Can Vision-Language Models Count? A Synthetic Benchmark and Analysis of Attention-Based Interventions

Saurav Sengupta * Nazanin Moradinasab * Jiebei Liu *
Donald E. Brown
School of Data Science, University of Virginia
{ss4yd, nm4wu, mcu2xn, deb}@virginia.edu

Abstract

Recent research suggests that Vision Language Models (VLMs) often rely on inherent biases learned during training when responding to queries about visual properties of images. These biases are exacerbated when VLMs are asked highly specific questions that require selective visual attention, a demand that mirrors cognitive challenges observed in human enumeration tasks. We build upon this research by developing a synthetic benchmark dataset and evaluation framework to systematically characterize how counting performance varies as image and prompt properties change. Using open-source VLMs, we analyze how performance shifts across controlled perturbations (e.g. number of objects, object color, background color, object texture, background texture, and prompt specificity) and examine corresponding changes in visual attention allocation. We further conduct exploratory attention reweighting experiments in the language model decoder to modulate focus on visual tokens at different layers and assess their effects on counting behavior. Our results reveal that counting accuracy degrades systematically with increasing visual and linguistic complexity echoing human limits and cognitive load effects known from human perception, while targeted attention reweighting yields modest but measurable improvements. Rather than competing on benchmark accuracy, we introduce a controlled diagnostic framework for analyzing VLM enumeration behavior. Through systematic experiments, we expose failure modes rooted in cross-modal binding that natural image benchmarks may not easily isolate, and provide preliminary empirical evidence that targeted attention reweighting in the language decoder can influence how models ground linguistic quantity concepts in visual representations. Code and data available here: https://github.com/ssen7/vlm-count-analysis

*These authors contributed equally.

1. Introduction

Specialized counting methods consistently outperform VLMs on controlled benchmarks, including SAM-based frameworks like PseCo [11], open-world counters like CountGD that use grounding and visual exemplars [2], and diffusion-based density estimators like CrowdDiff [17]. Yet despite this gap, robust counting remains an essential capability for VLMs. Unlike specialized counting tools that are trained for specific object types, VLMs need to handle open-ended questions where counting is just one part of a broader task. From both cognitive and architectural perspectives, accurate estimation of object quantities can serve as a foundational capability for general visual reasoning and downstream applications in robotics, medical imaging and more [19, 21]. We systematically evaluate how current open-source VLMs perform on counting tasks under varying prompt styles and image conditions, and analyze how attention patterns are associated with performance changes

Prior work shows that VLMs struggle with counting [1, 9]. Our work differs from existing approaches by introducing a fine-grained diagnostic framework that analyzes VLM counting behavior across four dimensions: (1) prompt specificity, (2) visual complexity (texture, color, background), (3) object counts ranges (0–50 in increments of 10), and (4) attention distribution over vision tokens. This controlled setup allows us to correlate specific visual and linguistic inputs (1–3) with their mechanistic effects on model internals (4), enabling a more precise understanding of both counting capabilities and failure modes. We evaluate three open-source VLMs that perform strongly on established benchmarks: **Qwen-VL** [23], **Kimi-VL-A3B** [22], and **InternVL3-9B** [6]. These models represent complementary architectural and inference paradigms, including dense transformer architecture and Mixture-of-Experts (MoE) design. For selected models, we evaluate dedicated reasoning and instruction-tuned checkpoints, allowing us to isolate whether explicit chain-of-thought inference confers any advantage in counting tasks. Their open-source nature

is central to our approach, as it enables direct introspection into model internals.

We further use a subset of our evaluation data to investigate how targeted modifications to attention over vision tokens affect counting performance. This is motivated in part by recent work showing that VLMs exhibit a systematic tendency to concentrate high attention weights on specific visual tokens that are largely irrelevant to the input query, called visual attention sinks [12]. Our primary evaluation uses synthetic images by design as synthetic stimuli allow controlled manipulation of factors that are typically entangled in real-world data, such as occlusion, texture, density, and background clutter. To assess whether our findings generalize, we validate selected interventions on a subset of the FSC-147 real-world counting benchmark [18]. Despite the increased visual complexity of natural scenes, we observe similar qualitative trends, suggesting that the identified failure modes are not artifacts of synthetic stimuli.

In summary, in this paper we:

1. Create a synthetic counting dataset that allows us to evaluate VLMs counting performance. The images, while unchallenging for modern computer vision, allow us to isolate specific characteristics of the image and control for other variables to investigate which characteristics of the input (prompt/image) can affect counting performance.
2. Characterize the chosen VLM's counting performance along a) increasing prompt specificity, b) object shapes and colors, c) background colors and textures, and d) distribution of attention over vision tokens.
3. We investigate how attention interventions over visual tokens in the language decoder,both globally and at individual layers, shape model counting behavior, probing whether redirecting attention toward object-bearing regions leads to measurable improvements in accuracy.

2. Related Work

Recent research has revealed significant limitations in Vision Language Models' counting capabilities. Vo et al.[24] demonstrate that state-of-the-art VLMs like Open AI's o3 [16] and Google DeepMind's Gemini 2.5 Pro [8] exhibit strong prior knowledge biases that severely compromise counting accuracy, as models default to memorized patterns rather than analyzing visual features. This bias extends to general visual understanding. Guo et al. [9] systematically show that VLMs struggle to count beyond 20 objects, with performance deteriorating as scene complexity increases. Aghisi at al. [1] identify that reasoning chains can partially improve counting, with middle transformer layers being most critical for accurate enumeration.

Attention mechanisms play a crucial role in VLM visual processing. Kang et al. [12] reveal that VLMs often allocate excessive attention to irrelevant sections of the image and propose a method to redistribute the attention to more relevant areas. An et al. [3] demonstrate that object hallucinations can be mitigated through specialized attention mechanisms that better integrate global and local visual features. Several studies have examined the vision understanding of VLMs by evaluating their Visual Question Answering (VQA) capabilities using synthetic imaging data [10, 14, 15] or by visualizing Attention Guided Class Activation Maps (AG-CAM) on chart-based data [7].

Our work builds upon these foundations by providing a more granular, diagnostic benchmark. We systematically isolate the impact of *visual properties* (e.g., textures, colors, shapes) to pinpoint failure modes related to visual complexity, while also varying input prompts along a specificity gradient, from generic to increasingly precise descriptions, to disentangle the contributions of linguistic and visual factors. To probe the attention mechanisms underlying counting failures, we implement two complementary intervention strategies: image-naive layer-wise modifications that systematically suppress or amplify attention across individual decoder layers, and mask-guided interventions that incorporate spatial object information to more precisely redirect attention toward object-bearing regions. Together, these interventions allow us to evaluate both the sources of counting failure and the conditions under which targeted attention modifications can improve accuracy.

3. Methodology

3.1. Synthetic Evaluation Dataset

We create a collection of synthetic datasets, each consisting of images and corresponding prompts, to facilitate a systematic examination of VLMs. Each dataset is designed to evaluate distinct aspects of the input data—both visual and textual. Our generation process begins with a **baseline dataset** consisting of 512×512 pixel images containing non-overlapping black circular objects on a pure white background. We generate 50 images for this initial configuration. From this baseline, we iterate by varying two primary factors: 1. **Object Numerosity:** We vary the object counts in buckets of 10, ensuring each bucket is equally represented. This allows us to evaluate how VLM performance degrades as the number of objects successively increases. 2. **Visual Properties:** We employ a controlled variable methodology. While preserving the object locations from the base dataset, we systematically vary only one feature at a time from the set: {`object shape`, `object color`, `object texture`, `background color`, `background texture`}. All other variables are held constant. This systematic generation framework allows us to measure and visualize variations in model performance and shifts in attention allocation. By manipulating a single dimension at a time, we can isolate the specific effects of each visual or textual property on the model's

counting capabilities. Refer to Supplementary Section 8.

3.2. Evaluation Criteria

We evaluate VLM counting performance using two primary metrics: Accuracy and Mean Relative Count Error (MRCE). MRCE is defined as:

$$\text{MRCE} = \frac{1}{N}\sum_{i=1}^{N}\frac{|c_{\text{pred}}^{(i)} - c_{\text{true}}^{(i)}|}{c_{\text{true}}^{(i)}} \quad (1)$$

where N is the number of samples, $c_{\text{pred}}^{(i)}$ is the predicted count for sample i, and $c_{\text{true}}^{(i)}$ is the ground truth count for sample i.

We assess these metrics across four primary experimental axes:

Increasing prompt specificity. We analyze model sensitivity to prompt phrasing using a "prompt ladder," where we progressively add descriptive details (e.g., color, texture) to a generic counting prompt. Example prompts are provided in Table 1.

Table 1. Example prompts used when image has different Object Texture.

ID	Example Prompt Text	Logical Role / Cognitive Cue
P1	`Count the number of distinct objects in this image...`	**Baseline:** Generic unconstrained prompt.
P2	`Count the number of {color} color objects in this image...`	**Single (Simple) Attribute:** Simple Cue (Color) - Replace {color} with object colors ("Blue-green" for default).
P3	`Count the number of objects with {pattern} pattern in this image ...`	**Single (Complex) Attribute:** Complex (Texture). Replace {pattern} with "dots", "linear gradient", "checkerboard","vertical stripe", etc.
P4	`Count the number of {pattern} pattern with {color} color objects in this image...`	**Compositional (Target):** Binding (Complex + Simple). Tests binding a simple cue with a complex one.
P5	`Count the number of {pattern} pattern with {color} color {shape} in this image...`	**Compositional (High Load):** Multi-attribute binding under high cognitive load.

Sensitivity to Visual Properties. We evaluate performance by systematically varying a single image characteristic at a time (e.g., `object color`, `object texture`, `background color`, `background texture`) while holding all other factors constant allowing us to isolate the impact of specific visual features on counting performance.

Object Counts Ranges. We analyze how VLM performance varies across different object count intervals (e.g., 0–9, 10–19, . . . , 40–50) to identify the threshold at which a model's counting capabilities begin to deteriorate.

Attention over vision tokens. We quantify how much attention is given to visual tokens in each of the above scenarios to glean insight as to how self-attention is choosing to allocate attention over objects in the image. We generate heatmaps using Layer-wise propagation of Visual Attention (LPV) [5] and GradCAM [20] over all decoder layers and calculate overlap between high attention areas and the objects in the image (See Supplementary Section 6).

3.3. Impact of attention redistribution over counting performance

VLMs exhibit systematic failures in counting tasks, with performance degrading sharply as object count increases (See Section 4.3). We hypothesize that this limitation stems from diffuse attention mechanisms that fail to distinguish and distinctly represent each object instance, leading to feature conflation and under-counting.

We systematically modify how much attention each layer of the language model decoder pays to each token in the vision input, measuring the resulting impact on accuracy and MRCE. We investigate five attention reweighting strategies that manipulate how VLMs allocate attention between visual and textual tokens during generation. Let $\mathbf{A} \in \mathbb{R}^{H\times Q\times K}$ denote the attention weight matrix for a given layer, where H is the number of attention heads, Q is the query sequence length, and K is the key sequence length. We denote the visual token positions as $V = \{v_{\text{start}}, \ldots, v_{\text{end}}\}$ where visual tokens typically occupy the prefix of the sequence.

Amplify. This strategy increases attention weights to visual tokens by a multiplicative factor $\alpha > 1$:

$$\tilde{A}_{h,i,j} = \begin{cases} \alpha \cdot A_{h,i,j} & \text{if } j \in V \\ A_{h,i,j} & \text{otherwise} \end{cases} \quad (2)$$

followed by renormalization: $\tilde{A}_{h,i,:} \leftarrow \tilde{A}_{h,i,:}/\sum_k \tilde{A}_{h,i,k}$. This strategy strengthens visual grounding by encouraging stronger connections to image features. We use $\alpha = 2.0$.

Suppress. Conversely, this strategy reduces attention to visual tokens, forcing greater reliance on linguistic context :

$$\tilde{A}_{h,i,j} = \begin{cases} \beta \cdot A_{h,i,j} & \text{if } j \in V \\ A_{h,i,j} & \text{otherwise} \end{cases} \quad (3)$$

where $0 < \beta < 1$, followed by renormalization. We set $\beta = 0.5$.

Focus. This strategy creates an extreme form of visual attention by largely eliminating attention to non-visual tokens:

$$\tilde{A}_{h,i,j} = \begin{cases} A_{h,i,j} & \text{if } j \in V \\ \epsilon & \text{otherwise} \end{cases} \tag{4}$$

where $\epsilon = 10^{-10}$ is a small constant to maintain numerical stability. After renormalization, attention is effectively concentrated solely on visual tokens, forcing direct visual conditioning at each generation step.

Balance. This strategy enforces a target distribution between visual and textual attention. Given the desired visual attention ratio r_v^{target} (we use $r_v^{\text{target}} = 0.4$) and the current ratio: $r_v^{\text{current}} = \frac{\sum_{j \in V} A_{h,i,j}}{\sum_k A_{h,i,k}}$, we apply a corrective scaling:

$$\tilde{A}_{h,i,j} = \begin{cases} \gamma \cdot A_{h,i,j} & \text{if } j \in V \\ A_{h,i,j} & \text{otherwise} \end{cases} \tag{5}$$

where $\gamma = r_v^{\text{target}} / r_v^{\text{current}}$, followed by renormalization. This preserves visual–textual attention balance and prevents over- or under-reliance on visual information.

Visual Mask Amplify. Let $\mathbf{M} \in \{0,1\}^{H_{\text{img}} \times W_{\text{img}}}$ denote a binary object mask obtained from an off-the-shelf segmentation model (e.g., SAM [13]). For a vision transformer with patch size p, we partition the image into a grid of $N_h \times N_w$ patches where $N_h = H_{\text{img}}/p$ and $N_w = W_{\text{img}}/p$.

For each visual token v_i corresponding to patch coordinates (r, c), we compute the **object overlap ratio**:

$$\rho_i = \frac{1}{p^2} \sum_{x=rp}^{(r+1)p-1} \sum_{y=cp}^{(c+1)p-1} \mathbf{M}(x, y) \tag{6}$$

which measures the fraction of the patch covered by object regions. We define the set of object-relevant tokens as:

$$V_{\text{obj}} = \{v_i \in V : \rho_i > \tau\} \tag{7}$$

where τ is an overlap threshold (we use $\tau = 0.1$ to capture patches with at least 10% object coverage). The visual mask amplify strategy then applies selective amplification:

$$\tilde{A}_{h,i,j} = \begin{cases} \alpha_{\text{obj}} \cdot A_{h,i,j} & \text{if } j \in V_{\text{obj}} \\ \alpha_{\text{bg}} \cdot A_{h,i,j} & \text{if } j \in V \setminus V_{\text{obj}} \\ A_{h,i,j} & \text{otherwise} \end{cases} \tag{8}$$

followed by renormalization. We test $\alpha_{\text{obj}} = 2.0$ to strongly emphasize objects and $\alpha_{\text{bg}} = 0.5$ to suppress background, creating a high-contrast attention distribution that prioritizes semantically meaningful content. We also test an ablation of this strategy without background suppression. In mask-based variants, object masks are obtained using an off-the-shelf segmentation model (SAM3 [4]). The masks serve as spatial priors to distinguish object regions from background regions when reweighting visual tokens. Together, these strategies enable controlled manipulation of the visual–textual attention trade-off, allowing systematic analysis of how attention allocation influences counting behavior.

4. Results

4.1. Effects of Prompt Specificity

Results for the effects of increasing linguistic details in the text prompt are shown in Table 2.

Table 2. Effect of prompt specificity on counting accuracy and MRCE. Prompt 1 (gray cells) serves as the baseline. For Accuracy: darker red indicates greater improvements, darker blue indicates larger drops. For MRCE: darker red indicates greater error reduction (better), darker blue indicates increased relative error (worse). "Bg" = *Background*, "Obj" = *Object*.

Cat.	Feat.	Prompts	Qwen32b		Qwen7b		InternVL		Kimi	
			Acc	MRCE	Acc	MRCE	Acc	MRCE	Acc	MRCE
Bg	color	P1	0.22	0.133	0.223	0.242	0.167	0.133	0.247	0.162
		P2	+0.047	-0.032	+0.013	-0.087	-0.010	+0.026	+0.020	-0.079
		P3	+0.033	-0.038	-0.003	-0.071	+0.013	+0.029	+0.033	-0.086
Bg	texture	P1	0.182	0.282	0.09	0.638	0.213	0.227	0.169	0.452
		P2	-0.003	-0.091	+0.078	-0.433	+0.009	-0.060	+0.095	-0.355
		P3	-0.018	-0.075	+0.078	-0.415	-0.020	+0.048	+0.076	-0.346
		P4	+0.042	-0.149	+0.062	-0.409	-0.013	+0.029	+0.069	-0.346
		P5	+0.057	-0.152	+0.057	-0.400	-0.011	+0.020	+0.067	-0.350
Obj	color	P1	0.24	0.104	0.163	0.274	0.22	0.100	0.246	0.129
		P2	-0.023	+0.021	+0.049	-0.115	-0.011	+0.030	-0.003	-0.013
		P3	-0.043	+0.002	+0.051	-0.118	-0.031	+0.034	0.000	-0.007
Obj	shape	P1	0.196	0.135	0.18	0.347	0.224	0.128	0.228	0.139
		P2	+0.004	-0.011	+0.020	-0.213	+0.020	+0.003	+0.016	+0.090
		P3	+0.024	+0.002	+0.004	-0.203	+0.024	+0.019	+0.028	-0.018
Obj	texture	P1	0.24	0.145	0.172	0.543	0.254	0.113	0.272	0.076
		P2	-0.084	+0.132	-0.006	-0.319	0.000	+0.048	-0.010	+0.067
		P3	-0.088	+0.107	-0.078	-0.266	-0.136	+0.309	-0.018	+0.027
		P4	-0.096	+0.144	-0.060	-0.238	-0.080	+0.201	-0.040	+0.090
		P5	-0.108	+0.172	-0.076	-0.219	-0.076	+0.194	-0.042	+0.098

Across all models, prompt specificity has a strongly asymmetric effect depending on the feature type. For background features, specificity consistently improves performance—background texture yields the largest gains, with Qwen7b and Kimi reducing MRCE by 0.433 and 0.355, respectively at P2, persisting through P5. In contrast, object texture is the only category where specificity monotonically degrades accuracy across all models (Qwen32b $\Delta Acc = -0.108$ at P5), even as MRCE improves for some models, suggesting errors become more systematic rather than random. Object color and shape show mixed, model-dependent responses. Qwen7b benefits substantially from shape specificity ($\Delta MRCE = -0.213$ at P2), while Kimi degrades under the same condition, and InternVL remains largely neutral, indicating greater robustness to prompt variation. Notably, model scale does not confer robustness: Qwen32b degrades substantially on object texture despite

being the largest model. Together, these results suggest that specificity aids counting when it simplifies visual segmentation (background cues), but becomes detrimental when it introduces competing object-level processing demands.

P1 (the simplest, most general prompt) succeeds because its very generality allows the model to deploy its most robust internal detector, bypassing the "cognitive sink" that *any* specific semantic cue (whether `texture`, `color`, or `shape`) creates in this task. We have direct visual evidence for this "sink" in Figure 1 when `shape` is added in P5 (`color+texture+shape`), attention overlays confirm the model's attention to the object's shape is absent, suppressed by the cognitive load of processing `texture` and `color`.

4.2. Effects of Visual Complexity

Table 3 presents the Mean Relative Count Error (MRCE) for Prompt 2—the single-attribute, color-focused prompt—across all background and object variations. The results show a clear trend: model performance degrades substantially as visual complexity increases. Models perform well under simple conditions, such as solid-colored backgrounds and plain, single-color objects, where MRCE is lowest. However, the error increases significantly when high-frequency textures (e.g., checkerboard, diagonal/vertical stripes, concentric rings) or visually heterogeneous objects (multicolor or complex patterns) are introduced. This pattern is consistent across most models, indicating that complex textures and patterns interfere with the models' ability to reliably segment and enumerate objects. Results for Prompts 1 and 3–5 are provided in the Supplementary.

4.3. Effects of Count Magnitude

To analyze the effect of count magnitude on model accuracy, we divided all images into five discrete count bins as shown in Table 4. The results are collapsed across all prompt formulations, where darker blue shades indicate smaller errors. We observe a consistent trend across all models: counting performance degrades monotonically with increasing object count. In the low-count regime (<10), most models exhibit minimal error (<0.1), indicating strong reliability when few instances are present. However, as the number of objects increases, error grows non-linearly—particularly beyond 30. When separating by visual feature type,simple Color category shows the smallest deviation. In contrast, texture categories consistently exhibit the largest variance, implying that complex surface patterns interfere with spatial grouping mechanisms. Among the models, Kimi-VL-A3B-Instruct and Qwen2.5-32B-Instruct maintain the lowest relative errors overall, whereas Qwen2.5-7B-Instruct and InternVL3-9B-Instruct exhibit greater sensitivity under large-count conditions. These results demonstrate a level-dependent counting robustness: performance remains stable for small sets but declines as visual density and textural complexity increase, emphasizing the need for count-adaptive attention strategies in future architectures.

Table 3. Mean Relative Count Error (lower is better) for Prompt 2 across all patterns.

Cat.	Feat.	Pattern	Qwen7b	Qwen32b	Intern	Kimi
Bg	Color	blue	0.129	0.109	0.134	0.075
Bg	Color	black	0.152	0.088	0.135	0.083
Bg	Color	green	0.159	0.109	0.134	0.062
Bg	Color	gray	0.156	0.106	0.160	0.072
Bg	Color	red	0.150	0.100	0.165	0.087
Bg	Color	yellow	0.176	0.095	0.221	0.113
Bg	Texture	lin. grad.	0.138	0.105	0.077	0.078
Bg	Texture	noise	0.120	0.096	0.145	0.071
Bg	Texture	rad. grad.	0.146	0.164	0.121	0.062
Bg	Texture	cr. hatch	0.172	0.137	0.133	0.053
Bg	Texture	ver. str.	0.210	0.130	0.179	0.062
Bg	Texture	checkerboa	0.232	0.135	0.188	0.113
Bg	Texture	hor. str.	0.245	0.182	0.115	0.154
Bg	Texture	con. rgs	0.239	0.193	0.253	0.068
Bg	Texture	diag. str	0.249	0.308	0.147	0.071
Bg	Texture	bubbles	0.264	0.176	0.300	0.204
Bg	Texture	dots	0.264	0.560	0.174	0.116
Obj	Color	green	0.097	0.086	0.085	0.066
Obj	Color	red	0.126	0.115	0.088	0.063
Obj	Color	blue	0.108	0.105	0.120	0.062
Obj	Color	white	0.099	0.101	0.102	0.109
Obj	Color	yellow	0.141	0.095	0.100	0.114
Obj	Color	light gray	0.170	0.113	0.187	0.079
Obj	Color	multicolor	0.362	0.259	0.223	0.308
Obj	Shape	star	0.130	0.117	0.077	0.076
Obj	Shape	polygon	0.110	0.105	0.140	0.089
Obj	Shape	rectangle	0.089	0.128	0.198	0.079
Obj	Shape	circle	0.168	0.131	0.173	0.073
Obj	Shape	triangle	0.161	0.137	0.066	0.360
Obj	Texture	lin. grad.	0.147	0.105	0.089	0.064
Obj	Texture	con. cir.	0.134	0.104	0.116	0.074
Obj	Texture	rad. grad.	0.137	0.164	0.095	0.068
Obj	Texture	ver. str.	0.124	0.130	0.125	0.095
Obj	Texture	checkerboa	0.191	0.206	0.113	0.072
Obj	Texture	hor. str.	0.167	0.182	0.175	0.079
Obj	Texture	zigzag	0.173	0.207	0.199	0.073
Obj	Texture	diag. str.	0.270	0.308	0.150	0.060
Obj	Texture	dots	0.232	0.560	0.177	0.139
Obj	Texture	cr. hatch	0.652	0.797	0.362	0.700

4.4. Reasoning vs Instruction Tuned Models

We additionally evaluate reasoning-enabled variants of two open-source reasoning VLMs, Qwen3b-Thinking and Kimi-VL-A3B-Thinking models, and compare them to instruction-tuned counterparts. The results show in Table 5. As shown, reasoning does not consistently improve counting performance. While reasoning models can reduce error for low object counts in some cases, they frequently produce unparsable or verbose outputs at higher counts, lead-

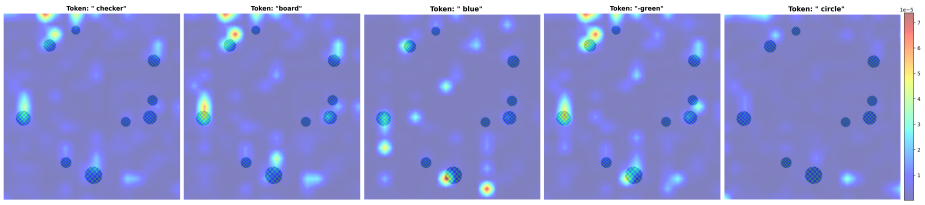

Figure 1. Token-level attention heatmaps for the compositional prompt 5 of object texture task for Kimi. The high load from texture and color suppresses attention to shape

Table 4. Mean Relative Count Error (lower is better). Darker blue cells indicate smaller errors.

Cat.	Feat.	Counts	Qwen7b	Qwen32b	Intern	Kimi
Bg	Color	<10	0.039	0.050	0.099	0.020
Bg	Color	10–19	0.068	0.067	0.082	0.085
Bg	Color	20–29	0.154	0.138	0.131	0.100
Bg	Color	30–39	0.226	0.124	0.189	0.125
Bg	Color	40–50	0.451	0.170	0.253	0.201
Bg	Texture	<10	0.137	0.155	0.134	0.077
Bg	Texture	10–19	0.150	0.167	0.117	0.143
Bg	Texture	20–29	0.315	0.202	0.176	0.182
Bg	Texture	30–39	0.411	0.188	0.226	0.232
Bg	Texture	40–50	0.533	0.255	0.323	0.222
Obj	Color	<10	0.080	0.076	0.081	0.044
Obj	Color	10–19	0.097	0.103	0.078	0.090
Obj	Color	20–29	0.182	0.116	0.109	0.107
Obj	Color	30–39	0.223	0.144	0.150	0.172
Obj	Color	40–50	0.395	0.142	0.186	0.190
Obj	Shape	<10	0.087	0.079	0.044	0.032
Obj	Shape	10–19	0.094	0.112	0.079	0.097
Obj	Shape	20–29	0.156	0.185	0.168	0.135
Obj	Shape	30–39	0.282	0.151	0.164	0.174
Obj	Shape	40–50	0.395	0.132	0.223	0.220
Obj	Texture	<10	0.152	0.199	0.085	0.073
Obj	Texture	10–19	0.210	0.237	0.183	0.130
Obj	Texture	20–29	0.368	0.273	0.294	0.139
Obj	Texture	30–39	0.422	0.247	0.341	0.156
Obj	Texture	40–50	0.515	0.323	0.412	0.162

ing to reduced effective coverage. For example, Qwen3-Thinking often fails to produce valid numerical answers in dense scenes, and Kimi-Instruct generally outperforms Kimi-Thinking overall. These results suggest that enumeration errors stem primarily from visual grounding limitations rather than insufficient linguistic reasoning.

4.5. Attention over Vision Tokens

To investigate how linguistic variations influence the model's spatial focus, we analyze the distribution of attention over vision tokens using five prompts for texture variations and three prompts for color variations applied to both the background and the object where each successive prompt adds more linguistic detail following our "prompt ladder" paradigm. In Figure 2, we present the

Table 5. Mean Relative Error (MRE) computed over parsable outputs. For Qwen-Thinking, images with > 10 objects resulted in interminable reasoning loops, while Kimi-Thinking performed noticably worse than Kimi-Instruct.

Qwen

Cat.	Feat.	Counts	Qwen3-T	Pars.	N-par.	Qwen7b
Bg	Color	<10	0.017	173	7	0.039
Bg	Color	10–50	0.114	85	635	–
Bg	Txtr	<10	0.058	495	55	0.137
Bg	Txtr	10–50	0.182	275	1925	–
Obj	Color	<10	0.028	204	6	0.080
Obj	Color	10–50	0.086	108	732	–
Obj	Shape	<10	0.019	147	0	0.087
Obj	Shape	10–50	0.113	86	517	–
Obj	Txtr	<10	0.051	479	21	0.152
Obj	Txtr	10–50	0.239	269	1731	–

Kimi

Cat.	Feat.	Counts	Kimi-T	Pars.	N-par.	Kimi
Bg	Color	<10	0.135	150	0	0.020
Bg	Color	10–50	0.424	534	66	0.128
Bg	Txtr	<10	0.336	587	3	0.077
Bg	Txtr	10–50	0.525	2358	42	0.195
Obj	Color	<10	0.130	210	0	0.044
Obj	Color	10–50	0.423	823	17	0.145
Obj	Shape	<10	0.114	150	0	0.032
Obj	Shape	10–50	0.411	568	32	0.164
Obj	Txtr	<10	0.193	475	5	0.073
Obj	Txtr	10–50	0.474	1821	99	0.158

IoU@50 overlap between the attention heatmap and image for both object and background. Both Qwen2.5-7B-Instruct and Kimi-VL-A3B-Instruct exhibit a consistent trend. For Background Texture images, prompts with moderate object-related detail (P2–P3) increase LPV overlap with object regions *(IoU@50, object)*, while decreasing LPV overlap with background regions *(IoU@50, background)*, relative to the baseline prompt (P1). These prompts (P2–P3) introduce explicit object descriptors such as color and shape, which help the models better localize the target regions—consistent with the performance gains under moderate specificity in Section 4.1. In contrast, prompts including additional background-related information (P4–P5) generally reduce object Grad-CAM attention *(IoU@50, object)* compared to the base prompt, particularly in the Background Texture condition for both models. For Object Texture, LPV overlap with the object consistently shows enhanced localization across P2–P5, reflecting stronger grounding when object-specific cues are present. However, beyond P3, the inclusion of fine-grained texture descriptors does not further reduce the relative count error; instead, error tends to rise again—aligning with the "cognitive sink" effect from Section 4.1. These results suggest that

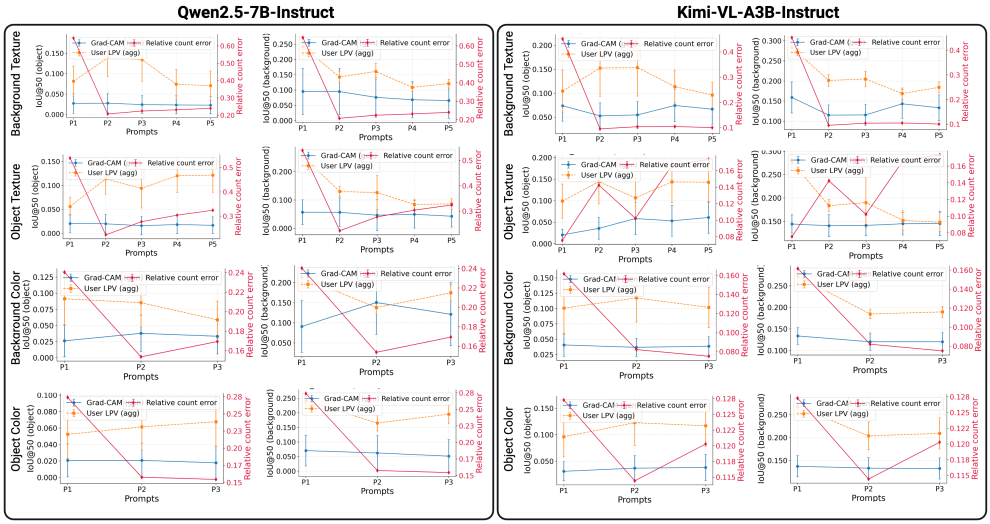

Figure 2. Visualization of the model's attention over object and background for all image types and increasing linguistic specificity in the prompt.

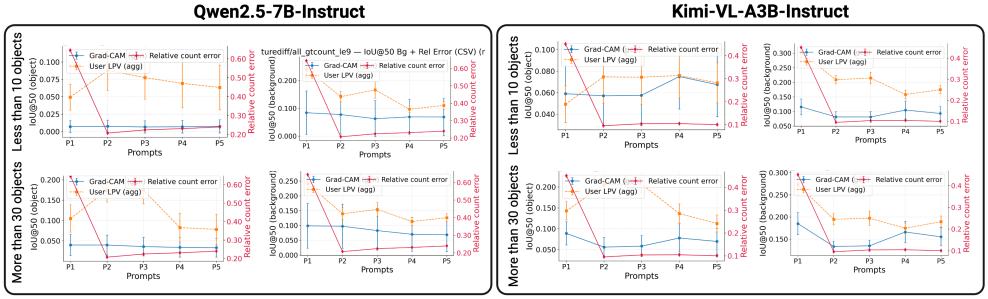

Figure 3. Visualization of the model's attention across different background-texture patterns for images containing fewer than 10 objects or more than 30 objects.

linguistic specificity aids in reducing counting error up to a moderate level, but excessive descriptive detail can counteract this benefit, highlighting a non-linear relationship between prompt richness and counting error reduction. Figure 3 shows that the number of objects in the image does not substantially influence the overall trend of visual attention. Both models exhibit consistent attention patterns for images containing fewer than 10 objects and for those with more than 30. In both settings, Grad-CAM and LPV attention distributions follow similar trajectories across prompts, though relative count error remains significantly lower in the <10 object condition compared to the >30 objects setting. This stability indicates that the models' spatial grounding behavior, as guided by linguistic cues, is largely invariant to object density. The model trends are presented in the supplementary material.

4.6. Impact of Attention Redistribution

We present the results of our layer wise attention reweighting experiments in Table 6. These interventions were tested on the `background texture` dataset and, to assess the transferability of our findings to natural images, on the FSC-147 counting benchmark, with results averaged over all images. We modify self-attention values either uniformly across all layers or in layer groups: `early` (layers 0-7 for Qwen, 0-8 for Kimi), `middle` (layers 8-24 for Qwen, 9-17 for Kimi), and `late` (layers 24-31 for Qwen, 18-26 for Kimi). Refer to Supplementary Sections 7.1 and 7.2 for detailed explanations of the experiments.

Across both synthetic and FSC-147 evaluations, mask-guided late-layer amplification emerges as the most consistent intervention: `late amplify visual mask bg suppress` is the only strategy to improve Qwen synthetic MRCE while also boosting FSC-147 accuracy, and `early amplify visual mask` similarly transfers to real-world data by improving both Qwen FSC-147 MRCE and accuracy. In contrast, image-naive strategies are largely model-specific — `uniform suppress` and `alternating amp sup` substantially reduce Kimi syn-

Table 6. Attention reweighting strategy performance on synthetic (background texture) and real-world (FSC-147) datasets for Qwen3-VL-8B (dense) and KimiVL-A3B (MoE). Results are mean MRCE and Accuracy across count buckets. **Bold** values indicate improvement over baseline. Arrows indicate direction of improvement (↓ lower is better, ↑ higher is better). "-" denotes undefined due to model collapse.

Group	Strategy	Qwen (Synthetic)		Kimi (Synthetic)		Qwen (FSC-147)		Kimi (FSC-147)	
		MRCE↓	Acc↑	MRCE↓	Acc↑	MRCE↓	Acc↑	MRCE↓	Acc↑
Baseline	baseline	0.080	0.330	0.140	0.320	0.168	0.220	0.301	0.160
Uniform	uniform_suppress	0.160	0.100	**0.090**	0.200	0.226	0.120	0.326	0.140
	uniform_amplify	0.250	0.070	0.350	0.160	0.192	**0.234**	0.347	**0.200**
	uniform_balance	0.080	**0.340**	0.320	0.210	**0.158**	0.163	0.694	**0.220**
	uniform_focus	—	0.000	0.860	0.020	—	0.000	0.374	**0.200**
Alternating	alternating_amp_sup	0.090	0.290	**0.090**	0.240	0.176	0.220	0.332	0.160
Progressive	progressive_visual_grow	0.080	**0.370**	0.160	0.260	**0.163**	**0.240**	0.694	**0.260**
	progressive_visual_fade	0.080	**0.370**	0.320	0.200	0.218	0.163	0.738	**0.200**
Early (image-naive)	early_visual_only	0.610	0.040	**0.100**	0.210	—	0.000	0.429	**0.180**
	extreme_visual_early	—	0.000	0.690	0.040	—	0.000	0.771	0.160
Early (mask-guided)	early_amplify_visual_mask	0.080	**0.350**	0.150	0.290	**0.143**	**0.224**	0.344	**0.184**
	early_amplify_visual_mask_bg_suppress	0.080	**0.360**	0.150	0.290	**0.160**	0.200	0.344	**0.184**
Middle (image-naive)	middle_visual_boost	0.230	0.080	0.420	0.170	0.185	0.208	0.307	**0.280**
Middle (mask-guided)	middle_amplify_visual_mask	0.270	0.110	0.220	0.240	0.195	**0.224**	**0.281**	**0.163**
	middle_amplify_visual_mask_bg_suppress	0.300	0.100	0.220	0.240	0.225	0.120	**0.281**	**0.163**
Late (image-naive)	late_visual_retention	0.090	0.330	0.280	0.200	**0.145**	**0.224**	0.725	**0.240**
	extreme_text_late	0.080	**0.340**	0.210	0.240	**0.164**	0.160	**0.297**	**0.220**
Late (mask-guided)	late_amplify_visual_mask	0.080	0.330	0.160	0.300	**0.163**	0.204	**0.296**	0.143
	late_amplify_visual_mask_bg_suppress	**0.070**	**0.350**	0.160	0.300	0.172	**0.260**	**0.296**	0.143

thetic MRCE (0.09 vs. 0.14 baseline) but degrade Qwen, suggesting Kimi's MoE architecture is more tolerant of global suppression than Qwen's dense design. Middle-layer interventions reveal an MRCE-accuracy tradeoff, improving Kimi FSC-147 MRCE while sharply degrading Qwen accuracy. Progressive and uniform strategies produce inconsistent cross-model effects, with `uniform balance` improving Qwen FSC-147 MRCE while catastrophically inflating Kimi's. Most strikingly, extreme attention redistribution strategies cause complete model collapse on Qwen while leaving Kimi comparatively unaffected. Taken together, these results indicate that spatial object mask guidance is the critical factor for robust, generalizable attention intervention — image-naive strategies may shift error distributions without reliably improving counting.

5. Conclusions and Future Work

We presented a systematic diagnostic study of VLM counting capabilities, introducing a controlled synthetic benchmark to isolate the effects of prompt specificity, visual properties, object count range, and attention distribution. Our prompt ladder experiments reveal that counting failures are not primarily driven by compositional reasoning, but by the cognitive load imposed by the prompt's primary segmentation cue. Attention analysis corroborates this, showing that models can perceptually attend to objects yet fail to enumerate them when competing semantic cues overwhelm the counting process. Our intervention experiments further show that mask-guided attention amplification in early and late decoder layers is the most robust strategy for improving counting accuracy, generalizing from synthetic stimuli to real-world FSC-147 images, while image-naive strategies remain model-specific and unreliable. Our findings motivate developing specialized attention mechanisms tailored to decoder architectures, and addressing the dissociation between counting accuracy and MRCE through hybrid training objectives. Interpretable probes could further identify network components responsible for enumeration versus classification, offering a path toward architectures whose functional organization more closely reflects the modular structure of human visual cognition, providing crucial insights for architectural improvements.

Limitations: While our controlled synthetic benchmarks enable precise failure mode analysis, extending this framework to real-world scenarios with occlusion, varying scales, and complex spatial arrangements remains essential. Our attention interventions assume standard transformer architectures; if frontier models employ attention variants (e.g., mixture of depths or sparse attention patterns), our strategies may not directly transfer. Finally, our analysis is limited to open-source VLMs; proprietary and larger-scale models may exhibit qualitatively different failure modes and responses to attention intervention.

References

[1] Simone Alghisi, Gabriel Roccabruna, Massimo Rizzoli, Seyed Mahed Mousavi, and Giuseppe Riccardi. [de— re] constructing vlms' reasoning in counting. *arXiv preprint arXiv:2510.19555*, 2025. 1, 2

[2] Niki Amini-Naieni, Tengda Han, and Andrew Zisserman. Countgd: Multi-modal open-world counting. *Advances in Neural Information Processing Systems*, 37:48810–48837, 2024. 1

[3] Wenbin An, Feng Tian, Sicong Leng, Jiahao Nie, Haonan Lin, QianYing Wang, Ping Chen, Xiaoqin Zhang, and Shijian Lu. Mitigating object hallucinations in large vision-language models with assembly of global and local attention. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 29915–29926, 2025. 2

[4] Nicolas Carion, Laura Gustafson, Yuan-Ting Hu, Shoubhik Debnath, Ronghang Hu, Didac Suris, Chaitanya Ryali, Kalyan Vasudev Alwala, Haitham Khedr, Andrew Huang, et al. Sam 3: Segment anything with concepts. *arXiv preprint arXiv:2511.16719*, 2025. 4

[5] Hila Chefer, Shir Gur, and Lior Wolf. Generic attention-model explainability for interpreting bi-modal and encoder-decoder transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 397–406, 2021. 3, 1

[6] Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, et al. Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 24185–24198, 2024. 1

[7] Lianghan Dong and Anamaria Crisan. Probing the visualization literacy of vision language models: the good, the bad, and the ugly. *arXiv preprint arXiv:2504.05445*, 2025. 2

[8] Google. Google deepmind: Gemini 2.5 pro, 2025. https://deepmind.google/models/gemini/pro/. 2

[9] Xuyang Guo, Zekai Huang, Zhenmei Shi, Zhao Song, and Jiahao 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. 1, 2

[10] Yifan Hou, Buse Giledereli, Yilei Tu, and Mrinmaya Sachan. Do vision-language models really understand visual language? *arXiv preprint arXiv:2410.00193*, 2024. 2

[11] Zhizhong Huang, Mingliang Dai, Yi Zhang, Junping Zhang, and Hongming Shan. Point segment and count: A generalized framework for object counting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17067–17076, 2024. 1

[12] Seil Kang, Jinyeong Kim, Junhyeok Kim, and Seong Jae Hwang. See what you are told: Visual attention sink in large multimodal models. *arXiv preprint arXiv:2503.03321*, 2025. 2

[13] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4015–4026, 2023. 4

[14] Kang-il Lee, Minbeom Kim, Seunghyun Yoon, Minsung Kim, Dongryeol Lee, Hyukhun Koh, and Kyomin Jung. Vlind-bench: Measuring language priors in large vision-language models. *arXiv preprint arXiv:2406.08702*, 2024. 2

[15] Tony Lee, Haoqin Tu, Chi H Wong, Wenhao Zheng, Yiyang Zhou, Yifan Mai, Josselin S Roberts, Michihiro Yasunaga, Huaxiu Yao, Cihang Xie, et al. Vhelm: A holistic evaluation of vision language models. *Advances in Neural Information Processing Systems*, 37:140632–140666, 2024. 2

[16] OpenAI. Open ai: Introducing openai o3 and o4-mini, 2025. https://openai.com/index/introducing-o3-and-o4-mini/. 2

[17] Yasiru Ranasinghe, Nithin Gopalakrishnan Nair, Wele Gedara Chaminda Bandara, and Vishal M Patel. Crowddiff: Multi-hypothesis crowd density estimation using diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12809–12819, 2024. 1

[18] Viresh Ranjan, Udbhav Sharma, Thu Nguyen, and Minh Hoai. Learning to count everything. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3394–3403, 2021. 2

[19] Ji Seung Ryu, Hyunyoung Kang, Yuseong Chu, and Sejung Yang. Vision-language foundation models for medical imaging: a review of current practices and innovations. *Biomedical Engineering Letters*, pages 1–22, 2025. 1

[20] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: visual explanations from deep networks via gradient-based localization. *International journal of computer vision*, 128(2):336–359, 2020. 3

[21] Rui Shao, Wei Li, Lingsen Zhang, Renshan Zhang, Zhiyang Liu, Ran Chen, and Liqiang Nie. Large vlm-based vision-language-action models for robotic manipulation: A survey. *arXiv preprint arXiv:2508.13073*, 2025. 1

[22] Kimi Team, Angang Du, Bohong Yin, Bowei Xing, Bowen Qu, Bowen Wang, Cheng Chen, Chenlin Zhang, Chenzhuang Du, Chu Wei, et al. Kimi-vl technical report. *arXiv preprint arXiv:2504.07491*, 2025. 1

[23] Qwen Team. Qwen2.5-vl, 2025. 1

[24] An Vo, Khai-Nguyen Nguyen, Mohammad Reza Taesiri, Vy Tuong Dang, Anh Totti Nguyen, and Daeyoung Kim. Vision language models are biased, 2025. 2