Counting Circuits: Mechanistic Interpretability of Visual Reasoning in Large Vision-Language Models

Liwei Che[*,1]   **Zhiyu Xue**[*,2]   **Yihao Quan**[*,1]   **Benlin Liu**[3,4]   **Zeru Shi**[1]   **Michelle Hurst**[1]
Jacob Feldman[1]   **Ruixiang Tang**[1]   **Ranjay Krishna**[3,4]   **Vladimir Pavlovic**[1]

[1]Rutgers University   [2]UC Santa Barbara   [3]University of Washington   [4]Allen Institute for AI

{lc1279, yq207, zs618, rt836, vladimir}@cs.rutgers.edu
michelle.hurst@rutgers.edu   jacob@ruccs.rutgers.edu
zhiyuxue@ucsb.edu   liubl@cs.washington.edu   ranjay@cs.washington.edu

[*]Equal contribution

Abstract

Counting serves as a simple but powerful test of a Large Vision-Language Model's (LVLM) reasoning; it forces the model to identify each individual object and then add them all up. In this study, we investigate how LVLMs implement counting using controlled synthetic and real-world benchmarks, combined with mechanistic analyses. Our results show that LVLMs display a human-like counting behavior, with precise performance on small numerosities and noisy estimation for larger quantities. We introduce two novel interpretability methods, Visual Activation Patching and HeadLens, and use them to uncover a structured "counting circuit" that is largely shared across a variety of visual reasoning tasks. Building on these insights, we propose a lightweight intervention strategy that exploits simple and abundantly available synthetic images to fine-tune arbitrary pretrained LVLMs exclusively on counting. Despite the narrow scope of this fine-tuning, the intervention not only enhances counting accuracy on in-distribution synthetic data, but also yields an average improvement of +8.36% on out-of-distribution counting benchmarks and an average gain of +1.54% on complex, general visual reasoning tasks for Qwen2.5-VL. These findings highlight the central, influential role of counting in visual reasoning and suggest a potential pathway for improving overall visual reasoning capabilities through targeted enhancement of counting mechanisms.

1   Introduction

Counting is one of the most fundamental yet revealing capacities of visual intelligence. Cognitive science reveals that human numerosity perception is shaped by severe information-processing constraints, resulting in near-perfect accuracy for small sets (subitizing) and noisy estimation for larger quantities [1]. Recent work further demonstrates that these discontinuities arise not from separate cognitive modules, but from resource-rational trade-offs [2] between precision, exposure time, and environmental statistics, suggesting that counting reflects a core organizing principle of visual reasoning rather than a task-specific heuristic [3].

Recent LVLMs [4, 5] have demonstrated remarkable generalization across diverse visual reasoning tasks [6–9]. However, our benchmarks (table 1) reveal a surprising phenomenon: these models struggle to count even fewer than ten simple black dots, a trivial task for human vision. This discrepancy raises fundamental questions regarding the nature of counting ability in LVLMs and its relation to general visual reasoning. Unlike standard recognition, counting cannot be bypassed through semantic memorization or dataset priors. It strictly requires models to individuate discrete entities, maintain intermediate representations, and aggregate information under architectural constraints.
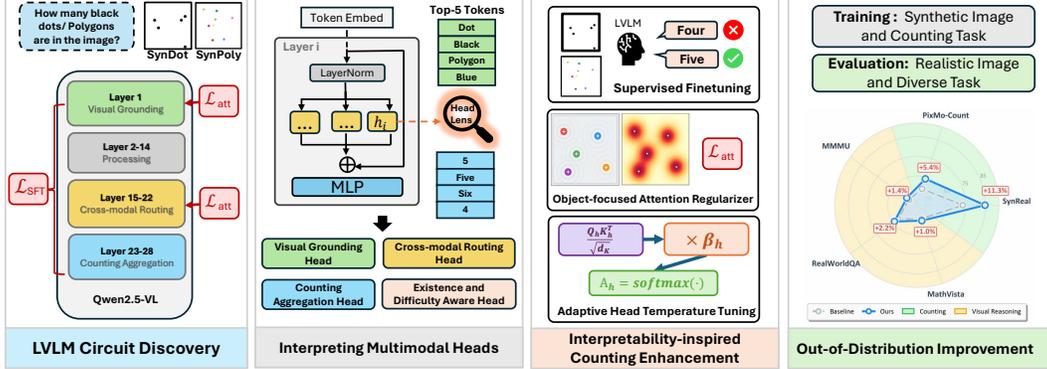
Figure 1: Overview of Main Contributions

Therefore, in this work, we utilize the counting task as a minimalist probe to investigate the internal visual reasoning process of LVLMs and explore how these mechanism discoveries can support more complex, high-level visual reasoning tasks.

Existing works often treat counting as an isolated task, attempting to improve the counting performance of LVLMs with methods like image preprocessing [10], fine-tuning [11], and attention intervention [12]. However, as specialized computer vision methods [13, 14] undoubtedly yield better performance, we argue that understanding and evaluating the visual reasoning process of LVLMs via counting should be the primary focus. Closer work such as CountScope [15] and [De|Re] [16] heavily rely on probing methods to understand the counting pattern of LVLMs, which is unreliable due to the simplicity of the representation of their synthetic data. More importantly, they fail to understand the circuit-level mechanism of counting. Therefore, in this work, we investigate how LVLMs execute counting tasks from the dual perspectives of human cognitive alignment and mechanistic interpretability. First, we leverage the controllability of synthetic images to extend activation patching [17] to multimodal inputs. By tracing the flow of counting-related information across model layers, we reveal the cross-modal information routing occurring in the middle layers and the emergence of counting answers in the later layers. To further isolate and verify the roles of different model components, we propose HeadLens, a novel tool that interprets the output of attention heads into semantic tokens. Through this, we identify four critical categories of attention heads: (1) *Visual Grounding Heads* (Early Layers): Extract foundational visual features (color, shape) directly from image tokens. (2) *Cross-Modal Routing Heads* (Middle Layers): Translate visual information into abstract numerical concepts. (3) *Counting Aggregation Heads* (Late Layers): Attend heavily to text prompts; their top-10 HeadLens decoded tokens highly correlate with the final prediction. (4) *Awareness Heads* (Late Layers): Two specialized heads encoding object existence and difficulty estimations.

Building on the interpretability findings, we design targeted intervention methods tailored to specific layers and heads via parameter tuning, attention regularization, and head temperature tuning. Utilizing 8,000 highly simplified synthetic images (e.g., black dots or colored polygons on a white canvas) that can be generated in minutes on a standard computer, our approach not only improves performance on synthetic data but also achieves up to an 8.36% average accuracy increase on out-of-distribution (OOD) real-world object counting. Furthermore, it yields a 1.54% average accuracy improvement across broader OOD visual tasks (e.g., general VQA, visual math). This demonstrates that counting serves as a vital foundational pillar of general visual intelligence and validates the existence of generalized visual reasoning mechanisms operating as internal circuits within LVLMs. Our main contributions are summarized as follows:

Cognitive Alignment: Our investigation reveals that LVLMs exhibit a striking discontinuity in visual counting that mirrors human behavior, characterized by precise subitizing for small sets and noisy estimation for larger quantities. We ground the model's behavioral failures in the topological entanglement of hidden state manifolds.

Methodological Innovation: We propose Visual Activation Patching (VAP) and HeadLens, two novel mechanistic interpretability techniques designed to isolate head-specific functions and perform circuit discovery for LVLMs.

Circuit Discovery: We identify four distinct functional categories of attention heads essential for visual counting: *visual grounding*, *cross-modal routing*, *counting aggregation*, and *difficulty/existence detection*. We further demonstrate that a great proportion of these specialized heads also serve as foundational components for general visual reasoning.

Performance Enhancement: We develop an interpretability-inspired intervention strategy. Using only the simplest synthetic images, we significantly enhance the model's OOD counting robustness and its performance on complex, general visual reasoning benchmarks.

2 Related Work

2.1 LVLMs for counting task.

The strong general visual capabilities of LVLMs have sparked considerable attention to their counting performance. Early exploration [11] utilizes additional counting data to fine-tune the CLIP for better counting performance. LVLM-count [10] utilizes SAM [18] and a pixel search algorithm to preprocess the image into pieces for better counting performance. Guo et al.[19] explore the relationship between object types and quantities, revealing the compositional counting failure modes of LVLMs. More recent works explore the counting task via internal states analysis. CountScope [15] verified that LVLMs will accumulate counting information along the layers and tokens in a causal order due to the auto-regressive pattern, yet failed to provide circuit-level functionality understanding on how the internal mechanism actually works. [Del|Re] [16] identified that the incorrect mapping of the last layer caused the counting error, while ignoring that the simplistic representational structure of the synthetic data renders probing results unreliable. An over 99% counting accuracy by linear probing can be trivially achieved even before the LVLM processes the input. More importantly, they failed to reveal the functions of architecture components such as attention heads and layers which are the fundamental building blocks of the transformer's computation. Sengupta et al. [12] propose attention-based interventions that redistribute attention weights to improve counting, but their analysis is limited to behavioral outcomes without dissecting which heads carry counting-specific signals or how they interact across layers. In contrast, our work goes beyond layer-level probing and behavioral intervention: we perform head-level circuit discovery via Visual Activation Patching and HeadLens, identify four functionally distinct head categories, and demonstrate that the discovered counting circuit generalizes to broader visual reasoning tasks.

2.2 Mechanistic Interpretability on LLM/LVLM.

Prior work on interpreting LLM/VLM internals on the layer level has explored probing the direct relationship between intermediate hidden states and final model outputs. Logit Lens [20] proposes to project hidden representations at various layers directly into the vocabulary space using the model's output head. It can reveal how information related to the next token emerges throughout layers. Tuned Lens [21] extends logit-based probing by learning a linear translator from hidden states to logits. It improves the fidelity of the probe but still operates at the level of entire layer activations. Existing head-level interpretability methods like AttentionLens [22] require training independent translators for each attention head. It is computationally expensive and risks masking the specialized role of a head by forcing it to mimic the final output. HeadLens overcomes this by leveraging the block-matrix additivity of the attention projection. It ensures the semantic space of each attention head, and reveals the authentic contribution of each head to the residual stream.

Beyond representation probing, recent advances in mechanistic interpretability employ causal interventions to isolate the functional roles of specific components. For instance, Meng et al. [23] introduce causal tracing to identify that mid-layer feed-forward networks (MLPs) are decisive in recalling factual associations, which enables direct model editing techniques such as Rank-One Model Editing (ROME). Similarly, Wang et al. [24] apply causal interventions to reverse-engineer a comprehensive attention circuit for the Indirect Object Identification (IOI) task in GPT-2, successfully categorizing specific attention heads into specialized functional classes (e.g., name mover heads, induction heads). These works highlight the necessity of isolating individual component contributions rather than treating layer activations as a monolith, further motivating our fine-grained, head-level analysis.

In the realm of Large Vision-Language Models (LVLMs), mechanistic interpretability is actively expanding to understand cross-modal interactions and visual token processing. Neo et al. [25] investigate how visual information evolves within the language model backbone of LLaVA, demonstrating that visual token representations gradually align with interpretable textual concepts in the vocabulary space across deeper layers. Furthermore, they reveal that the model extracts this localized object information at the final token position for prediction, mirroring factual recall in text-only models. Additionally, fine-grained analysis of attention mechanisms has proven crucial for diagnosing model failures [26]. Collectively, these studies underscore that understanding token-level routing and head-level attention mechanisms is essential for interpreting and improving LVLM behaviors.

3 Problem Formulation

To validate the intrinsic counting capabilities of LVLMs, we employ synthetic datasets that isolate numerical reasoning from confounding variables such as background clutter and complex textures. As in fig. 1, we use PIL [27] generated **SynDot** and **SynPoly**, comprising randomly distributed black dots or multi-colored polygons on a white canvas; and **SynReal**, which uses FLUX.1-dev [28] to generate photorealistic objects across six categories. Object counts range in $[1, 10]$; details are in section B. We evaluate counting performance on four axes (formally defined in section C.1): **Accuracy (ACC)** for exact-match precision; **MAE** and **RMSE** for deviation magnitude and stability; and **Off-by-one Accuracy (OBO)** for near-miss reliability.

Table 1: Performance comparison on synthetic benchmarks.

Model	Dataset	Acc ↑	MAE ↓	RMSE ↓	OBO ↑
Qwen2.5-VL-7B	SynDot	76.12	0.25	0.52	96.21
	SynPoly	53.74	0.94	1.75	82.27
	SynReal	73.48	0.79	2.13	91.01
Qwen3-VL-8B	SynDot	73.22	0.32	0.66	95.21
	SynPoly	70.34	0.52	1.49	94.02
	SynReal	88.79	0.15	0.47	98.88
LLaVA-1.5-7B	SynDot	24.07	4.46	6.58	39.22
	SynPoly	33.30	1.65	2.34	56.71
	SynReal	54.27	3.59	5.05	66.34

We evaluate the counting performance of three popular LVLMs, with the prompt "*What is the number of the {object} in the image? Answer with number only.*". As shown in table 1, we surprisingly find that these models fail significantly across all three benchmarks compared to humans, who can achieve flawless accuracy. A more counterintuitive observation is that LVLMs perform better on the more realistic SynReal, while struggling with the visually simpler SynDot and SynPoly. This discrepancy clearly exposes an inherent bias rooted in their training data distribution. This naturally raises a fundamental question: **Does the counting capability of LVLMs signify an emergent reasoning mechanism, or is it merely a statistical hallucination predicated on correlational visual cues and language priors?**

4 Do LVLMs know how to count? Or do they memorize?

In this section, we attempt to address the problem proposed in the initial evaluation. We observe that models often hesitate to provide precise numerical counts when faced with occlusion, blurriness, or high object density. To circumvent this, we reformulate the counting task into a binary 'Yes/No' verification framework. Specifically, for a given image, we employ the prompt: "*There are <K> <Object Name> in the image, Yes or No?*". For images with countable instances, we iteratively vary K from 0 to 100. For complex scenes (e.g., swarms of pigeons or dense crowds), we substitute K with quantitative descriptors such as 'tens of' or 'hundreds of'. We assign scores of 1 for 'Yes', 0 for 'No'.

As shown in fig. 2, counting uncertainty strongly correlates with the stability of the model's "Yes Band" (a contiguous interval of positive responses). Simple scenarios (e.g., few, unoccluded umbrellas) yield
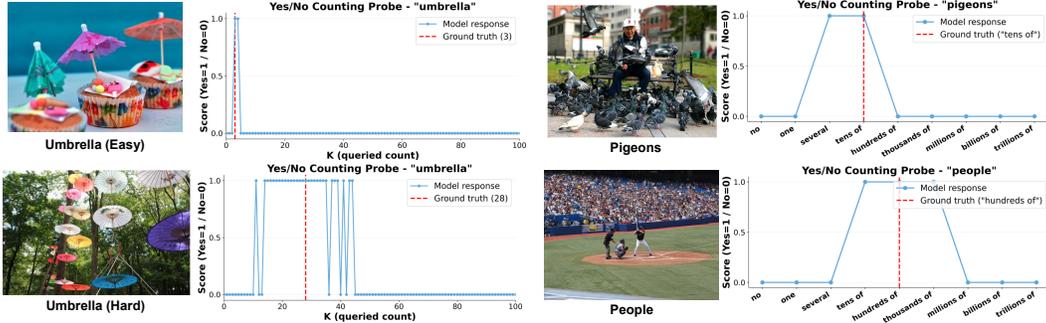
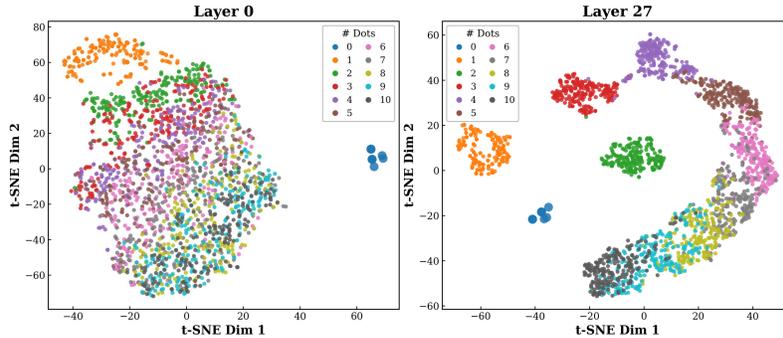Figure 2: Counting Uncertainty Curve by Yes/No Answer of Qwen2.5 VL 7B.

Figure 3: T-SNE visualization of model hidden states based on black dots data.

a concentrated, stable Yes Band, whereas challenging cases blur decision boundaries, causing the band to expand and oscillate. Nonetheless, LVLMs retain robust coarse-grained estimation, reliably distinguishing orders of magnitude (e.g., "tens" vs. "millions" of pigeons) despite lacking precision. These properties validate that: *LVLMs possess a visual subitizing and estimation capability similar to human behavior [1]. Instead of relying on discrete tokens, the model encodes quantity within a continuous latent space, enabling seamless cross-modal alignment between visual stimuli and linguistic quantifiers.*

Cognitive Alignment from the Representation Aspect. To further verify the counting ability alignment between LVLMs and human cognition, we project the hidden states of the SynDot dataset into a 2D space using t-SNE (fig. 3). As features propagate from the first to the final layer, *class 0* (background) and *class 1 - 4* (black dots number) form distinct, isolated clusters separated from other numbers. This topological isolation provides a mechanistic explanation for the model's subitizing capability, the accurate recognition of small quantities. However, as the object count increases, the corresponding clusters become progressively entangled and heavily overlapped. This spatial compression in the latent space directly accounts for the performance degradation on larger numbers, reflecting a natural cognitive shift from precise subitizing to approximate magnitude estimation.

5 Understand Layerwise Behavior of Counting

In this section, we investigate the underlying mechanisms that execute visual counting tasks inside LVLMs. We choose Qwen2.5-VL-7B as the default model for our interpretability studies due to its balance between architectural simplicity and good counting performance.

5.1 Information Flow and Cross-Modal Routing

To trace how counting information propagates from visual input to numerical output, we adapt activation patching [17, 29], a technique traditionally applied to text-only LLMs [30, 31], for LVLM analysis. By fixing the random seed in our synthetic datasets, we construct tightly controlled image pairs. Each pair consists of a clean image (e.g., three dots) and a corrupted counterpart with an

altered object count (e.g., five dots) that strictly preserves the original spatial layout. This minimal perturbation extends activation patching to the visual modality, allowing us to precisely isolate how the model's internal states respond to quantitative changes.

Visual Activation Patching. In an L-layer LVLM, the hidden state s_i^l encodes token x_i after being processed through layers 1 to l. We first record hidden states from a forward pass on the corrupted image, then re-run inference on the clean image while replacing s_i^l of a target token set at layer l with its corrupted counterpart. Whether the model's prediction flips from the clean to the corrupted label gives the *overwrite rate*, measuring the causal efficacy of the patched tokens. We partition input tokens into six groups: System Prompt, Image Tokens, User Instruction, Generated Tokens, Last Image Token ("|img_end|"), and Last Prompt Token (the "Assistant:" role tag), and use the prompt "*What is the number of the black dots in the image? Answer with the number only.*" so that the first generated token is the counting answer.
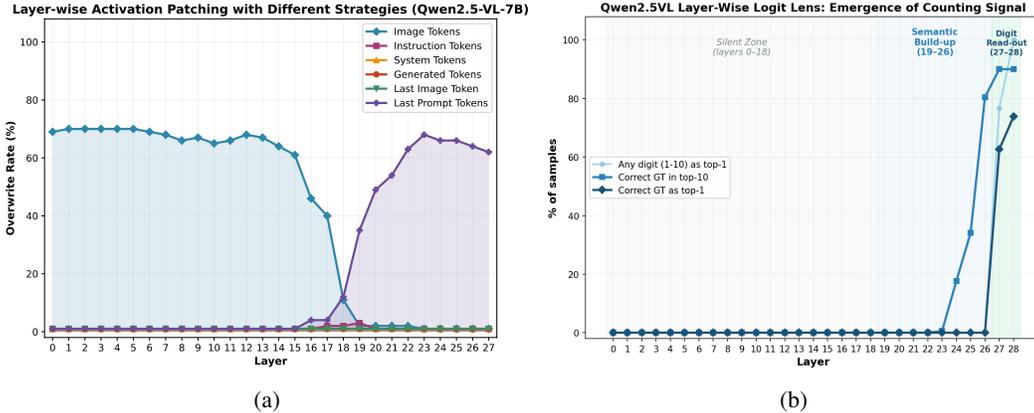

Figure 4: Left: Layer-wise Overwrite Rate of Different Input Tokens Patching Strategy; Right: Layer-wise Logit Lens Tracking Curve for Counting Number.

We run layer-wise activation patching on 100 black dot image pairs. As shown in fig. 4a, in early-to-middle layers ($L < 15$), counting information is predominantly anchored in *All Image Tokens* with high, stable causal influence. From Layer 15 onward, image token importance drops sharply while the *Last Prompt Token* ("Assistant:" tag) surges, peaking at Layer 23, revealing a clear "handover" mechanism. Layers 15–22 thus form the critical bottleneck for cross-modal routing, where spatially distributed visual features are compressed into the linguistic stream. The negligible impact of user instruction and system prompt tokens confirms that the numerical signal flows directly from visual representations to the response-triggering tokens. Similar patterns hold for other LVLMs (section D).

To corroborate that numerical representations emerge after the cross-modal routing phase, we employ a logit lens on the hidden state s_{-1}^l immediately preceding the generation of the final counting token. By projecting these intermediate representations through the model's terminal unembedding layer, we map s_{-1}^l directly into the vocabulary space to decode its latent semantic trajectory. As depicted in the fig. 4b, correct number tokens begin to surface within the top-10 logit predictions between layers 19 and 26, gaining pronounced significance from layer 23 onwards. By layers 27 and 28, the ground-truth counting token converges to the rank-1 position.

This layer-wise progression captures the internal patterns of how the model aggregates visual numerosity, translates it into a discrete linguistic space, and ultimately solidifies the final numerical output. To further investigate the mechanistic underpinnings of these patterns, the subsequent section aims to pinpoint the specific attention heads responsible for executing the counting task and characterize their precise functionalities.

6 Mechanistic Analysis on Attention Head Function

Isolating complete end-to-end circuits in LVLMs is intractable due to their massive scale and redundant backup heads [24]. Therefore, we focus on identifying and revealing the critical attention heads that dominate the counting task.

6.1 Important Attention Heads for Counting

We conduct head-level visual activation patching to identify the important attention heads for counting tasks. Specifically, we leverage 100 SynDot image pairs as the layerwise VAP but replace the target head activation. To prevent confounding effects from the residual stream, we define head activation strictly as the output of the projection layer. The importance score of each head is defined as the answer token logit difference change before and after the VAP. A positive importance score greater than 0.05 indicates that the target head plays a significant role in the counting task. There are $\frac{43}{784} = 5.5\%$ heads noted as important heads. We visualize the heads with positive importance score as a heatmap in fig. 5 (left). Next, we introduce a new interpretability tool named HeadLens and use it to reveal the functions of representative important heads based on their attention distribution and the semantic meaning of their output.

6.2 HeadLens: Decoding Individual Attention Heads

To isolate the semantic contribution of individual attention heads, we introduce a novel interpretability tool termed **HeadLens**. Utilizing the linear additivity of the multi-head self-attention (MHSA) projection, HeadLens enables granular, token-level semantic decoding without structural modifications.

Let $h_i(x) \in \mathbb{R}^{d_\text{head}}$ denote the output of the i-th attention head, where H is the total number of heads and $d_\text{model} = H \times d_\text{head}$. We first isolate the i-th head's contribution by constructing a zero-padded representation $\tilde{h}_i(x) \in \mathbb{R}^{d_\text{model}}$:

$$\tilde{h}_i(x) = [\ \underbrace{0,\ldots,0}_{(i-1)\times d_\text{head}}\ ,\ h_i(x),\ \underbrace{0,\ldots,0}_{(H-i)\times d_\text{head}}\ ] \tag{1}$$

Due to the properties of block matrices, the standard concatenated output of the MHSA mechanism can be equivalently formulated as the sum of these sparse representations. The final output of MHSA $\hat{x}$ is then obtained by applying the output projector matrix W_O:

$$\hat{x} = \left(\sum_{i=1}^{H} \tilde{h}_i(x)\right) W_O = \sum_{i=1}^{H} \left(\tilde{h}_i(x) W_O\right) \tag{2}$$

Rather than interpreting the aggregated hidden state $\hat{x}$, HeadLens operates directly on the isolated projection $\tilde{h}_i(x)W_O$. To translate this independent vector into human-interpretable concepts, we employ the learned affine translator [21] $T : \mathbb{R}^{d_\text{model}} \to \mathbb{R}^{d_\text{model}}$ (e.g., $T(z) = Az + b$) to map it into the final residual stream space. Utilizing the model's unembedding matrix $U \in \mathbb{R}^{|\mathcal{V}| \times d_\text{model}}$ and bias c, we formulate the head-specific residual transformation $r_i(x)$, its corresponding logits $\ell_i(x)$, and the token probability distribution $p_i(\cdot \mid x)$ as follows:

$$r_i(x) = T\big(\tilde{h}_i(x)W_O\big) \qquad \ell_i(x) = U\, r_i(x) + c \qquad p_i(\cdot \mid x) = \text{softmax}\big(\ell_i(x)\big)$$

By treating each $\tilde{h}_i(x)W_O$ as an independent semantic component, HeadLens effectively decodes the head activation into semantic tokens. We are particularly interested in tokens related to the visual features (e.g., color and shape) and counting (e.g., digits and numbers). We define the ratio of visual feature tokens and counting tokens in the top-10 HeadLens results as the Visual Grounding Score (VGS) and Counting Token Emergence Rate (CTER), respectively.

6.3 Revealing Attention Head Functionalities

In this section, we reveal the functionalities of four representative attention heads, including two counting heads (Cross-modal Routing Heads and Counting Aggregation Heads) and two special functional heads (Visual Grounding Heads and Awareness Heads) for the counting task. We provide the experiment details of head discovery in section D.1.

Counting Heads. We discuss two functionally distinct attention head groups for the counting task, Cross-modal Routing Heads and Counting Aggregation Heads. **Cross-modal Routing Heads** are defined as heads with high attention ratio on image, high head importance, and high top-10 HeadLens
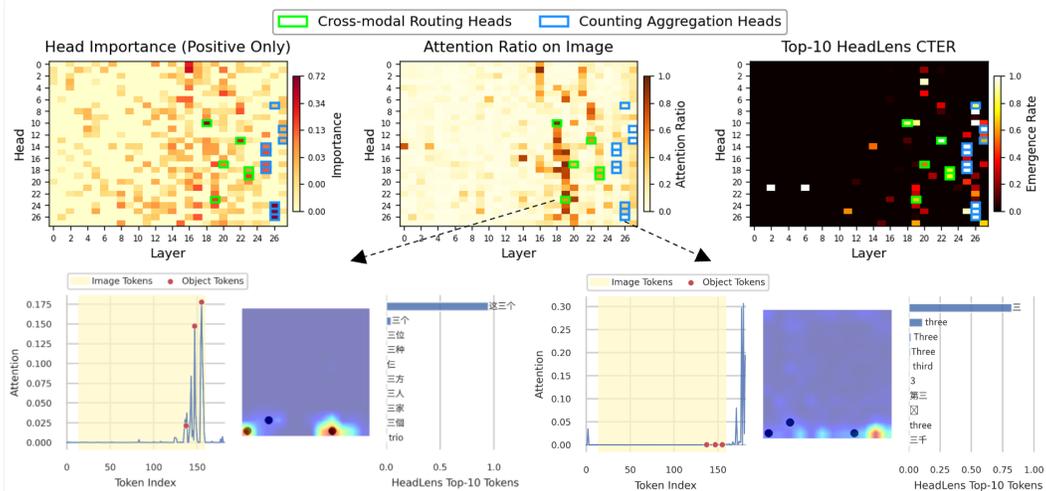
Figure 5: **Visualization of two functional groups of counting heads**. Left to right: Head importance (positive-only), attention ratio on image tokens, and top-10 HeadLens CTER across heads. We present the typical attention distribution and top-10 HeadLens results for Cross-modal Routing Heads (green boxes) and Counting Aggregation Heads (blue boxes), using L19H23(left) and L26H26(right). Both have "three" (三) in Chinese as the top-1 HeadLens token.

CTER. These heads directly extract information from image tokens and transfer visual information into the language related to the counting task. As shown in fig. 5, these heads mainly lie in Layers 18–24 noted with green boxes. We visualize L19H23 as a typical example. It attends to object-related image regions and top-10 HeadLens tokens are relevant to number (e.g., three), further confirming its role in bridging image and text. **Counting Aggregation Heads** are defined as heads with low image attention, high importance, and high top-10 HeadLens CTER. Despite extracting information from image tokens, these heads aggregate counting-relevant information already deposited in the residual stream by earlier layers rather than extracting visual information directly. We visualize L26H26, and its top-10 decoded token exhibit a remarkably high alignment with the model's final counting prediction while paying little attention to the image, confirming their role in counting aggregation.

Visual Grounding Heads. Concentrated primarily in the early stages of the network, specifically within Layer 1, these attention heads are dedicated to fundamental visual processing. They predominantly attend to object-specific image regions, serving as the primary mechanism for extracting basic, low-level visual attributes such as color, boundaries, and shape. We collect the HeadLens top-10 tokens of each head and compute the VGS accordingly. As shown in fig. 6, most visual grounding heads are located at layer 1, which has the highest average visual grounding score.

Existence and Difficulty Awareness Heads. We also identify two deep-layer attention heads that show awareness of object existence and the difficulty estimation of the counting. **L26H8** acts as a universal *existence detector*, outputting "1" whenever any object is present. **L23H19** is a difficulty-aware head and one of the best counting aggregation heads (32.9% top-1 accuracy). Besides number tokens, its top-10 tokens are mixed with "no" and "unnecessary" for $GT = 1$, indicating counting 1 dot is trivially easy. For $GT \geq 2$, the head switches to digit predictions with reasonable accuracy, and for higher counts begins mixing in Chinese difficulty-related tokens like difficult and impossible, alongside the numeric predictions.

7 Enhancing LVLMs' counting ability

Our findings reveal a step-by-step counting mechanism: early layers extract visual features, middle layers route information into a semantic space, and later layers aggregate the answer. Guided by these findings, we introduce two targeted interventions.
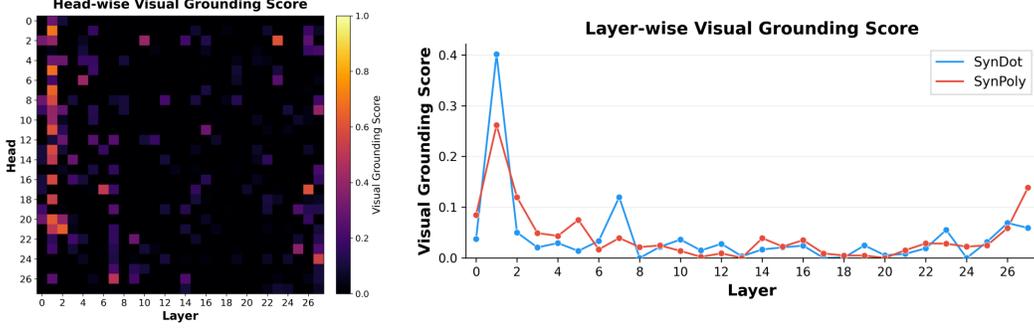
Figure 6: Visual Grounding Heads in Early Layers. Left: Head-wise heatmap based on Visual Grounding Score; Right: Layer-wise Visual Grounding Score.

7.1 Object-Focused Attention Regularizer

To improve the grounding of counting, we supervise the model's visual attention to be more "object-centric" using SynDot and SynPoly. Each image naturally provides object centers $\{c_k\}_{k=1}^N$, which we convert into a *soft instance prior* on the $H \times W$ patch grid. We define an unnormalized grid score $u(p)$ at patch p using a Gaussian kernel: $u(p) = \sum_{k=1}^N \exp\left(-\frac{\|p - \pi(c_k)\|_2^2}{2\sigma^2}\right)$, where $\pi(\cdot)$ maps pixels to patch coordinates and $\sigma = 1$ controls the supervision spread. We normalize $u(p)$ to obtain a target distribution $g \in \Delta^{|\mathcal{V}|-1}$ over the visual tokens $\mathcal{V}$.

We encourage the attention heads to allocate their weights on each image token $t \in \mathcal{T}$ to align with the instance prior g. For each layer l, we compute the average attention weights across all H heads and renormalize it over $\mathcal{V}$ to obtain the predicted distribution q_t^l:

$$q_t^l(j) = \frac{\sum_{h=1}^H a_h^l(t, j)}{\sum_{j' \in \mathcal{V}} \sum_{h=1}^H a_h^l(t, j')}, \quad j \in \mathcal{V}. \tag{3}$$

The focus loss is defined as the cross-entropy (equivalent to KL divergence) between g and q_t^l, where ϵ is a small number for stability:

$$\mathcal{L}_{\text{focus}} = \frac{1}{|\mathcal{L}||\mathcal{T}|} \sum_{l \in \mathcal{L}} \sum_{t \in \mathcal{T}} \left(-\sum_{j \in \mathcal{V}} g(j) \log\left(q_t^l(j) + \varepsilon\right) \right). \tag{4}$$

Guided by our circuit discovery (section 6), we explore applying attention supervision to the visual grounding heads at early layers and cross-modal routing heads with a high image attention ratio at late layers to verify if the model's counting mechanism can be precisely localized and regularized.

7.2 Adaptive Head Temperature Tuning

Our analysis reveals that cross-modal routing and counting aggregation heads exhibit highly targeted attention patterns. To amplify this intrinsic circuitry, we introduce Adaptive Head Temperature Tuning, which reduces the attention entropy of these critical heads to sharpen their focus on relevant tokens.

Instead of applying uniform temperature scaling, we adaptively modulate the pre-softmax logits of the identified target heads. For a given head h, we define an inverse temperature multiplier $\beta_h = \alpha \times \gamma_h$, where $\alpha \geq 1$ provides a baseline entropy reduction, and $\gamma_h \geq 0$ is the head's intrinsic importance score derived from our circuit analysis. The modified attention matrix is thus computed as:

$$A_h = \text{softmax}\left(\beta_h \frac{Q_h K_h^T}{\sqrt{d_k}}\right) \tag{5}$$

Applying this training-free technique exclusively to the routing and aggregation heads enhances the signal-to-noise ratio of the counting information flow.

9

7.3 Joint Optimization Objective

To integrate our targeted interventions, we combine standard Supervised Fine-Tuning (SFT) with our focus regularizer. The SFT process exclusively utilizes the SynDot and SynPoly datasets, restricting the autoregressive loss calculation strictly to the target counting tokens. The overall training objective is:

$$\mathcal{L} = \mathcal{L}_{\text{SFT}} + \lambda \, \mathcal{L}_{\text{focus}}, \tag{6}$$

where λ controls the regularization strength (set to 1 by default).

8 Experiments

Model and Data. We evaluate the proposed method on three different LVLMs: Qwen2.5-VL-7B, Qwen3-VL-8B and LLaVA-1.5-7B. The training data comprises 8000 synthetic images equally from SynDot and SynPoly (classes 1–10, evenly per class). For OOD counting, we evaluate on SynReal and PixMo-Count [32]. For visual reasoning, we use MMMU [6], RealWorldQA [8], and MathVista [7].

Training. We optimize with the joint loss (eq. (6)). For SFT, we fine-tune models using LoRA ($r = 64$) on the attention projection of all layers. We apply an attention regularizer to layers 2, 18-22 for Qwen2.5-VL-7B, layers 17-19 for Qwen3-VL-8B, and layers 0, 14 for LLaVA-1.5-7B, which have the most visual grounding heads and cross-modal routing heads. All models are trained for 2 epochs on a single NVIDIA H200 GPU. We use AdamW, BF16 precision, a batch size of 2, and a learning rate of 2×10^{-5} with linear decay and 3% warmup. We take the mean results with 3 random seeds. For head tuning, we use a global value $\alpha = 1.2$ across different models.

8.1 Evaluation Results

Counting Evaluation. Our method consistently improves OOD counting across all three backbones on both SynReal and PixMo-Count. As shown in Table table 2, the gains are consistent across all metrics, including higher accuracy and OBO, as well as lower MAE and RMSE. Notably, Qwen3-VL-8B, despite already being strong on SynReal, still shows clear gains, and LLaVA-1.5-7B also benefits substantially. These results show that our method transfers effectively to OOD counting rather than only fitting the training distribution. We provide the in-distribution (SynDot and SynPoly) and additional evaluation results in section E.2.

Backbone	Method	SynReal				PixMo-Count			
		Acc	MAE↓	RMSE↓	OBO↑	Acc	MAE↓	RMSE↓	OBO↑
Qwen2.5-VL-7B	Baseline	73.48	0.79	2.13	91.01	58.79	0.84	1.79	84.00
	Ours	**84.83**	**0.74**	**1.47**	**97.75**	**64.15**	**0.62**	**1.46**	**87.10**
Qwen3-VL-8B	Baseline	88.79	0.15	0.47	98.88	58.75	0.73	1.35	88.20
	Ours	**91.21**	**0.11**	**0.34**	**99.41**	**66.98**	**0.60**	**1.25**	**91.65**
LLaVA-1.5-7B	Baseline	54.27	3.59	5.05	66.34	31.88	1.81	3.09	59.77
	Ours	**60.11**	**1.56**	**2.08**	**91.01**	**32.64**	**1.61**	**2.59**	**62.43**

Table 2: Evaluation on Out-of-distribution Counting Benchmarks.

General Capabilities Evaluation. Our method consistently improves general visual capabilities across three backbones, though trained on the counting task only. As shown in Table table 3, all models achieve positive gains on MMMU, RealWorldQA, and MathVista, suggesting our intervention does not merely improve counting in isolation. Instead, it enhances the counting-related circuit which transfers to broader visual capabilities, indicating that counting may serve as a useful primitive for general visual reasoning. Notably, this improvement cannot be simply attributed to insufficient counting-related pretraining in the backbone. Qwen3-VL [33] emphasizes stronger visual grounding and reasoning in its model and training design than Qwen2.5-VL [5], especially with additional counting training data.

Backbone	Method	MMMU	RealWorldQA	MathVista	Δ (avg.)
Qwen2.5-VL-7B	Baseline	54.89	61.96	57.30	–
	Ours	**56.33**	**64.14**	**58.30**	+1.54
Qwen3-VL-8B	Baseline	58.33	70.05	66.30	–
	Ours	**60.74**	**71.83**	**67.90**	+1.93
LLaVA-1.5-7B	Baseline	44.44	56.21	23.90	–
	Ours	**44.56**	**56.48**	**26.30**	+0.93

Table 3: Evaluation on General Capability Benchmarks.

Figure 7: Jaccard Similarity for the top-20 most important heads of different tasks.

8.2 Ablation Study

We decompose our full method into its individual components: LoRA-based SFT, Object-Focused Attention Regularizer ($\mathcal{L}_{\text{focus}}$), and Adaptive Head Temperature Tuning (β_h). As shown in table 4, each component provides complementary gains. We provide additional ablation studies in section E.3.

Table 4: Component ablation on Qwen2.5-VL-7B across specialized and general benchmarks.

SFT	$\mathcal{L}_{\text{focus}}$	β_h	Synreal	PixMo-Count	MMMU	RealWorldQA	MathVista
Baseline			73.48	58.79	54.89	61.96	57.30
✓			79.78	61.67	56.00	63.14	58.10
✓	✓		82.34	63.32	56.11	64.05	58.20
✓	✓	✓	**84.83**	**64.15**	**56.33**	**64.14**	**58.30**

8.3 Mechanistic Overlap: Why Does Counting Enhancement Generalize?

Enhancing counting performance via our synthetic datasets (SynDot and SynPoly) unexpectedly improved OOD performance on broader tasks, including general VQA and mathematical reasoning. To investigate the origins of this transferability, we analyze five representative visual tasks: (1) **Counting** (SynDot and Pixmo-Count), (2) **Visual Attribution** (a custom color and shape diagnostic), (3) **Spatial Relations** (CLEVR [34]), (4) **General VQA**, and (5) **Math Reasoning** (MathVista).

To identify the underlying circuit mechanisms, we perform mean ablation at the head level. Specifically, we substitute each head's activation with its dataset-level mean and identify the Top-20 important heads based on the resulting logit degradation. We then use the Jaccard similarity J to quantify the mechanistic overlap across tasks. We use HS_A and HS_B to denote two arbitrary

important head sets.

$$J(HS_A, HS_B) = \frac{|HS_A \cap HS_B|}{|HS_A \cup HS_B|} \tag{7}$$

As shown in fig. 7, *Attribution Color* exhibits minimal overlap with other tasks, indicating reliance on pure perception. In contrast, *Counting* shares $> 33\%$ of its critical heads with reasoning-heavy tasks, explaining the OOD transferability. The high similarity between synthetic and real-world counting further confirms that our method targets fundamental enumeration circuits rather than dataset-specific artifacts. In contrast, applying the same intervention to visual attribution (color/shape recognition) yields no OOD reasoning gains (section F), reinforcing counting's critical role.

9 Conclusion

In this work, we show that counting is a meaningful probe of visual reasoning in LVLMs. Through controlled benchmarks and mechanistic analysis, we identify structured counting-related circuits and demonstrate that improving this primitive skill with lightweight synthetic-data intervention can yield gains beyond counting itself. These findings suggest that counting is not merely a narrow task, but an important building block of broader visual reasoning. Looking forward, an important direction is to examine whether these insights generalize to larger LVLMs and to other primitive visual skills, such as spatial reasoning and attribute comparison, toward a unified understanding of visual reasoning.

References

[1] S. J. Cheyette and S. T. Piantadosi, "A unified account of numerosity perception," *Nature human behaviour*, vol. 4, no. 12, pp. 1265–1272, 2020.

[2] F. Lieder and T. L. Griffiths, "Resource-rational analysis: Understanding human cognition as the optimal use of limited computational resources," *Behavioral and Brain Sciences*, vol. 43, p. e1, 2020, pMID: 30714890.

[3] S. J. Cheyette, S. Wu, and S. T. Piantadosi, "Limited information-processing capacity in vision explains number psychophysics." *Psychological Review*, vol. 131, no. 4, p. 891, 2024.

[4] H. Liu, C. Li, Q. Wu, and Y. J. Lee, "Visual instruction tuning," *Advances in neural information processing systems*, vol. 36, pp. 34 892–34 916, 2023.

[5] S. Bai, K. Chen, X. Liu, J. Wang, W. Ge, S. Song, K. Dang, P. Wang, S. Wang, J. Tang *et al.*, "Qwen2. 5-vl technical report," *arXiv preprint arXiv:2502.13923*, 2025.

[6] X. Yue, Y. Ni, K. Zhang, T. Zheng, R. Liu, G. Zhang, S. Stevens, D. Jiang, W. Ren, Y. Sun *et al.*, "Mmmu: A massive multi-discipline multimodal understanding and reasoning benchmark for expert agi," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2024, pp. 9556–9567.

[7] P. Lu, H. Bansal, T. Xia, J. Liu, C. Li, H. Hajishirzi, H. Cheng, K.-W. Chang, M. Galley, and J. Gao, "Mathvista: Evaluating mathematical reasoning of foundation models in visual contexts," *arXiv preprint arXiv:2310.02255*, 2023.

[8] xAI, "Grok-1.5 vision preview," https://x.ai/news/grok-1.5v, April 2024, accessed: [Insert Date Here].

[9] K. Wang, J. Pan, W. Shi, Z. Lu, H. Ren, A. Zhou, M. Zhan, and H. Li, "Measuring multimodal mathematical reasoning with math-vision dataset," *Advances in Neural Information Processing Systems*, vol. 37, pp. 95 095–95 169, 2024.

[10] M. F. Qharabagh, M. Ghofrani, and K. Fountoulakis, "Lvlm-count: Enhancing the counting ability of large vision-language models," *arXiv preprint arXiv:2412.00686*, 2024.

[11] R. Paiss, A. Ephrat, O. Tov, S. Zada, I. Mosseri, M. Irani, and T. Dekel, "Teaching clip to count to ten," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 3170–3180.

[12] S. Sengupta, N. Moradinasab, J. Liu, and D. E. Brown, "Can vision-language models count? a synthetic benchmark and analysis of attention-based interventions," *arXiv preprint arXiv:2511.17722*, 2025.

[13] N. Đukić, A. Lukežič, V. Zavrtanik, and M. Kristan, "A low-shot object counting network with iterative prototype adaptation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2023, pp. 18 872–18 881.

[14] Z. Huang, M. Dai, Y. Zhang, J. Zhang, and H. Shan, "Point segment and count: A generalized framework for object counting," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2024, pp. 17 067–17 076.

[15] H. Hasani, A. Izadi, F. Askari, M. Bagherian, S. Mohammadian, M. Izadi, and M. S. Baghshah, "Understanding counting mechanisms in large language and vision-language models," *arXiv preprint arXiv:2511.17699*, 2025.

[16] S. Alghisi, G. Roccabruna, M. Rizzoli, S. M. Mousavi, and G. Riccardi, "[de|re] constructing vlms' reasoning in counting," *arXiv preprint arXiv:2510.19555*, 2025.

[17] S. Heimersheim and N. Nanda, "How to use and interpret activation patching," *arXiv preprint arXiv:2404.15255*, 2024.

[18] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo *et al.*, "Segment anything," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2023, pp. 4015–4026.

[19] X. Guo, Z. Huang, Z. Shi, Z. Song, and J. Zhang, "Your vision-language model can't even count to 20: Exposing the failures of vlms in compositional counting," *arXiv preprint arXiv:2510.04401*, 2025.

[20] nostalgebraist, "Interpreting gpt: The logit lens," August 2020, accessed: 2024-12-21. [Online]. Available: https://www.lesswrong.com/posts/AcKRB8wDpdaN6v6ru/interpreting-gpt-the-logit-lens

[21] N. Belrose, Z. Furman, L. Smith, D. Halawi, I. Ostrovsky, L. McKinney, S. Biderman, and J. Steinhardt, "Eliciting latent predictions from transformers with the tuned lens," *arXiv preprint arXiv:2303.08112*, 2023.

[22] M. Sakarvadia, A. Khan, A. Ajith, D. Grzenda, N. Hudson, A. Bauer, K. Chard, and I. Foster, "Attention lens: A tool for mechanistically interpreting the attention head information retrieval mechanism," *arXiv preprint arXiv:2310.16270*, 2023.

[23] K. Meng, D. Bau, A. Andonian, and Y. Belinkov, "Locating and editing factual associations in gpt," *Advances in neural information processing systems*, vol. 35, pp. 17 359–17 372, 2022.

[24] K. Wang, A. Variengien, A. Conmy, B. Shlegeris, and J. Steinhardt, "Interpretability in the wild: a circuit for indirect object identification in gpt-2 small," *arXiv preprint arXiv:2211.00593*, 2022.

[25] C. Neo, L. Ong, P. Torr, M. Geva, D. Krueger, and F. Barez, "Towards interpreting visual information processing in vision-language models," in *The Thirteenth International Conference on Learning Representations*, 2025. [Online]. Available: https://openreview.net/forum?id=chanJGoa7f

[26] L. Che, T. Q. Liu, J. Jia, W. Qin, R. Tang, and V. Pavlovic, "Hallucinatory image tokens: A training-free eazy approach to detecting and mitigating object hallucinations in lvlms," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2025, pp. 21 635–21 644.

[27] A. Clark and Contributors, "Pillow (pil fork) documentation," 2015, accessed on <date of access>. [Online]. Available: https://pillow.readthedocs.io

[28] B. F. Labs, S. Batifol, A. Blattmann, F. Boesel, S. Consul, C. Diagne, T. Dockhorn, J. English, Z. English, P. Esser, S. Kulal, K. Lacey, Y. Levi, C. Li, D. Lorenz, J. Müller, D. Podell, R. Rombach, H. Saini, A. Sauer, and L. Smith, "Flux.1 kontext: Flow matching for in-context image generation and editing in latent space," 2025. [Online]. Available: https://arxiv.org/abs/2506.15742

[29] F. Zhang and N. Nanda, "Towards best practices of activation patching in language models: Metrics and methods," *arXiv preprint arXiv:2309.16042*, 2023.

[30] W. J. Yeo, R. Satapathy, and E. Cambria, "Towards faithful natural language explanations: A study using activation patching in large language models," in *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, 2025, pp. 10 436–10 458.

[31] C. Dumas, C. Wendler, V. Veselovsky, G. Monea, and R. West, "Separating tongue from thought: Activation patching reveals language-agnostic concept representations in transformers," in *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2025, pp. 31 822–31 841.

[32] M. Deitke, C. Clark, S. Lee, R. Tripathi, Y. Yang, J. S. Park, M. Salehi, N. Muennighoff, K. Lo, L. Soldaini *et al.*, "Molmo and pixmo: Open weights and open data for state-of-the-art vision-language models," in *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025, pp. 91–104.

[33] S. Bai, Y. Cai, R. Chen, K. Chen, X. Chen, Z. Cheng, L. Deng, W. Ding, C. Gao, C. Ge, W. Ge, Z. Guo, Q. Huang, J. Huang, F. Huang, B. Hui, S. Jiang, Z. Li, M. Li, M. Li, K. Li, Z. Lin, J. Lin, X. Liu, J. Liu, C. Liu, Y. Liu, D. Liu, S. Liu, D. Lu, R. Luo, C. Lv, R. Men, L. Meng, X. Ren, X. Ren, S. Song, Y. Sun, J. Tang, J. Tu, J. Wan, P. Wang, P. Wang, Q. Wang, Y. Wang, T. Xie, Y. Xu, H. Xu, J. Xu, Z. Yang, M. Yang, J. Yang, A. Yang, B. Yu, F. Zhang, H. Zhang, X. Zhang, B. Zheng, H. Zhong, J. Zhou, F. Zhou, J. Zhou, Y. Zhu, and K. Zhu, "Qwen3-vl technical report," 2025. [Online]. Available: https://arxiv.org/abs/2511.21631

[34] J. Johnson, B. Hariharan, L. Van Der Maaten, L. Fei-Fei, C. Lawrence Zitnick, and R. Girshick, "Clevr: A diagnostic dataset for compositional language and elementary visual reasoning," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2901–2910.

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

[36] W. Wang, Z. Gao, L. Gu, H. Pu, L. Cui, X. Wei, Z. Liu, L. Jing, S. Ye, J. Shao *et al.*, "Internvl3. 5: Advancing open-source multimodal models in versatility, reasoning, and efficiency," *arXiv preprint arXiv:2508.18265*, 2025.

[37] A. Treisman, "Feature binding, attention and object perception," *Philosophical Transactions of the Royal Society of London. Series B: Biological Sciences*, vol. 353, no. 1373, pp. 1295–1306, 1998.

[38] M. Minderer, A. Gritsenko, and N. Houlsby, "Scaling open-vocabulary object detection," *Advances in Neural Information Processing Systems*, vol. 36, pp. 72 983–73 007, 2023.

[39] Y. Li, S. Salehi, L. Ungar, and K. P. Kording, "Does object binding naturally emerge in large pretrained vision transformers?" *arXiv preprint arXiv:2510.24709*, 2025.

Appendix Table of Contents

A   Preliminaries of Large Vision Language Models

A typical Large Vision Language Model (LVLM) [5, 4, 32] consists of a vision encoder (e.g., vision transformer [35]), a cross-modal alignment module (e.g., linear layer or MLP), and a Large Language Model (LLM) backbone. Given an input image V, the LVLM first splits it into N_V image patches and then encodes them into a series of image tokens $[x_i^v]_{i=0}^{N_V}$. A paired instruction prompt T will be tokenized and processed as text tokens $[x_i^t]_{i=0}^{N_T}$, where N_T is the instruction token length. The final input sequences can also come with a system prompt claiming the LVLM's role in the conversation and tag prompt tokens such as "<image end>" and "Assistant:".

For a token x_i, its corresponding hidden states $s_i^l \in \mathcal{R}^{d_m}$ at the layer l of the LLM backbone is updated through multi-head self-attention(MHSA) and feed-forward(FF) sublayers with residual connection. For the MHSA layer with H heads, each head h applies causal attention scores $Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_h}} + M)V$ over the previous tokens. Here $Q, K, V \in \mathcal{R}^{n \times d_h}$ are the query, key, and value matrices projected from the input; M is the causal mask. We denote a_h^l as the attention scores of the head h at layer l, and $a_h^l(i, j)$ is the attention score from token x_i to token x_j.

A.1   Architecture Comparison for Different LVLMs

We discuss the architecture design difference between Qwen-2.5-VL, Qwen-3-VL and LLaVA-1.5 below. These three models have different designs on the vision encoder, cross-modal connector, and vision token construction methods.

Qwen2.5-VL. Qwen2.5-VL-7B utilizes a native dynamic-resolution Vision Transformer (ViT), which is integrated with window-based attention mechanisms to ensure that computational complexity scales linearly with the number of input patches. For the vision-language connector, the model employs a spatially-aware pooling strategy that groups and concatenates sets of four adjacent patch features, followed by a two-layer MLP to condense the visual token sequence. Additionally, the architecture incorporates Multimodal Rotary Position Embedding (MROPE). This embedding is aligned to absolute time and spatial coordinates, allowing the model to process spatial scales and temporal sequences (for video tasks) within a unified positional framework without requiring traditional coordinate normalization.

Qwen3-VL. The Qwen3-VL 8B is a dense vision-language foundation model that integrates a SigLIP-2-SO-400M vision encoder with an 8B Qwen3 language model backbone to enable sophisticated multimodal reasoning and agentic decision-making. Architecturally, it employs a two-layer MLP-based merger to compress 2×2 visual features into single tokens while incorporating a pioneering DeepStack integration that injects multi-level ViT features directly into the first three hidden layers of the LLM via lightweight residual connections. To achieve robust spatial-temporal modeling, the framework utilizes an Interleaved MROPE scheme that uniformly distributes temporal, horizontal, and vertical dimensions across the frequency spectrum, alongside an explicit text-based timestamp strategy (e.g., "<3.0 seconds>") for precise video grounding. This design natively supports interleaved contexts of up to 256K tokens, allowing the 8B model to maintain strong pure-text proficiency while achieving performance competitive with much larger previous-generation models on complex long-document and video understanding tasks.

LLaVA-1.5. The LLaVA 1.5 architecture is designed as a direct bridge between a CLIP ViT-L/14 vision encoder and a large language model. The connection is established through a lightweight linear projection layer, which serves as the sole interface to map visual features Z_v into the word embedding space of the language decoder. In this framework, the vision encoder functions as a visual tokenizer, converting images into a sequence of embeddings that the language model can process alongside text tokens. The model is trained end-to-end, focusing on the alignment between the visual features and the language model's latent space without the use of intermediate cross-attention modules or Q-formers.
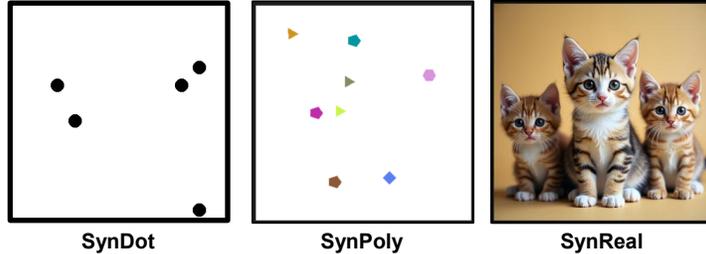
Figure 8: Data Samples of the three synthetic benchmarks.

B Datasets

B.1 SynDot

SynDot is a controlled synthetic dataset designed to isolate object counting from other visual complexities such as texture, color, and shape variation. Each image consists of a white canvas with a variable number of solid black circles (dots). Images are generated at a resolution of 336×336 pixels, where the canvas is partitioned into a grid of non-overlapping 28×28 patches. For each image, N patches are randomly selected (where $N \in \{1, 2, \ldots, 10\}$), and a black circle of fixed radius (default 4 pixels) is drawn at the center of each selected patch. This grid-based placement guarantees that dots never overlap, allowing the ground-truth count to be unambiguously determined. This design eliminates confounding visual factors and enables a focused evaluation of a model's fundamental numerosity perception. The training set contains 4000 images, and the test set comprises 2000 images.

B.2 SynPoly

SynPoly extends the synthetic paradigm to incorporate greater visual diversity while remaining fully controlled. Each image is a 336×336 white canvas populated with N randomly placed colorful polygons ($N \in \{1, 2, \ldots, 10\}$). For each polygon, the number of sides is uniformly sampled from $\{3, 4, 5, 6\}$ (i.e., triangles through hexagons), and the fill color is randomly drawn from the RGB space (excluding near-black and near-white values to ensure visibility). The polygon radius defaults to 8 pixels. By varying shape, color, and orientation simultaneously, SynPoly introduces within-image heterogeneity that more closely resembles real-world counting scenarios while still providing exact ground-truth labels. The test set contains 2000 images (200 per count), and a larger training set of 4000 images (400 per count) is prepared for supervised fine-tuning.

B.3 SynReal

SynReal bridges the gap between fully synthetic data and natural images by leveraging a state-of-the-art text-to-image diffusion model, FLUX.1-dev [28], to generate photorealistic images containing specified quantities of real-world objects. We define six common object categories(*cat*, *dog*, *bird*, *car*, *fish*, and *person*) and generate images for each category with object counts ranging from 1 to 10. Each image is produced at 1024×1024 resolution. The text prompt follows the template "*a photo of {N} {classname}*". For each category, 30 images are generated with distinct random seeds, yielding an evaluation set of 180 images. We conducted human verification for each of the generated images to ensure the label quality. SynReal enables evaluation of counting ability under realistic visual conditions, including complex textures, occlusions, varying poses, and natural backgrounds.

C Counting Evaluation Settings

C.1 Evaluation Metrics

To comprehensively assess the visual counting capabilities of LVLMs, we employ a multi-dimensional evaluation suite. Relying solely on exact-match accuracy is insufficient for visual counting, as it fails to distinguish between minor estimation errors and catastrophic reasoning failures. Therefore, our

metrics are designed to capture the exact precision, the magnitude of deviations, near-miss reliability, and the model's formatting compliance.

Let N denote the total number of evaluated examples in a given dataset. For the i-th example, let $y_i \in \mathbb{N}$ represent the ground-truth object count, and p_i denote the model's predicted count. To account for inference failures, where the model generates a vague, purely descriptive, or unparsable response instead of a discrete number, we assign a default penalty value of $p_i = -1$. We utilize the indicator function $\mathbf{1}[\cdot]$, which returns 1 if the inner condition is true and 0 otherwise.

The four evaluation metrics are formally defined as follows:

Accuracy (ACC). Accuracy serves as the strictest measure of counting proficiency, calculating the proportion of predictions that exactly match the ground-truth label.

$$\text{ACC} = \frac{1}{N} \sum_{i=1}^{N} \mathbf{1}[\, p_i = y_i \,]$$

Mean Absolute Error (MAE). Since visual counting is inherently a discrete regression task, MAE quantifies the average magnitude of numerical deviation from the ground truth. It provides an intuitive measure of how far off the model's predictions are on average, treating all linear errors equally.

$$\text{MAE} = \frac{1}{N} \sum_{i=1}^{N} |p_i - y_i|$$

Root Mean Square Error (RMSE). Unlike MAE, RMSE quadratically penalizes larger discrepancies before averaging. This metric is particularly sensitive to outliers or catastrophic counting failures (e.g., hallucinating a count of 10 when only 2 objects exist), making it a crucial indicator of the model's worst-case robustness.

$$\text{RMSE} = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (p_i - y_i)^2}$$

Off-by-one Accuracy (OBO). In both human cognition (the Approximate Number System) and computer vision, minor counting errors (± 1) frequently occur due to occlusion, ambiguous boundaries, or subitizing thresholds. OBO measures the proportion of predictions that fall within a strict ± 1 margin of error, serving as a proxy for "near-miss" reliability and functional estimation ability.

$$\text{OBO} = \frac{1}{N} \sum_{i=1}^{N} \mathbf{1}[\, |p_i - y_i| \leq 1 \,]$$

D Additional Interpretability Results

D.1 Revealing Attention Head Functionalities (Extended)

In this section, we detail the experimental design and evaluation criteria for head function discovery. To systematically categorize the functional roles of individual attention heads, we evaluate them across three dimensions: importance scores, attention distribution patterns, and top-10 HeadLens projections.

Based on the overall distribution (visualized in fig. 9), we set an importance score threshold of 0.05, identifying 43 out of 784 heads (5.5%) as functionally critical.

For the attention distribution analysis, we first examine the allocation of weights between image and text tokens. Within the visual tokens, we further quantify the proportion of attention directed toward object-relevant regions. Consequently, our primary filtering step isolates heads that both exceed the 0.05 importance threshold and allocate over 40% of their total attention mass to image tokens. These heads are more likely to have visual-centric functions. We further evaluate the ratio of ground truth number tokens in top-10 HeadLens results and the top-1 HeadLens counting accuracy for each head. We categorize the identified influential heads into two functional groups based on their behavior
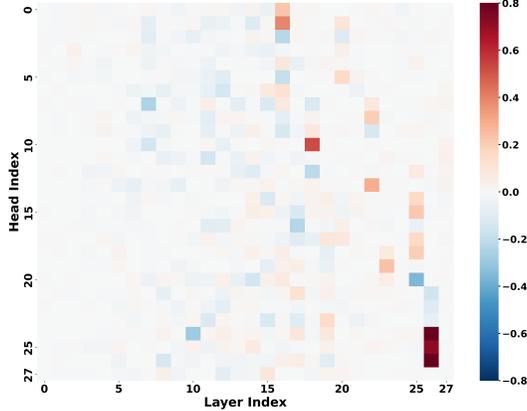
Figure 9: Head Importance Heatmap for Qwen2.5-VL on SynDot

Table 5: Top-20 attention heads ranked by importance and categorized by HeadLens projection. **Importance**: importance score from head ablation; **Img_Attn**: fraction of attention on image tokens; **Obj_in_Img**: fraction of image attention on object-relevant regions; **GT@10**: ground-truth count in top-10 HeadLens projections; **Top-1 Acc**: top-1 HeadLens projection accuracy. Rows in green denote *Counting Aggregation Heads* and rows in blue denote *Cross-Modal Routing Heads*.

Rank	ID	Import.	Img_Attn	Obj_in_Img	GT@10	Top-1 Acc
1	L26H26	0.801	0.034	0.032	0.83	0.55
2	L26H24	0.792	0.301	0.005	0.83	0.53
3	L26H25	0.716	0.314	0.023	0.92	0.77
4	L18H10	0.521	0.937	0.129	0.13	0.03
5	L16H1	0.393	0.903	0.260	0.10	0.00
6	L22H13	0.289	0.538	0.094	0.68	0.48
7	L23H19	0.232	0.470	0.033	0.32	0.17
8	L16H0	0.222	0.469	0.173	0.10	0.00
9	L22H8	0.198	0.708	0.024	0.07	0.00
10	L20H5	0.161	0.105	0.059	0.34	0.22
11	L25H17	0.157	0.155	0.041	0.69	0.30
12	L19H23	0.154	0.535	0.107	0.30	0.14
13	L20H1	0.104	0.015	0.083	0.21	0.10
14	L19H17	0.098	0.942	0.481	0.10	0.05
15	L19H24	0.087	0.689	0.014	0.10	0.00
16	L20H17	0.085	0.405	0.041	0.14	0.02
17	L22H7	0.084	0.564	0.059	0.10	0.00
18	L23H18	0.075	0.426	0.006	0.26	0.03
19	L16H18	0.064	0.515	0.029	0.11	0.00
20	L20H3	0.054	0.067	0.066	0.29	0.18

across three metrics: (1) Counting Aggregation Heads (CAH), which exhibit high alignment with ground-truth counts in HeadLens projections ($Top\text{-}1\ Acc > 0.1$) despite lower direct image attention; and (2) Cross-Modal Routing Heads (CMR), which prioritize extracting spatial and object-relevant visual features ($Img\_Attn > 0.5$) but do not directly encode numerical information. We demonstrate the top-20 most important heads and their features in table 5. Note that in practice, we combine the five metrics and their top-10 HeadLens frequency dictionary as the head categorization feature.

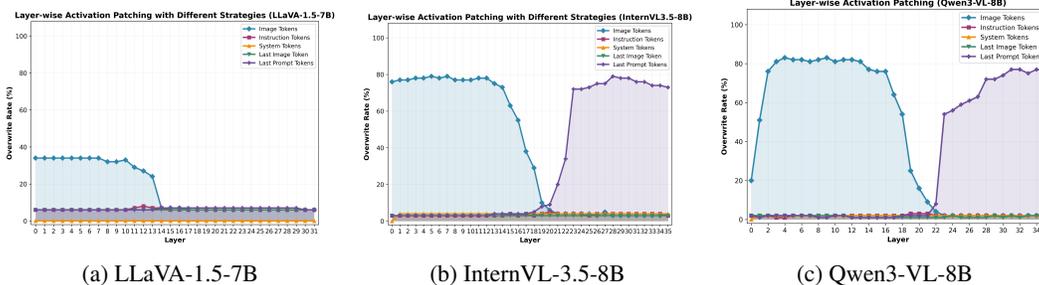(a) LLaVA-1.5-7B   (b) InternVL-3.5-8B   (c) Qwen3-VL-8B

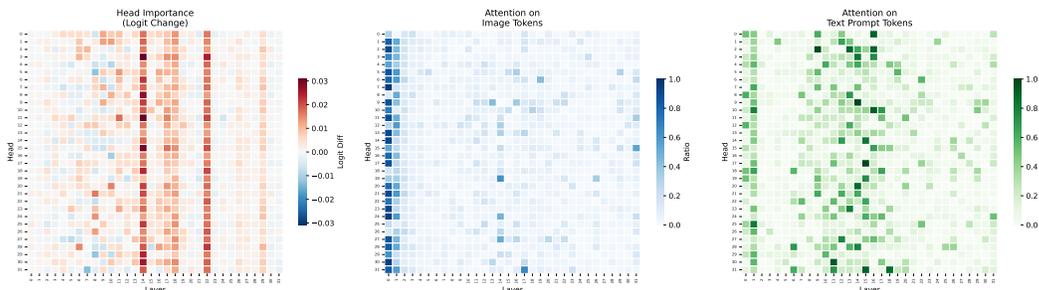Figure 10: Layer-wise VAP on SynDot for LLaVA-1.5-7B, InternVL3.5-8B and Qwen3-VL-8B.

Figure 11: LLaVA1.5-7B: Head-wise HeatMap on Head Importance, Attention Ratio on Image Tokens and Text Tokens (from left to right).

D.2 Additional Layer-wise Visual Activation Patching Result

We apply the same Layer-wise Visual Activation Patching experiment on LLaVA-1.5-7B, InternVL-3.5 [36] and Qwen3-VL-8B. It can be observed that both InternVL-3.5 and Qwen3-VL shows exactly the same cross-modal routing patterns as Qwen2.5-VL. The LLaVA-1.5 also shows the overwrite rate declining for the image tokens, indicating the counting information is transferred from visual information into the generated token directly. We hypothesize that LLaVA 1.5 fails to route counting information from the image tokens to the final prompt token because its corresponding circuitry remains underdeveloped during training, as illustrated in fig. 15.

D.3 Additional Head-wise Analysis

Similarly, we apply the head-level activation patching detailed in section 6 to LLaVA1.5-7B and Qwen3-VL-8B to identify the attention heads critical for counting. As illustrated in fig. 11, the most vital layers in LLaVA 1.5 are layers 14 and 22, while its first layer heads allocate the highest attention weights to image tokens. For Qwen3-VL (fig. 12), the behavior mirrors Qwen2.5-VL: functionally
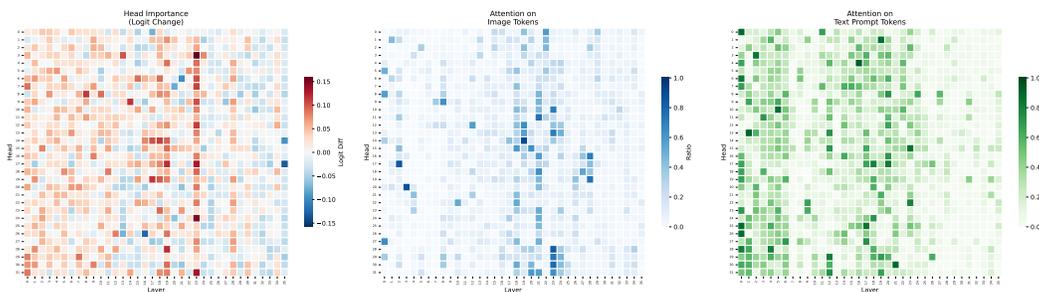

Figure 12: Qwen3-VL-8B: Head-wise HeatMap on Head Importance, Attention Ratio on Image Tokens and Text Tokens (from left to right).
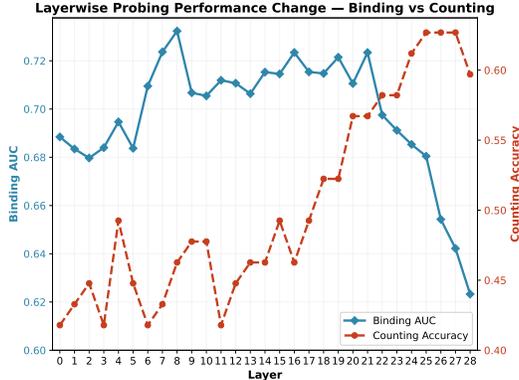
Figure 13: Layer-wise Probing Performance Curves for Binding and Counting.

important heads predominantly cluster in the middle layers and maintain the strongest focus on visual tokens.

D.4 Probing the Dual-Stage Mechanism: Perception and Abstraction

We posit that visual counting is not a monolithic process but comprises two distinct cognitive stages: Perception, which necessitates robust *object binding* [37] to individuate distinct instances from the background; and Reasoning, which abstracts these visual signals into a *numerical quantity*. To empirically validate this hypothesis, we design two targeted probing tasks using the SynDot and CountBenchQA dataset [11] for both synthetic and real-world objects.

Data Curation Pipeline. To isolate object-specific representations, we employ a two-stage preprocessing pipeline. First, OwlViT-v2 [38] performs open-vocabulary detection to localize bounding boxes for the target noun (e.g., "cats"). Subsequently, the SAM [18] generates precise pixel-level masks for each detected box. This yields a set of instance-specific binary masks $\mathcal{M} = \{m_1, ..., m_N\}$ for each image, serving as the ground truth for our probes.

Task A: Object Binding Probe (Perception). This task evaluates whether the model's internal representations can distinguish between independent entities. Given two visual tokens x_i^v and x_j^v, the probe predicts whether they belong to the same object instance. To capture relational subspace similarities that linear classifiers might miss, we follow [39] to employ a Quadratic Probe with rank $r = 64$. The similarity score is computed as:

$$\text{Score}(x_i^v, x_j^v) = (W x_i^v)^\top (W x_j^v) \tag{8}$$

where $W \in \mathbb{R}^{r \times d}$ is the learnable projection matrix. We measure performance using **ROC AUC**, quantifying the separability of instance-level features.

Task B: Numerosity Probe (Abstraction). This task assesses the model's ability to abstract a global count from visual features. We first compute the mean-pooled embedding of all tokens falling within the union of object masks, explicitly excluding background tokens to focus on object-centric representations. These aggregated embeddings are fed into an MLP classifier to predict the ground-truth count. Performance is evaluated via Accuracy, serving as a proxy for the distinctiveness of the model's internal numerical manifold.

As illustrated in fig. 13, the blue and red curves denote the layer-wise binding and counting performances, respectively. The binding AUC grows gradually through the early and middle layers before declining in the late layers as semantic information aggregates. Conversely, counting accuracy starts low but surges during the cross-modal routing stage between layers 15 and 23. This precise trajectory directly corroborates our layer-wise functional findings detailed in section 5.1.

D.5 AttentionLens vs HeadLens

In this section we provide a detailed comparison between HeadLens (ours) and AttentionLens [22], another head-level interpretability method. We first formalize both training pipelines, then argue that

21

AttentionLens's unconstrained per-head probing introduces systematic noise that undermines circuit discovery, and finally show that HeadLens is orders of magnitude more efficient.

All experiments below are conducted on Qwen2.5-VL-7B-Instruct (28 layers $\times$ 28 heads, $d_{\text{head}} = 128$, $d_{\text{model}} = 3584$, $|\mathcal{V}| = 152{,}064$) using 2000 SynPoly samples (200 per count class, classes 1–10).

D.5.1 Training Formulations.

HeadLens extends the Tuned Lens [21] to the per-head level. *Phase 1 (per-layer training):* For each layer l, a single affine translator $T_l(\mathbf{x}) = W_l \mathbf{x} + \mathbf{b}_l$, $W_l \in \mathbb{R}^{d_{\text{model}} \times d_{\text{model}}}$, $\mathbf{b}_l \in \mathbb{R}^{d_{\text{model}}}$, is trained to minimize:

$$\mathcal{L}_{\text{HL}}^{(l)} = D_{\text{KL}}\Big(\text{softmax}(\text{lm\_head}(\text{RMSNorm}(T_l(\mathbf{a}_l)))) \, \big\| \, \text{softmax}(\mathbf{y}_{\text{final}}) \Big), \tag{9}$$

where $\mathbf{a}_l \in \mathbb{R}^{d_{\text{model}}}$ is the *full* attention output at layer l (post-W_O, pre-residual), and $\mathbf{y}_{\text{final}}$ is the model's final logits. All components except $(W_l, \mathbf{b}_l)$ are frozen; W_l is initialized to identity and $\mathbf{b}_l$ to zero, ensuring the lens begins as a no-op.

Phase 2 (per-head decoding, no additional training): To decode head (l, h), we isolate its contribution via zero-padding and project through the corresponding W_O slice:

$$\mathbf{p}_{l,h} = W_O[:, \, h \cdot d_{\text{head}} : (h+1) \cdot d_{\text{head}}] \cdot \mathbf{z}_{l,h}, \tag{10}$$

where $\mathbf{z}_{l,h} \in \mathbb{R}^{d_{\text{head}}}$ is the raw head output. Then the trained translator and frozen unembedding produce:

$$\text{logits}_{l,h} = \text{lm\_head}(\text{RMSNorm}(T_l(\mathbf{p}_{l,h}))) \in \mathbb{R}^{|\mathcal{V}|}. \tag{11}$$

HeadLens thus decodes every head through the model's *own computational pathway*, reflecting each head's actual contribution to the residual stream.

AttentionLens trains a separate linear probe for *every* head (l, h):

$$f_{l,h}(\mathbf{z}_{l,h}) = W_{l,h}\, \mathbf{z}_{l,h} + \mathbf{b}_{l,h}, \quad W_{l,h} \in \mathbb{R}^{d_{\text{model}} \times d_{\text{head}}}, \ \mathbf{b}_{l,h} \in \mathbb{R}^{d_{\text{model}}}, \tag{12}$$

trained to minimize:

$$\mathcal{L}_{\text{AL}}^{(l,h)} = D_{\text{KL}}\Big(\text{softmax}(\text{lm\_head}(\text{RMSNorm}(f_{l,h}(\mathbf{z}_{l,h})))) \, \big\| \, \text{softmax}(\mathbf{y}_{\text{final}}) \Big). \tag{13}$$

Like HeadLens, AttentionLens reuses the model's frozen RMSNorm and lm_head for decoding. However, the critical architectural difference is that it *bypasses* the model's output projection W_O: instead of projecting head activations through the architectural W_O slice, it learns an unconstrained linear map from $\mathbb{R}^{d_{\text{head}}}$ to $\mathbb{R}^{d_{\text{model}}}$. Each probe has $\sim$0.46M parameters (784 probes in total, $\sim$362M parameters overall) and is individually optimized per head.

D.5.2 Critique: Forced Semantic Uniformity Undermines Circuit Discovery.

The critical methodological concern with AttentionLens lies in eq. (13): *every* head is trained via KL divergence to approximate the model's final output distribution. This training objective implicitly assumes that each head's activation should be interpretable as a complete vocabulary distribution resembling the model's prediction. However, this assumption is fundamentally flawed:

- **Not every head participates in the current task.** In a 784-head model, only a small fraction ($\sim$5.5% as identified by activation patching in section D.1) causally contributes to counting. The remaining heads serve other functions (syntactic parsing, positional encoding, copy behavior) or are largely inactive for the given input.

- **Not every participating head is responsible for the final semantic output.** Even among task-relevant heads, many perform intermediate operations—e.g., cross-modal routing or visual grounding—whose internal representations are meaningful but do not directly resemble the final token distribution.

Despite this, the KL-divergence training in eq. (13) *forces* each probe to fit the final distribution regardless. With no architectural constraint tying the learned map to W_O, each probe has full freedom to discover whatever linear relationship between $\mathbb{R}^{d_{\text{head}}}$ and $\mathbb{R}^{d_{\text{model}}}$ best approximates the target, even
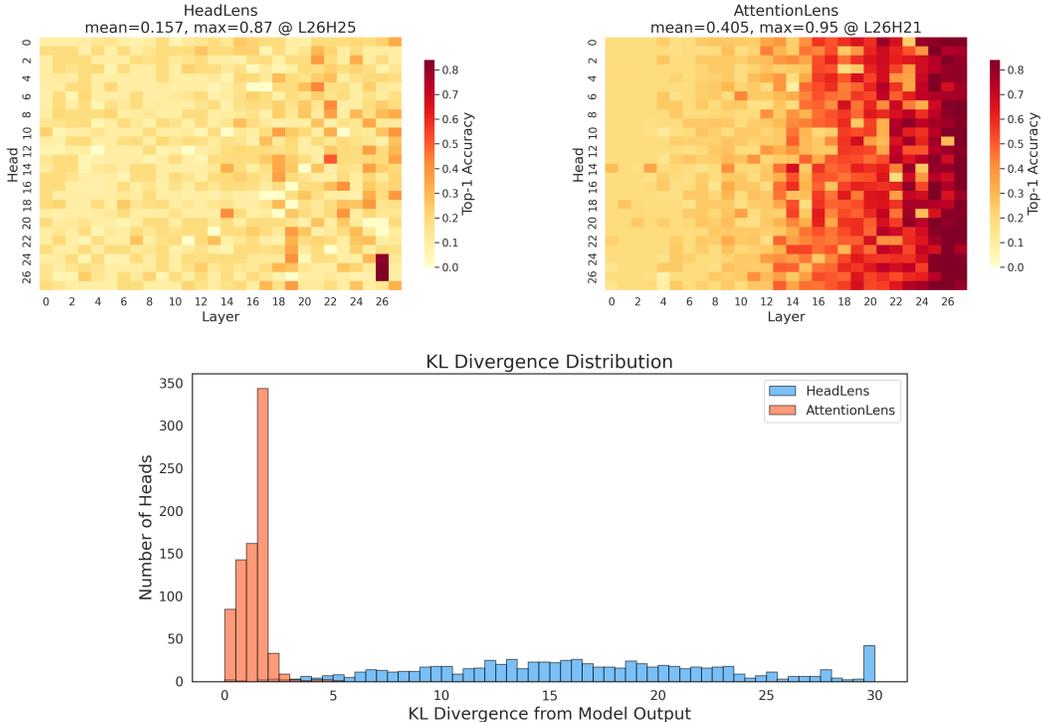
Figure 14: **Comparison of HeadLens and AttentionLens on SynPoly.** *Top*: Head-wise top-1 accuracy on model final output (not ground truth counting). HeadLens (left) produces a sparse signal where only a few heads in Layer 26 stand out, which is aligned with our head importance heatmap in fig. 9. While AttentionLens (right) spreads high accuracy across the majority of heads, especially in layers 14–27. *Bottom*: KL divergence distribution across all 784 heads. AttentionLens concentrates nearly all heads into a narrow low-KL band (KL $<$ 3), collapsing the dynamic range needed to distinguish critical from non-critical heads. HeadLens preserves a broad distribution that naturally separates the counting circuit from irrelevant heads.

for heads whose actual W_O projection contributes nothing task-relevant. The result, as shown in fig. 14, is that AttentionLens makes nearly *every* head appear to produce a distribution closely aligned with the model's final output.

This "semantic flattening" effect is directly visible in fig. 14. Under AttentionLens, the top-1 accuracy on model response heatmap shows a large, densely activated region spanning most of layers 14–27, with a mean top-1 accuracy of 0.405 (compared to HeadLens's 0.157). The KL divergence histogram further reveals the problem: AttentionLens compresses virtually all 784 heads into a narrow low-KL band (KL $<$ 3), while HeadLens preserves a broad spread from 0 to 30. Under AttentionLens, the few genuinely important counting heads (L26H24–26) are buried within hundreds of heads that *appear* equally capable.

For mechanistic interpretability, this is counterproductive. The goal of circuit discovery is to identify the *minimal* set of attention heads that causally drive a behavior, which requires a method that clearly separates contributing heads from non-contributing ones. To quantify this, we compute the best-to-mean VGS ratio as a discriminability measure: HeadLens yields $0.772/0.134 = 5.8\times$, while AttentionLens yields only $0.639/0.229 = 2.8\times$. The sparser HeadLens signal makes it straightforward to threshold and identify the counting circuit; the flatter AttentionLens signal requires additional filtering that reintroduces subjectivity.

D.5.3 HeadLens: Faithfulness via Architectural Consistency.

HeadLens avoids the above pitfall because its translator is trained on the *full aggregate* attention output $\mathbf{a}_l$ (eq. (9)) rather than on individual heads. The translator learns to bridge the representational

gap between intermediate attention outputs and the final layer, a correction that is *shared* across all heads within a layer. When this shared translator is subsequently applied to a *single* head's projected output $\mathbf{p}_{l,h}$, it does not force that head to mimic the final distribution. Instead, heads that genuinely contribute counting-relevant information through W_O produce strong decoded signals, while heads that contribute little or encode information in subspaces projected away by W_O naturally produce weak or incoherent signals.

This property follows from the causal structure of the transformer itself: the residual stream at each layer is the sum of all head contributions, and the final output is causally determined by these accumulated contributions passed through RMSNorm and lm_head. By preserving this pathway, HeadLens measures what each head actually *writes into* the residual stream rather than what can be *extracted from* its activation by an unconstrained probe. The distinction is precisely the difference between *causal contribution* and *information existence*—the former is what circuit discovery requires.

D.5.4 Computational Cost.

Beyond the methodological concerns, AttentionLens incurs substantial computational overhead due to the sheer number of independently trained probes (table 6). Although both methods reuse the model's frozen lm_head and have comparable total parameter counts ($\sim$360M), AttentionLens must train 784 independent probes (each $\mathbb{R}^{d_{\text{head}}} \to \mathbb{R}^{d_{\text{model}}}$) with $\sim$235K total optimization steps, whereas HeadLens trains only 28 shared translators (each $\mathbb{R}^{d_{\text{model}}} \to \mathbb{R}^{d_{\text{model}}}$) with $\sim$14K steps.

Table 6: Computational cost comparison between HeadLens and AttentionLens.

	HeadLens	AttentionLens	Ratio
# Lenses trained	28	784	28$\times$
Parameters per lens	$\sim$12.8M	$\sim$0.46M	0.04$\times$
Total parameters	$\sim$360M	$\sim$362M	$\sim$1$\times$
Training steps	14,000	235,200	17$\times$
Wall time (SynPoly)	37.9 s	2,385.1 s	63$\times$
Wall time (SynDot)	41.9 s	2,414.5 s	58$\times$

In practice, HeadLens completes full-model analysis in under one minute, whereas AttentionLens requires $\sim$40 minutes on the same hardware (NVIDIA RTX A6000)—a $\sim$**60**$\times$ speedup. Despite comparable total parameter counts, the 28$\times$ difference in the number of independently trained lenses dominates the wall-clock cost due to repeated optimizer initialization, data loading, and per-probe forward/backward passes. This efficiency gap is especially consequential for iterative workflows in mechanistic interpretability, where practitioners repeatedly analyze head behaviors across different inputs, tasks, and model checkpoints.

D.5.5 Summary.

Both methods localize the same core counting circuit (Layer 26, Heads 24–26), confirming the robustness of this finding. However, HeadLens is strongly preferred for circuit discovery:

1. **Faithful decoding**: HeadLens measures each head's causal contribution through the model's own pathway ($W_O \to T_l \to \text{RMSNorm} \to \text{lm\_head}$), whereas AttentionLens bypasses W_O with a freely learned linear map, conflating information *existence* with information *usage*.

2. **Discriminability**: By not forcing every head to mimic the final output, HeadLens naturally produces a sparse signal (best/mean VGS ratio 5.8$\times$ vs. 2.8$\times$) that cleanly separates counting-critical heads from irrelevant ones. AttentionLens's forced semantic uniformity makes nearly all heads appear meaningful, introducing noise that hinders both circuit discovery and functional decomposition.

3. **Efficiency**: Despite comparable total parameter counts, HeadLens runs $\sim$60$\times$ faster by training only 28 shared translators instead of 784 independent probes, making it practical for routine, large-scale head-level analysis.
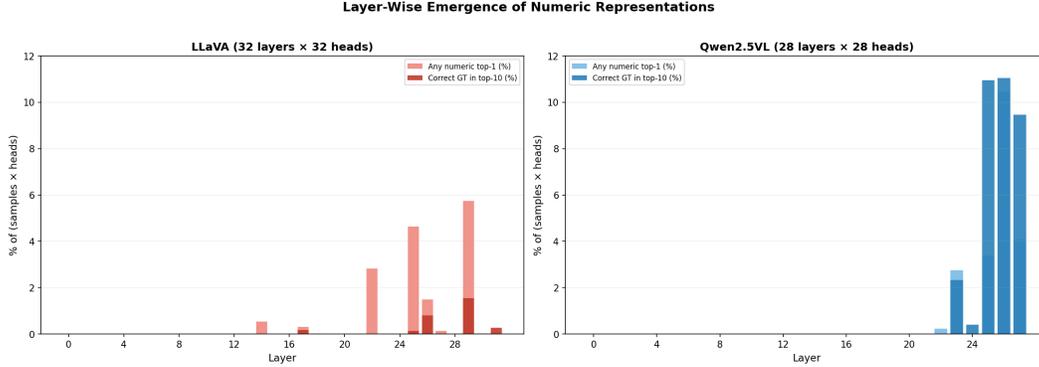
Figure 15: Layer-wise emergence of numeric representations for both models. Both show zero numeric signal in the first 70% of layers. Qwen's emergence is more concentrated (layers 25–27) and achieves far higher correct-in-top-10 rates (up to 11%) versus LLaVA's peak of 1.6%.

D.6 HeadLens Results Comparison Between LLaVA1.5-7B and Qwen2.5-VL-7B

We compare the architecture difference between LLaVA1.5-7B and Qwen2.5-VL-7B based on the HeadLens discovery based on SynDot dataset.

Layer-Wise Emergence Comparison. As in fig. 15, both models share the pattern of late emergence-numeric representations appear only in the final 20% of the network. However, Qwen's emergence is more concentrated and more powerful. While LLaVA shows scattered, weak numeric signals across layers 14–31, Qwen is completely silent until layer 22 and then explodes with a strong signal across layers 23–27. Qwen's peak per-layer correct-in-top-10 rate (11%) is $7\times$ higher than LLaVA's peak (1.6%), consistent with its distributed counting architecture. Qwen compresses its counting computation into a narrower, deeper band of layers ($79 - 96\%$ depth) but deploys far more heads within that band. LLaVA spreads weak signals over a wider layer range ($44 - 97\%$ depth) but never achieves meaningful concentration. This suggests Qwen's architecture enables more efficient "late-stage" counting circuits.

E   Additional Experiment

E.1   In Distribution Counting Evaluation

Model	Method	SynDot				SynPoly			
		Acc	MAE↓	RMSE↓	OBO↑	Acc	MAE↓	RMSE↓	OBO↑
Qwen2.5-VL-7B	Baseline	76.12	0.25	0.52	96.21	53.74	0.94	1.75	82.27
	Ours	**89.20**	**0.11**	**0.33**	**100.00**	**79.10**	**0.21**	**0.47**	**99.60**
Qwen3-VL-8B	Baseline	73.22	0.32	0.66	95.21	70.34	0.52	1.49	94.02
	Ours	**95.40**	**0.05**	**0.21**	**100.00**	**93.40**	**0.07**	**0.26**	**100.00**
LLaVA-1.5-7B	Baseline	24.07	4.46	6.58	39.22	33.30	1.65	2.34	56.71
	Ours	**75.30**	**0.26**	**0.53**	**98.90**	**72.50**	**0.29**	**0.56**	**98.70**

Table 7: In-distribution counting benchmarks with full metrics.

Across three backbones, our method consistently improves in-distribution counting on both SynDot and SynPoly. The gain is not only in exact accuracy, but also in the size of counting errors: MAE/RMSE drop clearly, which means mistakes become much smaller rather than only shifting between nearby counts. In addition, OBO increases to 100% for the stronger models, showing that predictions are almost always within one count of the correct answer.

Table 8: CountBenchQA results.

Model	Method	Acc ↑	MAE ↓	RMSE ↓	OBO ↑
Qwen2.5-VL-7B	Baseline	80.44	0.66	4.34	89.61
	Ours	**82.08**	**0.58**	**4.28**	**91.24**
Qwen3-VL-8B	Baseline	87.98	0.23	0.86	95.52
	Ours	**90.22**	**0.20**	**0.85**	**95.93**
LLaVA-1.5-7B	Baseline	42.16	1.38	2.08	59.67
	Ours	**47.86**	**1.15**	**1.78**	**67.01**

A notable pattern is that SynPoly benefits more than SynDot for the same backbone. SynPoly contains more varied shapes and layouts, so it is a harder in-distribution subset. The larger improvement there suggests the method helps with structured scenes and complex visual grouping, not only with simple dot patterns.

E.2 Additional Counting Evaluation

E.2.1 Counting Evaluation on CountBenchQA

As shown in table 8, our method yields consistent improvements over the baseline across all three backbones on the real-world CountBenchQA benchmark [11]. Notably, LLaVA-1.5-7B benefits the most, with accuracy increasing by $+5.70$ percentage points (from 42.16% to 47.86%), MAE decreasing from 1.38 to 1.15, RMSE dropping from 2.08 to 1.78, and OBO rising substantially from 59.67% to 67.01%. This large margin suggests that the weaker baseline has more room for improvement, and our interpretability-guided training effectively strengthens its visual counting circuits. For the stronger Qwen-series models, the gains are moderate but consistent: Qwen3-VL-8B achieves $+2.24$ accuracy improvement (from 87.98% to 90.22%) while Qwen2.5-VL-7B improves by $+1.64$ points. Importantly, even for these already-strong baselines, all four metrics improve simultaneously, confirming that the enhancement is not limited to exact-match accuracy but extends to reducing error magnitude (MAE/RMSE) and improving near-miss reliability (OBO). These results demonstrate that the counting ability acquired from synthetic training transfers effectively to real-world images with natural backgrounds, complex textures, and diverse object categories.

E.2.2 Counting Beyond 10

To further verify that our method enhances intrinsic counting capabilities rather than inducing rote memorization, we expand the object range of the SynDot dataset to [1, 30]. Specifically, we train Qwen3 exclusively on the 1 to 10 range and evaluate its performance on two unseen intervals: 11 to 20 and 21 to 30. As shown in table 9, our approach consistently improves overall counting accuracy across these extrapolated ranges. This confirms a genuine enhancement of the model's foundational numerosity mechanisms.

Table 9: Performance comparison over larger counting ranges for Qwen3-VL-8B. **Ours** are finetuned on the number range 1-10 only.
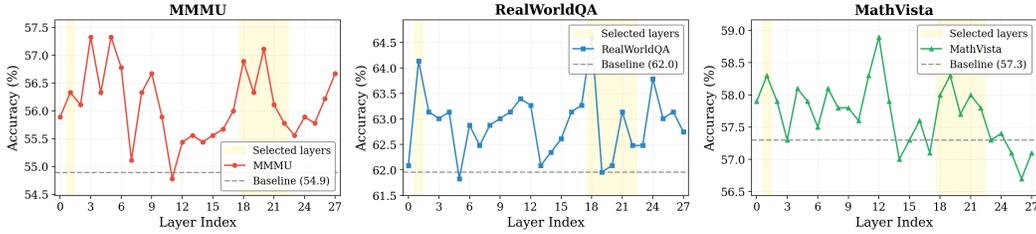
Model	Method	Range	Acc ↑	MAE ↓	RMSE ↓	OBO ↑
Qwen3-VL-8B	Baseline	11-20	35.00	0.90	1.25	80.02
		21-30	10.42	2.49	3.02	32.37
	Ours	11-20	**48.16**	**0.59**	**0.86**	**93.45**
		21-30	**18.27**	**1.61**	**2.05**	**52.81**

E.3 Additional Ablation Study

We provide additional ablation studies in this section for the layer choice of Attention Regularizer and the value choice of α for head temperature tuning.
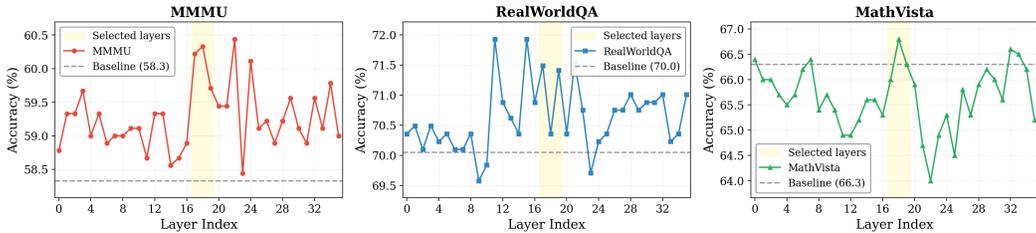
E.3.1 Layer Choice for Attention Regularizer

(a) Layer-wise ablation analysis for **Qwen2.5-VL-7B**.

(b) Layer-wise ablation analysis for **Qwen3-VL-8B**.

(c) Layer-wise ablation analysis for **LLaVA-1.5-7B**.

Figure 16: **Layer-wise ablation of the attention regularizer for three LVLMs.**

fig. 16 illustrates the performance variations across three visual reasoning tasks when applying the object focus regularizer to individual layers. We optimize the model using both SFT and $\mathcal{L}_{\text{focus}}$ on the SynDot and SynPoly datasets, isolating the regularization to a single layer per trial. Notably, applying this attention regularizer to the critical functional layers identified in our prior interpretability analysis consistently yields positive performance gains. This directly validates the practical utility of our mechanistic findings.

E.3.2 Impact of α on Adaptive Head Temperature Tuning

We study the effect of the baseline inverse temperature coefficient α in Adaptive Head Temperature Tuning (HTT). Recall that each target head's logits are scaled by $\beta_h = \alpha \times \gamma_h$, where γ_h is the head importance score from our circuit analysis. When $\alpha = 1.0$, the attention distribution remains unmodified, reducing to the SFT+$\mathcal{L}_{\text{focus}}$ baseline. We sweep $\alpha \in \{1.1, 1.2, 1.3\}$ and additionally evaluate whether combining HTT with Head Importance Reweighting (HIR)—which re-scales the output contributions of the identified counting heads proportionally to their importance scores—yields further gains. All experiments are conducted on Qwen2.5-VL-7B and evaluated on three general visual reasoning benchmarks (MMMU, RealWorldQA, MathVista) to assess whether sharpening target heads preserves or improves general capabilities beyond counting.

Table 10: Ablation study of head temperature tuning (HTT) coefficient α and head importance reweighting (HIR) on Qwen2.5-VL-7B. The baseline method is SFT+$\mathcal{L}_{\text{focus}}$ with $\alpha = 1.0$.

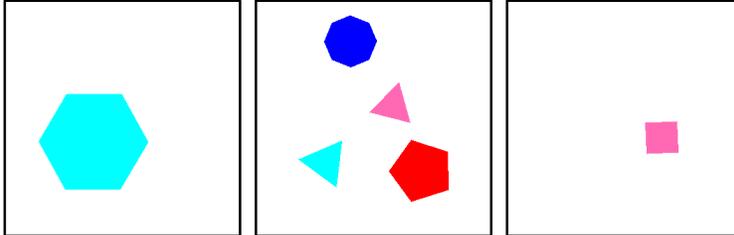| Method	α	MMMU	RealWorldQA	MathVista	Average
SFT + $\mathcal{L}_{\text{focus}}$	1.0	56.11	64.05	58.20	59.45
+ HTT	1.1	56.14	63.94	58.10	59.39
+ HTT	1.2	56.33	64.12	58.20	59.55
+ HTT	1.3	**56.43**	64.05	57.90	59.46
+ HTT + HIR	1.2	56.33	**64.14**	**58.30**	**59.59**

Figure 17: Data sample of SynColorShape.

As shown in table 10, moderate temperature sharpening ($\alpha = 1.2$) yields the best trade-off, improving the average score from 59.45 to 59.55 with consistent gains on MMMU ($+0.22$) and RealWorldQA ($+0.07$) without any degradation on MathVista. A milder scaling ($\alpha = 1.1$) produces negligible change, suggesting that the entropy reduction is too small to meaningfully amplify the counting circuit. A more aggressive scaling ($\alpha = 1.3$) boosts MMMU to the highest single-benchmark score (56.43) but causes a slight decline on MathVista (-0.30), indicating that over-sharpening attention may suppress non-counting heads that contribute to mathematical reasoning. This highlights a precision–generality trade-off: excessively concentrating attention on counting-critical heads risks narrowing the information bandwidth available for other reasoning pathways.

Combining HTT ($\alpha = 1.2$) with HIR achieves the best overall performance (59.59 average), simultaneously improving all three benchmarks. This validates that the two interventions are complementary: HTT sharpens *where* the counting heads attend, while HIR amplifies *how much* their outputs contribute to the residual stream, together strengthening the signal-to-noise ratio of the counting circuit without harming general capabilities.

F Control Experiment: Visual Recognition Training Does Not Improve Visual Reasoning

A central finding of our work is that enhancing counting ability generalizes to broader visual reasoning benchmarks (table 3). A natural question arises: *is this transfer a generic property of any visual fine-tuning, or is it specific to counting as a reasoning-intensive task?* To answer this, we conduct a controlled experiment using a pure visual recognition task—color and shape identification—that requires perception but minimal reasoning.

F.1 Setup

Data. We construct a synthetic color-shape recognition dataset (**SynColorShape**) following the same generation protocol as our counting datasets. Each image is a 336×336 white canvas containing a single filled polygon drawn at a random position, random radius (30–80 px), and random rotation. We use 10 colors (red, blue, green, yellow, orange, purple, pink, cyan, brown, gray) $\times$ 8 shapes (triangle, square, pentagon, hexagon, octagon, circle, star, diamond) = 80 combinations, uniformly sampled to produce 4,960 training samples (62 per combination). Each sample is paired with either a color or shape question (50/50 split), and all ground-truth answers are single tokens.

Task and Prompts. The color-shape task asks the model to identify a single visual attribute of one object. Three prompt templates are randomly assigned per sample for each task type:

Color recognition:

- "What is the color of the {shape} in the image? Answer with the color name only."
- "Identify the color of the {shape} shown. Answer with the color name only."
- "What color is the {shape} in this image? Answer with the color name only."

Shape recognition:

- "What is the shape of the {color} object in the image? Answer with the shape name only."
- "Identify the shape of the {color} figure shown. Answer with the shape name only."
- "What shape is the {color} object in this image? Answer with the shape name only."

All prompts end with a constraint phrase to enforce single-word answers. We apply the same answer-only supervision strategy as our counting SFT: cross-entropy loss is computed exclusively on the 1–2 answer tokens, with all image, prompt, and special tokens masked.

Training. We fine-tune Qwen2.5-VL-7B-Instruct using LoRA ($r=16$, $\alpha=32$) on all linear layers with the same optimizer setting (AdamW, $\mathrm{lr} = 2 \times 10^{-5}$, cosine schedule). We additionally train a second-stage attention regularization model (LoRA $r=8$, $\alpha=16$) that applies KL-divergence supervision on the top-20 important heads identified via activation patching, pushing their attention toward the object mask. This mirrors the two-stage pipeline used in our counting method.

F.2 Results

The color-shape SFT dramatically improves the target recognition task: shape accuracy rises from 78.1% to 99.8% (with diamond going from 0% to 100%), and color accuracy reaches a perfect 100%. However, the critical observation is its effect on general reasoning benchmarks:

Table 11: Change in general reasoning benchmarks (Δ from each method's own baseline) for Qwen2.5-VL-7B. Color-shape recognition training shows no positive transfer, while counting training yields consistent gains.

Training Task	ΔMMMU	ΔMathVista	ΔRealWorldQA	Δ (avg.)
Color-Shape SFT	-0.1	-0.7	-0.2	-0.3
Color-Shape SFT + AttnReg	-0.2	-0.8	$+0.2$	-0.3
Counting (Ours)	**+1.44**	**+1.00**	**+2.18**	**+1.54**

Despite successfully learning the target visual recognition task, color-shape training produces *no positive transfer* to general reasoning benchmarks. MMMU, MathVista, and RealWorldQA all remain within measurement noise or show slight degradation ($\Delta_{\mathrm{avg}} \approx -0.3\%$). By contrast, our counting-based training with the same LoRA fine-tuning paradigm achieves a consistent $+1.54\%$ average improvement across the same benchmarks.

F.3 Analysis

This control experiment provides direct evidence that the generalization observed with counting training is **not** a generic byproduct of visual fine-tuning. While both tasks involve synthetic images, single-token answers, and the same training pipeline, only counting training transfers to broader reasoning. This aligns with our mechanistic analysis in fig. 7: the Jaccard similarity between *Attribution* (color/shape) and reasoning tasks is remarkably low, confirming that recognition relies on perception-only circuits with minimal overlap to the shared reasoning sub-networks.

The key distinction is that counting requires *visual reasoning*: the model must individuate objects, maintain a running tally, and abstract a numerical quantity, operations that engage cross-modal routing and aggregation heads shared with other reasoning tasks. In contrast, color and shape recognition

are fundamentally perceptual; they require identifying a visual attribute of a single object, which is largely resolved by early-layer feature extraction without engaging the deeper reasoning circuitry.

This finding strengthens our central claim: counting serves as a uniquely effective proxy task for enhancing visual reasoning in LVLMs precisely because it activates and refines the shared computational circuits underlying general visual understanding.

G   Case Study

We visualize the attention maps of the baseline and our proposed method to gain insights into the model's counting behavior. For each case, we plot (i) the visual attention distribution at Layer 2 (the layer where we apply *Object-Focused Attention Regularizer* and *Adaptive Head Temperature Tuning*) and (ii) the attention distribution averaged over all layers.
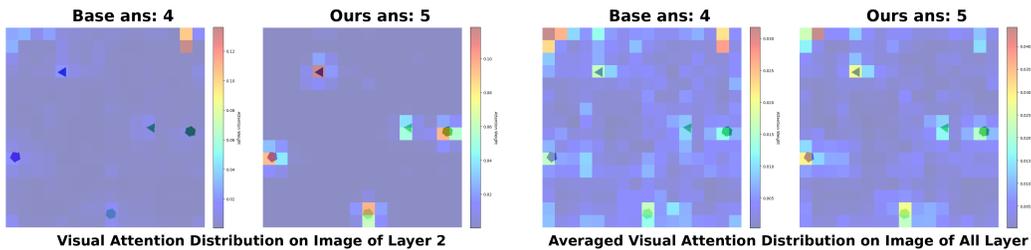

Figure 18: SynPoly case studies at GT=5. **Prompt:** How many colorful polygons are in the image? Answer with only an integer. **Observation:** Base predicts 4 and Ours predicts 5. Base attention is more spread with weak peaks, while ours shows sharper peaks aligned with the dots already at Layer 2 and remains aligned in the all-layer average. **Conclusion:** Our method strengthens early visual grounding for counting, making the attention more instance-aligned even when both models output the correct number.
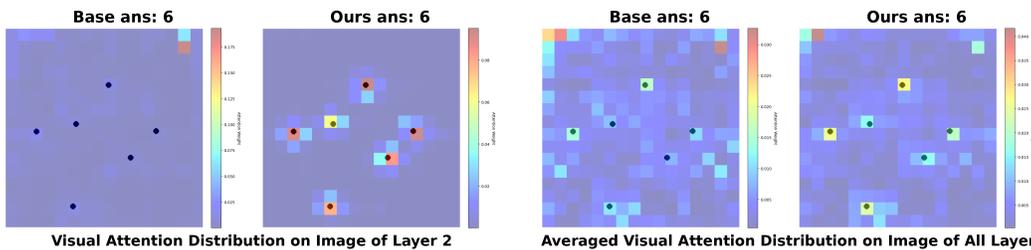

Figure 19: SynDot case studies at GT=6. **Prompt:** How many black dots are in the image? Answer with only an integer. **Observation:** Base predicts 6 and Ours also predicts 6. However, base attention is more spread with weak peaks, while ours shows sharper peaks aligned with the dots already at Layer 2 and remains aligned in the all-layer average. **Conclusion:** Our method strengthens early visual grounding for counting, making the attention more instance-aligned even when both models output the correct number.