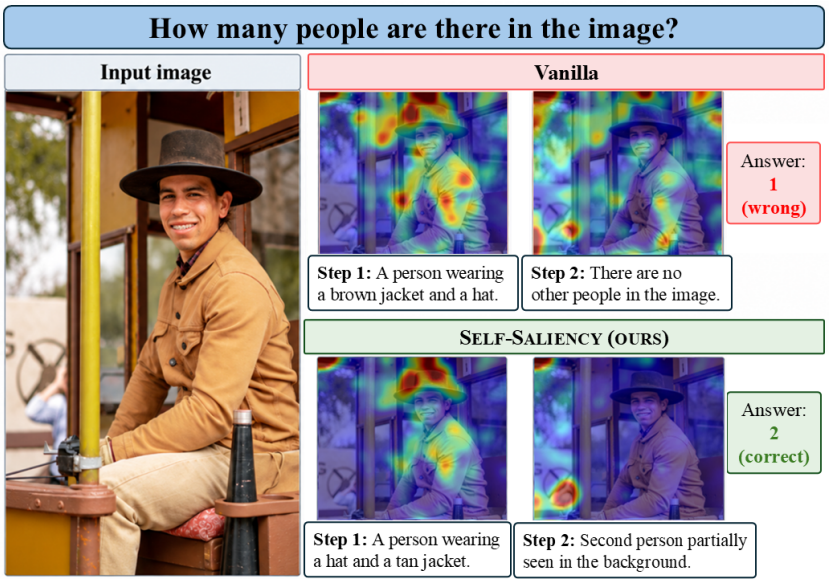


Figure 1: SELF-SALIENCY is trained to increase the alignment between visual attention and the regions referenced in its reasoning chain, creating two complementary pressures: (1) generate claims that can be grounded in the image, and (2) attend to the regions those claims refer to.

We evaluate SELF-SALIENCY on this broad suite and find that it substantially outperforms relevant baselines, with particularly strong gains on tasks that require attending to specific image regions. Surprisingly, these improvements are not limited to natural images. Although the grounding model was trained primarily on natural-image data, SELF-SALIENCY also improves performance on visual mathematical reasoning tasks.

We analyze whether training primarily aligns the generated text with existing attention patterns or shifts attention toward the generated reasoning. SELF-SALIENCY produces shorter reasoning steps that refer to larger image regions. To isolate attention changes from this shift in text, we teacher-force the pre- and post-training models on the same reasoning chains. While grounded-region attention does not increase in the two heads used for the saliency reward, it increases significantly when averaged across all heads in the relevant layer.

Finally, we analyze visual attention patterns across several VLM architectures and identify a consistent geometric bias: VLMs allocate disproportionately high attention to regions near the image border, whereas human annotations of task-relevant regions are more concentrated toward the image center. Motivated by this observation, we demonstrate that simply steering attention toward the image center does not improve downstream performance over the baselines, indicating that SELF-SALIENCY's gains cannot be explained by a generic center bias. Finally, SELF-SALIENCY outperforms a baseline that grounds image-level text, suggesting that grounding the model's reasoning chain provides a more effective training signal.

2 RELATED WORK

Reasoning VLMs. Recent work has increasingly focused on training VLMs to perform explicit reasoning over images. Several studies investigate the construction of multimodal chain-of-thought data for this purpose (Yang et al., 2025; Huang et al., 2026). Others modify the training procedure itself to improve reasoning behavior (Zhang et al., 2025a; Wang et al., 2025a).

Training VLMs to relocate attention. Several previous studies have proposed methods for redistributing visual attention. VAR (Kang et al., 2025a) and LASER (Yuan et al., 2026) transfer attention mass from sink tokens to non-sink tokens. VGA (Zhao et al., 2026) operates at inference time, steering attention toward image patches whose visual representations are more strongly asso-

ciated with the objects referenced in the question. RAL (Li et al., 2026a) encourages the model's attention distribution to move toward attention patterns of correct reasoning chains.

Most closely related to our approach, several prior works steer attention toward content-dependent ground-truth regions. KLAL (Esmaeilkhani & Latecki, 2026) and Saliency-R1 (Gong et al., 2026) use human-annotated bounding boxes as supervision targets. EASE (Hu et al., 2026) prompts GPT to identify phrases that are important for answering the question, grounds these phrases using a grounding model, and treats the resulting regions as ground truth. To the best of our knowledge, SELF-SALIENCY is the first method to derive grounded supervision directly from the reasoning chain generated by the policy being trained.

Visual attention analysis. In recent years, several studies have investigated how visual attention is distributed across an image. Darcet et al. (2024) and Gong et al. (2024) argue that a large fraction of visual attention is assigned to outlier tokens with large norms. Kang et al. (2025a) define *visual sink tokens* as tokens that consistently receive high attention due to their intrinsic properties, rather than their relevance to the generated text. They further suggest that such sink behavior arises from large activations in specific dimensions of the tokens' hidden states. Other studies show that LLaVA (Liu et al., 2024a) allocates a disproportionate amount of attention to the lower part of the image (Zhang et al., 2025c; Zhao et al., 2025). In Section 5, we extend this line of analysis to several VLMs and show that visual attention is consistently concentrated along the geometric border of the image.

3 METHOD

3.1 PRELIMINARIES

Image Patches. Most modern VLMs begin image processing by dividing the image into a grid of disjoint patches. A vision transformer jointly processes the patches and outputs a contextualized representation. These visual representations are then passed through an adapter that maps them into visual tokens. Thus, each visual token retains a spatial correspondence to an original image patch.

Attention Weights. When generating a new token, the self-attention mechanism computes a weight for each preceding token. These attention weights determine the contribution of each preceding token's value vector to the output representation of the current token at that layer. Attention weights are therefore often used as a proxy for which tokens the model is focusing on.

3.2 SETTING

To measure how well the model's attention aligns with the image regions referenced in its reasoning chain, we define a saliency score function. We first decompose each reasoning chain into atomic reasoning steps and retain only those that describe observations about the image.[1] We compute the saliency score at the step level and apply it only to these observation steps.

3.3 SALIENCY SCORE

Our saliency score pipeline consists of three components: localization of mentioned objects, saliency-map extraction, and saliency-map scoring. Figure 2 demonstrates our pipeline.

Throughout this section, we represent an image as a set of patches, $I = \{p_1, \ldots, p_{|I|}\}$.

Localization of mentioned objects. For each *observe* step s, we use a phrase-grounding model on the text of the step to localize the image regions corresponding to the objects mentioned in the step. The grounding model produces a set of bounding boxes $b_1, \ldots, b_N$, each represented as a set of image patches $b_i \subseteq I$. We define the grounded image region associated with step s as the union of all predicted bounding boxes: $u_s = \bigcup_{i=1}^{N} b_i$.

Attention-map extraction. Since current VLMs operate on image patches rather than individual pixels (Section 3.1), we define saliency at the patch level. We use the attention weights of selected

[1] See Appendix A.1 for details on reasoning-step segmentation and classification.

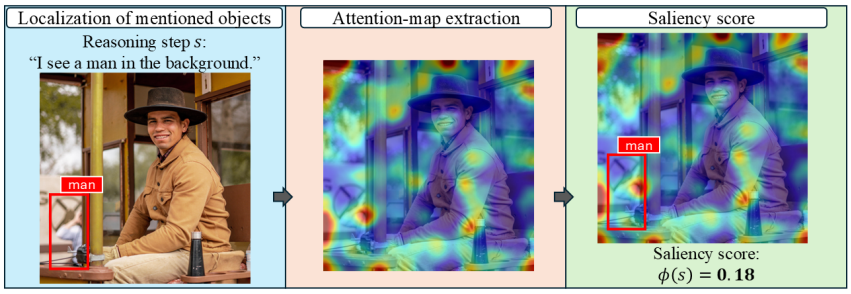


Figure 2: Our saliency-scoring pipeline. Given an input image and reasoning step, we extract bounding boxes corresponding to the mentioned image regions, compute a saliency map from the model's visual attention, and calculate the saliency score based on the overlap between the bounding boxes and the saliency map.

attention heads (see Section 3.5) as a proxy for model saliency. Specifically, for a generated token t and an image patch p, we define $sal_t(p)$ as the attention weight assigned by t to the visual token corresponding to p. For a reasoning step $s = \langle t_1, \ldots, t_{|s|} \rangle$, we define the saliency of patch p as the mean saliency across all tokens in the step: $sal_s(p) = \frac{1}{|s|} \sum_{i=1}^{|s|} sal_{t_i}(p)$.

Saliency score function. For each observation step s, we combine its grounded image region u_s with its patch-level saliency map $sal_s(p)$. We define the saliency score $\phi(s)$ as the mean saliency within the grounded region, normalized by the maximum saliency[2] over the entire image:

$$\phi(s) := \frac{\frac{1}{|u_s|} \sum_{p \in u_s} sal_s(p)}{\max_{p \in I} sal_s(p)} \tag{1}$$

3.4 Rewards

To apply our saliency score during training, we propose a post-training procedure based on reinforcement learning with verifiable rewards (RLVR), using GRPO (Shao et al., 2024b). In this setting, the reward is computed deterministically from the model's generated response. The most common reward in multimodal reasoning evaluates whether the model's final answer matches the ground-truth answer (Zhou et al., 2025). We augment this objective with an additional reward that utilizes our saliency score.

We use the following reward components. **(1) Saliency reward.** Given a reasoning chain $c = \langle s_1, \ldots, s_{|c|} \rangle$ and its set of extracted observation steps $O \subseteq c$, we define the saliency reward as the mean saliency score across all observation steps: $R_{sal} = \frac{1}{|O|} \sum_{s \in O} \phi(s)$. **(2) Accuracy rewards.** We use two complementary accuracy rewards that compare the model's final answer with the ground-truth answer: a direct-match reward, R_{direct}, and an LLM-as-a-judge reward, R_{llm}. **(3) Format reward.** The format reward, R_{format} is defined as 1 if the output is valid and 0 otherwise. A valid output must contain a `<think>...</think>` block followed by the final answer.

The final reward is a weighted linear combination of the individual reward components mentioned above: $R = \alpha_{sal} R_{sal} + \alpha_{direct} R_{direct} + \alpha_{llm} R_{llm} + \alpha_{format} R_{format}$.

3.5 Head Selection

Following prior work (Kang et al., 2025b), we compute the saliency reward using only a small set of selected attention heads. To identify these heads, we run the base model (Qwen3-VL-8B-Instruct; Bai et al., 2025) on the Visual-CoT dataset (Shao et al., 2024a) and compute the saliency reward R_{sal} (Section 3.4) for each sample and attention head. We then measure, for each head,

[2]We also experimented with normalizing by the mean saliency over the entire image, but found this variant to perform worse. See Appendix C for the full analysis.

Type	Benchmarks
Mathematical reasoning	MathVista (Lu et al., 2024), MathVision (Wang et al., 2024), MathVerse (Zhang et al., 2024), WeMath (Qiao et al., 2025), MMK12 (Meng et al., 2025)
Logical & algorithmic puzzles	LogicVista (Xiao et al., 2024), AlgoPuzzleVQA (Ghosal et al., 2024), VisuLogic (Xu et al., 2026), DailyClue (Li et al., 2026b)
Spatial & 3D reasoning	CV-Bench (Zhu et al., 2025), OmniSpatial (Jia et al., 2026)
High-resolution visual search	V^* (Wu & Xie, 2024), HR-Bench 4K&8K (Wang et al., 2025c), MME-RealWorld (Zhang et al., 2025d)
Hallucination & visual illusions	POPE (Li et al., 2023), HallusionBench (Guan et al., 2024), IllusionVQA (Shahgir et al., 2024)
Fine-grained perception & chart reading	ChartQA (Masry et al., 2022), SalBench (Huynh et al., 2025)
General & knowledge-intensive VQA	MMStar (Chen et al., 2024), MME (Fu et al., 2025), RealWorldQA (xAI, 2024), MMMU-Pro (Yue et al., 2025), ScienceQA-IMG (Lu et al., 2022)

Table 1: The 25 benchmarks in our evaluation suite, grouped by the capability they primarily target.

the correlation between its saliency reward and model accuracy, and select the two heads with the highest correlation.[3] This procedure selects heads 28 and 31 in layer 22, out of 36 layers.

This selection is consistent with the findings of Gandikota & Bau (2026), who show that the strongest alignment between attention and the image region being described occurs in layers 20–28. In particular, their Figure 1B shows that, for Qwen3-8B-VL, layer-wise residual-stream steering alters the model's spatial referent most strongly at layer 22.

4 Experiments

4.1 Training

To bootstrap the model's reasoning capabilities we start by training the base model, Qwen3-VL-8B-Instruct (Bai et al., 2025) using supervised fine-tuning (SFT) on the LLaVA-CoT (Xu et al., 2025) and Mulberry-SFT (Yao et al., 2025) datasets. We then start GRPO training from the resulting checkpoint on the Saliency-R1-8K (Gong et al., 2026) dataset, holding out 100 of its $8{,}080$ samples, using group size of $G = 8$. We set $\alpha_{format} = \alpha_{direct} = \alpha_{llm} = 1$ and $\alpha_{sal} = 0.4$.[4]

Auxiliary models. For grounding we use Grounding-DINO (Liu et al., 2024b), and as the LLM judge needed for R_{llm} we use GPT-4o mini (OpenAI, 2024).

Infrastructure. Training runs on a single node of eight H100 GPUs in bfloat16 under DeepSpeed ZeRO-3: six GPUs hold the policy, one serves generation with vLLM, and one hosts the grounding model. A full run takes approximately 42 seconds per optimizer step, or about 46 hours.

4.2 Evaluation

Benchmarks. Recent studies have introduced a large number of visual question answering (VQA) benchmarks (Kim et al., 2025). However, most papers evaluate on only a small subset, raising the possibility of benchmark-selection bias. For example, VGA (Zhao et al., 2026), Saliency-R1 (Gong et al., 2026), and EASE (Hu et al., 2026) evaluate on 3, 10, and 11 benchmarks, respectively, with only a single benchmark shared across all three. To mitigate this concern, we evaluate on all benchmarks used in recent work, together with several recently introduced benchmarks, yielding a comprehensive suite of 25 benchmarks listed in Table 1. We evaluate with lmms-eval (Zhang et al., 2025b).

Baselines. We first consider several baselines that do not explicitly intervene on visual attention: the vanilla **Qwen3-VL-8B-Instruct** model, our **coldstart** checkpoint (the checkpoint after SFT we

[3] We additionally verify that this relationship transfers across datasets by observing a similar correlation on a separate dataset.

[4] Selected using a hyperparameter sweep.

Table 2: Results across 25 benchmarks. Best per benchmark in **bold**. Mean score is computed over all 25 benchmarks after rescaling MME to $[0, 100]$ (raw score / 2800). Mean rank is the average over per-benchmark ranks (1 = best).

	No attention intervention			*Prior work*			*Ours*
Benchmark	Qwen3-VL 8B-Instruct	Coldstart	No-Sal	VGA	Saliency R1	EASE	SELF-SALIENCY
AlgoPuzzleVQA ↑	32.8	36.9	39.9	34.3	38.7	36.4	**41.4**
ChartQA ↑	85.0	88.8	89.4	84.9	89.7	88.9	**90.0**
CV-Bench ↑	86.0	83.4	84.9	**86.1**	83.8	84.2	85.9
DailyClue ↑	**35.4**	32.4	34.2	34.2	32.0	32.7	31.2
HallusionBench ↑	**63.2**	59.7	57.1	62.0	58.3	62.6	55.7
HRBench4k ↑	78.0	76.9	77.4	78.1	78.9	76.9	**79.5**
HRBench8k ↑	74.1	72.9	72.0	74.2	**76.0**	72.8	74.5
IllusionVQA ↑	40.0	41.9	41.5	40.1	**43.8**	40.5	43.5
LogicVista ↑	52.5	54.9	55.1	51.3	**56.5**	52.9	56.2
MathVerse ↑	34.4	51.0	**52.8**	33.8	49.5	50.4	50.9
MathVision ↑	29.6	33.5	37.5	31.2	33.9	35.9	**41.5**
MathVista ↑	73.5	69.5	71.8	**73.8**	71.0	70.0	70.4
MME ↑	2375.9	2371.7	2376.1	2379.8	**2395.7**	2395.4	2355.1
MME-RW ↑	63.7	64.2	66.7	63.8	66.4	65.1	**68.7**
MMK12 ↑	**68.2**	63.1	62.3	67.9	61.1	63.1	62.0
MMMUPro ↑	40.4	48.6	**48.6**	41.9	47.7	47.1	48.4
MMStar ↑	64.0	69.8	**70.9**	64.7	69.9	69.5	69.6
Omnispatial ↑	47.4	50.3	48.2	46.8	50.2	50.2	**51.0**
SalBench ↑	43.3	60.7	59.8	43.3	62.7	58.6	**64.7**
POPE ↑	88.8	87.4	88.2	88.8	88.5	88.5	**88.9**
RealWorldQA ↑	68.5	67.1	67.8	68.4	**68.8**	67.5	68.2
ScienceQA ↑	94.4	96.3	97.2	94.4	**97.6**	96.2	97.4
VisuLogic ↑	**26.6**	24.9	24.8	26.4	24.9	24.4	24.4
V* ↑	84.3	79.6	79.6	**85.3**	80.6	81.2	82.7
WeMath ↑	69.4	68.6	74.3	68.2	73.8	71.1	**75.5**
Mean score ↑	61.1	62.7	63.5	61.2	63.6	62.9	**64.3**
Mean rank ↓	4.42	4.64	3.68	4.48	3.26	4.50	**3.02**
Number of Wins ↑	4	0	3	3	6	0	**9**

use to initialize GRPO), and a model trained using the same procedure as SELF-SALIENCY but without the saliency reward, i.e., using only the accuracy and format rewards (**No-Sal**). The latter tests whether the observed performance gains can be attributed simply to additional post-training rather than to the saliency objective itself.

We also compare against methods from prior work that steer visual attention toward different image regions. **VGA** (Zhao et al., 2026) operates at inference time, steering attention toward image patches whose visual tokens are more strongly related to the objects referenced by the question. **Saliency-R1** (Gong et al., 2026) and **EASE** (Hu et al., 2026) steer attention toward bounding boxes annotated by human annotators (Saliency-R1) or generated by GPT (EASE).[5]

4.3 RESULTS

We show the results in Table 2. SELF-SALIENCY scores the highest mean score, the lowest mean rank and is most often the winner. The largest improvement is observed on benchmarks that require low-level visual saliency and multi-step reasoning (AlgoPuzzleVQA, SalBench, MME-RealWorld). Surprisingly, we also observe substantial gains on several mathematical reasoning benchmarks (We-Math, MathVision). This result is unexpected, as Grounding-DINO was trained on natural images, and we would therefore expect its localization predictions to be less reliable for mathematical diagrams. For qualitative examples of SELF-SALIENCY, see Figure 3.

[5]Since all baselines except EASE reported results on Qwen2.5, and EASE did not release trained checkpoints, we reimplemented and trained the baselines using their publicly available code. See Appendix D for further implementation details.



Figure 3: Qualitative examples of how SELF-SALIENCY correctly identifies low-level visual information.

Table 3: Text and attention statistics before and after training, averaged over 100 held-out validation samples. In-region share: attention in grounded regions divided by all visual attention. SELF-SALIENCY primarily adapts its generated text to the selected heads' attention patterns, while also increasing attention to grounded regions across layer 22.

	Text-related		Attention-related	
	Completion length	Box union area	In-region share (selected heads)	In-region share (layer-22 heads)
Coldstart	230.0	0.533	0.462	0.612
SELF-SALIENCY	149.6	0.566	0.462	0.621

4.4 TRAINED MODEL ANALYSIS

The model can increase our reward in two complementary ways: by adapting its generated text to regions that are easier to ground and already highly attended, or by shifting visual attention toward the regions described in its reasoning. We investigate which mechanism dominates after training.

We compare SELF-SALIENCY with the coldstart checkpoint on 100 held-out validation samples. Text statistics are computed on each model's own generations. To isolate changes in attention from changes in generated text, we additionally teacher-force both models on the same set of 200 reasoning chains – the 100 chains generated by each model – and compute attention statistics on this fixed set.

Table 3 shows that SELF-SALIENCY generates shorter reasoning chains that refer to larger image regions. With the text held fixed, attention to grounded regions does not increase in the two heads used to compute the saliency reward. However, averaged across all heads in layer 22, grounded-region attention increases significantly ($p < 0.001$). These results suggest that SELF-SALIENCY primarily improves the reward by adapting its generated reasoning, while also inducing a broader shift toward grounded regions across the relevant layer.

5 ATTENTION ANALYSIS AND ABLATIONS

In this section, we analyze the visual attention patterns of commonly used VLMs to better understand the training signal provided by SELF-SALIENCY. Prior work has suggested that visual attention tends to concentrate on background regions (Gong et al., 2024; Darcet et al., 2024) or regions that

Table 4: Attention enrichment for different models.

Model	En border ring	En center	Largest En
Qwen3-VL-8B-Instruct	1.20	0.92	Top left, 6.64
InternVL3.5-8B	1.18	0.95	Bottom left, 2.29
GLM-4.1V-Thinking	1.50	0.82	Top right, 7.79
Nemotron-Omni	1.43	0.87	Top right, 3.86
Human boxes	0.53	1.14	Non-border, 1.87

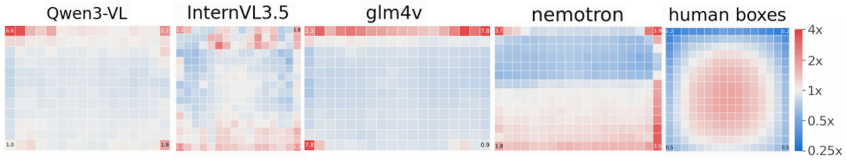


Figure 4: Saliency maps for different models and human-annotated bounding boxes. For each model, each cell represents the fraction of the model's total visual attention assigned to the corresponding image patch. For the human annotations (right), each cell represents the fraction of images in which the corresponding patch falls within a human-annotated bounding box.

are visually confusing or semantically related to the target object (Wang et al., 2026).[6] We propose a framing that does not contradict previous findings, but is more useful for our purposes: VLM attention is disproportionately concentrated along the geometric border of the image.

5.1 Experimental Setup

We use 1,800 (image, question) pairs from the Visual-CoT dataset (Shao et al., 2024a) to prompt four released instruction-tuned VLMs: Qwen3-VL-8B-Instruct (Bai et al., 2025), InternVL3.5-8B (Wang et al., 2025b), GLM-4.1V-Thinking (Hong et al., 2025) and Nemotron-Omni (Deshmukh et al., 2026).[7] For every image we run a forward pass and record, at every layer and head, the attention each image patch receives, averaged over the generated tokens – and renormalized within the image.[8]

Given a set of patches, we define its *enrichment* (En) as the fraction of total visual attention assigned to the set, normalized by the fraction of image patches it contains. An enrichment of $En = 1$ indicates no effect, while $En > 1$ and $En < 1$ indicate disproportionately high and low attention, respectively. We record two sets of patches: the border ring (all patches on the border of the image) and the center (all patches but the border ring).

5.2 Results

Table 4 shows enrichment results. Across all models, attention is more enriched along the border ring than in the center of the image. Moreover, in every model, the patch with the highest enrichment is located at a corner of the grid. Figure 4 visualizes the enrichment of each image patch for every model.

Relevant image areas are in the center. We use the bounding-box annotations from Visual-CoT, in which human annotators identify the image regions they consider important for answering each question. Over the same set of images, we compute an analogous patch-level statistic: we measure the fraction of images in which each patch falls within an annotated bounding box. The bottom row of Table 4 and the rightmost plot in Figure 4 present the results. While models allocate more attention to image boundaries, human annotations indicate that regions near the center are more relevant for answering questions.

[6] LLaVA was shown to attend to the lower portion of the image (Zhang et al., 2025c; Zhao et al., 2025); here we focus on other models.

[7] See Appendix E for a list of each model's characteristics.

[8] For models with image-dependent grid shape, we take the most common grid shape, and use area-weighted averaging to map all grids to a fixed-size averaged grid.

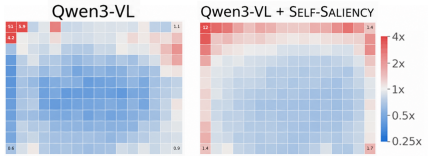


Figure 5: The fraction of the model's total visual attention assigned to the corresponding image patch, for the heads used to extract saliency (layer 22, heads 28+31), before and after training.

Table 5: Performance across our 25-benchmark suite for the ablation study.

	Qwen3-VL-8B Instruct	coldstart	no_sal	center_rect	question_boxes	SELF-SALIENCY
Mean score (↑)	61.14	62.69	63.47	63.39	63.33	**64.26**
Mean rank (↓)	3.90	4.06	3.34	3.68	3.38	**2.64**
Wins no. (↑)	7	0	3	2	5	**9**

The most attended patch is typically outside the grounded area. Because attention peaks near image corners while relevant regions tend to lie near the center, the maximum-attention patch usually falls outside the grounded region. Consequently, increasing attention within the grounded area raises the numerator of ϕ (Equation 1) without increasing its max-normalized denominator.

SELF-SALIENCY reduces corner-focused attention. Figure 5 shows the average attention heatmaps of the two selected heads before and after training. After training, attention is redistributed primarily from corner sinks to the image border, with a smaller fraction moving toward the center.

5.3 ABLATION STUDY

To isolate the contribution of different components of our method, we train two variants using the same procedure as SELF-SALIENCY, while changing only the source of the target bounding boxes. Motivated by the findings in Section 5.2, **center_rect** uses a fixed rectangle centered in the image. If this variant performs comparably to SELF-SALIENCY, it would suggest that the gains primarily result from shifting attention away from the grid corner. In contrast, **question_boxes** applies the grounding model to the input question rather than to the model's reasoning steps, allowing us to test whether chain-level grounding provides additional benefit over grounding image-level text.

We present the results in Table 5 and show a clear advantage for SELF-SALIENCY over the ablation variants. This finding indicates that grounding the model's reasoning chain provides a meaningful signal for alignment with visual attention.

6 CONCLUSION

We introduced SELF-SALIENCY, a post-training method for vision-language models. Rather than prescribing target regions from the input image and question alone, SELF-SALIENCY segments the model's reasoning chain into steps, grounds the objects mentioned in each observation step, and rewards the model for attending to those regions during GRPO training. We utilize a suite of 25 visual reasoning benchmarks and show that SELF-SALIENCY achieves the highest mean score, best mean rank, and most wins, outperforming prior attention-steering methods. The gains are largest on tasks that depend on fine-grained visual evidence, and, unexpectedly, extend to mathematical reasoning benchmarks. After training, the model produces shorter reasoning chains which refer to larger image regions, and pays more attention to the grounded regions across relevant layers.

Additionally, we find a consistent geometric bias of attention towards the corners of the image, while human-annotated relevant regions lie mainly near the center. SELF-SALIENCY substantially reduces this bias in selected heads. However, our ablations show that the gains cannot be explained by a simple attention alignment with the image center, nor by grounding question text alone. Grounding the model's own reasoning steps provides a more effective training signal.

AI USE STATEMENT

We used generative AI tools to assist with code generation and to improve the wording of text initially written by the authors. We also used AI tools to support literature discovery, while the literature review itself was conducted primarily through standard scholarly search and alerting tools. In addition, we used generative AI to create synthetic data for training the reasoning-step classifier described in Appendix A.1.

Generative AI was not used to generate paper sections, develop the core ideas of the work, or determine research directions. The authors take full responsibility for all content in the paper, including any text, claims, code, or other artifacts produced with the assistance of generative AI tools.

ETHICS STATEMENT

This work does not involve the collection of new data or interactions with human participants. All experiments are conducted using existing publicly available datasets, which we use in accordance with their respective licenses and terms of use. We do not introduce or release new datasets containing personal or sensitive information. Accordingly, we do not identify additional ethical concerns beyond those already associated with the underlying datasets and models used in this work.

REPRODUCIBILITY STATEMENT

We provide the experimental details necessary to reproduce our results throughout the main text and appendix, including the model configurations, training procedure, datasets, evaluation protocols, and implementation details. We have aimed to report all relevant experimental settings and design choices that may affect the results. Upon acceptance, we will publicly release our code and trained model weights to further facilitate reproduction of our experiments.

REFERENCES

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*, 2025.

Lin Chen, Jinsong Li, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Zehui Chen, Haodong Duan, Jiaqi Wang, Yu Qiao, Dahua Lin, et al. Are we on the right way for evaluating large vision-language models? *Advances in Neural Information Processing Systems*, 37:27056–27087, 2024.

Timothée Darcet, Maxime Oquab, Julien Mairal, and Piotr Bojanowski. Vision transformers need registers. In *International conference on learning representations*, volume 2024, pp. 2632–2652, 2024.

Amala Sanjay Deshmukh, Kateryna Chumachenko, Tuomas Rintamaki, Matthieu Le, Tyler Poon, Danial Mohseni Taheri, Ilia Karmanov, Guilin Liu, Jarno Seppanen, Arushi Goel, et al. Nemotron 3 nano omni: efficient and open multimodal intelligence. *arXiv preprint arXiv:2604.24954*, 2026.

Parsa Esmaeilkhani and Longin Jan Latecki. Direct visual grounding by directing attention of visual tokens. In *2026 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 5787–5797. IEEE, 2026.

Chaoyou Fu, Peixian Chen, Yunhang Shen, Yulei Qin, Mengdan Zhang, Xu Lin, Jinrui Yang, Xiawu Zheng, Ke Li, Xing Sun, Yunsheng Wu, Rongrong Ji, Caifeng Shan, and Ran He. Mme: A comprehensive evaluation benchmark for multimodal large language models. In *Advances in Neural Information Processing Systems*, volume 38. Curran Associates, Inc., 2025. doi: 10.52202/085713-4899. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/d79a27cf2772fe00be7f341efc0eb517-Paper-Datasets_and_Benchmarks_Track.pdf.

Rohit Gandikota and David Bau. Gaze heads: How vlms look at what they describe. *arXiv preprint arXiv:2606.14703*, 2026.

Deepanway Ghosal, Vernon Toh Yan Han, Chia Yew Ken, and Soujanya Poria. Are language models puzzle prodigies? algorithmic puzzles unveil serious challenges in multimodal reasoning. *arXiv preprint arXiv:2403.03864*, 2024.

Shizhan Gong, Minda Hu, Qiyuan Zhang, Chen Ma, and Qi Dou. Saliency-r1: Enforcing interpretable and faithful vision-language reasoning via saliency-map alignment reward. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 24095–24106, June 2026.

Xuan Gong, Tianshi Ming, Xinpeng Wang, and Zhihua Wei. Damro: Dive into the attention mechanism of lvlm to reduce object hallucination. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 7696–7712, 2024.

Tianrui Guan, Fuxiao Liu, Xiyang Wu, Ruiqi Xian, Zongxia Li, Xiaoyu Liu, Xijun Wang, Lichang Chen, Furong Huang, Yaser Yacoob, et al. Hallusionbench: an advanced diagnostic suite for entangled language hallucination and visual illusion in large vision-language models. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14375–14385. IEEE, 2024.

Wenyi Hong, Wenmeng Yu, Xiaotao Gu, Guo Wang, Guobing Gan, Haomiao Tang, Jiale Cheng, Ji Qi, Junhui Ji, Lihang Pan, et al. GLM-4.5V and GLM-4.1V-Thinking: Towards versatile multimodal reasoning with scalable reinforcement learning. *arXiv preprint arXiv:2507.01006*, 2025.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Ruina Hu, Chen Wang, Lai Wei, Jionghao Bai, Bin Yu, Weiran Huang, Kai Wang, and Yue Wang. Attend to evidence: Evidence-anchored spatial attention supervision for multimodal rlvr. *arXiv preprint arXiv:2605.30912*, 2026.

Wenxuan Huang, Bohan Jia, Shaosheng Cao, Zheyu Ye, Zhe Xu, Yao Hu, Shaohui Lin, et al. Vision-r1: Incentivizing reasoning capability in multimodal large language models. In *International Conference on Learning Representations*, volume 2026, pp. 63794–63812, 2026.

Ngoc Dung Huynh, Phuc H Le-Khac, Wamiq Reyaz Para, Ankit Singh, and Sanath Narayan. Vision-language models can't see the obvious. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 24159–24169, 2025.

Mengdi Jia, Zekun Qi, Shaochen Zhang, Wenyao Zhang, Xinqiang Yu, Jiawei He, He Wang, and Li Yi. Omnispatial: Towards comprehensive spatial reasoning benchmark for vision language models. In *International Conference on Learning Representations*, volume 2026, pp. 35634–35670, 2026.

Seil Kang, Jinyeong Kim, Junhyeok Kim, and Seong Jae Hwang. See what you are told: Visual attention sink in large multimodal models. In *International Conference on Learning Representations*, volume 2025, pp. 87676–87703, 2025a.

Seil Kang, Jinyeong Kim, Junhyeok Kim, and Seong Jae Hwang. Your large vision-language model only needs a few attention heads for visual grounding. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9339–9350. IEEE, 2025b.

Byeong Su Kim, Jieun Kim, Deokwoo Lee, and Beakcheol Jang. Visual question answering: A survey of methods, datasets, evaluation, and challenges. *ACM Computing Surveys*, 57(10):1–35, 2025.

Bangzheng Li, Jianmo Ni, Chen Qu, Ian Miao, Liu Yang, Xingyu Fu, Muhao Chen, and Derek Zhiyuan Cheng. Reinforced attention learning. *arXiv preprint arXiv:2602.04884*, 2026a.

Xiaomin Li, Tala Wang, Zichen Zhong, Ying Zhang, Zirui Zheng, Takashi Isobe, Dezhuang Li, Huchuan Lu, You He, and Xu Jia. Seek-and-solve: Benchmarking mllms for visual clue-driven reasoning in daily scenarios. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 15499–15517, 2026b.

Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Xin Zhao, and Ji-Rong Wen. Evaluating object hallucination in large vision-language models. In *Proceedings of the 2023 conference on empirical methods in natural language processing*, pp. 292–305, 2023.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 26286–26296. IEEE, 2024a.

Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Qing Jiang, Chunyuan Li, Jianwei Yang, Hang Su, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. In *European conference on computer vision*, pp. 38–55. Springer, 2024b.

Pan Lu, Swaroop Mishra, Tanglin Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. Learn to explain: Multimodal reasoning via thought chains for science question answering. *Advances in neural information processing systems*, 35:2507–2521, 2022.

Pan Lu, Hritik Bansal, Tony Xia, Jiacheng Liu, Chunyuan Li, Hannaneh Hajishirzi, Hao Cheng, Kai-Wei Chang, Michel Galley, and Jianfeng Gao. Mathvista: Evaluating mathematical reasoning of foundation models in visual contexts. In *International Conference on Learning Representations*, volume 2024, pp. 23439–23554, 2024.

Ahmed Masry, Jia Qing Tan, Shafiq Joty, Enamul Hoque, et al. Chartqa: A benchmark for question answering about charts with visual and logical reasoning. In *Findings of the association for computational linguistics: ACL 2022*, pp. 2263–2279, 2022.

Fanqing Meng, Lingxiao Du, Zongkai Liu, Zhixiang Zhou, Quanfeng Lu, Daocheng Fu, Tiancheng Han, Botian Shi, Wenhai Wang, Junjun He, et al. Mm-eureka: Exploring the frontiers of multimodal reasoning with rule-based reinforcement learning. *arXiv preprint arXiv:2503.07365*, 2025.

OpenAI. GPT-4o mini: Advancing Cost-Efficient Intelligence. `https://openai.com/index/gpt-4o-mini-advancing-cost-efficient-intelligence/`, July 2024. Accessed: 2026-09-13.

Runqi Qiao, Qiuna Tan, Guanting Dong, Minhui Wu, Chong Sun, Xiaoshuai Song, Jiapeng Wang, Zhuoma GongQue, Shanglin Lei, YiFan Zhang, Zhe Wei, Miaoxuan Zhang, Runfeng Qiao, Xiao Zong, Yida Xu, Peiqing Yang, Zhimin Bao, Muxi Diao, Chen Li, and Honggang Zhang. We-math: Does your large multimodal model achieve human-like mathematical reasoning? In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 20023–20070. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.acl-long.983. URL `https://aclanthology.org/2025.acl-long.983/`.

Haz Sameen Shahgir, Khondker Salman Sayeed, Abhik Bhattacharjee, Wasi Uddin Ahmad, Yue Dong, and Rifat Shahriyar. IllusionVQA: A challenging optical illusion dataset for vision language models. In *First Conference on Language Modeling*, 2024. URL `https://openreview.net/forum?id=7ysaJGs7zY`.

Hao Shao, Shengju Qian, Han Xiao, Guanglu Song, Zhuofan Zong, Letian Wang, Yu Liu, and Hongsheng Li. Visual cot: Advancing multi-modal language models with a comprehensive dataset and benchmark for chain-of-thought reasoning. *Advances in Neural Information Processing Systems*, 37:8612–8642, 2024a.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Yang Wu, and Daya Guo. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024b.

Haozhe Wang, Chao Qu, Zuming Huang, Wei Chu, Fangzhen Lin, and Wenhu Chen. Vl-rethinker: Incentivizing self-reflection of vision-language models with reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 38, pp. 30865–30891. Curran Associates, Inc., 2025a. doi: 10.52202/085713-1037. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/2c84844a559e4f962752570bff456ae4-Paper-Conference.pdf.

Ke Wang, Junting Pan, Weikang Shi, Zimu Lu, Houxing Ren, Aojun Zhou, Mingjie Zhan, and Hongsheng Li. Measuring multimodal mathematical reasoning with math-vision dataset. *Advances in Neural Information Processing Systems*, 37:95095–95169, 2024.

Weiyun Wang, Zhangwei Gao, Lixin Gu, Hengjun Pu, Long Cui, Xingguang Wei, Zhaoyang Liu, Linglin Jing, Shenglong Ye, Jie Shao, et al. Internvl3. 5: Advancing open-source multimodal models in versatility, reasoning, and efficiency. *arXiv preprint arXiv:2508.18265*, 2025b.

Wenbin Wang, Liang Ding, Minyan Zeng, Xiabin Zhou, Li Shen, Yong Luo, Wei Yu, and Dacheng Tao. Divide, conquer and combine: A training-free framework for high-resolution image perception in multimodal large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 7907–7915, 2025c.

Zichuan Wang, Songlin Yang, Bo Peng, Zhenchen Tang, Yang Li, Beibei Dong, and Jing Dong. Same attention, different truths: Put logit-lens over visual attention to detect and mitigate lvlm object hallucination. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 25315–25325, 2026.

Penghao Wu and Saining Xie. V*: Guided visual search as a core mechanism in multimodal llms. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13084–13094. IEEE, 2024.

xAI. Grok-1.5 vision preview. https://x.ai/news/grok-1.5v, April 2024. Accessed: 2026-09-23.

Yijia Xiao, Edward Sun, Tianyu Liu, and Wei Wang. Logicvista: Multimodal llm logical reasoning benchmark in visual contexts. *arXiv preprint arXiv:2407.04973*, 2024.

Guowei Xu, Peng Jin, Ziang Wu, Hao Li, Yibing Song, Lichao Sun, and Li Yuan. Llava-cot: Let vision language models reason step-by-step. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 2087–2098. IEEE, 2025.

Weiye Xu, Jiahao Wang, Weiyun Wang, Zhe Chen, Wengang Zhou, Aijun Yang, Lewei Lu, Houqiang Li, Xiaohua Wang, Xizhou Zhu, Wenhai Wang, Jifeng Dai, and Jinguo Zhu. Visulogic: A benchmark for evaluating visual reasoning in multi-modal large language models. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 25966–26003, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/2bbc73b3d3c2de43743ce2d82c8f3d7d-Paper-Conference.pdf.

Yi Yang, Xiaoxuan He, Hongkun Pan, Xiyan Jiang, Yan Deng, Xingtao Yang, Haoyu Lu, Dacheng Yin, Fengyun Rao, Minfeng Zhu, et al. R1-onevision: Advancing generalized multimodal reasoning through cross-modal formalization. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 2376–2385. IEEE, 2025.

Huanjin Yao, Jiaxing Huang, Wenhao Wu, Jingyi Zhang, Yibo Wang, Shunyu Liu, Yingjie Wang, YuXin Song, Haocheng Feng, Li Shen, and Dacheng Tao. Mulberry: Empowering mllm with o1-like reasoning and reflection via collective monte carlo tree search. In *Advances in Neural Information Processing Systems*, volume 38, pp. 29918–29952. Curran Associates, Inc., 2025. doi: 10.52202/085713-1004. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/2b0c4fc1bd52329279c51ad2ddec9ace-Paper-Conference.pdf.

Bowen Yuan, Zijian Wang, Yadan Luo, Shijie Wang, and Zi Huang. Laser: A corrective lens for lvlms via visual attention preservation and sink suppression. In *European Conference on Computer Vision*, pp. 355–372. Springer, 2026.

Xiang Yue, Tianyu Zheng, Yuansheng Ni, Yubo Wang, Kai Zhang, Shengbang Tong, Yuxuan Sun, Botao Yu, Ge Zhang, Huan Sun, et al. Mmmu-pro: A more robust multi-discipline multimodal understanding benchmark. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 15134–15186, 2025.

Jingyi Zhang, Jiaxing Huang, Huanjin Yao, Shunyu Liu, Xikun Zhang, Shijian Lu, and Dacheng Tao. R1-vl: Learning to reason with multimodal large language models via step-wise group relative policy optimization. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 1859–1869. IEEE, 2025a.

Kaichen Zhang, Bo Li, Peiyuan Zhang, Fanyi Pu, Joshua Adrian Cahyono, Kairui Hu, Shuai Liu, Yuanhan Zhang, Jingkang Yang, Chunyuan Li, et al. Lmms-eval: Reality check on the evaluation of large multimodal models. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pp. 881–916, 2025b.

Qizhe Zhang, Aosong Cheng, Ming Lu, Renrui Zhang, Zhiyong Zhuo, Jiajun Cao, Shaobo Guo, Qi She, and Shanghang Zhang. Beyond text-visual attention: Exploiting visual cues for effective token pruning in vlms. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 20857–20867. IEEE, 2025c.

Renrui Zhang, Dongzhi Jiang, Yichi Zhang, Haokun Lin, Ziyu Guo, Pengshuo Qiu, Aojun Zhou, Pan Lu, Kai-Wei Chang, Yu Qiao, et al. Mathverse: Does your multi-modal llm truly see the diagrams in visual math problems? In *European Conference on Computer Vision*, pp. 169–186. Springer, 2024.

YiFan Zhang, Huanyu Zhang, Haochen Tian, Chaoyou Fu, Shuangqing Zhang, Junfei Wu, Feng Li, Kun Wang, Qingsong Wen, Zhang Zhang, et al. Mme-realworld: Could your multimodal llm challenge high-resolution real-world scenarios that are difficult for humans? In *International Conference on Learning Representations*, volume 2025, pp. 89655–89701, 2025d.

Jianfei Zhao, Feng Zhang, Xin Sun, Chong Feng, and Zhixing Tan. Tell model where to look: Mitigating hallucinations in mllms by vision-guided attention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 32582–32591, 2026.

Qiyan Zhao, Xiaofeng Zhang, Yiheng Li, Yun Xing, Xiaosong Yuan, Feilong Tang, Sinan Fan, Xuhang Chen, Da-Han Wang, and Xu-Yao Zhang. Mca-llava: Manhattan causal attention for reducing hallucination in large vision-language models. In *Proceedings of the 33rd ACM International Conference on Multimedia*, pp. 3981–3990, 2025.

Guanghao Zhou, Panjia Qiu, Cen Chen, Jie Wang, Zheming Yang, Jian Xu, and Minghui Qiu. Reinforced mllm: A survey on rl-based reasoning in multimodal large language models. *arXiv preprint arXiv:2504.21277*, 2025.

Nannan Zhu, Yonghao Dong, Teng Wang, Xueqian Li, Shengjun Deng, Yijia Wang, Zheng Hong, Tiantian Geng, Guo Niu, Hanyan Huang, et al. Cvbench: Benchmarking cross-video synergies for complex multimodal reasoning. *arXiv preprint arXiv:2508.19542*, 2025.

A Further Implementation Details

A.1 Reasoning Steps Segmentation and Classification

Given a reasoning chain c generated in response to a question q, we first decompose it into an ordered sequence of atomic reasoning steps, $c = \langle s_1, \ldots, s_{|c|} \rangle$. Each step is itself a sequence of tokens, $s_i = \langle t_1, \ldots, t_{|s_i|} \rangle$, and is assigned a functional label l_i.

We define four functional categories for reasoning steps: *plan*, a forward-looking statement describing an intended action (e.g., "let me check the left side of the image"); *observe*, a direct statement about the visual content of the image (e.g., "there is a red car on the left"); *deduce*, a conclusion derived from previous reasoning steps rather than from newly introduced visual evidence (e.g., "therefore it is a traffic sign"); and *none*, a statement that does not contribute to the reasoning process (e.g., "the user asked where the cat is").

Segmentation. We segment the reasoning chain at sentence-final punctuation marks and line breaks, and discard fragments shorter than ten characters.

Classification. We classify each reasoning step using an encoder-only model trained by distilling labels produced by a proprietary model.[9]

After classification, we retain only steps labeled as *observe*, as these steps explicitly describe visual evidence from the image.

A.2 Training

Coldstart. For the coldstart, we use LoRA (Hu et al., 2022) of rank 128 on all linear layers, learning rate 5×10^{-5}, and cosine schedule with 5% warmup, training for two epochs.

GRPO training. We run GRPO (Shao et al., 2024b) for three epochs. Each step draws 6 prompts and samples $G = 8$ rollouts per prompt at temperature 1.0, for 48 completions per step (one completion per device, gradient accumulation 8). Prompts are truncated at $2{,}048$ tokens and completions at $1{,}024$, and images are resized so that the longer side is at most 512 pixels. We update only LoRA adapters of rank 16 ($\alpha = 32$, dropout 0.05) on the query and value projections, using AdamW at a learning rate of 1×10^{-5} decayed linearly to zero with no warmup, no weight decay, and gradient-norm clipping at 1.0. Advantages are normalized by the standard deviation of the group's rewards, and we use no KL penalty towards a reference policy ($\beta = 0$), so the reward of Section 3.4 is the entire training signal.

Judge model. As the LLM judge needed for R_{llm} we use GPT-4o mini (OpenAI, 2024) prompted with the question, the ground-truth answer and the model's extracted answer span, and asked for a correctness rating on a 1–5 scale that we rescale to $[0, 1]$.

A.3 Evaluation

Evaluation protocol. We evaluate with lmms-eval (Zhang et al., 2025b), temperature 0, single beam with `max_new_tokens`$= 4096$, `repetition_penalty`$= 1.05$, batch size 1, in bfloat16. Images are processed at lmms-eval's default budget of $1{,}605{,}632$ pixels, raised to $3{,}211{,}264$ for the three benchmarks whose subject is resolution (MME-RealWorld, HR-Bench 4K/8K). Benchmarks whose scoring requires answer extraction or an LLM judge use GPT-4o mini (OpenAI, 2024).

Benchmark splits. We use the following splits for the evaluation benchmarks: *soft_localization* for IllusionVQA, *testmini_vision_only* for MathVerse, *testmini* for MathVision, *testmini* for MathVista, *standard* for MMMUPro, *test* for Omnispatial, *p3* for SalBench, *img* for ScienceQA, *testmini* for WeMath.

B Extended Results

In this section, we report the results over our 25-benchmark suite, with bootstrap standard errors.

Error bars. Each reported score carries a nonparametric bootstrap estimate of its question-sampling uncertainty: we resample the run's units with replacement 10,000 times, recompute the score, and report the standard deviation of the resampled scores. The unit is whatever the benchmark's aggregation treats as independent – the question for most tasks, the image for MME (which asks two questions per image), and the question rather than the row for HR-Bench (where each is asked four times under cyclic option rotation); groups are resampled independently to their original sizes, so that macro-averaged metrics such as MMStar's nested average are preserved. Differences between models use a paired bootstrap over the questions both models answered, which is markedly tighter than combining the two marginal bars, since questions the models score identically contribute no uncertainty.

[9] The resulting classifier achieves 91.9% accuracy on the held-out test set and 93% accuracy on a human-labeled test set.

Table 6: Results across 25 benchmarks (mean ± standard error). Best per benchmark in **bold**. Mean score is computed over all 25 benchmarks after rescaling MME to $[0, 100]$ (raw score / 2800). Mean rank is the average over per-benchmark ranks (1 = best).

	No attention intervention			*Prior work*			*Ours*
Benchmark	Qwen3-VL 8B-Instruct	Coldstart	No-Sal	VGA	Saliency-R1	EASE	SELF-SALIENCY
AlgoPuzzleVQA ↑	32.8 ±1.11	36.9 ±1.14	39.9 ±1.15	34.3 ±1.12	38.7 ±1.15	36.4 ±1.13	**41.4** ±1.16
ChartQA ↑	85.0 ±0.71	88.8 ±0.63	89.4 ±0.62	84.9 ±0.72	89.7 ±0.61	88.9 ±0.63	**90.0** ±0.60
CV-Bench ↑	86.0 ±0.68	83.4 ±0.72	84.9 ±0.70	**86.1** ±0.67	83.8 ±0.72	84.2 ±0.71	85.9 ±0.68
DailyClue ↑	**35.4** ±1.85	32.4 ±1.82	34.2 ±1.84	34.2 ±1.84	32.0 ±1.81	32.7 ±1.82	31.2 ±1.80
HallusionBench ↑	**63.2** ±1.56	59.7 ±1.59	57.1 ±1.61	62.0 ±1.57	58.3 ±1.60	62.6 ±1.57	55.7 ±1.57
HRBench4k ↑	78.0 ±2.70	76.9 ±2.70	77.4 ±2.63	78.1 ±2.68	78.9 ±2.57	76.9 ±2.69	**79.5** ±2.60
HRBench8k ↑	74.1 ±2.75	72.9 ±2.77	72.0 ±2.79	74.2 ±2.77	**76.0** ±2.66	72.8 ±2.86	74.5 ±2.73
IllusionVQA ↑	40.0 ±1.55	41.9 ±1.56	41.5 ±1.56	40.1 ±1.55	**43.8** ±1.57	40.5 ±1.55	43.5 ±1.57
LogicVista ↑	52.5 ±2.36	54.9 ±2.35	55.1 ±2.35	51.3 ±2.36	**56.5** ±2.35	52.9 ±2.36	56.2 ±2.35
MathVerse ↑	34.4 ±1.69	51.0 ±1.78	**52.8** ±1.78	33.8 ±1.69	49.5 ±1.78	50.4 ±1.78	50.9 ±1.78
MathVision ↑	29.6 ±2.62	33.5 ±2.71	37.5 ±2.78	31.2 ±2.66	33.9 ±2.72	35.9 ±2.76	**41.5** ±2.83
MathVista ↑	73.5 ±1.40	69.5 ±1.46	71.8 ±1.42	**73.8** ±1.39	71.0 ±1.44	70.0 ±1.45	70.4 ±1.44
MME ↑	2375.9 ±36.2	2371.7 ±35.1	2376.1 ±36.3	2379.8 ±36.2	**2395.7** ±34.0	2395.4 ±33.1	2355.1 ±35.1
MME-RW ↑	63.7 ±0.31	64.2 ±0.31	66.7 ±0.31	63.8 ±0.31	66.4 ±0.31	65.1 ±0.31	**68.7** ±0.30
MMK12 ↑	**68.2** ±1.04	63.1 ±1.08	62.3 ±1.08	67.9 ±1.04	61.1 ±1.09	63.1 ±1.08	62.0 ±1.09
MMMUPro ↑	40.4 ±1.18	48.6 ±1.20	**48.6** ±1.20	41.9 ±1.19	47.7 ±1.20	47.1 ±1.20	48.4 ±1.20
MMStar ↑	64.0 ±1.34	69.8 ±1.29	**70.9** ±1.26	64.7 ±1.33	69.9 ±1.28	69.5 ±1.28	69.6 ±1.28
Omnispatial ↑	47.4 ±1.28	50.3 ±1.28	48.2 ±1.28	46.8 ±1.27	50.2 ±1.28	50.2 ±1.28	**51.0** ±1.28
SalBench ↑	43.3 ±0.97	60.7 ±0.96	59.8 ±0.96	43.3 ±0.97	62.7 ±0.95	58.6 ±0.97	**64.7** ±0.94
POPE ↑	88.8 ±0.33	87.4 ±0.35	88.2 ±0.34	88.8 ±0.33	88.5 ±0.34	88.5 ±0.34	**88.9** ±0.33
RealWorldQA ↑	68.5 ±1.68	67.1 ±1.70	67.8 ±1.69	68.4 ±1.68	**68.8** ±1.68	67.5 ±1.70	68.2 ±1.68
ScienceQA ↑	94.4 ±0.51	96.3 ±0.42	97.2 ±0.37	94.4 ±0.51	**97.6** ±0.34	96.2 ±0.42	97.4 ±0.35
VisuLogic ↑	**26.6** ±1.40	24.9 ±1.37	24.8 ±1.37	26.4 ±1.39	24.9 ±1.37	24.4 ±1.36	24.4 ±1.36
V* ↑	84.3 ±2.64	79.6 ±2.92	79.6 ±2.92	**85.3** ±2.57	80.6 ±2.87	81.2 ±2.84	82.7 ±2.74
WeMath ↑	69.4 ±1.10	68.6 ±1.11	74.3 ±1.05	68.2 ±1.12	73.8 ±1.05	71.1 ±1.09	**75.5** ±1.03
Mean score ↑	61.1	62.7	63.5	61.2	63.6	62.9	**64.3**
Mean rank ↓	4.42	4.64	3.68	4.48	3.26	4.50	**3.02**
Number of wins ↑	4	0	3	3	6	0	**9**

Results. We show results in Table 6.

C EXPERIMENTING WITH A DIFFERENT SALIENCY SCORING FUNCTION

Definition. We define a variant of our saliency score ϕ (Equation 1) in which the denominator is the mean saliency over the image rather than the maximum saliency:

$$\phi_{\text{mean}}(s) := \frac{\frac{1}{|u_s|}\sum_{p\in u_s} sal_s(p)}{\frac{1}{|I|}\sum_{p\in I} sal_s(p)}.$$

Analogously to the saliency reward, we define the corresponding mean-denominator saliency reward as

$$R_{sal_\text{mean}} = \frac{1}{|O|}\sum_{s\in O}\phi_{\text{mean}}(s)$$

Training. We follow the same training procedure described in Section 4.1, while calibrating the reward weight α_{sal_mean} relative to the tuned value of α_{sal}. To approximately match the strength of the training signal applied by GRPO, we measure the within-group standard deviation of R_{sal} and R_{sal_mean} over groups of completions sampled from the cold-start model, and set

$$\alpha_{sal_\text{mean}} = \alpha_{sal} \times \frac{\text{std}(R_{sal})}{\text{std}(R_{sal_\text{mean}})}.$$

Table 7: Results across 25 benchmarks, including SELF-SALIENCY$_{\text{mean}}$. Best per benchmark in **bold**.

		No attention intervention			*Saliency-based rewards*	
Benchmark	Split	Qwen3-VL 8B-Instruct	Coldstart	No-Sal	SELF-SALIENCY$_{\text{mean}}$	SELF-SALIENCY
AlgoPuzzleVQA ↑	–	32.8	36.9	39.9	38.8	**41.4**
ChartQA ↑	–	85.0	88.8	89.4	89.7	**90.0**
CV-Bench ↑	–	**86.0**	83.4	84.9	84.4	85.9
DailyClue ↑	–	**35.4**	32.4	34.2	32.7	31.2
HallusionBench ↑	image	**63.2**	59.7	57.1	55.7	55.7
HRBench ↑	4k	78.0	76.9	77.4	76.8	**79.5**
HRBench ↑	8k	74.1	72.9	72.0	**74.5**	**74.5**
IllusionVQA ↑	soft_localization	40.0	41.9	41.5	**44.1**	43.5
LogicVista ↑	–	52.5	54.9	55.1	53.1	**56.2**
MathVerse ↑	testmini_vision_only	34.4	51.0	**52.8**	50.8	50.9
MathVision ↑	testmini	29.6	33.5	37.5	36.8	**41.5**
MathVista ↑	testmini	**73.5**	69.5	71.8	70.8	70.4
MME ↑	–	2375.9	2371.7	**2376.1**	2359.4	2355.1
MME-RW ↑	–	63.7	64.2	66.7	66.6	**68.7**
MMK12 ↑	–	**68.2**	63.1	62.3	60.2	62.0
MMMUPro ↑	standard	40.4	48.6	**48.6**	48.3	48.4
MMStar ↑	–	64.0	69.8	**70.9**	70.3	69.6
Omnispatial ↑	test	47.4	50.3	48.2	50.0	**51.0**
SalBench ↑	p3	43.3	60.7	59.8	61.9	**64.7**
POPE ↑	–	88.8	87.4	88.2	88.2	**88.9**
RealWorldQA ↑	–	**68.5**	67.1	67.8	68.2	68.2
ScienceQA ↑	img	94.4	96.3	97.2	96.9	**97.4**
VisuLogic ↑	–	**26.6**	24.9	24.8	23.3	24.4
V* ↑	–	**84.3**	79.6	79.6	79.6	82.7
WeMath ↑	testmini	69.4	68.6	74.3	73.4	**75.5**
Mean score ↑	–	61.1	62.7	63.5	63.2	**64.3**
Mean rank ↓	–	3.24	3.52	2.68	3.26	**2.30**

Given our tuned value of $\alpha_{sal} = 0.4$, this yields $\alpha_{sal_\text{mean}} = 0.033$. We denote the resulting checkpoint by SELF-SALIENCY$_{\text{mean}}$.

Evaluation. We evaluate SELF-SALIENCY$_{\text{mean}}$ on the same suite of 25 benchmarks used in Section 4.2.

Results. Results are presented in Table 7. Unlike SELF-SALIENCY, SELF-SALIENCY$_{\text{mean}}$ does not improve over the *no_sal* baseline, which is trained without attention steering.

D REIMPLEMENTING BASELINES WITHOUT A PUBLISHED CHECKPOINT

In this section, we describe our reimplementation of baselines without a published checkpoint.

D.1 VGA

VGA (Zhao et al., 2026) is training-free, so our replication is an inference-time patch on Qwen3-VL rather than a retrained checkpoint. We keep the published hyperparameters – the guidance strength β, the top-K of the object-agnostic map, and the head-balancing scheme. Two points still required a decision, in each case because the original is backbone-specific or under-specified. We flag them in the following paragraphs.

The layer window. The window over which guidance is applied is backbone-specific in the original, and the rule for choosing it – start where attention to the sink position begins to dominate – fixes neither a quantity nor a threshold. On Qwen3-VL-8B its two natural readings disagree, selecting layer 3 or layer 7 against the published 4. Rather than choose, we swept five windows spanning both ($\text{start} \in \{2, 4, 7\}$, $\text{end} \in \{16, 20\}$, $\beta \in \{0.2, 0.25\}$) over 13 benchmarks at 100 documents each. The published window scores highest and the spread across all five is 0.005, so we report VGA as published; the result is not an artefact of under-tuning.

The object's first token. The map reads the probability of the object's first token at each visual position, but under BPE a word and its space-prefixed form are distinct tokens and it is not stated which is intended. We score the plain, space-prefixed and capitalised forms and keep the largest, which can only raise a real object above the floor.

D.2 SALIENCY-R1

We reimplement Saliency-R1 (Gong et al., 2026) from the authors' public repository[10], changing as little as the backbone swap allows. Their released checkpoints are Qwen2.5-VL-3B and -7B; no Qwen3-VL model is released.

Cold start. The cold-start stage described in Section 4.1 *is* theirs, simply on Qwen3-VL-8B-Instruct instead of Qwen2.5-VL-7B-Instruct. The Saliency-R1 row and all our own rows therefore begin from the same weights, and the comparison isolates the RL stage.

Reward. Their saliency map is computed as released: for each rollout, the attention of the answer tokens to the reasoning span is composed with the reasoning span's attention to the image tokens, weighted by the corresponding value vectors, projected through the output projection and the unembedding, summed over answer tokens, rectified, and reshaped onto the merged patch grid. The reward is the fraction of that map's mass falling inside the question-level box shipped with Saliency-R1-8K, and zero for completions that fail their format check. Our only edit to this code is that rather than retaining per-step attention throughout sampling, we sample first and recover the same attention rows from a single teacher-forced pass over the sampled sequence. Under causal attention the two are algebraically identical, and the second fits 1,024-token completions in memory.

GRPO configuration. We keep their published command: the same corpus and the same 100-sample held-out split (seed 42), their system prompt, learning rate 1×10^{-5}, three epochs, $G = 8$ rollouts at temperature 1.0, LoRA on the query and value projections at the library defaults their command leaves unset (rank 16, $\alpha = 32$, dropout 0.05), no KL penalty, and their global batch of 64 completions per optimizer step – redistributed as one completion per device with gradient accumulation 8 over eight GPUs, rather than their four and two, for memory. All eight GPUs train, giving 8 prompts per step and $2{,}991$ optimizer steps for three epochs rather than the $3{,}990$ of Section 4.1. We deviate from their command in three places. First, prompts and completions are capped at $2{,}048$ and $1{,}024$ tokens instead of $1{,}024$ and 512, matching our other runs; the cap was rarely active, with mean completion length between 162 and 247 tokens and fewer than 0.5% of rollouts reaching it. Second, we add the same exact-match reward R_{direct} that every other arm receives, so the reward here is $R_{\text{format}} + R_{\text{direct}} + R_{\text{llm}} + R_{\text{sal}}^{\text{S-R1}}$ and this arm differs from ours in the attention term alone; we leave their weight of 1 on $R_{\text{sal}}^{\text{S-R1}}$ rather than retuning it to our α_{sal}. Third, training-time sampling uses no repetition penalty, where their command sets 1.05. We use the same correctness judge they use, GPT-4o-mini with their prompt.

D.3 EASE

Although EASE reports results on Qwen3-VL-8B, the authors did not release their evaluation protocol, training data, or a trained checkpoint. We contacted the authors by email but did not receive a response. We therefore reimplemented and trained EASE ourselves and evaluated it using our unified evaluation protocol.

What we run, and on what data. We use their code as released, in their framework, in a separate environment; the files implementing the method itself are untouched, and our changes are confined to their reward interface and an import fix. They release neither the evidence-box annotations nor the code of the three-stage pipeline that produced them, which is described in prose only. We therefore run EASE on Saliency-R1-8K, which already ships a box for every one of its $8{,}080$ rows, bypassing their annotation pipeline entirely. This has a second benefit: EASE and our method then differ in exactly one thing – where the box signal enters – on the same corpus, from the same cold-start checkpoint, with the same held-out split and the same judge.

[10] `https://github.com/peterant330/Saliency_R1`

Table 8: Models used in the attention focus experiment.

Model	Vision encoder	Grid shape	Language model
Qwen3-VL-8B-Instruct	ViT	Image-dependent	Qwen3-8B
InternVL3.5-8B	InternViT-300M	16×16	Qwen3-8B
GLM-4.1V-Thinking	AIMv2-Huge	Image-dependent	GLM-4-9B-0414
Nemotron-Omni	C-RADIOv4-H	Image-dependent	Nemotron

Verification. Because EASE is a method inside a framework that is not ours, a single EASE number would confound the two. We therefore also trained their DAPO baseline in the same framework, on the same data, from the same cold start. On the natural-image benchmarks where both arms were scored, EASE $-$ DAPO is $+2.52$ points, against the $+2.5$ to $+3.1$ over DAPO that they report: the effect replicates, on our corpus, in their framework. One qualification belongs with that number – DAPO is a weak reference here, scoring slightly below the cold start it started from, so part of the margin is EASE avoiding damage rather than adding value; measured against the cold start directly, EASE is $+2.07$.

E Attention Analysis Model Details

Table 8 shows the characteristics of each of the models we use in the experiment in Section 5.