SPAR: SELF-REFERENTIAL PROMPT-CONDITIONED ATTENTION REWEIGHTING FOR VLM DECODING

Anonymous authors
Paper under double-blind review

ABSTRACT

Vision-language models often fail on fine-grained visual queries even when the required evidence is present in the image. We show that this failure arises from a decoder-side routing problem rather than a lack of visual perception. Across VLM families, early and final decoder layers fall into a Prompt-Invariant Sink Regime, where attention repeatedly collapses onto stable visual sink tokens across prompts. In contrast, middle layers form a Prompt-Conditioned Evidence Window, where attention is less sink-dominated and more responsive to the input query. Motivated by this depth asymmetry, we introduce SPAR, a self-referential and training-free attention reweighting method for frozen VLMs. SPAR estimates a region of interest from the model's own middle-layer visual attention and uses this prompt-conditioned evidence signal to guide visual attention during decoding. Across multiple benchmarks and VLM families, SPAR improves answer accuracy while increasing evidence retention and reducing mid-to-final evidence drop. Ablations over ROI construction, random controls, and layer choice show that the gains come from prompt-conditioned middle-layer evidence rather than generic attention perturbation or early-layer attention cues. Our results suggest that reliable fine-grained VLM decoding requires preserving evidence-bearing middle-layer signals through later decoder computation.

1 INTRODUCTION

Vision-language models (VLMs) (Liu et al., 2023; 2024b; Li et al., 2023a; Wang et al., 2024; Yang et al., 2025a; Chen et al., 2024b; Liu et al., 2025) have become a standard interface for image-conditioned reasoning, supporting open-ended visual question answering, document understanding, object recognition, and multimodal instruction following. Despite this progress, VLMs remain unreliable on fine-grained visual queries where the answer depends on small, localized, or visually subtle evidence (Tong et al., 2024; Kim & Ji, 2024; Liao et al., 2025; Wu & Xie, 2024; Zhu et al., 2024). Typical examples include reading a short text span, identifying a small object, verifying a local attribute, or distinguishing between visually similar alternatives (Singh et al., 2019; Li et al., 2023b; Hudson & Manning, 2019; Lu et al., 2022; Yin et al., 2024; Yu et al., 2023). These failures are often surprising: the relevant evidence is present in the image, and the model may even expose useful visual attention at intermediate depths, yet the final answer remains incorrect.

A common assumption behind many inference-time fixes is that such failures can be addressed by making the model attend more strongly to a query-relevant region of interest (ROI) (Zhao et al., 2025; Esmaeilkhani & Latecki, 2025; Shi et al., 2025). Under this view, the central problem is localization: once the correct region is identified, increasing attention to that region should improve the answer. However, our analysis suggests that this view is incomplete. Stronger ROI attention alone does not fully explain reliable fine-grained decoding (Geigle et al., 2024). In several settings, changing attention statistics or increasing attention to a candidate region improves internal attention measures without consistently improving final answer accuracy. This motivates a more specific question: *when the model has already exposed useful visual evidence in its own computation, why is that evidence not preserved until the final answer?*

We identify the failure as a decoder-side routing problem. Across VLM families, visual attention over decoder depth exhibits a consistent asymmetry. Early and final decoder layers fall into what we call the *Prompt-Invariant Sink Regime* (PISR): attention repeatedly collapses onto stable visual sink

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.

tokens that remain similar across prompts. In contrast, middle layers form a *Prompt-Conditioned Evidence Window* (PCEW): attention is less dominated by prompt-invariant sinks and becomes more responsive to the input query. In other words, middle layers are where the model most clearly exposes query-conditioned visual evidence, while early and final layers are more prone to prompt-invariant sink behavior. This depth asymmetry suggests that the problem is not simply whether the model can find evidence, but whether evidence-bearing middle-layer signals can survive later decoder computation.

We support this diagnosis with two complementary analyses. First, we measure prompt-invariant sink overlap by computing how consistently the same visual tokens are selected as high-attention sink tokens across different prompts for the same image. Second, we analyze the spectral structure of visual attention logits by stacking prompt- and step-conditioned attention vectors and measuring the singular value concentration ρ_k. As shown in Figure 1, layers with high Sink IoU also exhibit high spectral concentration, and the depth profile follows an early-high, middle-low, final-rebound pattern. This indicates that visual sinks are not merely high-attention heatmap artifacts, but stable low-rank routing directions in the decoder. Figure 2 further shows that this Sink IoU–ρ_1 alignment persists across multiple VLM families and benchmarks, supporting the generality of the PISR–PCEW distinction.

Motivated by this diagnosis, we introduce SPAR, short for Self-Referential Prompt-Conditioned Attention Reweighting. SPAR is a training-free inference-time method for frozen VLMs. It estimates an ROI from the model's own prompt-conditioned middle-layer attention and uses this self-referential evidence signal to reweight visual attention during decoding. The method is designed to be minimally invasive: it preserves the full visual token sequence, modifies no model parameters, and changes only the allocation of visual attention at inference time. The central principle is simple: use the PCEW as an evidence source and preserve its signal through later decoding.

Our experiments show that SPAR improves both final task performance and internal evidence routing. Across LLaVA (Liu et al., 2023; 2024b; Li et al., 2024), Qwen-VL (Wang et al., 2024; Bai et al., 2023; Yang et al., 2025a), InternVL (Chen et al., 2024b; Zhu et al., 2025), and NVILA (Liu et al., 2025), SPAR improves benchmark scores over the corresponding baseline and consistently outperforms a same-size random ROI control. These results show that SPAR's gains are not caused by arbitrary visual-token reweighting, but depend on the self-referential ROI estimated from prompt-conditioned middle-layer attention. We further show that SPAR increases evidence retention R and reduces mid-to-final evidence drop D, connecting external answer improvements to the decoder-side routing problem identified in our diagnosis.

Our contributions are summarized as follows:

- We identify a depth-asymmetric decoder routing failure in VLMs: early and final layers exhibit a Prompt-Invariant Sink Regime, while middle layers form a Prompt-Conditioned Evidence Window.
- We show that prompt-invariant visual sinks are aligned with low-rank spectral dominance, indicating that they are structured routing directions rather than arbitrary high-attention artifacts.
- We propose SPAR, a self-referential and training-free attention reweighting method that uses the model's own middle-layer attention to estimate evidence-bearing visual tokens.
- We demonstrate that SPAR improves answer accuracy together with evidence-retention metrics, and provide controls showing that the gains depend on prompt-conditioned middle-layer evidence rather than random visual-token reweighting.

2 Related Work

Visual attention sinks and evidence routing in VLMs. Modern VLMs achieve strong general multimodal performance, but remain unreliable on fine-grained queries that require small or localized visual evidence, such as scene text, subtle attributes, and object-level verification (Liu et al., 2023; 2024b; Chen et al., 2024b; Li et al., 2024; Bai et al., 2023; Wang et al., 2024; Yang et al., 2025a; Liu et al., 2025). Recent analyses suggest that these failures are not only caused by the vision encoder, but also by how decoder layers route attention over visual tokens. Attention sinks, originally studied in

language models, have also been observed in multimodal decoders (Touvron et al., 2023; Grattafiori et al., 2024; Chiang et al., 2023), where a small set of visual tokens repeatedly absorbs attention mass even when it is not query-relevant (Kang et al., 2025a; Kaduri et al., 2025; Feng et al., 2025). Closest to our setting, prior VLM analyses show that visual attention can be depth-dependent: mid layers often become more prompt-responsive, while early and late layers can collapse onto prompt-invariant visual regions. Our work builds on this view but goes beyond measuring sink mass or ROI attention alone. We show that late visual sinks form stable low-rank directions and that selectively controlling these directions is necessary to jointly improve evidence retention and answer accuracy.

Inference-time visual-token and attention interventions. A broad class of training-free VLM methods modifies inference by changing which visual tokens are processed or how decoder attention is allocated (Liu et al., 2024c; Kang et al., 2025b; Oota et al., 2024; Mahajan et al., 2025; Chen et al., 2025a; Wu et al., 2026; Jiang et al., 2025). Token selection and compression methods, such as FastV, VisionZip, and SparseVLM, reduce visual-token redundancy to improve efficiency, often by pruning or merging tokens based on attention, similarity, or text-guided relevance (Chen et al., 2024a; Yang et al., 2025b; Zhang et al., 2024; Khaki et al., 2025; Xing et al., 2024; Bolya et al., 2022). These methods primarily change the visual token set or compute budget. In contrast, attention steering methods keep the frozen VLM intact but modify decoder attention logits or attention mass to improve grounding and reduce hallucination (Li et al., 2023b; Huang et al., 2024b; Chen et al., 2025b; Liu et al., 2024a; Bai et al., 2024; Huang et al., 2024a; Leng et al., 2024; Liu et al., 2024c). ROI-guided steering assumes that increasing attention to a query-relevant region is sufficient (Zhao et al., 2025; Esmaeilkhani & Latecki, 2025; Shi et al., 2025), but our experiments show that generic ROI seeding or mid-layer attention copying can improve internal attention metrics without reliably improving accuracy. We therefore target a more specific mechanism: preserving prompt-conditioned evidence that appears in middle decoder layers but is under-used during later decoding. SPAR differs from token pruning and generic ROI amplification by estimating a self-referential ROI from PCEW attention and applying balanced push-pull reweighting to visual-token attention logits, without removing visual tokens or modifying model parameters.

3 Diagnosing Low-Rank Visual Sinks

We first diagnose how visual attention is routed across decoder depth. Our goal is to determine whether visual attention sinks are isolated high-attention artifacts, or whether they reflect a structured routing pattern that competes with query-relevant evidence. To this end, we analyze two complementary properties: prompt invariance and spectral concentration.

Prompt-invariant sink overlap. For a fixed image, we evaluate the model under multiple prompts and identify visual tokens that receive high attention. If the same visual tokens repeatedly appear as high-attention tokens across different prompts, we regard them as prompt-invariant sink tokens. We quantify this behavior using Sink IoU, defined as the intersection-over-union between the top-attended visual-token sets obtained from different prompts for the same image. High Sink IoU indicates that attention is not primarily determined by the prompt, but repeatedly collapses onto similar visual locations.

Spectral concentration of visual attention. We further ask whether these prompt-invariant sinks induce a low-dimensional structure in attention space. For each decoder layer and head, we collect pre-softmax attention logits from text-query positions to visual-token keys across prompts and decoding steps. Let $a_{p,t}^{(\ell,h)} \in \mathbb{R}^N$ denote the visual-key logit vector at layer ℓ, head h, prompt instance p, and decoding step t, where N is the number of visual tokens. We stack these vectors into a matrix

$$A^{(\ell,h)} \in \mathbb{R}^{M \times N}, \tag{1}$$

where each row corresponds to a prompt–step pair. We then compute its singular value decomposition,

$$A^{(\ell,h)} = U \Sigma V^\top, \tag{2}$$

and define the top-k spectral concentration as

$$\rho_k(A) = \frac{\sum_{j=1}^{k} \sigma_j^2}{\sum_{j=1}^{\min(M,N)} \sigma_j^2}. \tag{3}$$

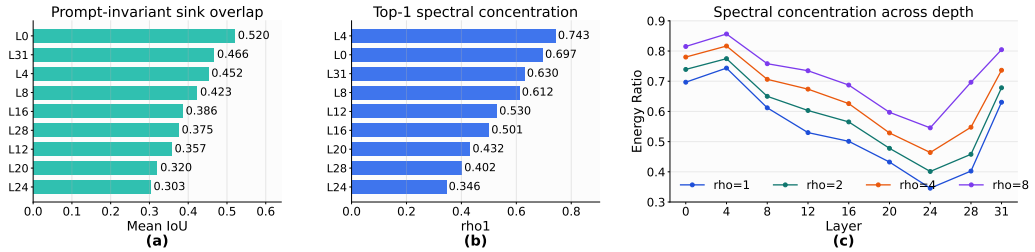


Figure 1: Prompt-invariant visual sinks form structured low-rank attention patterns. (a) Layers ranked by mean Sink IoU, measuring how consistently the same visual tokens are selected as sinks across different prompts for the same image. (b) Layers ranked by top-1 spectral concentration ρ_1, computed from stacked visual-key attention logits. (c) Spectral concentration ρ_k across decoder depth for multiple ranks k. Layers with high Sink IoU also exhibit high spectral concentration, and the depth profile follows an early-high, middle-low, final-rebound pattern. This suggests that prompt-invariant sinks are not arbitrary heatmap artifacts, but stable low-rank routing directions in the decoder.

High ρ_k for small k indicates that visual attention logits are dominated by a small number of stable directions. When such concentration aligns with high Sink IoU, we interpret it as evidence that prompt-invariant sinks form structured low-rank routing directions (Xiao et al., 2023; Gu et al., 2024) rather than arbitrary attention artifacts.

Depth asymmetry: PISR and PCEW. Figure 1 reveals a clear depth asymmetry. Early and final layers show high Sink IoU and high spectral concentration, indicating that they fall into a Prompt-Invariant Sink Regime (PISR). In this regime, attention repeatedly collapses onto similar visual sink tokens, regardless of the prompt. In contrast, middle layers show lower Sink IoU and lower spectral concentration. We refer to this middle-depth interval as the Prompt-Conditioned Evidence Window (PCEW), where visual attention is less sink-dominated and more responsive to the input query. This distinction motivates the central design principle of SPAR: use PCEW attention as a self-referential evidence signal, rather than relying on early or final layers that are more strongly dominated by prompt-invariant sinks.

Generality across models and benchmarks. We next test whether the Sink IoU–ρ_1 relationship is specific to one architecture or dataset. Figure 2 shows the layer-wise alignment between prompt-invariant sink overlap and spectral concentration across LLaVA, Qwen-families and two benchmarks. Across all evaluated settings, layers with stronger prompt-invariant sink behavior also show stronger low-rank spectral dominance. This supports the view that PISR and PCEW are not artifacts of a single model, but recurring depth-dependent regimes in VLM decoders. Importantly, we use this analysis as a structural diagnostic rather than a causal claim. The causal relevance of PCEW-based evidence and PISR-driven sink behavior is tested later through SPAR and its control ablations.

4 METHOD

We propose SPAR, Self-Referential Prompt-Conditioned Attention Reweighting, a training-free inference-time method for frozen VLMs. SPAR is motivated by the depth asymmetry identified in Section 3: middle decoder layers expose comparatively prompt-conditioned visual evidence, while early and final layers are more strongly affected by prompt-invariant sink behavior. SPAR uses the model's own middle-layer attention to estimate a region of interest (ROI), and then uses this self-referential ROI to softly reweight visual attention during generation.

4.1 OVERVIEW

Given an image I and a query Q, SPAR operates in two passes over the same input. The first pass estimates an ROI over visual tokens without modifying the model. The second pass performs autoregressive generation while applying a lightweight push-pull bias to visual-token attention logits

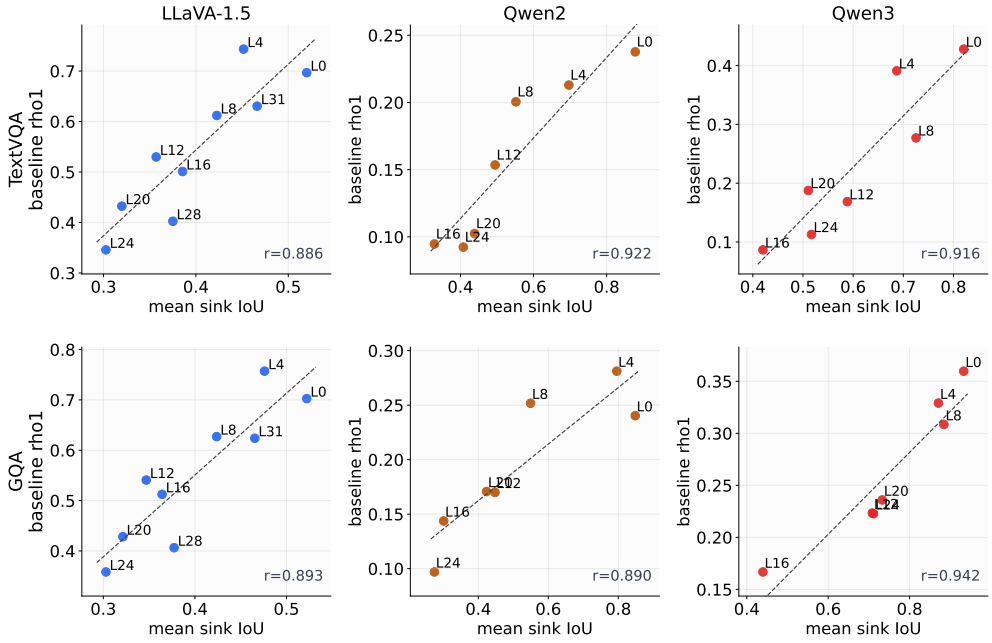


Figure 2: Sink overlap and spectral dominance are aligned across VLM families and benchmarks. Each point denotes a decoder layer, with the x-axis showing mean Sink IoU and the y-axis showing top-1 spectral concentration ρ_1. Across LLaVA-1.5, Qwen2-VL, and Qwen3-VL on TextVQA and GQA, layers that repeatedly select the same sink tokens across prompts also exhibit stronger low-rank spectral dominance. Dashed lines indicate fitted linear trends, and r denotes Pearson correlation. Because attention scales differ across architectures, we interpret the association within each model rather than comparing absolute ρ_1 values across models.

at a single decoder layer. SPAR does not update model parameters, introduce trainable modules, prune visual tokens, or mask the visual sequence. It only reweights attention allocation over visual-token key positions. This design has three intended properties. First, it is *self-referential*: the ROI is estimated from the target VLM's own internal attention. Second, it is *prompt-conditioned*: the ROI is derived from middle-layer attention, where visual attention is more responsive to the input query. Third, it is *minimally invasive*: reweighting is applied at a single decoder layer while the rest of the model remains unchanged.

4.2 Depth Partition

To apply the same procedure across VLMs with different numbers of decoder layers, we define early, middle, and final regions by normalized decoder depth. For a model with L decoder layers, we use a 25/50/25 partition:

$$\mathcal{L}_{\text{early}} = \{0, \ldots, \lfloor 0.25L \rfloor - 1\}, \tag{4}$$

$$\mathcal{L}_{\text{mid}} = \{\lfloor 0.25L \rfloor, \ldots, \lfloor 0.75L \rfloor - 1\}, \tag{5}$$

$$\mathcal{L}_{\text{final}} = \{\lfloor 0.75L \rfloor, \ldots, L - 1\}. \tag{6}$$

The middle region $\mathcal{L}_{\text{mid}}$ corresponds to the Prompt-Conditioned Evidence Window (PCEW), where attention is less dominated by prompt-invariant sinks and is more responsive to the input query. For the 32-layer LLaVA-1.5 backbone, this partition gives

$$\mathcal{L}_{\text{early}} = \{0, \ldots, 7\}, \quad \mathcal{L}_{\text{mid}} = \{8, \ldots, 23\}, \quad \mathcal{L}_{\text{final}} = \{24, \ldots, 31\}.$$

The same normalized partition is used for other VLM backbones.

4.3 Self-Referential ROI Estimation

Let the VLM represent an image as a sequence of N visual tokens,

$$\mathcal{V} = \{v_1, \ldots, v_N\}.$$

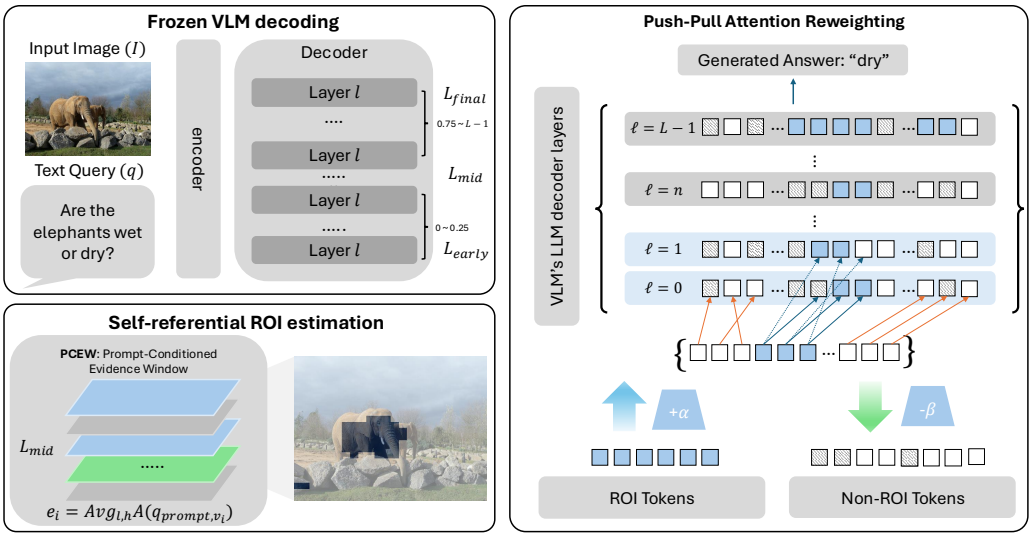


Figure 3: Overview of SPAR. (a) SPAR operates on a frozen VLM during decoding. (b) In the first pass, SPAR estimates a self-referential ROI by averaging PCEW attention from the final prompt token to visual tokens and selecting the top-k visual tokens. (c) In the second pass, SPAR applies a push-pull bias at a single target decoder layer: ROI visual-token logits receive $+\alpha$, while non-ROI visual-token logits receive $-\beta$. The intervention reweights visual attention without pruning or masking visual tokens.

For each image–query pair, SPAR first performs one unmodified forward pass and captures attention from the middle-layer window $\mathcal{L}_{\text{mid}}$. We use the final prompt token as the query position for ROI estimation. This token is the last textual token before answer generation begins, and its attention reflects how the decoder prepares to use visual information for the response. Let q_{prompt} denote this final prompt-token position. For each visual token v_i, we compute an evidence score by averaging attention from q_{prompt} to v_i across all heads and middle layers:

$$e_i = \frac{1}{|\mathcal{L}_{\text{mid}}|H} \sum_{\ell \in \mathcal{L}_{\text{mid}}} \sum_{h=1}^{H} A^{\ell,h}(q_{\text{prompt}}, v_i), \tag{7}$$

where $A^{\ell,h}(q_{\text{prompt}}, v_i)$ is the post-softmax attention weight from the final prompt token to visual token v_i at layer ℓ and head h, and H is the number of attention heads. We define the self-referential ROI as the top-k visual tokens by evidence score:

$$\mathcal{R} = \text{TopK}(\mathbf{e}, k), \tag{8}$$

where $\mathbf{e} = [e_1, \ldots, e_N]$. Unless otherwise stated, we use $k = 48$ for $N = 576$ visual tokens, corresponding to 8.3% of the visual token sequence. For backbones with a different number of visual tokens, we use the same token ratio. The resulting ROI $\mathcal{R}$ is computed once per sample and reused during generation.

4.4 Push-Pull Attention Reweighting

After estimating $\mathcal{R}$, SPAR performs generation with a soft attention intervention at a single target decoder layer $\ell^\star$. Let $z_{\ell^\star,q,i}$ denote the pre-softmax attention logit from query position q to visual token v_i at layer $\ell^\star$. SPAR modifies visual-token attention logits as

$$z'_{\ell^\star,q,i} = z_{\ell^\star,q,i} + \alpha \cdot \mathbf{1}[v_i \in \mathcal{R}] - \beta \cdot \mathbf{1}[v_i \in \mathcal{V} \setminus \mathcal{R}], \tag{9}$$

where $\alpha, \beta > 0$. For clarity, both α and β are defined as non-negative bias magnitudes: ROI visual tokens receive a $+\alpha$ bias, while non-ROI visual tokens receive a $-\beta$ bias. In all main experiments, we use $\alpha = 5$ and $\beta = 2$. The intervention is applied uniformly across query positions at the target layer and affects only visual-token key positions. Text-token keys remain unchanged. Because the

operation is applied before the softmax, it reweights the attention distribution without removing any token from the context. Thus, SPAR preserves the full visual token sequence while biasing the attention budget toward the self-referential ROI.

4.5 Practical Inference

SPAR adds one ROI-estimation pass before generation. In the first pass, the model computes the self-referential ROI from middle-layer attention using Eq. (7) and Eq. (8). In the second pass, the model generates the answer while applying the push-pull reweighting in Eq. (9) at the target layer. The base VLM remains frozen throughout, and SPAR introduces no additional trainable parameters.

5 Experiments

We evaluate SPAR on a diverse set of vision-language benchmarks covering fine-grained visual question answering, general multimodal reasoning, hallucination-oriented evaluation, and visually subtle paired-image discrimination. Our experiments are designed to answer three questions. First, does SPAR improve final answer accuracy across benchmarks and VLM backbones? Second, are the gains due to the self-referential ROI estimated from PCEW attention, rather than a generic attention perturbation? Third, do the accuracy improvements coincide with better evidence routing, as measured by higher evidence retention R and lower mid-to-final evidence drop D? Unless otherwise stated, SPAR uses the self-referential ROI estimator in Section 4.3, selects the top-$k = 48$ visual tokens for 576-token visual sequences, and applies push-pull reweighting with $\alpha = 5$ and $\beta = 2$. For models with a different number of visual tokens, we keep the same ROI ratio.

5.1 Routing metric definitions

To test whether SPAR improves internal evidence flow, we measure how well visual evidence exposed in middle layers is preserved through later decoder layers. Following the evidence-retention formulation used in prior attention-retention analysis, we aggregate attention from answer-token query positions to visual-token key positions. For each decoder layer ℓ, this gives a normalized visual attention distribution $A_\ell \in \mathbb{R}^V$, where V is the number of visual tokens and $\sum_{i=1}^{V} A_\ell[i] = 1$. Given an ROI token set $\mathcal{R} \subseteq \{1, \dots, V\}$, we define the ROI attention mass at layer ℓ as

$$M_{\text{roi}}(\ell) = \sum_{i \in \mathcal{R}} A_\ell[i]. \tag{10}$$

We define the middle-layer peak ROI mass and the average final-layer ROI mass as

$$M_{\text{mid}} = \max_{\ell \in \mathcal{L}_{\text{mid}}} M_{\text{roi}}(\ell), \qquad M_{\text{final}} = \frac{1}{|\mathcal{L}_{\text{final}}|} \sum_{\ell \in \mathcal{L}_{\text{final}}} M_{\text{roi}}(\ell). \tag{11}$$

The mid-to-final evidence drop is

$$D = M_{\text{mid}} - M_{\text{final}}, \tag{12}$$

where lower D indicates less loss of ROI attention from middle to final layers. The evidence retention ratio is

$$R = \frac{M_{\text{final}}}{M_{\text{mid}} + \epsilon}, \tag{13}$$

where ϵ is a small constant for numerical stability. Higher R indicates stronger preservation of middle-layer evidence.

5.2 Main results and ROI specificity

Table 1 reports the main benchmark (Hudson & Manning, 2019; Liu et al., 2024d; Li et al., 2023b; Lu et al., 2022; Singh et al., 2019; Fu et al., 2026; Tong et al., 2024) results and the same-size random ROI control. We evaluate SPAR on four VLM backbones: LLaVA-1.5-7B (Liu et al., 2024b), Qwen2-VL-7B (Wang et al., 2024), InternVL3-2B (Zhu et al., 2025), and NVILA-8B (Liu et al., 2025). For compactness, the table abbreviates MMBench as MMB, ScienceQA as SQA, TextVQA as TVQA, MMVP-pair as MMVP-p, and MMVP-Ori-pair as MMVP-Ori-p. SPAR obtains the best score among

Table 1: Main benchmark results and random ROI control. All values are accuracy or percentage scores unless otherwise noted. MME reports the official raw score.

Model	Method	GQA	MMB	POPE	SQA	TVQA	MME	MMVP	MMVP-p	MMVP-Ori	MMVP-Ori-p
LLaVA-1.5	Base	61.80	72.80	74.65	62.22	55.62	1786.69	61.48	24.44	29.00	6.67
LLaVA-1.5	SPAR	**64.35**	**73.14**	**75.05**	**63.15**	**57.20**	**1792.61**	**62.96**	**31.11**	**33.33**	**7.33**
LLaVA-1.5	Random	60.45	72.35	73.65	60.97	53.60	1783.64	58.24	23.33	27.32	5.95
Qwen2-VL	Base	66.35	78.62	76.85	54.24	76.91	1541.08	64.44	39.26	38.43	30.67
Qwen2-VL	SPAR	**66.40**	**85.05**	**77.72**	**56.10**	**77.26**	**1552.03**	**67.03**	**41.48**	**40.12**	**32.00**
Qwen2-VL	Random	66.30	77.42	76.23	52.24	76.42	1489.37	62.48	31.85	35.72	25.81
InternVL3	Base	55.85	74.30	87.97	56.11	67.57	1751.69	76.30	54.81	43.48	40.00
InternVL3	SPAR	**56.75**	**77.45**	**88.89**	**61.28**	**68.20**	**1759.23**	**77.41**	**57.04**	**44.18**	**41.33**
InternVL3	Random	54.95	68.00	87.17	51.11	66.98	1743.82	75.19	54.07	43.13	39.33
NVILA	Base	59.55	64.30	86.79	40.48	68.23	1604.93	77.41	57.78	51.33	29.33
NVILA	SPAR	**61.00**	**73.20**	**88.02**	**42.19**	**70.21**	**1613.34**	**78.15**	**58.30**	**53.00**	**32.00**
NVILA	Random	58.70	59.15	85.81	38.14	65.60	1591.49	73.70	47.41	39.33	20.67

Table 2: Evidence routing metrics. Higher R and lower D indicate better preservation of PCEW evidence through later decoding. We report both raw values and changes relative to the baseline.

Model	Benchmark	Base R	SPAR R	$\Delta R \uparrow$	Base D	SPAR D	$\Delta D \downarrow$
LLaVA-1.5-7B	TextVQA	0.816	**0.961**	**+0.145**	0.138	**0.030**	**-0.108**
LLaVA-1.5-7B	POPE	0.724	**0.958**	**+0.234**	0.200	**0.030**	**-0.170**
LLaVA-1.5-7B	MMVP	0.772	**0.975**	**+0.203**	0.165	**0.017**	**-0.148**
LLaVA-1.5-7B	GQA	0.747	**0.927**	**+0.180**	0.176	**0.049**	**-0.127**
Qwen2-VL	TextVQA	0.768	**0.838**	**+0.070**	0.178	**0.123**	**-0.055**
Qwen2-VL	POPE	0.722	**0.827**	**+0.105**	0.191	**0.119**	**-0.072**
Qwen2-VL	MMVP	0.595	**0.668**	**+0.073**	0.221	**0.175**	**-0.046**
Qwen2-VL	GQA	0.658	**0.725**	**+0.067**	0.230	**0.184**	**-0.046**

the compared variants across the reported model–benchmark settings. In contrast, the same-size random ROI control is consistently worse than SPAR and often below the unmodified baseline. The random control uses the same number of visual tokens and the same push-pull intervention as SPAR, but samples the ROI uniformly from the visual-token sequence. This comparison shows that SPAR's gain is not a generic consequence of perturbing attention logits; it depends on targeting the self-referential ROI estimated from PCEW attention.

5.3 Does SPAR improve evidence routing?

Beyond final answer accuracy, SPAR is designed to preserve visual evidence exposed in the PCEW through later decoding. We therefore evaluate whether the performance gains are accompanied by improved routing metrics. A successful intervention should increase R, indicating stronger evidence retention, and decrease D, indicating less mid-to-final evidence loss. Table 2 shows that SPAR consistently improves evidence routing across benchmarks and model families. On LLaVA-1.5-7B, SPAR increases R and reduces D on TextVQA, POPE, MMVP, and GQA. The same trend is also observed on Qwen2-VL. These results connect the benchmark improvements in Table 1 to the internal routing problem identified in Section 3: SPAR helps preserve PCEW evidence through later decoding.

5.4 Why use middle-layer attention?

SPAR estimates its ROI from middle-layer PCEW attention. This design follows from our diagnosis: middle layers are less dominated by prompt-invariant sinks and are more responsive to the input query. We test this design choice by comparing different ROI construction strategies while keeping the intervention itself fixed. In all variants, we use the same top-k token budget and the same push-pull reweighting rule. Only the source used to construct the ROI is changed. It is important to distinguish the *ROI construction layers* from the *intervention layer*. The layer range in the "Construction" column of Table 3 denotes the decoder layers whose attention is used to estimate the ROI. It does not denote where the push-pull intervention is applied. For each ROI source s, we sweep the target intervention

layer $\ell \in \{0, \ldots, L-1\}$ over the full decoder depth and compute

$$\Delta_s(\ell) = \mathrm{Acc}(s, \ell) - \mathrm{Acc}_{\mathrm{base}}. \tag{14}$$

Thus, this ablation asks which ROI source provides the most reliable evidence signal for downstream reweighting, rather than which single target layer is best. We summarize this layer sweep using three quantities. $\Delta\mathrm{Acc}_{\ell^\star}$ is the accuracy change at the representative target layer used by SPAR. The positive-layer rate P_+ is the fraction of target layers that improve accuracy:

$$P_+(s) = \frac{1}{L} \sum_{\ell=0}^{L-1} \mathbf{1}\left[\Delta_s(\ell) > 0\right]. \tag{15}$$

The worst-layer drop W_{drop} is the largest degradation observed over the layer sweep:

$$W_{\mathrm{drop}}(s) = \left| \min_{\ell} \Delta_s(\ell) \right|. \tag{16}$$

These metrics are distinct from R and D: they measure the robustness of an ROI source across target intervention layers, rather than the preservation of ROI attention mass through depth.

Table 3 compares seven ROI construction strategies. Early, middle, and final variants construct ROIs from the corresponding decoder-depth regions. The all-layer variant averages attention over the entire decoder, while the single-middle variant uses only one representative middle layer. The shuffled-middle control preserves the middle-layer score distribution but destroys spatial correspondence, and the random control selects the same number of visual tokens without using attention. A more detailed description of each variant is provided in Appendix D.

Table 3: ROI source ablation and layer-sweep robustness. All variants use the same top-k token budget and the same push-pull intervention. $\Delta\mathrm{Acc}_{\ell^\star}$ reports the accuracy change at the representative target layer. P_+ measures the fraction of target layers that improve accuracy. W_{drop} measures the worst degradation over the layer sweep. These metrics are distinct from the evidence-routing metrics R and D.

ROI source	Construction	$\Delta\mathrm{Acc}_{\ell^\star} \uparrow$	$P_+ \uparrow$	$W_{\mathrm{drop}} \downarrow$
Early-layer average	$L0$–7	-0.556	53.1%	3.856
Middle-layer average	$L8$–23	**+0.925**	**79.6%**	**0.925**
Final-layer average	$L24$–31	+0.463	56.6%	1.748
All-layer average	$L0$–31	+0.564	50.0%	1.491
Single middle layer	$L16$	+0.720	65.6%	1.234
Shuffled middle average	shuffled $L8$–23	+0.514	50.0%	1.483
Random same-size ROI	random top-k	-1.250	45.4%	1.578

The middle-layer average is the most reliable ROI source in Table 3. It achieves the largest representative-layer gain, the highest positive-layer rate, and the smallest worst-layer drop. These results support the PCEW-based design of SPAR: middle-layer attention provides a more stable prompt-conditioned evidence signal than early, final, all-layer, shuffled, or random alternatives. We further observe that SPAR does not rely on a single isolated target layer. The target-layer sweep shows a broad effective band, approximately $L8$–$L20$ for LLaVA-1.5-7B, where push-pull reweighting consistently improves performance. This band overlaps with the PCEW identified by our diagnostic analysis. Thus, SPAR exploits a depth-level routing regime rather than a single layer-specific artifact. Full layer-sweep details are provided in Appendix D.

6 Conclusion

We studied fine-grained VLM failures from the perspective of decoder-side evidence routing. Our analysis shows a depth-asymmetric pattern: early and final layers tend to exhibit prompt-invariant sink behavior, while middle layers form a Prompt-Conditioned Evidence Window that exposes query-relevant visual evidence. Motivated by this observation, we proposed SPAR, a self-referential attention reweighting method that estimates an ROI from the model's own middle-layer attention and uses it to guide decoding. Across multiple VLM backbones and benchmarks, SPAR improves answer accuracy while increasing evidence retention and reducing mid-to-final evidence drop. Random ROI comparisons and ROI-source ablations further suggest that the improvement depends on using prompt-conditioned middle-layer evidence rather than arbitrary visual-token reweighting. SPAR remains limited by the quality of evidence exposed in middle layers and adds an additional ROI-estimation pass; we discuss these limitations and future directions in Appendix I.

AI USE STATEMENT

We received assistance from generative AI tools with text editing and code-related tasks.

ETHICS STATEMENT

This work studies inference-time attention reweighting for existing vision-language models and evaluates the method on established benchmarks. It does not introduce new training data or new model checkpoints. The model and benchmark assets and their available license and usage information are documented in Appendix A.9. SPAR is intended to improve the use of visual evidence, but better evidence routing should not be interpreted as a guarantee of factual correctness or suitability for high-stakes applications. Any deployment in sensitive settings should therefore involve task-specific evaluation and appropriate human oversight. The method's limitations and additional inference cost are discussed in Appendix I.

REPRODUCIBILITY STATEMENT

Section 4 specifies SPAR's two-pass inference procedure, including self-referential ROI estimation and the attention-logit intervention. Appendix A documents the default hyperparameters, decoder-depth partition, model backbones, decoding conventions, benchmark-specific evaluation protocols, answer normalization, and implementation details. The hardware and software environment and measured computational costs are reported in Appendix A.8. Appendices C and D further detail the computation of evidence-routing metrics and the ROI-source construction and layer-sweep protocols, respectively. The main Base, SPAR, and random-ROI comparisons use matched decoding and answer-normalization protocols (Appendix A.3); method-specific decoding for additional baselines is documented in Appendix G.

REFERENCES

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. Qwen technical report. *arXiv preprint arXiv:2309.16609*, 2023.

Zechen Bai, Pichao Wang, Tianjun Xiao, Tong He, Zongbo Han, Zheng Zhang, and Mike Zheng Shou. Hallucination of multimodal large language models: A survey. *arXiv preprint arXiv:2404.18930*, 2024.

Daniel Bolya, Cheng-Yang Fu, Xiaoliang Dai, Peizhao Zhang, Christoph Feichtenhofer, and Judy Hoffman. Token merging: Your vit but faster. *arXiv preprint arXiv:2210.09461*, 2022.

Haoran Chen, Junyan Lin, Xinghao Chen, Yue Fan, Jianfeng Dong, Xin Jin, Hui Su, Jinlan Fu, and Xiaoyu Shen. Multimodal language models see better when they look shallower. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 6688–6706, 2025a.

Liang Chen, Haozhe Zhao, Tianyu Liu, Shuai Bai, Junyang Lin, Chang Zhou, and Baobao Chang. An image is worth 1/2 tokens after layer 2: Plug-and-play inference acceleration for large vision-language models. In *European Conference on Computer Vision*, pp. 19–35. Springer, 2024a.

Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, et al. Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 24185–24198, 2024b.

Zhiyuan Chen, Yuecong Min, Jie Zhang, Bei Yan, Jiahao Wang, Xiaozhen Wang, and Shiguang Shan. A survey of multimodal hallucination evaluation and detection. *arXiv preprint arXiv:2507.19024*, 2025b.

Wei-Lin Chiang, Zhuohan Li, Ziqing Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E Gonzalez, et al. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality. *See https://vicuna. lmsys. org (accessed 14 April 2023)*, 2(3):6, 2023.

Parsa Esmaeilkhani and Longin Jan Latecki. Direct visual grounding by directing attention of visual tokens. *arXiv preprint arXiv:2511.12738*, 2025.

Wenfeng Feng, Hongxiang Wang, Jianlong Wang, Xin Zhang, Jingjing Zhao, Yueyue Liang, Xiang Chen, and Duokui Han. EDIT: Enhancing vision transformers by mitigating attention sink through an encoder-decoder architecture. *arXiv preprint arXiv:2504.06738*, 2025.

Chaoyou Fu, Peixian Chen, Yunhang Shen, Yulei Qin, Mengdan Zhang, Xu Lin, Jinrui Yang, Xiawu Zheng, Ke Li, Xing Sun, et al. Mme: A comprehensive evaluation benchmark for multimodal large language models. *Advances in Neural Information Processing Systems*, 38, 2026.

Gregor Geigle, Radu Timofte, and Goran Glavaš. Does object grounding really reduce hallucination of large vision-language models? *arXiv preprint arXiv:2406.14492*, 2024.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Xiangming Gu, Tianyu Pang, Chao Du, Qian Liu, Fengzhuo Zhang, Cunxiao Du, Ye Wang, and Min Lin. When attention sink emerges in language models: An empirical view. *arXiv preprint arXiv:2410.10781*, 2024.

Qidong Huang, Xiaoyi Dong, Pan Zhang, Bin Wang, Conghui He, Jiaqi Wang, Dahua Lin, Weiming Zhang, and Nenghai Yu. Opera: Alleviating hallucination in multi-modal large language models via over-trust penalty and retrospection-allocation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13418–13427, 2024a.

Wen Huang, Hongbin Liu, Minxin Guo, and Neil Gong. Visual hallucinations of multi-modal large language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 9614–9631, 2024b.

Drew A Hudson and Christopher D Manning. Gqa: A new dataset for real-world visual reasoning and compositional question answering. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 6700–6709, 2019.

Zhangqi Jiang, Junkai Chen, Beier Zhu, Tingjin Luo, Yankun Shen, and Xu Yang. Devils in middle layers of large vision-language models: Interpreting, detecting and mitigating object hallucinations via attention lens. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 25004–25014, 2025.

Omri Kaduri, Shai Bagon, and Tali Dekel. What's in the image? a deep-dive into the vision of vision language models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 14549–14558, 2025.

Seil Kang, Jinyeong Kim, Junhyeok Kim, and Seong Jae Hwang. See what you are told: Visual attention sink in large multimodal models. *arXiv preprint arXiv:2503.03321*, 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 *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 9339–9350, 2025b.

Samir Khaki, Junxian Guo, Jiaming Tang, Shang Yang, Yukang Chen, Konstantinos N. Plataniotis, Yao Lu, Song Han, and Zhijian Liu. SparseVILA: Decoupling visual sparsity for efficient VLM inference. *arXiv preprint arXiv:2510.17777*, 2025.

Jeonghwan Kim and Heng Ji. Finer: Investigating and enhancing fine-grained visual concept recognition in large vision language models. *arXiv preprint arXiv:2402.16315*, 2024.

Sicong Leng, Hang Zhang, Guanzheng Chen, Xin Li, Shijian Lu, Chunyan Miao, and Lidong Bing. Mitigating object hallucinations in large vision-language models through visual contrastive decoding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13872–13882, 2024.

Feng Li, Renrui Zhang, Hao Zhang, Yuanhan Zhang, Bo Li, Wei Li, Zejun Ma, and Chunyuan Li. Llava-next-interleave: Tackling multi-image, video, and 3d in large multimodal models. *arXiv preprint arXiv:2407.07895*, 2024.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*, pp. 19730–19742. PMLR, 2023a.

Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Wayne Xin Zhao, and Ji-Rong Wen. Evaluating object hallucination in large vision-language models. *arXiv preprint arXiv:2305.10355*, 2023b.

Yuan-Hong Liao, Rafid Mahmood, Sanja Fidler, and David Acuna. Can large vision-language models correct semantic grounding errors by themselves? In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 14667–14678, 2025.

Hanchao Liu, Wenyuan Xue, Yifei Chen, Dapeng Chen, Xiutian Zhao, Ke Wang, Liping Hou, Rongjun Li, and Wei Peng. A survey on hallucination in large vision-language models. *arXiv preprint arXiv:2402.00253*, 2024a.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36:34892–34916, 2023.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 26296–26306, 2024b.

Shi Liu, Kecheng Zheng, and Wei Chen. Paying more attention to image: A training-free method for alleviating hallucination in lvlms. In *European Conference on Computer Vision*, pp. 125–140. Springer, 2024c.

Yuan Liu, Haodong Duan, Yuanhan Zhang, Bo Li, Songyang Zhang, Wangbo Zhao, Yike Yuan, Jiaqi Wang, Conghui He, Ziwei Liu, et al. Mmbench: Is your multi-modal model an all-around player? In *European conference on computer vision*, pp. 216–233. Springer, 2024d.

Zhijian Liu, Ligeng Zhu, Baifeng Shi, Zhuoyang Zhang, Yuming Lou, Shang Yang, Haocheng Xi, Shiyi Cao, Yuxian Gu, Dacheng Li, et al. Nvila: Efficient frontier visual language models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 4122–4134, 2025.

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.

Shweta Mahajan, Hoang Le, Hyojin Park, Farzad Farhadzadeh, Munawar Hayat, and Fatih Porikli. Attention guided alignment in efficient vision-language models. *arXiv preprint arXiv:2511.17793*, 2025.

Subba Reddy Oota, Emin Çelik, Fatma Deniz, and Mariya Toneva. Speech language models lack important brain-relevant semantics. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 8503–8528, 2024.

Yuheng Shi, Xiaohuan Pei, Minjing Dong, and Chang Xu. Catching the details: Self-distilled RoI predictors for fine-grained MLLM perception. *arXiv preprint arXiv:2509.16944*, 2025.

Amanpreet Singh, Vivek Natarajan, Meet Shah, Yu Jiang, Xinlei Chen, Dhruv Batra, Devi Parikh, and Marcus Rohrbach. Towards vqa models that can read. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 8317–8326, 2019.

Shengbang Tong, Zhuang Liu, Yuexiang Zhai, Yi Ma, Yann LeCun, and Saining Xie. Eyes wide shut? exploring the visual shortcomings of multimodal llms. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9568–9578, 2024.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, et al. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*, 2024.

Hongxuan Wu, Yukun Zhang, and Xueqing Zhou. How vision becomes language: A layer-wise information-theoretic analysis of multimodal reasoning. *arXiv preprint arXiv:2602.15580*, 2026.

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

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. Efficient streaming language models with attention sinks. *arXiv preprint arXiv:2309.17453*, 2023.

Long Xing, Qidong Huang, Xiaoyi Dong, Jiajie Lu, Pan Zhang, Yuhang Zang, Yuhang Cao, Conghui He, Jiaqi Wang, Feng Wu, et al. Pyramiddrop: Accelerating your large vision-language models via pyramid visual redundancy reduction. *arXiv preprint arXiv:2410.17247*, 2024.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025a.

Senqiao Yang, Yukang Chen, Zhuotao Tian, Chengyao Wang, Jingyao Li, Bei Yu, and Jiaya Jia. Visionzip: Longer is better but not necessary in vision language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 19792–19802, 2025b.

Shukang Yin, Chaoyou Fu, Sirui Zhao, Ke Li, Xing Sun, Tong Xu, and Enhong Chen. A survey on multimodal large language models. *National Science Review*, 11(12):nwae403, 2024.

Weihao Yu, Zhengyuan Yang, Linjie Li, Jianfeng Wang, Kevin Lin, Zicheng Liu, Xinchao Wang, and Lijuan Wang. Mm-vet: Evaluating large multimodal models for integrated capabilities. *arXiv preprint arXiv:2308.02490*, 2023.

Yuan Zhang, Chun-Kai Fan, Junpeng Ma, Wenzhao Zheng, Tao Huang, Kuan Cheng, Denis Gudovskiy, Tomoyuki Okuno, Yohei Nakata, Kurt Keutzer, et al. Sparsevlm: Visual token sparsification for efficient vision-language model inference. *arXiv preprint arXiv:2410.04417*, 2024.

Jianfei Zhao, Feng Zhang, Xin Sun, Chong Feng, and Zhixing Tan. Tell model where to look: Mitigating hallucinations in MLLMs by vision-guided attention. *arXiv preprint arXiv:2511.20032*, 2025.

Fengbin Zhu, Ziyang Liu, Xiang Yao Ng, Haohui Wu, Wenjie Wang, Fuli Feng, Chao Wang, Huanbo Luan, and Tat Seng Chua. Mmdocbench: Benchmarking large vision-language models for fine-grained visual document understanding. *arXiv preprint arXiv:2410.21311*, 2024.

Jinguo Zhu, Weiyun Wang, Zhe Chen, Zhaoyang Liu, Shenglong Ye, Lixin Gu, Hao Tian, Yuchen Duan, Weijie Su, Jie Shao, et al. Internvl3: Exploring advanced training and test-time recipes for open-source multimodal models. *arXiv preprint arXiv:2504.10479*, 2025.

A EXPERIMENTAL PROTOCOL AND EVALUATION DETAILS

This appendix provides the experimental details needed to reproduce the main results and interpret the reported scores. We describe the SPAR configuration, depth partition, model backbones, benchmark-specific evaluation protocols, answer normalization, and implementation conventions. Unless otherwise stated, all comparisons use the same base model, preprocessing, prompt format, decoding setting, and answer normalization within each benchmark. For all benchmarks, we use the same evaluation split and scoring protocol for the baseline and SPAR comparisons. When subset-based diagnostic analyses are reported, the subset construction is held fixed across all compared methods.

A.1 SPAR CONFIGURATION

Table 4 summarizes the default SPAR configuration used in the main experiments. SPAR estimates the ROI from the model's own middle-layer attention and applies a push-pull bias to visual-token attention logits at a single target decoder layer.

Table 4: Default SPAR configuration. For model families with different numbers of decoder layers or visual tokens, we use the same normalized depth partition and the same ROI token ratio.

Component	Setting
Method	SPAR: Self-Referential Prompt-Conditioned Attention Reweighting
Base model state	Frozen; no parameter updates
Number of passes	Two passes: ROI estimation pass + generation pass
ROI source	Model's own middle-layer attention
Depth partition	Early 25%, middle 50%, final 25% of decoder layers
PCEW / middle layers for LLaVA-1.5-7B	$L8$–$L23$
Early layers for LLaVA-1.5-7B	$L0$–$L7$
Final layers for LLaVA-1.5-7B	$L24$–$L31$
Visual tokens for LLaVA-1.5-7B	576 visual tokens, 24×24 grid
ROI budget	Top-48 visual tokens, 8.3% of 576 visual tokens
ROI budget for other backbones	Same visual-token ratio as LLaVA-1.5-7B
ROI query position	Final prompt token before answer generation
Attention score for ROI	Average over middle layers and attention heads
Push strength	$\alpha = 5$ for ROI visual tokens
Pull strength	$\beta = 2$ for non-ROI visual tokens
Bias convention	ROI receives $+\alpha$; non-ROI visual tokens receive $-\beta$
Intervention target	Single decoder layer $\ell^\star$
Default target layer for LLaVA-1.5-7B	$\ell^\star = 20$ unless otherwise stated
Token handling	No visual-token pruning; no hard masking
Text-token keys	Unchanged
Trainable parameters	None

A.2 DEPTH PARTITION ACROSS MODEL FAMILIES

SPAR relies on the Prompt-Conditioned Evidence Window (PCEW), which is defined by normalized decoder depth rather than by model-specific layer indices. For a model with L decoder layers, we use the 25/50/25 partition:

$$\mathcal{L}_{\text{early}} = \{0, \ldots, \lfloor 0.25L \rfloor - 1\}, \tag{17}$$

$$\mathcal{L}_{\text{mid}} = \{\lfloor 0.25L \rfloor, \ldots, \lfloor 0.75L \rfloor - 1\}, \tag{18}$$

$$\mathcal{L}_{\text{final}} = \{\lfloor 0.75L \rfloor, \ldots, L - 1\}. \tag{19}$$

The middle region $\mathcal{L}_{\text{mid}}$ is used for self-referential ROI estimation. For 32-layer LLaVA-1.5-7B, this gives early layers $L0$–$L7$, middle layers $L8$–$L23$, and final layers $L24$–$L31$. The same normalized rule is applied to other VLM backbones.

A.3 MODELS AND DECODING

Table 5 summarizes the VLM backbones used in the main benchmark evaluation. For each model family, we follow the corresponding recommended image preprocessing, tokenizer, and chat template. For the main Base, SPAR, and random-ROI comparisons, decoding settings and answer normalization

are held fixed within each benchmark. Method-specific decoding for the additional VCD/OPERA comparison is specified in Appendix G.

Table 5: Model backbones used in the main benchmark evaluation. All models are evaluated with the same SPAR protocol: self-referential ROI estimation from middle-layer attention followed by single-layer push-pull reweighting.

Model family	Checkpoint / variant	Notes
LLaVA-1.5	LLaVA-1.5-7B	Main LLaVA-family backbone
Qwen2-VL	Qwen2-VL-7B	Main Qwen-family backbone
InternVL3	InternVL3-2B	Additional VLM backbone for cross-family evaluation
NVILA	NVILA-8B	Additional VLM backbone for cross-family evaluation

Unless a benchmark requires a different format, we use deterministic decoding for evaluation. For multiple-choice benchmarks, we use the benchmark-specific prompt format and parse the model output into the corresponding option space. For binary yes/no benchmarks, we map model outputs to the allowed labels using deterministic normalization. All reported baseline, SPAR, and random ROI results use the same decoding and normalization protocol within each benchmark.

A.4 Benchmarks and evaluation protocols

We evaluate SPAR on benchmarks that cover fine-grained visual question answering, hallucination resistance, multimodal reasoning, broad perception/cognition diagnostics, and paired-image visual discrimination. Unless otherwise stated, all scores are computed with the corresponding official evaluation protocol, recommended preprocessing, and benchmark-specific answer formatting.

GQA. GQA (Hudson & Manning, 2019) evaluates compositional visual reasoning over objects, attributes, and relations. We report standard accuracy on the evaluated split. Predictions are normalized according to the benchmark evaluation interface before scoring.

MMBench. MMBench (Liu et al., 2024d) evaluates broad multimodal understanding with multiple-choice questions. We report benchmark accuracy under the official evaluation setting. The model output is mapped to the corresponding answer option before scoring.

POPE. POPE (Li et al., 2023b) is a hallucination-oriented benchmark with binary yes/no questions about object existence. We report accuracy after mapping model outputs to $\{\text{yes}, \text{no}\}$ using a deterministic normalization rule. When F1 is reported in supplementary analyses, it is computed using the same binary normalization.

ScienceQA. ScienceQA (Lu et al., 2022) is a multimodal multiple-choice reasoning benchmark. We report multiple-choice accuracy. The model output is parsed as either a choice letter or answer string and mapped to the corresponding option under the official evaluation logic.

TextVQA. TextVQA (Singh et al., 2019) evaluates question answering that often requires reading scene text or localized textual evidence. We report VQA-style exact-match accuracy. Predictions are normalized using standard text canonicalization such as lowercasing and punctuation removal, following the official evaluation implementation.

MME. MME (Fu et al., 2026) is a broad diagnostic suite covering perception-oriented and cognition-oriented categories. We report the official aggregated raw score. Because MME is not expressed as a percentage accuracy, we do not average it with other benchmark scores.

MMVP. MMVP (Tong et al., 2024) evaluates sensitivity to subtle visual differences using paired-image or paired-question settings. We report both image-level accuracy and pairwise accuracy when available. Pairwise accuracy is stricter: a pair is counted as correct only when both corresponding predictions are correct.

MMVP-Ori. MMVP-Ori (Tong et al., 2024) follows the same evaluation interface as MMVP but uses the original benchmark organization. We report both image-level and pairwise accuracy. Because each pair contributes a coarse unit to the final score, small changes in the number of correct pairs can produce visible percentage changes.

MMVet. When MMVet (Yu et al., 2023) is included, we report the official score produced by the benchmark evaluator and use the recommended answer normalization and scoring pipeline.

A.5 Answer normalization and score reporting

We follow benchmark-specific answer normalization rules. For open-ended VQA-style datasets, predictions are canonicalized before scoring using lowercasing, punctuation removal, and benchmark-specific string normalization. For multiple-choice datasets, the model output is mapped to the corresponding option label before computing accuracy. For binary yes/no datasets, outputs are deterministically mapped to the allowed label set. For MME, we report the official raw score and do not convert it into percentage accuracy.

When a table contains both percentage-style scores and MME, the caption explicitly notes that MME is reported on a different scale. When a table contains pairwise results, the caption specifies whether the score is image-level or pair-level accuracy.

A.6 Implementation details

SPAR is implemented as an inference-time attention-logit modification. The first pass computes the ROI from middle-layer attention. The second pass runs generation with push-pull reweighting at the target decoder layer. The intervention modifies only visual-token key logits and leaves text-token keys unchanged. No model weights are updated.

In implementation, the push-pull bias is applied to the pre-softmax attention logits:

$$z'_{\ell^\star,q,i} = z_{\ell^\star,q,i} + \alpha \cdot \mathbf{1}[v_i \in \mathcal{R}] - \beta \cdot \mathbf{1}[v_i \in \mathcal{V} \setminus \mathcal{R}]. \tag{20}$$

Both α and β are non-negative bias magnitudes. The default setting is $\alpha = 5$, $\beta = 2$, and $k = 48$ for 576 visual tokens.

A.7 Reporting conventions

Throughout the paper, numbers in parentheses denote absolute changes relative to the corresponding baseline. For example, $64.35\ (+2.55)$ means that the method improves the baseline score by 2.55 points. For routing metrics, higher R and lower D indicate better evidence preservation. For benchmark scores, higher is better unless otherwise specified.

A.8 Compute Resources

SPAR is an inference-time method and does not require model training or fine-tuning. The main additional cost comes from one ROI-estimation pass before generation. Table 6 summarizes the compute environment and runtime measurements used for the completed SPAR runs.

Output-length-controlled latency and memory. We additionally profile inference with the output length fixed to 1, 8, 32, or 128 tokens. Baseline and SPAR use the same model and hardware within this profiling study. SPAR's total latency includes one prompt-only ROI-analysis pass and one autoregressive generation; the ROI is computed once per sample and reused throughout generation. These controlled-length measurements complement the representative runtime summary in Table 6.

The prompt-only ROI analysis takes approximately 0.224–0.291 seconds per sample. The measured total latency ratio is $2.15\times$ for one-token output and $1.06\times$ for 128-token output. The ratios are not strictly monotonic at intermediate lengths: they are $1.20\times$ at 8 tokens and $1.33\times$ at 32 tokens. Peak allocated memory increases by 0.06–0.41 GiB across the reported settings. These measurements show that the relative cost is particularly pronounced for very short answers and is smaller in the measured 128-token setting; they do not imply a uniform overhead across workloads.

Table 6: Compute resources for the main SPAR experiments. SPAR inference time includes one ROI-estimation pass and one intervened generation pass.

Item	Value
GPU type	NVIDIA RTX A6000
Number of GPUs per run	1
GPU memory	48 GB per GPU; 46,068 MiB reported
CPU / RAM	2× Intel Xeon Gold 6248R, 96 logical CPUs / 503 GiB RAM
Python version	3.10.18
PyTorch version	2.1.2+cu121
CUDA version	CUDA 12.1 runtime; NVIDIA driver 535.183.01
Transformers version	4.37.2
Model precision	FP16 for LLaVA-1.5 SPAR runs
Baseline inference time per sample	≈ 0.26 s/sample
SPAR inference time per sample	≈ 0.54 s/sample
Approx. overhead from ROI-estimation pass	≈ 0.24 s/sample, about $0.94\times$ baseline
Measured GPU-hours for representative completed runs	≈ 11.1 single-GPU hours for SPAR GQA+POPE sweeps

Table 7: Output-length-controlled latency and memory profiling. The latency ratio is the total SPAR time divided by the baseline time. Peak memory denotes peak allocated GPU memory.

Output	Total latency (s/sample)		SPAR/Base	Peak memory (GiB)	
tokens	Base	SPAR	ratio	Base	SPAR
1	0.222	0.477	2.15×	13.79	14.20
8	0.980	1.178	1.20×	14.22	14.28
32	3.069	4.088	1.33×	14.23	14.30
128	12.178	12.851	1.06×	14.32	14.40

A.9 Existing Assets and Licenses

We use existing VLM backbones and public evaluation benchmarks. Table 8 summarizes the assets used in this work and the license or usage information available from the corresponding official repositories, model cards, dataset cards, or project pages. For assets whose official page does not explicitly specify a license, we report this status and note the relevant source dependencies.

Table 8: Existing assets used in the paper. License and usage terms are reported according to the corresponding official pages, model cards, dataset cards, or project repositories.

Asset	Type	Version / source	License / usage terms
LLaVA-1.5-7B	VLM backbone	LLaVA-1.5 model card	Llama 2 Community License
Qwen2-VL	VLM backbone	Official Hugging Face model card	Apache-2.0
Qwen3-VL	VLM backbone	Official Hugging Face model card	Apache-2.0
GQA	Benchmark	Official GQA dataset page	Not explicitly specified; images are from COCO/Flickr and scene graphs are based on Visual Genome
MMBench	Benchmark	Hugging Face dataset card	CC BY 4.0
POPE	Benchmark	GitHub repository	MIT
ScienceQA	Benchmark	GitHub repository	CC BY-NC-SA 4.0 for dataset; MIT for code
TextVQA	Benchmark	Hugging Face dataset card	CC BY 4.0
MME	Benchmark	Hugging Face dataset mirror	Apache-2.0 in HF mirror; original image/source terms may still apply
MMVP	Benchmark	Hugging Face dataset card	MIT
MMVet	Benchmark	GitHub repository	CC BY-NC 4.0 for dataset; Apache-2.0 for code

B Additional Diagnostic Results

This appendix provides additional diagnostic analyses supporting the PISR–PCEW distinction introduced in Section 3. The main paper presents compact diagnostic figures showing that prompt-invariant sink overlap and low-rank spectral concentration are aligned across decoder depth. Here, we include extended diagnostic plots that are not duplicated in the main text. These analyses are intended as structural diagnostics, not as standalone causal evidence. The causal relevance of PCEW-based ROI estimation is evaluated in the main experiments through SPAR, random ROI controls, and ROI-source ablations.

B.1 Full prompt-invariant sink profiles

We measure prompt-invariant sink behavior by comparing high-attention visual-token sets across different prompts for the same image. For a given image I, prompt p, and decoder layer ℓ, let $\mathcal{S}_\ell^p(I)$ denote the set of top-attended visual tokens at that layer. We compute pairwise overlap between prompts using intersection-over-union:

$$\mathrm{SinkIoU}_\ell(I; p_a, p_b) = \frac{|\mathcal{S}_\ell^{p_a}(I) \cap \mathcal{S}_\ell^{p_b}(I)|}{|\mathcal{S}_\ell^{p_a}(I) \cup \mathcal{S}_\ell^{p_b}(I)|}. \tag{21}$$

We then average over prompt pairs and images:

$$\mathrm{SinkIoU}_\ell = \mathbb{E}_I \mathbb{E}_{p_a \neq p_b} \left[\mathrm{SinkIoU}_\ell(I; p_a, p_b) \right]. \tag{22}$$

High Sink IoU indicates that the same visual tokens repeatedly attract attention across different prompts, suggesting prompt-invariant sink behavior.

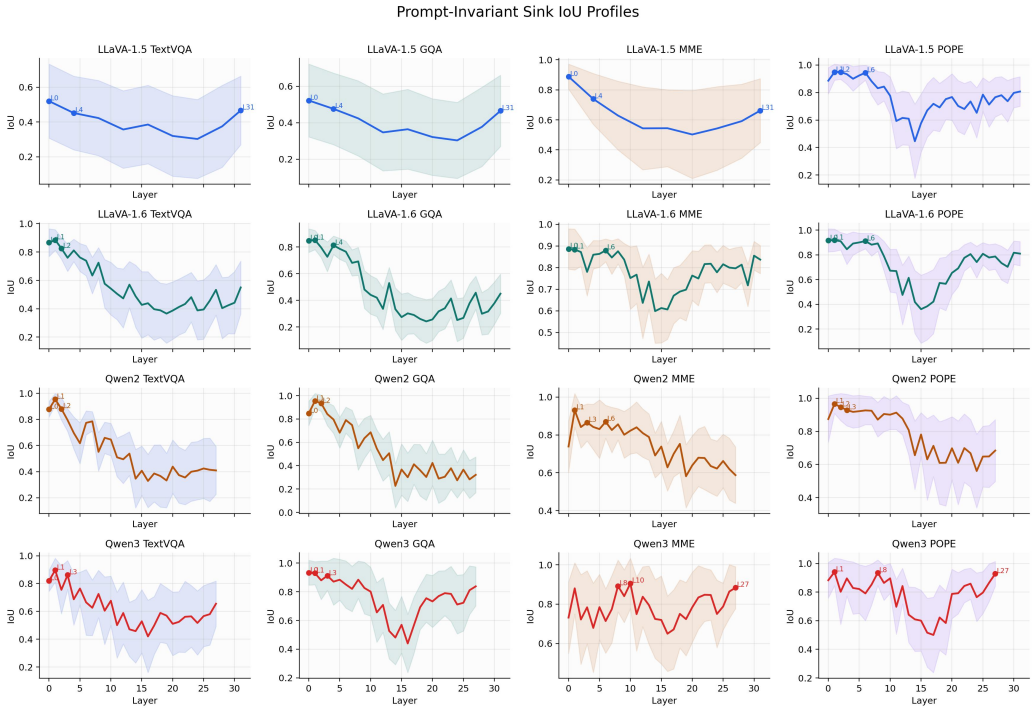


Figure 4: Full prompt-invariant Sink IoU profiles across model families and benchmarks. Each curve shows how consistently the same visual tokens are selected as high-attention sink tokens across different prompts for the same image. Higher values indicate stronger prompt-invariant sink behavior.

Figure 4 provides the full Sink IoU profiles across model families and benchmarks. The compact figures in the main text show representative ranking and alignment results, while this appendix figure shows the complete depth-wise behavior. Across settings, early and late regions often exhibit stronger prompt-invariant overlap than the middle region, supporting the Prompt-Invariant Sink Regime (PISR) and Prompt-Conditioned Evidence Window (PCEW) distinction.

B.2 Additional Sink IoU–spectral alignment results

The main paper shows a compact cross-model alignment between Sink IoU and top-1 spectral concentration ρ_1. Here, we provide the extended Qwen-family alignment results across additional benchmarks. Each point corresponds to a decoder layer. The x-axis is mean Sink IoU, and the y-axis is top-1 spectral concentration ρ_1. A positive association indicates that layers repeatedly selecting the same sink tokens across prompts also exhibit stronger low-rank spectral dominance.

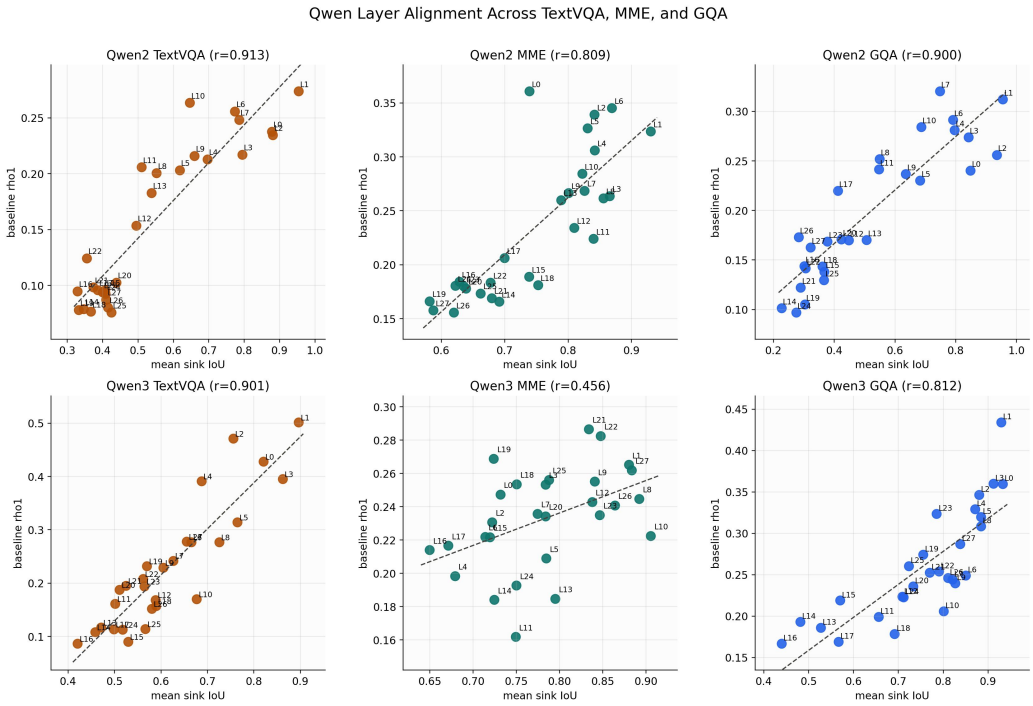


Figure 5: Additional Sink IoU–ρ_1 alignment results for Qwen-family VLMs across benchmarks. Each point denotes a decoder layer. The positive association between Sink IoU and spectral concentration supports the generality of the PISR–PCEW distinction beyond a single backbone.

Figure 5 extends the alignment analysis beyond the compact plot in the main paper. The same qualitative trend appears across Qwen-family models: layers with stronger prompt-invariant sink overlap tend to have stronger low-rank spectral concentration. This supports the use of spectral concentration as a complementary diagnostic for identifying sink-dominated layers. We note that the strength of this association can vary by benchmark, which is expected because task format and answer type affect decoder attention dynamics.

B.3 Spectral dominance rankings for Qwen-family models

We further report layer rankings by top-1 spectral concentration for Qwen-family models. These rankings are not intended to replace the Sink IoU–ρ_1 alignment analysis. Instead, they show that low-rank spectral dominance is concentrated in a structured subset of layers, and that the exact ranking can be architecture-specific.

Figure 6 shows that Qwen-family models also exhibit layer-dependent spectral concentration. The highest-ρ_1 layers are not uniformly distributed across depth, indicating that low-rank attention structure is layer-specific. This supports the broader diagnostic claim that decoder layers form distinct routing regimes, while also showing that exact layer indices should not be transferred directly across backbones.

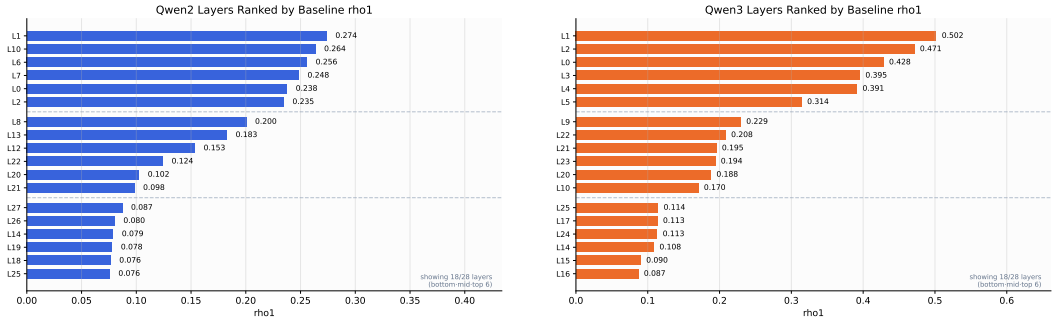


Figure 6: Layer rankings by top-1 spectral concentration ρ_1 for Qwen2-VL and Qwen3-VL. The rankings show that spectral dominance is concentrated in a subset of layers rather than being uniform across decoder depth. The exact high-ρ_1 layers differ across architectures, which motivates using normalized depth partitions and model-specific diagnostics rather than assuming identical layer indices across VLM families.

B.4 Interpretation and Limitations of the Diagnostics

The diagnostics in this appendix are used to characterize decoder-layer regimes, not to prove causality by themselves. High Sink IoU indicates that attention repeatedly selects similar visual tokens across prompts. High ρ_k indicates that visual-attention logits are concentrated in a low-dimensional subspace. When both are high, we interpret the layer as belonging to a sink-dominated regime. When both are lower, as in the middle-depth region, we interpret the layer as more prompt-conditioned and less dominated by stable sink patterns.

These diagnostics motivate the design of SPAR, but SPAR's causal relevance is evaluated through intervention experiments. In particular, the main text compares SPAR against same-size random ROI controls and ROI-source ablations. Those experiments test whether using PCEW-derived visual evidence improves both final performance and evidence-routing metrics. Thus, the diagnostic results in this appendix support the PISR–PCEW framing, while the main experiments test whether that framing leads to an effective intervention.

C Evidence Routing Metric Details

This appendix provides additional details for the evidence-routing metrics used in Section 5.3. The main text reports the per-benchmark routing results. Here, we describe the metric computation protocol and provide model-level averages that summarize the overall routing trend without duplicating the main table.

C.1 Metric Computation Protocol

For each image–query example, we measure how much attention mass assigned to an ROI in middle layers is preserved in final layers. Given an ROI token set $\mathcal{R}$, the layer-wise ROI attention mass is

$$M_{\text{roi}}(\ell) = \sum_{i \in \mathcal{R}} A_\ell[i], \tag{23}$$

where A_ℓ is the normalized visual attention distribution at decoder layer ℓ, aggregated over answer-token query positions and attention heads.

We define

$$M_{\text{mid}} = \max_{\ell \in \mathcal{L}_{\text{mid}}} M_{\text{roi}}(\ell), \qquad M_{\text{final}} = \frac{1}{|\mathcal{L}_{\text{final}}|} \sum_{\ell \in \mathcal{L}_{\text{final}}} M_{\text{roi}}(\ell). \tag{24}$$

The retention ratio and evidence drop are

$$R = \frac{M_{\text{final}}}{M_{\text{mid}} + \epsilon}, \qquad D = M_{\text{mid}} - M_{\text{final}}, \tag{25}$$

where ϵ is a small constant for numerical stability. Higher R and lower D indicate better evidence preservation.

For paired diagnostic comparisons, the ROI token set is held fixed within each example when comparing baseline and SPAR. This ensures that changes in R and D reflect changes in decoder routing rather than changes in the definition of the ROI. All routing metrics use the same early/middle/final depth partition as the main method.

C.2 Model-level routing summary

The main text reports per-benchmark R and D values. To summarize the overall trend without repeating the main table, Table 9 reports model-level averages across the diagnostic benchmarks used for routing analysis. SPAR improves the average retention ratio R and reduces the average evidence drop D for both model families.

Table 9: Model-level average evidence-routing metrics over TextVQA, POPE, MMVP, and GQA. These are the benchmarks for which we compute the full R, D routing diagnostics for both LLaVA-1.5-7B and Qwen2-VL. Higher R and lower D indicate better evidence preservation.

Model	Base R	SPAR R	$\Delta R \uparrow$	Base D	SPAR D	$\Delta D \downarrow$
LLaVA-1.5-7B	0.765	**0.955**	**+0.191**	0.170	**0.032**	**-0.138**
Qwen2-VL	0.686	**0.765**	**+0.079**	0.205	**0.150**	**-0.055**

C.3 Interpreting R and D

The retention ratio R and evidence drop D are complementary. A high R indicates that final layers preserve a large fraction of the strongest middle-layer ROI attention mass. A low D indicates that the absolute amount of ROI attention lost from middle to final layers is small. Thus, the most desirable pattern is $R \uparrow$ and $D \downarrow$.

SPAR achieves this pattern on both model families in Table 9. The improvement is larger for LLaVA-1.5-7B, but Qwen2-VL shows the same direction of change. This supports the interpretation that SPAR improves evidence flow through the decoder rather than merely changing final answer probabilities.

D ROI Source Construction and Layer-Sweep Protocol

This appendix details how we construct the ROI-source variants used in the ROI-source ablation and how we compute the layer-sweep robustness metrics. The main text reports the compact comparison showing that middle-layer PCEW attention provides the strongest ROI source. Here, we describe the construction of each ROI variant, the layer-sweep protocol, and the broad effective layer band observed for SPAR.

D.1 ROI source variants

SPAR estimates a self-referential ROI from middle-layer PCEW attention. To test whether this design choice is important, we compare it against alternative ROI sources constructed from different decoder-depth regions. All ROI variants use the same top-k token budget and the same push-pull intervention rule. Thus, the only difference across variants is the source used to score visual tokens.

For a source layer set $\mathcal{L}_s$, we compute a visual-token evidence score

$$e_i^{(s)} = \frac{1}{|\mathcal{L}_s| H} \sum_{\ell \in \mathcal{L}_s} \sum_{h=1}^{H} A^{\ell,h}(q_{\text{prompt}}, v_i), \tag{26}$$

and select the top-k tokens as the ROI:

$$\mathcal{R}_s = \text{TopK}(\mathbf{e}^{(s)}, k). \tag{27}$$

For random ROI, no attention score is used for ROI estimation; the same number of visual tokens is sampled uniformly from the visual-token sequence. For shuffled ROI, we first compute the middle-layer evidence score and then randomly permute the evidence scores across visual-token positions before selecting top-k tokens.

Table 10: ROI source variants used in the ROI-source ablation. All variants use the same top-k token budget and the same push-pull intervention; only the source of the ROI differs. Layer ranges are shown for the 32-layer LLaVA-1.5-7B backbone.

ROI source	Layer range / construction	Description
Early-layer average	$L0$–$L7$	ROI is computed from early-layer attention.
Middle-layer average	$L8$–$L23$	ROI is computed from PCEW attention; this is the default SPAR source.
Final-layer average	$L24$–$L31$	ROI is computed from final-layer attention.
All-layer average	$L0$–$L31$	ROI is computed by averaging attention across all decoder layers.
Single middle layer	$L16$	ROI is computed from one representative middle layer.
Shuffled middle-layer average	shuffled $L8$–$L23$	Middle-layer evidence scores are computed and then spatially shuffled.
Random same-size ROI	random top-k	ROI positions are sampled uniformly without using attention scores.

D.2 Layer-sweep protocol

For each ROI source s, we evaluate how the intervention behaves when applied at different target decoder layers. Let $\ell \in \{0, \ldots, L-1\}$ denote the target intervention layer. For each source s and target layer ℓ, we run the same push-pull intervention and compute the accuracy change relative to the unmodified baseline:

$$\Delta_s(\ell) = \mathrm{Acc}(s, \ell) - \mathrm{Acc}_{\mathrm{base}}. \tag{28}$$

This sweep is used to measure not only whether an ROI source can improve performance at a representative layer, but also whether the source remains stable across decoder depth. A strong ROI source should improve accuracy at the representative target layer, remain helpful across many target layers, and avoid severe degradation at suboptimal target layers.

D.3 Layer-sweep summary metrics

We summarize the layer sweep using three quantities. These metrics are distinct from the evidence-routing metrics R and D used in the main text.

Representative-layer accuracy change. We report the accuracy change at the representative target layer $\ell^\star$:

$$\Delta\mathrm{Acc}_{\ell^\star}(s) = \Delta_s(\ell^\star). \tag{29}$$

This is the accuracy change under the default SPAR target-layer setting.

Positive-layer rate. We define the positive-layer rate as the fraction of decoder layers where the intervention improves accuracy:

$$P_+(s) = \frac{1}{L} \sum_{\ell=0}^{L-1} \mathbf{1}\left[\Delta_s(\ell) > 0\right]. \tag{30}$$

Higher P_+ indicates that the ROI source remains helpful across a larger portion of the decoder.

Worst-layer drop. We define the worst-layer drop as the largest accuracy degradation observed across the layer sweep:

$$W_{\mathrm{drop}}(s) = \left| \min_{\ell} \Delta_s(\ell) \right|. \tag{31}$$

Lower W_{drop} indicates that the ROI source is less likely to cause severe degradation when applied at a suboptimal target layer.

D.4 Broad effective layer band

The target-layer sweep shows that SPAR does not rely on a single isolated intervention layer. For LLaVA-1.5-7B, the intervention is consistently effective across a broad middle-depth band, approximately $L8$–$L20$. This region overlaps with the PCEW used for ROI estimation, suggesting that decoder layers in and around the PCEW provide a stable regime for evidence-preserving attention reweighting.

We emphasize that this sweep is used as a sensitivity analysis. The main comparison uses a fixed representative target layer rather than selecting the best target layer separately for each benchmark. The broad improvement band reduces sensitivity to the exact layer choice and supports the interpretation that SPAR exploits a depth-level routing regime rather than a single layer-specific artifact.

D.5 Why middle-layer averaging is used

The ROI-source ablation in the main text shows that middle-layer averaging is more reliable than early-layer, final-layer, all-layer, single-layer, shuffled, or random alternatives. This supports the PCEW-based design of SPAR.

The comparison is useful for three reasons. First, it tests whether middle layers provide a stronger prompt-conditioned evidence signal than early or final layers. Second, it tests whether averaging over the PCEW is more stable than using a single middle layer. Third, it tests whether the spatial structure of the middle-layer evidence map matters by comparing against shuffled and random controls.

The middle-layer average is effective because it combines multiple prompt-conditioned evidence signals while reducing noise from any single layer. By contrast, early layers can be dominated by low-level or prompt-invariant attention patterns, and final layers can be affected by later-stage sink behavior. All-layer averaging can dilute the middle-layer evidence signal by mixing it with less informative early or final attention. The shuffled and random controls remove the spatial correspondence between the evidence score and the visual-token positions, testing whether SPAR depends on the actual evidence-bearing locations rather than only the score distribution or token budget.

E ROI Specificity Control

This appendix provides an additional control analysis for the self-referential ROI used by SPAR. The goal is to verify whether the gain comes from targeting the ROI estimated from middle-layer PCEW attention, rather than from reweighting an arbitrary subset of visual tokens. We use a same-size random ROI control that matches SPAR in both token budget and push-pull reweighting strength.

Table 11 reports a representative fine-grained setting. The SPAR row corresponds to the main configuration used in Table 1. Thus, the score of 57.20 is the TextVQA result of the main SPAR setting, not a separate auxiliary run. The random control uses the same ROI size as SPAR, but replaces the self-referential ROI with randomly selected visual tokens.

Table 11 shows that SPAR improves the baseline by $+1.58$ points, while the same-size random ROI control decreases performance by -2.02 points. Because the random control uses the same token budget and the same push-pull strengths as SPAR, the gap cannot be explained by ROI size or intervention strength alone. Instead, the result supports the interpretation that the ROI estimated from PCEW attention contains task-relevant visual evidence.

This control is intentionally simple. Its purpose is not to exhaustively sweep all ROI budgets or mask variants, but to isolate whether the spatial identity of the ROI matters. The degradation under the

Table 11: ROI specificity control. SPAR uses the self-referential ROI estimated from middle-layer PCEW attention. The random control uses the same ROI size and the same push-pull strengths, but replaces the ROI with randomly selected visual tokens. Numbers in parentheses denote absolute changes relative to the baseline.

Benchmark	Method	Score ↑	Δ
TextVQA	Baseline	55.62	–
TextVQA	SPAR (main configuration)	**57.20**	**+1.58**
TextVQA	Random same-size ROI	53.60	-2.02

random same-size ROI suggests that reweighting arbitrary visual tokens can steer attention away from evidence-bearing regions. In contrast, SPAR benefits from using the model's own middle-layer evidence signal to identify the visual tokens to reweight.

F Hyperparameter Sensitivity

This appendix examines SPAR's sensitivity to the push–pull strengths α, β and the ROI budget k. The main experiments use $\alpha = 5$, $\beta = 2$, and an ROI ratio of approximately 8.3%. We first isolate the effects of the push and pull terms, then report a cross-benchmark coefficient sweep and a separate local ROI-budget study. The coefficient and budget sweeps vary these choices separately; they are not a joint search over all hyperparameters.

F.1 Ablation design

SPAR applies the following visual-token logit reweighting at the target decoder layer:

$$z'_{\ell^\star,q,i} = z_{\ell^\star,q,i} + \alpha \cdot \mathbf{1}[v_i \in \mathcal{R}] - \beta \cdot \mathbf{1}[v_i \in \mathcal{V} \setminus \mathcal{R}]. \tag{32}$$

Here, α promotes attention to ROI visual tokens, while β suppresses attention to non-ROI visual tokens. Both α and β are non-negative bias magnitudes.

We compare the default SPAR setting with variants that isolate the push term, isolate the pull term, weaken the intervention, or make it stronger. All variants use the same PCEW-derived ROI, the same ROI budget, and the same target intervention layer. Only α and β are changed.

Table 12: Push-pull strength sensitivity sets on MMVP. All variants use the same PCEW-derived ROI, ROI budget, and target intervention layer. The default SPAR setting is $\alpha = 5, \beta = 2$.

Variant	α	β	Acc. ↑	R ↑	D ↓
No reweighting	0	0	61.48	0.77	0.16
Pull only	0	2	54.07	0.61	0.21
Push only	5	0	57.49	0.79	0.13
Weak push-pull	3	1	61.91	0.83	0.09
SPAR	5	2	**62.96**	**0.97**	**0.01**
Strong push-pull	7	3	59.39	0.75	0.11
Aggressive push-pull	9	4	55.93	0.78	0.12

F.2 Purpose of the ablation

This ablation separates three possible effects of the push-pull intervention.

First, the push-only variant tests whether increasing attention to ROI visual tokens is sufficient. Second, the pull-only variant tests whether suppressing non-ROI visual tokens is sufficient. Third, the full push-pull variants test whether SPAR benefits from jointly promoting ROI visual tokens and reducing attention allocated to non-ROI visual tokens.

The strength sweep also tests whether SPAR is sensitive to the exact choice of α and β. The weak, default, strong, and aggressive variants use the same ROI and target layer, changing only the

magnitude of the attention-logit bias. This allows us to distinguish balanced attention reweighting from overly weak or overly aggressive intervention.

F.3 Cross-benchmark sensitivity to push–pull strengths

We extend the coefficient analysis beyond the MMVP setting in Table 12. Within each model–benchmark setting, the ROI source, ROI budget, target intervention layer, evaluated examples, and decoding protocol are held fixed; only the coefficient pair (α, β) changes. Table 13 reports all nine tested pairs rather than selecting a separate pair for each benchmark. The default pair $(5, 2)$ reproduces the corresponding scores in Table 1.

Table 13: Cross-benchmark coefficient sensitivity. Parentheses show percentage-point changes relative to the baseline in the same column. MMVP-pair denotes strict pairwise accuracy, distinct from the image-level accuracy in Table 12. The default pair is marked in the first column; it need not maximize every benchmark.

	Qwen2-VL	LLaVA-1.5		
(α, β)	GQA	GQA	MMVP-pair	MMBench
Base	66.35	61.80	24.44	72.80
$(3, 1)$	67.15 (+0.80)	62.47 (+0.67)	25.10 (+0.66)	72.93 (+0.13)
$(3, 2)$	66.95 (+0.60)	62.47 (+0.67)	31.11 (+6.67)	72.93 (+0.13)
$(4, 1.5)$	66.55 (+0.20)	62.80 (+1.00)	27.77 (+3.33)	73.14 (+0.34)
$(5, 1)$	66.75 (+0.40)	63.47 (+1.67)	27.77 (+3.33)	73.09 (+0.29)
(5,2), default	66.40 (+0.05)	64.35 (+2.55)	31.11 (+6.67)	73.14 (+0.34)
$(5, 3)$	66.55 (+0.20)	63.13 (+1.33)	27.77 (+3.33)	73.01 (+0.21)
$(6, 2.5)$	66.55 (+0.20)	63.13 (+1.33)	25.67 (+1.23)	73.09 (+0.29)
$(7, 2)$	66.35 (+0.00)	61.80 (+0.00)	25.67 (+1.23)	73.18 (+0.38)
$(7, 3)$	66.75 (+0.40)	61.47 (-0.33)	25.10 (+0.66)	73.09 (+0.29)

Of the 36 reported coefficient–setting combinations, 33 improve on the baseline and 35 are non-degrading. The only decrease is 0.33 percentage points for $(7, 3)$ on LLaVA-GQA. The default pair improves all four settings without per-benchmark coefficient retuning, although other pairs perform better on Qwen2-GQA or MMBench. These results support stability over the tested coefficient neighborhood, rather than the claim that a single pair is universally optimal. They complement the one-sided and strong-intervention controls in Table 12.

F.4 Sensitivity to ROI budget

We conduct a separate local one-factor study in which only the number of selected visual tokens is varied. We evaluate $k \in \{24, 48, 72, 96\}$, corresponding to ROI ratios from 4.17% to 16.67%. Within each benchmark, the PCEW source, target intervention layer, evaluated examples, and decoding protocol are held fixed. This study evaluates sensitivity to the ROI budget within its own controlled setting; its absolute scores are not presented as reproductions of the representative main configuration in Table 1. In particular, "default budget" below refers to $k = 48$, not to equivalence of the complete experimental configuration.

Table 14: Local ROI-budget sensitivity. Parentheses show percentage-point changes relative to the baseline in the same column. Comparisons are made within this auxiliary study.

ROI tokens	ROI ratio	GQA	POPE	MMVP strict-pair
Base	–	61.80	74.65	24.44
24	4.17%	61.92 (+0.12)	75.02 (+0.37)	31.04 (+6.60)
48 (default budget)	8.33%	62.02 (+0.22)	75.05 (+0.40)	31.11 (+6.67)
72	12.50%	61.97 (+0.17)	75.25 (+0.60)	31.13 (+6.69)
96	16.67%	61.94 (+0.14)	74.94 (+0.29)	31.11 (+6.67)

All 12 benchmark–budget combinations improve on their respective baselines. Across the four tested budgets, the score ranges are 0.10, 0.31, and 0.09 percentage points on GQA, POPE, and MMVP strict-pair, respectively. The default budget achieves the highest GQA score in this study and is within 0.20 and 0.02 points of the highest POPE and MMVP strict-pair scores. Increasing the budget from 72 to 96 tokens does not improve any of the three scores. These observations support low local sensitivity to ROI size under the evaluated conditions, not universal optimality of a particular token count.

G Comparison with Related Decoding Methods

We compare SPAR with VCD (Leng et al., 2024) and OPERA (Huang et al., 2024a) using their official implementations within the same LLaVA-1.5 benchmark and evaluation harness. Base, SPAR, and VCD use greedy decoding. OPERA uses its official five-beam setting because beam search is integral to its penalty-and-rollback procedure. This is therefore a comparison under a common evaluation harness with method-specific decoding, not a comparison with matched decoding strategies or inference budgets.

Table 15: Comparison with VCD and OPERA on LLaVA-1.5. All scores are percentages; MMVP-pair denotes strict pairwise accuracy. Decoding settings are listed explicitly because OPERA uses beam search. Bold denotes the highest or joint-highest score in each column.

Method	Decoding	GQA	POPE	MMVP-image	MMVP-pair
Base	Greedy	61.80	74.65	61.48	24.44
SPAR	Greedy	**64.35**	**75.05**	**62.96**	**31.11**
VCD	Greedy + contrast	58.57	74.54	56.67	26.07
OPERA	Official beam-5	61.07	**75.05**	61.53	28.70

SPAR achieves the highest or joint-highest score on all four reported metrics. It exceeds VCD on all four and OPERA on GQA, MMVP-image, and MMVP-pair, while tying OPERA on POPE. These empirical comparisons are restricted to the evaluated backbone, benchmarks, and decoding configurations. They do not establish a compute-matched ranking or superiority over methods not directly evaluated here.

H Qualitative Evidence Routing Example

This appendix provides a qualitative example illustrating how SPAR changes evidence routing during decoding. The example is intended as a visual complement to the quantitative results in the main text. It shows how the baseline model can under-use localized visual evidence, while SPAR uses the PCEW-derived ROI to guide attention toward the relevant region.

Figure 7 illustrates the main intuition behind SPAR. The image contains several signs, but the answer requires reading a small local region: the blue sign indicating “End of bus lane.” The baseline model attends to the image but produces the broader answer “Road,” suggesting that the local evidence is not sufficiently preserved for the final response. SPAR estimates an ROI from PCEW attention and applies push-pull reweighting during decoding. In this example, the resulting attention maps remain more aligned with the sign region, and the model outputs the more specific answer “Bus lane.”

This example should be interpreted as a qualitative illustration rather than standalone evidence. The quantitative effects of SPAR are evaluated in the main benchmark results, random ROI controls, and evidence-routing metrics.

I Limitations, Future Work

I.1 Limitations

SPAR targets a specific decoder-side routing issue: visual evidence that appears in middle decoder layers can be under-used during later decoding. Accordingly, the method is most effective when the

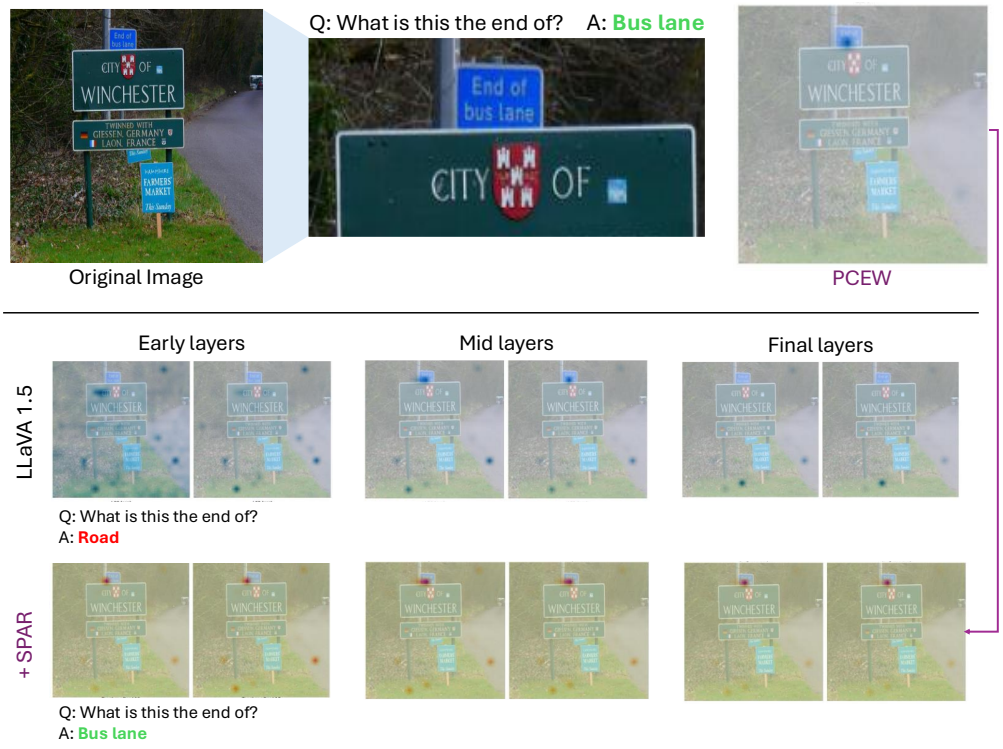


Figure 7: Qualitative evidence-routing example. The question asks, "What is this the end of?" The relevant evidence is the small sign reading "End of bus lane." The baseline LLaVA-1.5 prediction is "Road," while SPAR predicts "Bus lane." The top row shows the original image, a zoomed evidence region, and the PCEW-derived ROI visualization. The lower panels compare attention maps across early, middle, and final layers for the baseline and SPAR. SPAR uses the middle-layer evidence signal to reweight visual attention, helping the decoder preserve attention to the evidence-bearing sign region through later layers.

required evidence is exposed in the Prompt-Conditioned Evidence Window and can be represented by a compact visual-token ROI. When the required evidence is not captured by middle-layer attention, is spatially diffuse, or requires reasoning over many distant regions, the benefit of SPAR may be smaller.

SPAR also adds one ROI-estimation pass before generation. Although the method does not update model parameters and does not introduce trainable modules, this additional pass increases inference cost relative to the unmodified baseline. The relative overhead depends on output length: in the controlled-length measurements of Table 7, SPAR takes $2.15\times$ the baseline time at one token and $1.06\times$ at 128 tokens. This additional cost remains important for short-answer and latency-sensitive applications.

Finally, SPAR is an inference-time intervention. It improves visual evidence routing but does not change the underlying model's learned knowledge, reasoning ability, or answer calibration. Therefore, SPAR should be viewed as an evidence-preserving decoding method rather than a complete solution to all VLM reasoning failures.

I.2 Future work

Future work can extend SPAR in several directions. First, the ROI-estimation pass could be made more efficient by caching middle-layer attention or approximating the PCEW evidence map with a lightweight predictor. Second, the target intervention layer and push-pull strengths could be adapted dynamically based on the input, while still avoiding test-set-specific tuning. Third, SPAR could be

combined with complementary methods for answer calibration, multi-step reasoning, or visual-token compression. Finally, applying the same PCEW-based analysis to video-language models and multi-image reasoning models may reveal whether similar evidence-preservation failures occur in longer visual contexts.