When Sinks Help or Hurt: Unified Framework for Attention Sink in Large Vision-Language Models

Jiho Choi[1], Jaemin Kim[2]
Sanghwan Kim[⋆3], Seunghoon Hong[⋆1], and Jin-Hwi Park[⋆2]

[1] KAIST, South Korea
[2] Chung-Ang University, South Korea
[3] Technical University of Munich, Germany
{jiho.choi,seunghoon.hong}@kaist.ac.kr
{kimjm1213,jinhwipark}@cau.ac.kr
sanghwan.kim@tum.de

Abstract. Attention sinks are defined as tokens that attract disproportionate attention. While these have been studied in single modality transformers, their cross-modal impact in Large Vision-Language Models (LVLM) remains largely unexplored: are they redundant artifacts or essential global priors? This paper first categorizes visual sinks into two distinct categories: ViT-emerged sinks (V-sinks), which propagate from the vision encoder, and LLM-emerged sinks (L-sinks), which arise within deep LLM layers. Based on the new definition, our analysis reveals a fundamental performance trade-off: while sinks effectively encode global scene-level priors, their dominance can suppress the fine-grained visual evidence required for local perception. Furthermore, we identify specific functional layers where modulating these sinks most significantly impacts downstream performance. To leverage these insights, we propose Layer-wise Sink Gating (LSG), a lightweight, plug-and-play module that dynamically scales the attention contributions of V-sink and the rest visual tokens. LSG is trained via standard next-token prediction, requiring no task-specific supervision while keeping the LVLM backbone frozen. In most layers, LSG yields improvements on representative multimodal benchmarks, effectively balancing global reasoning and precise local evidence.[1]

Keywords: Attention Sinks · Large Vision-Language Models

1 Introduction

Transformers are widely known to exhibit the *attention sink* phenomenon, where a small subset of tokens attracts disproportionately large attention weights despite possessing limited semantic content [6,10,38,40]. In Large Language Models (LLMs), these sinks often manifest in punctuation or special symbols (e.g., `[BOS]`

⋆ Corresponding authors.
[1] Code will be released at `https://github.com/JH-GEECS/lsg_public`

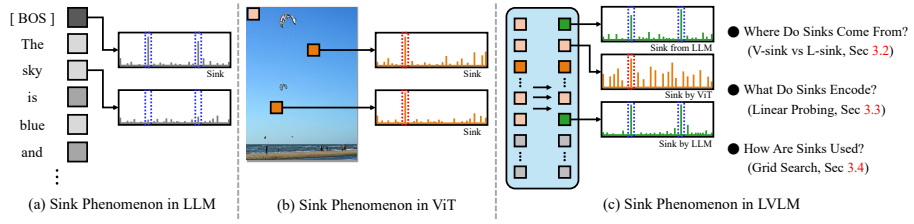


Fig. 1: Attention sink phenomena across model modalities. (a) In LLMs, certain tokens consistently receive disproportionately high attention scores across layers. (b) In ViTs, a similar pattern appears where background patches accumulate high attention. (c) In LVLMs, visual sink tokens among the vision token sequence arise from two distinct sources: those inherited from the vision encoder and those newly formed within the LLM layers, motivating three questions investigated in this work: (1) where sinks originate, (2) what information sinks encode, and (3) how sinks affect downstream task performance.

token) and are often characterized by massive activations in specific hidden dimensions [32,40]. A similar behavior exists in Vision Transformers (ViTs), where background patches with high ℓ_2 norms absorb attention to facilitate global context propagation [6, 13, 18]. While recent works [2, 15, 16, 23, 30, 37] have begun to investigate how these tokens interact within multimodal architectures (e.g., LLaVA [20]), it remains unexplored whether they represent undesirable artifacts to be removed or essential components encoding useful global priors.

To clarify the role of sink tokens in multimodal tasks, we first categorize them into two fundamentally distinct types based on their computational origins: *ViT-emerged sinks* (V-sinks), which originate from the vision encoder and persist as high-norm representations, and *LLM-emerged sinks* (L-sinks), which arise within the deeper layers of the LLM through sharp, dimension-specific activations. While both types consistently attract significant attention during generation, they stem from different computational origins and display unique activation patterns. Our analysis of the information encoded within V-sinks, L-sinks, and ordinary tokens reveals a distinct distribution of semantic attributes across each category such as count, size, color, and shape. This suggests that these token types can be actively utilized depending on the character of downstream task. To validate this, we perform oracle interventions by modulating the Key scaling of specific token groups. These experiments expose a fundamental trade-off: while sinks are vital for tasks reliant on global context, they can become detrimental to local tasks requiring fine-grained perception. Furthermore, we identify specific *functional layers* where modulating these sinks is impactful, suggesting that optimal sink utilization is highly sensitive to both task requirements and layer depth.

To leverage these insights, we propose **Layer-wise Sink Gating (LSG)**, a lightweight, plug-and-play module that dynamically adjusts the attention contributions of V-sinks, L-sinks, and ordinary tokens. Conditioned on the hidden state of the final input token, which aggregates both visual context and query semantics [17, 25, 42], LSG predicts layer-specific key scaling factors. This small gating module is trained via standard next-token prediction loss while

the LVLM backbone remains frozen, requiring no task-specific supervision. Our framework enables the model to balance global knowledge and local perception in a content-aware manner. We evaluate LSG on representative multimodal benchmarks, demonstrating improvements, particularly in vision-centric tasks.

Our contribution can be summarized as follows. (1) We provide a unified analysis of attention sinks in LVLMs, characterizing two distinct categories: *ViT-emerged sinks* (V-sinks) and *LLM-emerged sinks* (L-sinks). (2) We reveal a fundamental trade-off in multimodal tasks: while sink tokens facilitate global scene understanding, ordinary tokens are more effective for fine-grained perception tasks relying on precise local evidence. (3) We introduce **Layer-wise Sink Gating (LSG)**, a lightweight module that dynamically scales sink contributions, yielding performance gains on representative multimodal benchmarks.

2 Related Work

2.1 Attention Sink in Large Language Models

The attention sink phenomenon has been observed in transformer-based language models, where disproportionately large attention scores are allocated to a small subset of tokens (e.g., `[BOS]` token or punctuation tokens) despite their limited semantic significance [3, 38, 40]. Recent work [32] shows that this property arises from massive activations of specific dimensions in the hidden states. Additionally, an investigation of the different characteristics of attention sinks is conducted by [10, 28], which demonstrates that they behave more like key biases rather than contributing to value computation. Recent works [11, 21, 31] suggest that sink tokens serve as indispensable structural biases, motivating the performance improvements without additional training.

2.2 Attention Sink in Vision Encoders

For ViT-based vision encoders [5, 26, 34], a similar phenomenon has also been observed, where high-norm tokens tend to emerge in low-informative background regions. Some works [6, 13] find that they contain global information across the entire image and propose register tokens to reduce artifacts in the feature maps, which improves dense prediction performance (e.g., semantic segmentation and object detection). In addition, Wang *et al.* [18] reveals that the `[CLS]` token tends to rely on these sink tokens, and Lu *et al.* [22] modifies the attention dynamics to take advantage of the different roles of massive and artifact tokens.

2.3 Attention Sink in LVLMs

Recent studies [16, 23] have explored the attention sink phenomenon in Large Vision-Language Models (LVLMs) with conflicting interpretations. Kang *et al.* [16] argue that sink tokens are largely irrelevant to performance and propose redistributing their attention via threshold-based head selection. Conversely, Luo *et*

al. [23] suggest these tokens encode vital global information, advocating the active adjustment of sink contributions based on task-specific (global vs. local) requirements. From a reliability perspective, Wang *et al*. [37] investigate how sinks may exacerbate hallucination errors and their potential utility in hallucination-based attacks. Given these divergent perspectives, it remains an open question whether attention sinks constitute redundant artifacts to be eliminated or functional anchors to be strategically utilized. To bridge this gap, we provide a unified analysis of attention sinks across both vision and language modalities. We then propose a layer-wise sink-gating mechanism for task-aware modulation. This approach is motivated by recent findings [29] indicating that LLM layers exhibit distinct functional specializations, such as counting, grounding, and OCR, requiring different levels of sink-based information. A structured comparison with Kang *et al*. [16] and Luo *et al*. [23] is provided in Appendix B.

3 Analysis of Visual Sink Tokens

3.1 Preliminaries

Recent Large Vision-Language Model (LVLM) [1, 19, 20] typically consist of (i) a **vision encoder** $\mathcal{E}$ [27, 41], (ii) a MLP-based **projector** $\mathcal{P}$ that maps vision tokens to the language embedding space [20, 24], and (iii) a **Large Language Model (LLM)** $\mathcal{M}$ [35, 39]. Given an input image $\mathbf{I}$, the vision encoder outputs n patch-level representations of dimension D_v, $\mathbf{V} = [\mathbf{v}_1, \ldots, \mathbf{v}_n] \in \mathbb{R}^{n \times D_v}$. The projector then maps these representations to match the LLM hidden dimension D, producing **visual tokens** $\mathbf{X}_{\text{vis}} \in \mathbb{R}^{n \times D}$. The LLM takes as input a single concatenated sequence of **system tokens** $\mathbf{X}_{\text{sys}}$, **visual tokens** $\mathbf{X}_{\text{vis}}$, and **query tokens** $\mathbf{X}_{\text{txt}}$, and operates in an autoregressive manner. In this paper, we analyze how visual tokens—especially those fed into the LLM after the projector—are grouped and utilized across layers within this LVLM architecture.

Layer-wise Hidden Representations. Let the hidden states of the entire input sequence at the ℓ-th LLM layer be denoted by $\mathbf{H}^\ell \in \mathbb{R}^{T \times D}$, where $T = |\mathcal{I}_{\text{sys}}| + |\mathcal{I}_{\text{vis}}| + |\mathcal{I}_{\text{txt}}|$ is the total sequence length. The ℓ-th layer hidden state can be written with a residual formulation as:

$$\mathbf{H}^\ell \;=\; \mathbf{H}^{\ell-1} + F^\ell\big(\mathbf{H}^{\ell-1}\big)\,, \tag{1}$$

where F^ℓ is a transformer block that includes self-attention and an MLP (FFN). All of our layer-wise analyses begin from $\mathbf{H}^\ell$, and we introduce additional notation only as needed based on this definition.

Sink Identification and Token Grouping. Recent work [32] observed that Transformer-based models can exhibit abnormally large activation values in specific hidden dimensions. Concretely, such behavior appears consistently in a fixed set of dimensions within each model across different modality (e.g., dimension 650 in CLIP-L, or dimensions 1415 and 2533 in LLaMA2-7B). Following [16], we call these dimensions the **sink dimensions** $\mathcal{D}_{\text{sink}}$. Building on this finding,

we identify **sink tokens** in both the ViT and the LLM by applying thresholding to the activation magnitude on sink dimensions (threshold selection and its justification are detailed in Appendix A.1).

Let $\mathbf{h}_j^\ell \in \mathbb{R}^D$ be the hidden state of token j at layer ℓ. Given a threshold τ, we define the set of sink-token indices at layer ℓ as:

$$\hat{\mathcal{I}}^\ell = \left\{ j \in \mathcal{I} \;\middle|\; \max_{d \in \mathcal{D}_{\text{sink}}} \left|(\mathbf{h}_j^\ell)[d]\right| \geq \tau \right\}. \tag{2}$$

Referring to this criterion [16, 23], we partition $\mathcal{I}_{\text{vis}}$ into three groups:

- **ViT-emerged sinks** [23] (V-sink, $\hat{\mathcal{I}}_{\text{vit}}$): Token indices that are already classified as sinks in the vision encoder (right before the projector), using a criterion analogous to Eq. (2). After being passed into the LLM, we treat them as a fixed set of indices.
- **LLM-emerged sinks** [16] (L-sink, $\hat{\mathcal{I}}_{\text{llm}}^\ell$): Vision tokens that are ordinary at the ViT output stage, but become sink tokens inside the LLM after passing through attention/MLP computations, i.e., they satisfy Eq. (2) at some LLM layer ℓ. These are identified in a layer-wise manner.
- **Ordinary visual tokens** ($\mathcal{I}_{\text{ord}}^\ell$): The remaining visual tokens that do not belong to the two sink groups, defined as $\mathcal{I}_{\text{ord}}^\ell = \mathcal{I}_{\text{vis}} \setminus (\hat{\mathcal{I}}_{\text{vit}} \cup \hat{\mathcal{I}}_{\text{llm}}^\ell)$.

Token group statistics, including average sink counts per sample, are reported in Appendix A.2.

3.2 Two Sources of Visual Sinks: ViT vs. LLM

Unless otherwise stated, all analyses in this section are conducted with LLaVA-1.5-7B [20] (CLIP ViT-L/14 + Llama 2-7B). The layer-wise norm and attention patterns (Figure 2) and the group-wise activation profiles (Figure 3) are replicated on LLaVA-OneVision-7B (SigLIP + Qwen2) in Appendix E, confirming that the sink taxonomy generalizes across architectures.

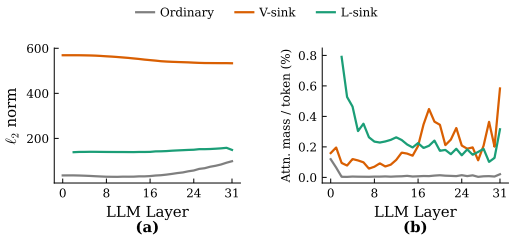


Fig. 2: Layer-wise salience pattern of visual token groups. (a) V-sinks and L-sinks maintain higher ℓ_2 norms. (b) They also receive a significantly larger percentage of attention mass compared to ordinary visual tokens.

Prior interpretability studies [14, 17, 25] have shown that the final input token's attention over visual tokens critically mediates image-to-text information flow in LVLMs. We analyze this attention for the three token groups ($\hat{\mathcal{I}}_{\text{vit}}$, $\hat{\mathcal{I}}_{\text{llm}}^\ell$, $\mathcal{I}_{\text{ord}}^\ell$) using 300 GQA [12] samples, measuring the attention weight from the final input token to each group per layer (with head-wise and sample-wise averaging). As shown in Figure 2, both sink groups receive significantly larger per-token attention weights than ordinary tokens, consistent with their abnormally large norms [6, 32].

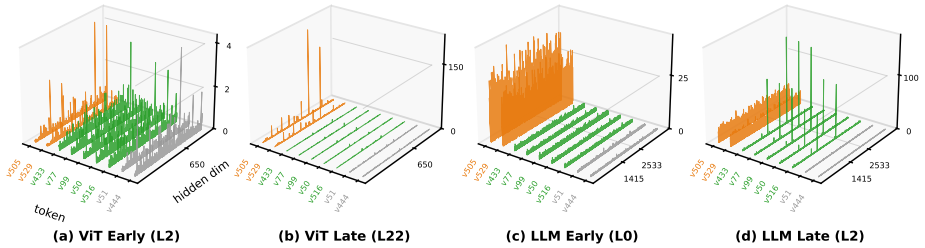


Fig. 3: Activation pattern of visual tokens across LVLM stack at different depth level. Each line represents one token, colored by category: **orange** = ViT-emerged sinks, **green** = LLM-emerged sinks, gray = ordinary tokens. Sink dimensions: 650 for CLIP-ViT-L; 1415, 2533 for LLaMA-2-7B. The input image is shown in Figure S3.

Although both sink groups attract strong attention from the final input token, their origins differ. Prior work has shown that attention sinks arise from massive activations in a few fixed hidden dimensions, leading to large key norms and disproportionate attention weight [32]. The same phenomenon has been observed in both vision transformers [6] and LVLMs [16].

V-sinks carry large activation values from the vision encoder and enter the LLM with uniformly high norms (Figure 3 (c)) [23]. In the ViT, these tokens develop sharp spikes in a few sink dimensions under bidirectional attention (Figure 3 (a,b)). However, the two-layer MLP projector, which lacks a residual connection, applies a linear transformation followed by a nonlinearity, spreading the concentrated activation across output dimensions. As a result, V-sinks arrive in the LLM embedding space with uniformly elevated norms rather than dimension-specific spikes. L-sinks, by contrast, appear ordinary at the ViT output but develop sharp dimension-specific activations at deeper LLM layers (Figures 3 (b) and 3 (d)). These activations occur in the same $\mathcal{D}_{\text{sink}}$ as text sink tokens (e.g., BOS) and are written by early-layer FFNs and maintained through residual connections, mirroring the text-sink formation mechanism [16]. Because V-sinks and L-sinks follow fundamentally different formation mechanisms, one shaped by ViT bidirectional attention and the projector, the other by the LLM's own text-sink mechanism, they are expected to carry different information despite both attracting strong attention.

3.3 What Do Visual Sinks Encode? A Linear Probing Approach

As discussed in Section 2, high-norm outlier tokens in vision transformers aggregate global image information [6], and Luo *et al.* [23] further confirmed this by isolating visual tokens through modified attention masks and mapping them to output vocabulary. However, both studies characterize sink tokens only at the ViT output or through indirect decoding, without examining how their information content changes across LLM layers. Meanwhile, Kang *et al.* [16] identified L-sinks that emerge within the LLM decoder and showed that masking them causes little performance degradation, yet what these L-sinks actually encode remains uncharacterized.

Fu *et al.* [8] show that vision representations are broadly preserved across LLM layers, while Feucht *et al.* [7] demonstrate that token identities can be

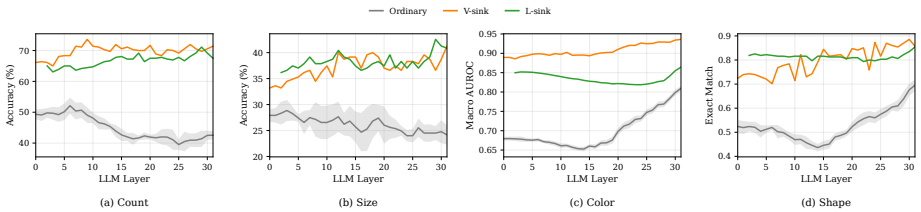


Fig. 4: Layer-wise linear probing on CLEVR scene attributes. Each panel probes one property (count, size, color, shape) from pooled hidden states across LLM layers of LLaVA-1.5-7B: V-sinks and L-sinks follow the sink criterion of Eq. (2) and the V-/L-sink partition defined below it, while the ordinary curve averages five independent random samplings of the ordinary group with $\pm 1\sigma$ shading.

rapidly erased in early layers. Since visual tokens are projected into the language embedding space via an MLP connector [24], V-sinks may undergo similar restructuring, yet no prior work has examined them separately.

To address this gap, we design a linear probing study on the CLEVR dataset that tracks V-sinks ($\hat{\mathcal{I}}_{\text{vit}}$), L-sinks ($\hat{\mathcal{I}}^{\ell}_{\text{llm}}$), and ordinary visual tokens ($\mathcal{I}^{\ell}_{\text{ord}}$) across LLM layers, probing four scene-level properties: object *count*, *color*, *shape*, and *size*. These cover different abstraction levels, from holistic scene understanding (counting) to per-object categorical attributes (color, shape) and spatial reasoning (relative size), letting us ask: (1) what granularity of scene information do sink tokens retain, and (2) does this information degrade, persist, or strengthen across successive LLM layers?

Experimental Setup. We define four tasks spanning different prediction types. Count and Size are multi-class classification problems: Count predicts the total number of objects in the scene, while Size filters the largest single object by bounding-box ratio and assigns it to one of 5 bins. Color and Shape are multi-label classification problems: Color predicts $y \in \{0,1\}^8$ indicating the presence of 8 colors, and Shape predicts $y \in \{0,1\}^3$ indicating the presence of 3 shapes. As a baseline, we construct an ordinary sampled group by excluding all tokens identified as ViT sinks or LLM sinks at any layer, then randomly sampling 5 tokens per image ($\sim$1% of 576 patches in CLIP ViT-L/14 336px [27]) from this pool and averaging their hidden states. The same token indices are used across all LLM layers to ensure that layer-wise trends reflect the evolution of identical tokens. To control for sampling variance, we repeat this construction with five independent random samplings and report the mean with ± 1 standard deviation (shaded) for the ordinary group.

Results. Figure 4 summarizes the layer-wise probing results. For question (1), both V-sink and L-sink groups outperform ordinary visual tokens across all four tasks, with the advantage most evident in early-to-middle layers. The gap is largest for count prediction (Figure 4a), a holistic property that no single patch can capture alone, confirming that sink tokens concentrate scene-level summary information. The advantage also holds for size bin classification (Figure 4b), which requires spatial reasoning about relative object scale. For color and shape (Figures 4c, 4d), sink tokens lead in the early and middle layers, while ordinary tokens gradually close the gap at deeper layers as the LLM's own processing enriches their representations.

For question (2), probing accuracy for sink tokens either remains stable or increases at deeper layers across all four tasks, showing almost no degradation despite the layer-wise transformations. Unlike the token-identity erasure reported for language tokens in early layers [7], the scene-level information in sink tokens is preserved throughout all LLM layers.

Taken together, these results establish that both V-sinks and L-sinks carry rich, multi-granularity scene information that persists throughout the LLM. This confirms that L-sinks, despite following the text-sink formation mechanism, do not lose the visual information present in their representations. However, the presence of information does not imply that the model actively exploits it during inference [8]. We therefore turn to attention-level interventions to examine whether and how the information in each sink type is utilized.

3.4 Layer-wise Intervention on Sink vs. Ordinary Attention

Our probing results show that sink tokens stably encode global attributes across LLM layers. Combined with evidence that visual processing becomes functionally specialized at deeper layers [29], a natural question arises: at which layers, and how, does the global information in sinks actually contribute to visual reasoning?

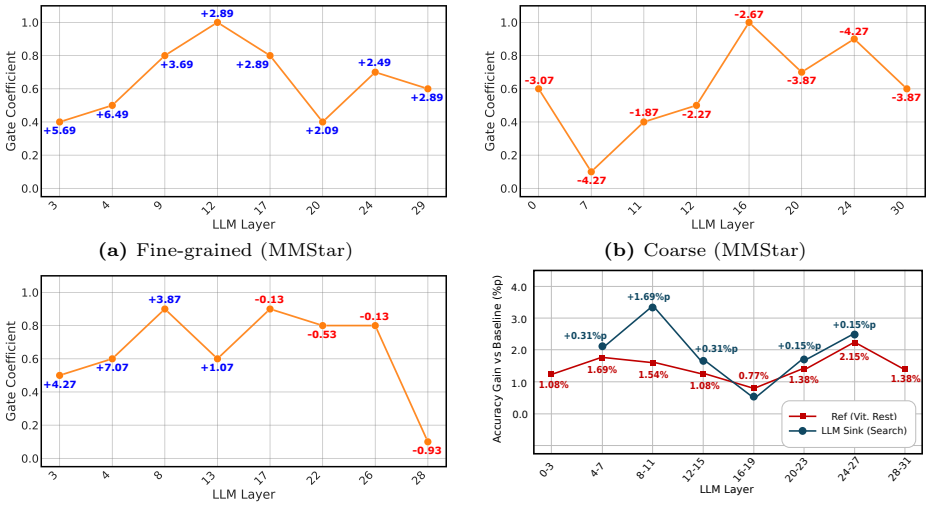
Prior work probes information flow via hard attention knockouts [9, 14] or continuous rescaling [36]. The former masks attention connections at selected layers to identify which token-to-token interactions are essential for a correct prediction, whereas the latter multiplies attention weights by learned or fixed scalars and observes the resulting performance change. Since our question concerns the balance between sink and ordinary attention rather than which specific interactions are essential, we follow the continuous rescaling approach.

Our formulation is closest to Wang *et al.* [36], but differs in the axis of intervention: Wang *et al.* [36] assigns a scalar to each attention head independently to identify which heads are influential for specific visual semantics, whereas we assign a scalar to each token group and apply it uniformly across all heads within a layer, in order to probe how the relative attention allocation between sink and ordinary tokens affects downstream reasoning at each layer.

Formulation (Key Gating). To intervene, we introduce a Key Gating that smoothly scales the attention contribution of a specific token group. Consider a standard attention operation at layer ℓ, where the previous hidden state $\mathbf{H}^{(\ell-1)} \in \mathbb{R}^{T\times D}$ is linearly projected into query $\mathbf{Q}^{(\ell)}$, key $\mathbf{K}^{(\ell)}$, and value $\mathbf{V}^{(\ell)}$. For simplicity, we present the single-head form, but the intervention is applied identically to all heads in the layer. To control token group-wise attention contribution, we intervene by multiplying the key vectors by scalar coefficients. Let the token-wise coefficient vector be $\mathbf{s} \in \mathbb{R}^T$. The intervened attention is defined as:

$$\mathrm{Attn}\big(\mathbf{Q}^{(\ell)}, \mathrm{diag}(\mathbf{s}) \cdot \mathbf{K}^{(\ell)}, \mathbf{V}^{(\ell)}\big) = \mathrm{SoftMax}\left(\frac{\mathbf{Q}^{(\ell)}\big(\mathrm{diag}(\mathbf{s}) \cdot \mathbf{K}^{(\ell)}\big)^\top}{\sqrt{D}}\right)\mathbf{V}^{(\ell)}. \quad (3)$$

Here, $\mathrm{diag}(\mathbf{s})$ is a diagonal matrix whose entries correspond to the coefficient assigned to each token according to its group. We choose to intervene on the Key



(a) Fine-grained (MMStar) **(b)** Coarse (MMStar)

(c) Logical (MMStar) **(d)** Relation (CVBench)

Fig. 5: Layer-wise optimal gate coefficients from key-gating intervention (a–c) Best sink gate per 4-layer block with accuracy change vs. baseline (positive, negative). (d) Stage-2 sweep splitting Rest into L-sinks and ordinary tokens.

projection because scaling the keys directly modulates the pre-SoftMax logits, and therefore controls the attention distribution itself: how much probability mass each token group receives. Hereafter, we refer to the per-group scalars in $\mathbf{s}$ as gate coefficients, and the resulting relative proportion of attention mass allocated to sink versus ordinary tokens as the attention contribution ratio.

Experimental Setup. We sweep all 32 self-attention layers (L0–L31) individually: for each layer, we apply the intervention to that layer alone while keeping all others at the default setting (i.e., no intervention). Within each intervened layer, the same gate coefficients are applied uniformly across all heads. We sweep the coefficients from (0.0, 1.0) to (1.0, 0.0) in steps of 0.1 and record task accuracy for each setting. For visualization, we group the 32 layers into blocks of four and report the best-performing layer within each block.

Token Grouping. In Stage 1, we partition visual tokens into V-sinks ($\hat{\mathcal{I}}_{\text{vit}}$) vs. the rest (L-sinks + ordinary). Our probing shows V-sinks stably encode global information across LLM layers, and Luo *et al.* [23] confirmed that V-sinks alone can suffice for global reasoning tasks. In Stage 2, we further split L-sinks from ordinary tokens for a finer local sweep.

Target Tasks. We evaluate on two benchmarks that span fine-grained perception and high-level reasoning. CVBench [33] provides a vision-centric assessment of fine-grained capabilities such as depth, spatial relations, distance estimation, and counting, while MMStar [4] covers a broader set of task categories ranging from fine-grained perception to logical reasoning.

Observations. Figure 5 summarizes the layer-wise optimal gate coefficients. We highlight three key findings:

– **Observation 1: Task-dependency.** Rebalancing sink vs. ordinary attention does not uniformly help or hurt; its effect reverses across tasks. For

fine-grained perception (Figure 5a), adjusting the ratio improves accuracy at every layer block (up to +6.49%), whereas for coarse perception (Figure 5b), every adjustment degrades performance (up to −4.27%), indicating that the default balance already suits this task and any shift disrupts it.
- **Observation 2: Layer-dependency.** Even within a single task, the effect of sink strengthening can reverse across depths: in logical reasoning (Figure 5c), early layers benefit substantially (+7.07% at L4–L8) while late layers are hurt by the same direction (L22–L26).
- **Observation 3: Limited benefit of 3-group decomposition.** In Stage 2, splitting the residual group into L-sinks and ordinary tokens and re-sweeping around the Stage-1 optimum yields additional gains only for specific task×layer combinations (Figure 5d), consistent with recent finding [8] that accessible information is not necessarily utilized (a detailed analysis is provided in Appendix C.3).

Takeaway. Sink tokens carry rich scene-level information that persists across all LLM layers, yet the grid search shows that exploiting it is a double-edged sword: the effect of rebalancing reverses across tasks and even across depths within a single task (Observations 1–2). A fixed coefficient cannot accommodate this variability, motivating a learned mechanism that predicts when and where to adjust sink attention.

4 Layer-wise Sink Gating

We introduce a lightweight gating module inserted between LLM layers, trained solely with the standard next-token prediction loss and no direct supervision from the grid-search oracle coefficients. The module predicts input-dependent, per-layer ratios that a fixed grid search cannot provide. Figure 6 illustrates the architecture.

4.1 Gating Signal

The gating module requires a per-layer input that encodes both visual and textual information. As discussed in Section 3.2, the final input token progressively aggregates visual information across layers [17, 25], and Zhao *et al.* [42] further showed that this hidden state predicts higher-order model judgments via linear probing. We therefore use $\mathbf{h}^{\ell}_{\text{last}} \in \mathbb{R}^D$ as the gating signal at each layer.

4.2 Gating Formulation

Following Observation 3 in Section 3.4, we adopt a 2-group formulation: V-sinks ($\hat{\mathcal{I}}_{\text{vit}}$) vs. the remaining visual tokens ($\mathcal{I}^{\ell}_{\text{rest}}$), where the rest group includes both L-sinks and ordinary tokens. As shown in Figure 6 (right), the gating module g_ℓ, a lightweight 2-layer MLP, maps the gating signal at layer ℓ to allocation logits for layer ℓ+1:

$$\mathbf{z}^{(\ell+1)} = g_\ell\big(\mathbf{h}^{\ell}_{\text{last}}\big) \in \mathbb{R}^2 \tag{4}$$

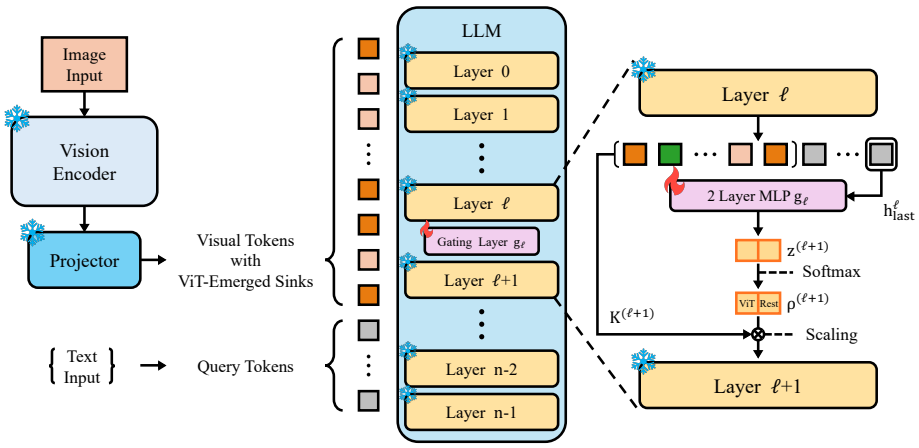


Fig. 6: Layer-wise Sink Gating (LSG). Left: the pretrained LVLM remains frozen; a gating module (lightweight MLP) g_ℓ is inserted between consecutive LLM layers. Right: g_ℓ takes $\mathbf{h}^{\ell}_{\text{last}}$ and predicts key-scaling ratios $\rho^{(\ell+1)}$ for V-sink vs. remaining visual tokens.

The attention contribution ratio between the two groups is obtained via softmax:

$$\rho^{(\ell+1)}_{\text{vit}},\ \rho^{(\ell+1)}_{\text{rest}} = \text{Softmax}\left(\mathbf{z}^{(\ell+1)}\right), \quad \rho^{(\ell+1)}_{\text{vit}} + \rho^{(\ell+1)}_{\text{rest}} = 1 \tag{5}$$

The key vectors of each visual token j at layer $\ell{+}1$ are scaled accordingly:

$$\tilde{\mathbf{K}}^{(\ell+1)}_j = \rho^{(\ell+1)}_{g(j)} \cdot \mathbf{K}^{(\ell+1)}_j \tag{6}$$

where $g(j) \in \{\text{vit}, \text{rest}\}$ denotes the group membership of token j. This extends Eq. 3 from Section 3.4 by replacing the manually swept coefficient with a learned, input-conditioned gate. Ablations on alternative gating signals and token grouping strategies are reported in Appendix D.2.

4.3 Training

All parameters of the pretrained LVLM are frozen; only the gating MLP g is trained with the standard next-token prediction loss:

$$\mathcal{L} = -\sum_t \log p_\theta(x_t \mid x_{<t}) \tag{7}$$

where θ denotes the parameters of the gating MLP. No task-specific labels or auxiliary losses are used. Full training configuration is provided in Appendix A.

5 Experiment

Experimental Setup. We apply the gating module (Section 4) to LLaVA-1.5-7B and train it on 10K stratified samples from Cambrian-7M for 2 epochs. Each gating module contains ∼262K parameters (4096→64→2, softmax output)

Table 1: Per-layer oracle (best of 11 gate values per task) vs. learned gate (single layer, NTP-trained). Δ (%p) relative to baseline. $n^{\geq 0}/n^{<0}$: sub-tasks with non-negative / negative Δ out of 10 categories.

		Per-layer Oracle			Learned Gate		
Block	Layer	MMStar	CVBench	$n^{\geq 0}/n^{<0}$	MMStar	CVBench	$n^{\geq 0}/n^{<0}$
Baseline		33.27	57.29		–	–	
L0–3	L3	+3.19	+2.76	9/1	+0.34	+1.67	8/2
L4–7	L7	+2.26	+2.66	9/1	+1.00	+0.49	8/2
L8–11	L10	+2.13	+3.98	7/3	+0.72	+2.14	8/2
L12–15	L13	+1.06	+0.99	7/3	−0.24	+0.33	6/4
L16–19	L19	+0.39	+2.60	8/2	+0.76	+0.50	**10/0**
L20–23	L23	−0.14	+1.06	7/3	+0.34	+0.19	9/1
L24–27	L24	−0.54	+0.32	5/5	+0.27	−0.02	7/3
L28–31	L30	−0.07	+0.88	8/2	+0.47	+0.10	8/2

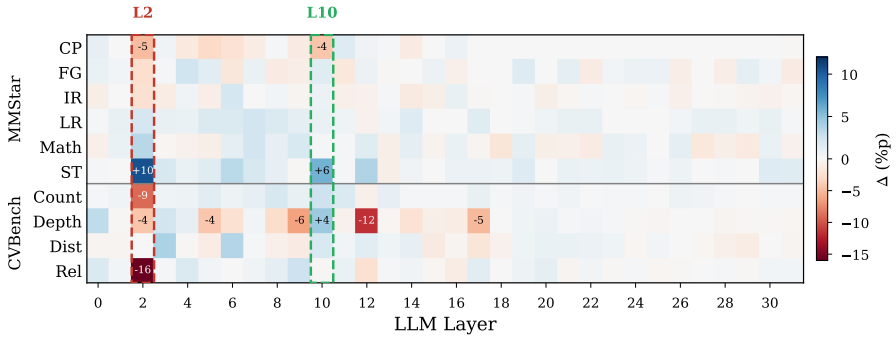


Fig. 7: Learned gate Δ (%p) across all 32 layers and 10 sub-tasks. Dashed lines mark L2 (red) and L10 (green).

and is initialized to ρ_{vit}=0.5, allocating equal attention to V-sink and remaining tokens (Section 4.2). We train a separate gate at each of the 32 LLM layers independently. All results report accuracy changes (%p) relative to the unmodified baseline on MMStar (6 sub-tasks) and CVBench (4 sub-tasks).

5.1 Single-Layer Results.

Table 1 reports the best single-layer result within each 4-layer block, alongside the per-layer oracle that selects the best of 11 gate values independently per task at the same layer. Seven of eight blocks yield positive gains under the learned gate; L10 achieves the largest improvement at +0.72 on MMStar and +2.14 on CVBench, while L19 maintains non-negative gains across all 10 sub-tasks. Notably, these gains emerge from NTP training on just 10K general-purpose samples, without access to task identifiers. The oracle represents an upper bound, as it selects the best coefficient with access to ground-truth accuracy per task; the learned gate, which predicts from hidden states alone, is analyzed against this bound in Appendix C.4. For comparison, a parameter-matched LoRA shows that these gains arise from the sink-specific structure rather than from the added pa-

Table 2: Greedy multi-layer gate stacking. Starting from the single best layer (L10), gates are added greedily in order of marginal gain, using only the SL checkpoints without retraining. Added: layer appended at this step. Active layers: full set of stacked gates. All Δ in %p relative to baseline. $n^{\geq 0}/n^{<0}$: tasks with non-negative / negative Δ out of 10.

Step	Added	Active layers	ΔMMStar	ΔCVBench	$n^{\geq 0}/n^{<0}$
Baseline			0.00	0.00	
(single, L10)			+0.72	+2.14	8/2
1	L10	{L10}	+0.72	+2.14	8/2
2	L3	{L3, L10}	+0.59	+2.98	7/3
3	L7	{L3, L7, L10}	+1.17	+2.41	7/3
4	L19	{L3, L7, L10, L19}	+0.74	+2.47	7/3
5	L6	{L3, L6, L7, L10, L19}	**+1.55**	**+3.08**	**8/2**

rameters, as analyzed in detail in Appendix D.1. We also evaluate on additional benchmarks including OCRBench and MathVista (Appendix D.4).

Figure 7 expands the per-task accuracy changes of the learned gate to all 32 layers and 10 sub-tasks. Effective layers concentrate in L3–19, while early layers such as L2 produce large but opposing shifts across sub-tasks that hurt overall performance. We analyze the source of this layer-level variation in Section 5.4.

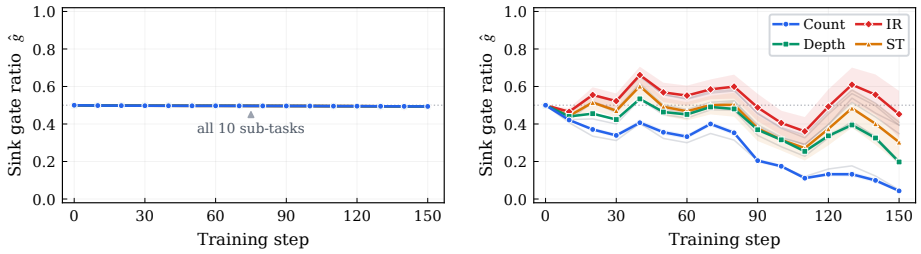
5.2 Multi-Layer Stacking

The single-layer results in Table 1 show that multiple layers independently yield positive gains. We test whether these gains compound by stacking independently trained single-layer checkpoints without retraining: starting from L10 (the single best layer), we greedily add gates in descending order of their single-layer gain and evaluate the accumulated set jointly.

Table 2 reports the results. The final 5-layer configuration {L3, L6, L7, L10, L19} achieves MMStar +1.55 and CVBench +3.08, exceeding L10's single-layer best of +0.72/+2.14 by a clear margin. At the same time, the stacked gains fall well below the sum of individual single-layer gains, indicating that the contributions of different layers are not independent. Adding L19 at Step 4, for instance, slightly reduces the combined score relative to Step 3, while the subsequent addition of L6 produces the largest single-step increase. This sub-additivity reflects the distribution shift each gating module faces at inference, where hidden states have already been modified by other active gating modules not present during training. Nonetheless, the five-layer configuration achieves a combined improvement that no single layer reaches alone, confirming that independently trained gates can be combined effectively. The stack can involve two further choices, in how layers are selected and in how their gates are trained. For selection, choosing the five layers from held-out training signals alone, with no downstream-task information, yields a stack whose performance is on par with the greedy one (Appendix D.3). For training, the stacked gates can also be trained jointly rather than kept independent (Appendix D.3).

Table 3: Comparison with prior sink-handling methods on LLaVA-1.5-7B. *Input-cond.*: modulation can be conditioned on the input; *Layer-wise*: modulation can vary across layers. Δ (%p) relative to the unmodified baseline (MMStar 33.27, CVBench 57.29). † Single-MLP adaptation of DIYSink's CoT routing.

Method	Input-cond.	Layer-wise	ΔMMStar	ΔCVBench
VAR [16]	×	×	−0.01	+0.75
Sink-to-the-front [23]	×	×	−0.13	−0.39
DIYSink (CoT)† [23]	✓	×	+0.25	+0.41
LSG (L10)	✓	✓	**+0.72**	**+2.14**
LSG (5-layer)	✓	✓	**+1.55**	**+3.08**



(a) LLM Layer 2 **(b)** LLM Layer 10

Fig. 8: Learned gate trajectory during training. Solid lines: predicted V-sink ratio ρ_{vit} averaged over evaluation samples per sub-task (shaded: ±1 std).

5.3 Comparison with prior sink-handling methods

Table 3 compares LSG against sink-related prior works that handle L-sink [16] and V-sink [23]. LSG at L10 alone exceeds all of them on both MMStar and CVBench, and the 5-layer stack widens the margin to +1.55/+3.08. These gaps reflect each prior method covering only part of Section 3.4's (task, layer) landscape, where sink modulation needs to both strengthen and suppress depending on the task (Obs. 1) and the LLM layer (Obs. 2): VAR is suppress-only, sink-to-the-front is static, and DIYSink (CoT) is input-conditioned but input-level only. In contrast, LSG combines input-conditioned and layer-wise modulation through a lightweight learned gate. Reproduction protocols are detailed in Appendix B.1.

5.4 Gate Behavior Across LLM Layers

To examine why gating effectiveness varies across layers, we record the gate's V-sink ratio $\rho_{\text{vit}}^{(\ell)}$ per evaluation sample at each training checkpoint and aggregate by sub-task. Figure 8 contrasts L2 and L10.

At L2 (left), ρ_{vit} remains near the 0.5 initialization with negligible cross-task variance throughout training—the gate does not learn to differentiate inputs. At L10 (right), the gate separates into two regimes: CVBench sub-tasks converge to low sink ratios—Count and Relation near 0.05, Depth and Distance near 0.20—while MMStar sub-tasks settle between 0.30 and 0.45. Within CVBench, these two pairs form well-separated clusters with within-task standard deviation below

0.01. This contrast aligns with the performance gap: L10, where the gate differentiates inputs, achieves 8 non-negative sub-tasks out of 10 (Table 1), whereas L2, where the gate remains near initialization, achieves only 3 non-negative sub-tasks out of 10; as Figure 7 shows, ST shifts by +10 %p while Rel drops by −16 %p at the same layer.

Cross-modal information transfer concentrates in layers 4–20 of LLaVA-1.5 [14]. L2 falls below this range—at such an early depth, the final input token carries little visual information, leaving the gate no informative signal to differentiate inputs. Conversely, beyond L20, cross-modal transfer has largely concluded, so adjusting the V-sink-to-rest attention ratio no longer induces meaningful representational changes. This accounts for the concentration of effective layers in L3–19 visible in Figure 7: the gate is effective in the layers where cross-modal transfer is still underway.

5.5 Cross-Architecture Generalization

We examine whether our findings hold beyond LLaVA-1.5. On LLaVA-OneVision-7B [19] (SigLIP [41], Qwen2-7B [39]) architecture, the V-sink/L-sink taxonomy and the oracle key-gating sweep reproduce the same task- and layer-dependent structure (Appendix E). On two further LLaVA variants with different LLM and vision-encoder backbones, the oracle sweep again reproduces this structure, and a learned single-layer LSG gives same-sign gains on both benchmarks (Appendix F), providing preliminary evidence that LSG transfers.

6 Conclusion

In this work, we analyze attention sinks in LVLMs and identify two categories with distinct computational origins: ViT-emerged sinks (V-sinks) and LLM-emerged sinks (L-sinks). These sinks carry global scene priors that benefit coarse reasoning but degrade local perception. Based on this observation, we introduced Layer-wise Sink Gating, a per-layer MLP that scales the Key contributions of sink versus ordinary visual tokens. LSG is trained with next-token prediction loss alone, requires no task labels, and keeps the backbone frozen. The learned gates concentrate their effects in the mid-layers where cross-modal information transfer is active and yield gains across the evaluated benchmarks.

Acknowledgements

This work was partially funded by the ERC (853489 – DEXIM) and the Alfried Krupp von Bohlen und Halbach Foundation, which we thank for their generous support. It was also supported by the Institute of Information & Communications Technology Planning & Evaluation (IITP) grant (No. RS-2026-25518317, Development of AI memory mechanism that reflects human cognitive principles) and by the National Research Foundation of Korea (NRF) grant (No. RS-2026-25495084), both funded by the Korea government (MSIT). We also thank Jaehoon Yoo for helpful discussions.

References

1. Bai, S., Chen, K., Liu, X., Wang, J., Ge, W., Song, S., Dang, K., Wang, P., Wang, S., Tang, J., Zhong, H., Zhu, Y., Yang, M., Li, Z., Wan, J., Wang, P., Ding, W., Fu, Z., Xu, Y., Ye, J., Zhang, X., Xie, T., Cheng, Z., Zhang, H., Yang, Z., Xu, H., Lin, J.: Qwen2.5-vl technical report. arXiv preprint arXiv:2502.13923 (2025)
2. Bai, S., Liu, Y., Han, Y., Zhang, H., Tang, Y., Zhou, J., Lu, J.: Self-calibrated clip for training-free open-vocabulary segmentation. IEEE Transactions on Image Processing (2025)
3. Cancedda, N.: Spectral filters, dark signals, and attention sinks, 2024. URL https://arxiv. org/abs/2402.09221 (2024)
4. Chen, L., Li, J., Dong, X., Zhang, P., Zang, Y., Chen, Z., Duan, H., Wang, J., Qiao, Y., Lin, D., et al.: Are we on the right way for evaluating large vision-language models? Advances in Neural Information Processing Systems **37**, 27056–27087 (2024)
5. Cherti, M., Beaumont, R., Wightman, R., Wortsman, M., Ilharco, G., Gordon, C., Schuhmann, C., Schmidt, L., Jitsev, J.: Reproducible scaling laws for contrastive language-image learning. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 2818–2829 (2023)
6. Darcet, T., Oquab, M., Mairal, J., Bojanowski, P.: Vision transformers need registers. In: The Twelfth International Conference on Learning Representations (2024)
7. Feucht, S., Atkinson, D., Wallace, B.C., Bau, D.: Token erasure as a footprint of implicit vocabulary items in llms. In: Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing. pp. 9727–9739 (2024)
8. Fu, S., Guillory, D., Darrell, T., et al.: Hidden in plain sight: Vlms overlook their visual representations. In: Second Conference on Language Modeling (2025)
9. Geva, M., Bastings, J., Filippova, K., Globerson, A.: Dissecting recall of factual associations in auto-regressive language models. In: Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing. pp. 12216–12235 (2023)
10. Gu, X., Pang, T., Du, C., Liu, Q., Zhang, F., Du, C., Wang, Y., Lin, M.: When attention sink emerges in language models: An empirical view. In: The Thirteenth International Conference on Learning Representations (2025)
11. Han, F., Yu, X., Tang, J., Rao, D., Du, W., Ungar, L.: Zerotuning: Unlocking the initial token's power to enhance large language models without training. arXiv preprint arXiv:2505.11739 (2025)
12. Hudson, D.A., Manning, C.D.: 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)
13. Jiang, N., Dravid, A., Efros, A.A., Gandelsman, Y.: Vision transformers don't need trained registers. In: The Thirty-ninth Annual Conference on Neural Information Processing Systems (2025)
14. Kaduri, O., Bagon, S., Dekel, T.: 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)
15. Kan, Z., Li, X., Liu, Y., Yang, X., Jiang, X., Liu, Y., Jiang, D., Sun, X., Liao, Q., Yang, W.: Rar: Reversing visual attention re-sinking for unlocking potential in multimodal large language models. In: The Fourteenth International Conference on Learning Representations (2026)

16. Kang, S., Kim, J., Kim, J., Hwang, S.J.: See what you are told: Visual attention sink in large multimodal models. In: The Thirteenth International Conference on Learning Representations (2025)
17. Kim, J., Kang, S., Park, J., Kim, J., Hwang, S.J.: Interpreting attention heads for image-to-text information flow in large vision–language models. In: Mechanistic Interpretability Workshop at NeurIPS 2025 (2025)
18. Lappe, A., Giese, M.A.: Register and [cls] tokens induce a decoupling of local and global features in large vits. In: The Thirty-ninth Annual Conference on Neural Information Processing Systems (2025)
19. Li, B., Zhang, Y., Guo, D., Zhang, R., Li, F., Zhang, H., Zhang, K., Zhang, P., Li, Y., Liu, Z., Li, C.: LLaVA-onevision: Easy visual task transfer. Transactions on Machine Learning Research (2025), https://openreview.net/forum?id=zKv8qULV6n
20. Liu, H., Li, C., Wu, Q., Lee, Y.J.: Visual instruction tuning. Advances in neural information processing systems **36**, 34892–34916 (2023)
21. Liu, X., Chen, G., Wang, W.: Sinktrack: Attention sink based context anchoring for large language models. In: The Fourteenth International Conference on Learning Representations (2026)
22. Lu, A., Liao, W., Wang, L., Yang, H., Shi, J.: Artifacts and attention sinks: Structured approximations for efficient vision transformers. arXiv preprint arXiv:2507.16018 (2025)
23. Luo, J., Fan, W.C., Wang, L., He, X., Rahman, T., Abolmaesumi, P., Sigal, L.: To sink or not to sink: Visual information pathways in large vision-language models. In: The Fourteenth International Conference on Learning Representations (2026), https://openreview.net/forum?id=sQGlhjKUC0
24. Merullo, J., Castricato, L., Eickhoff, C., Pavlick, E.: Linearly mapping from image to text space. In: The Eleventh International Conference on Learning Representations (2023)
25. Neo, C., Ong, L., Torr, P., Geva, M., Krueger, D., Barez, F.: Towards interpreting visual information processing in vision-language models. In: The Thirteenth International Conference on Learning Representations (2025)
26. Oquab, M., Darcet, T., Moutakanni, T., Vo, H., Szafraniec, M., Khalidov, V., Fernandez, P., Haziza, D., Massa, F., El-Nouby, A., et al.: Dinov2: Learning robust visual features without supervision. Transactions on Machine Learning Research Journal (2024)
27. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. In: International conference on machine learning. pp. 8748–8763. PmLR (2021)
28. Ruscio, V., Nanni, U., Silvestri, F.: What are you sinking? a geometric approach on attention sink. arXiv preprint arXiv:2508.02546 (2025)
29. Shi, C., Yu, Y., Yang, S.: Vision function layer in multimodal llms. arXiv preprint arXiv:2509.24791 (2025)
30. Srikrishnan, T.A., Shah, D., Reinhardt, S.K.: Blindsight: Harnessing sparsity for efficient vlms. arXiv preprint arXiv:2507.09071 (2025)
31. Su, Z., Yuan, K.: Kvsink: Understanding and enhancing the preservation of attention sinks in kv cache quantization for llms. arXiv preprint arXiv:2508.04257 (2025)
32. Sun, M., Chen, X., Kolter, J.Z., Liu, Z.: Massive activations in large language models. arXiv preprint arXiv:2402.17762 (2024)

33. Tong, P., Brown, E., Wu, P., Woo, S., Iyer, A.J.V., Akula, S.C., Yang, S., Yang, J., Middepogu, M., Wang, Z., et al.: Cambrian-1: A fully open, vision-centric exploration of multimodal llms. Advances in Neural Information Processing Systems **37**, 87310–87356 (2024)
34. Touvron, H., Cord, M., Jégou, H.: Deit iii: Revenge of the vit. In: European conference on computer vision. pp. 516–533. Springer (2022)
35. Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., et al.: Llama 2: Open foundation and fine-tuned chat models. arXiv preprint arXiv:2307.09288 (2023)
36. Wang, Q., Hu, J., Jiang, M.: V-seam: Visual semantic editing and attention modulating for causal interpretability of vision-language models. In: Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing. pp. 17407–17431 (2025)
37. Wang, Y., Zhang, M., Sun, J., Wang, C., Yang, M., Xue, H., Tao, J., Duan, R., Liu, J.: Mirage in the eyes: Hallucination attack on multi-modal large language models with only attention sink. In: 34th USENIX Security Symposium (USENIX Security 25). pp. 3707–3726 (2025)
38. Xiao, G., Tian, Y., Chen, B., Han, S., Lewis, M.: Efficient streaming language models with attention sinks. arXiv preprint arXiv:2309.17453 (2023)
39. Yang, A., Yang, B., Hui, B., Zheng, B., Yu, B., Zhou, C., Li, C., Li, C., Liu, D., Huang, F., Dong, G., Wei, H., Lin, H., Tang, J., Wang, J., Yang, J., Tu, J., Zhang, J., Ma, J., Xu, J., Zhou, J., Bai, J., He, J., Lin, J., Dang, K., Lu, K., Chen, K., Yang, K., Li, M., Xue, M., Ni, N., Zhang, P., Wang, P., Peng, R., Men, R., Gao, R., Lin, R., Wang, S., Bai, S., Tan, S., Zhu, T., Li, T., Liu, T., Ge, W., Deng, X., Zhou, X., Ren, X., Zhang, X., Wei, X., Ren, X., Fan, Y., Yao, Y., Zhang, Y., Wan, Y., Chu, Y., Liu, Y., Cui, Z., Zhang, Z., Fan, Z.: Qwen2 technical report. arXiv preprint arXiv:2407.10671 (2024)
40. Yu, Z., Wang, Z., Fu, Y., Shi, H., Shaikh, K., Lin, Y.C.: Unveiling and harnessing hidden attention sinks: Enhancing large language models without training through attention calibration. arXiv preprint arXiv:2406.15765 (2024)
41. Zhai, X., Mustafa, B., Kolesnikov, A., Beyer, L.: Sigmoid loss for language image pre-training. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 11975–11986 (2023)
42. Zhao, Q., Xu, M., Gupta, K., Asthana, A., Zheng, L., Gould, S.: The first to know: How token distributions reveal hidden knowledge in large vision-language models? In: European Conference on Computer Vision. pp. 127–142. Springer (2024)