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

Supplementary Material

6. Layer-wise Propagation of Visual attention

Gradient-weighted attention. Inspired by Chefer et al. [5] propose, we propose a lightweight gradient-weighted relevance propagation(LPV) for autoregressive VLMs that turns layer-wise attentions into token-level relevance maps by using gradient weighting and cross-layer diffusion. For each Transformer layer ℓ, let $A^{(\ell)} \in \mathbb{R}^{H\times S\times S}$ be the multi-head attention (post-softmax) and let

$$G^{(\ell)} \;=\; \frac{\partial \mathcal{L}}{\partial A^{(\ell)}}$$

be its gradient obtained from a token-level cross-entropy loss on selected output positions. Noisy negative signals in the gradient are suppressed using a ReLU:

$$\tilde{G}^{(\ell)} \;=\; \mathrm{ReLU}\Big(G^{(\ell)}\Big).$$

We then form a gradient-weighted attention map by element-wise interaction and heads averaging:

$$H^{(\ell)} \;=\; \frac{1}{H}\sum_{h=1}^{H}\big(A_h^{(\ell)} \odot \tilde{G}_h^{(\ell)}\big).$$

To preserve self-information paths, we add the identity and apply row-normalize to obtain a row-stochastic per-layer relevance transition:

$$M^{(\ell)}(i,j) \;=\; \frac{H^{(\ell)}(i,j)+\delta_{ij}}{\sum_k\big(H^{(\ell)}(i,k)+\delta_{ik}\big)}.$$

Cross-layer joint relevance. We compose the last K layers to aggregate both deep semantics and shallow localization:

$$C \;=\; \prod_{\ell=L-K+1}^{L} M^{(\ell)}.$$

For any given output token at index t, the corresponding row $C[t,:]$ gives a fine-grained relevance distribution over all input tokens (textual and visual) that contribute to the prediction of that specific token.

Differences from Chefer et al. [5]. Our formulation follows the gradient-weighted attention idea and cross-layer composition of Chefer et al., but is tailored to autoregressive VLMs: (1) **Token-specific supervision.** We supervise a token-level cross-entropy on a *selected* set of decoding steps and collect $\{\nabla A^{(\ell)}\}$ in a single backward pass, enabling token/time-step-specific attribution and natural multi-token aggregation. (2) **Per-layer row-stochastic transitions.** We explicitly enforce row-stochastic $M^{(\ell)}$ by adding the identity and applying row normalization at each layer, which stabilizes deep products and improves robustness. (3) **Controllable depth.** We optionally compose only the last K layers to trade interpretability depth for stability and speed without modifying model internals.

Why not plain attention rollout? Our method provides more faithful and selective relevance maps compared to simpler methods like Attention Rollout. Plain Attention rollout composes raw attentions and is class-/token-agnostic. It highlights tokens the model "looked at" (high A), but not necessarily tokens that *influenced* the specific prediction. Our gradient-weighted diffusion emphasizes attention edges that are both *used* (large A) and *useful for the current prediction* (large positive ∇A), yielding more selective and faithful relevance maps that adapt naturally to multi-token, multi-modal settings.

7. Attention Reweighting in Qwen Models

7.1. Attention Reweighting in Grouped Query Attention Architecture

Qwen 2.5 and Qwen 3 models employ Grouped Query Attention (GQA) , which differs from standard Multi-Head Attention by using fewer key-value heads than query heads to reduce computational cost. Specifically, with $H = 32$ attention heads and $K = 8$ key-value heads, each KV head is shared across $G = H/K = 4$ query heads. This architecture necessitates special handling during attention reweighting. After projecting inputs through W_q, W_k, and W_v, the resulting tensors have shapes $\mathbf{Q} \in \mathbb{R}^{B\times L\times Hd}$ and $\mathbf{K}, \mathbf{V} \in \mathbb{R}^{B\times L\times Kd}$, where B is batch size, L is sequence length, and d is head dimension. Before computing attention, we reshape these to $\mathbf{Q} \in \mathbb{R}^{B\times H\times L\times d}$ and $\mathbf{K}, \mathbf{V} \in \mathbb{R}^{B\times K\times L\times d}$. The key-value heads must then be repeated via repeat_kv$(\mathbf{V}, G)$ to obtain $\mathbf{V}' \in \mathbb{R}^{B\times H\times L\times d}$ by expanding along a new dimension and reshaping: $\mathbf{V}' = \text{reshape}(\mathbf{V}[:,:,\text{None},:,:].expand(B,K,G,L,d), [B,H,L,d])$. This ensures dimensional compatibility when applying reweighted attention weights $\tilde{\mathbf{A}} \in \mathbb{R}^{B\times H\times L\times L}$ to compute the output $\mathbf{O} = \tilde{\mathbf{A}}\mathbf{V}'$. Our implementation maintains a full cache of value projections across generation steps, repeats the KV heads

appropriately, applies the reweighting strategy to the attention weights, recomputes the attention output with modified weights, and finally applies the output projection W_o. This approach preserves the efficiency benefits of GQA while enabling fine-grained control over visual-textual attention distribution across all H query heads.

7.2. Attention Reweighting Experiments

We experiment with more comprehensive layer based configurations to measure the impact of changing attention values over different layers. In all experiments we use the same prompt: `"Count the number of objects in this image. Answer the count within curly brackets, eg. {10}"`. We used only the `background texture` dataset for this evaluation because that provided a more challenging visual use case.

7.2.1. Experimental Configurations

Layer Groups.
- **Early (0–7)**: Feature extraction—low-level visual patterns and syntactic structure
- **Middle (8–23)**: Semantic integration—multimodal fusion and object relationships
- **Late (24–31)**: High-level reasoning—global reasoning and linguistic refinement

We evaluate 19 attention reweighting configurations across six categories:

Baseline.
- `baseline`: No modification—standard model inference

Uniform Strategies (all layers).
- `uniform_amplify`: $2\times$ visual attention throughout
- `uniform_suppress`: $0.5\times$ visual attention throughout
- `uniform_focus`: Exclusive visual attention throughout
- `uniform_balance`: Maintained 40% visual ratio throughout

Progressive Strategies (layer-wise transitions).
- `progressive_visual_fade`: Strong $\rightarrow$ balanced $\rightarrow$ weak visual (amplify/balance/suppress)
- `progressive_visual_grow`: Weak $\rightarrow$ balanced $\rightarrow$ strong visual (suppress/balance/amplify)

Localized Strategies (group-specific).
- `early_visual_only`: Focus early layers, suppress middle-late
- `middle_visual_boost`: Amplify middle layers, balance elsewhere
- `late_visual_retention`: Amplify late layers, balance elsewhere
- `extreme_visual_early`: Focus first 37.5% of layers (0–11), balance rest
- `extreme_text_late`: Suppress final 37.5% of layers (20–31), balance rest

Alternating Strategy.
- `alternating_amp_sup`: Layer-by-layer amplify/suppress alternation

Object-Aware Strategies (segmentation-guided).
- `early/middle/late_amplify_visual_mask`: Object amplification ($2\times$) with no background suppression in respective layer groups
- `early/middle/late_amplify_visual_mask_bg_suppress`: Object amplification ($2\times$) with strong background suppression ($0.5\times$) in respective layer groups

Findings: Tab. 7 presents Qwen3-VL-8B-Instruct MRCE and accuracy results for the background texture dataset across different object count buckets. We do not include the `bubbles` (Fig. 7k) texture since it confuses all VLMs into counting the bubbles as objects leading to high errors.

Progressive growth strategies (`progressive_visual_grow`, mean MRCE 0.37) and late-stage object masked amplification with background suppression (`late_amplify_visual_mask_bg_suppress`, mean MRCE 0.35) consistently outperform the baseline (0.33 MRCE), showing that gradual, layer-wise amplification of visual tokens while suppressing background information improves counting accuracy. The performance degradation across count buckets—from near-perfect accuracy in the 0-10 range (MRCE $\sim$0.02, Acc $\sim$0.87-0.90) to substantial errors in the 40-50 range (MRCE $\sim$0.06-0.16, Acc $\sim$0.08)—confirms that higher counts remain challenging even with attention reweighting, though our best methods maintain more graceful degradation than baseline. Critically, our experiments with aggressive early-layer interventions (`early_visual_only`, `extreme_visual_early`) demonstrate the importance of preserving text-visual alignment, as these methods catastrophically fail (mean MRCE 0.04-0.02) by oversuppressing linguistic representations.

Tab. 8 shows results from Qwen2.5-VL-7B-Instruct. The results on the background texture dataset reveal substantially different dynamics compared to Qwen 3-VL-8B, with most attention reweighting strategies achieving marginal improvements over baseline (mean MRCE 0.14 across top performers). Object mask-guided interventions including `early_amplify_visual_mask`, `late_amplify_visual_mask_bg_suppress`, and `alternating_amp_sup` all converge to 0.14 mean

Table 7. Qwen 3-VL-8B-Instruct MRCE and Accuracy results for different object count buckets averaged over the `background texture` dataset. Results sorted by Mean MRCE over all buckets. We see that object mask guided attention reweighting methods beat baseline.

Attn. Strat.	0-10		10-20		20-30		30-40		40-50		Mean	
	MRCE	Acc.	MRCE	Acc.	MRCE	Acc.	MRCE	Acc.	MRCE	Acc.	MRCE	Acc.
late_amplify_visual_mask_bg_suppress	0.02	0.87	0.05	0.5	0.09	0.17	0.09	0.11	0.12	0.11	0.07	0.35
late_amplify_visual_mask	0.02	0.87	0.05	0.44	0.1	0.13	0.09	0.15	0.14	0.06	0.08	0.33
baseline	0.02	0.87	0.05	0.42	0.09	0.2	0.1	0.12	0.14	0.06	0.08	0.33
early_amplify_visual_mask	0.02	0.89	0.05	0.47	0.09	0.2	0.1	0.13	0.14	0.06	0.08	0.35
early_amplify_visual_mask_bg_suppress	0.01	0.91	0.05	0.45	0.08	0.25	0.1	0.1	0.15	0.08	0.08	0.36
extreme_text_late	0.02	0.85	0.05	0.43	0.1	0.18	0.08	0.17	0.15	0.08	0.08	0.34
uniform_balance	0.02	0.88	0.05	0.43	0.09	0.14	0.1	0.14	0.15	0.11	0.08	0.34
progressive_visual_fade	0.02	0.85	0.04	0.48	0.08	0.25	0.09	0.13	0.18	0.16	0.08	0.37
progressive_visual_grow	0.02	0.9	0.04	0.49	0.09	0.22	0.1	0.16	0.17	0.06	0.08	0.37
late_visual_retention	0.02	0.86	0.05	0.41	0.08	0.19	0.11	0.12	0.18	0.09	0.09	0.33
alternating_amp_sup	0.03	0.77	0.07	0.28	0.1	0.16	0.09	0.15	0.17	0.07	0.09	0.29
uniform_suppress	0.1	0.42	0.17	0	0.2	0	0.17	0.02	0.14	0.06	0.16	0.1
middle_visual_boost	0.13	0.27	0.18	0.05	0.2	0.04	0.28	0.01	0.35	0.05	0.23	0.08
uniform_amplify	0.14	0.25	0.2	0.05	0.22	0.01	0.29	0	0.38	0.03	0.25	0.07
middle_amplify_visual_mask	0.1	0.39	0.17	0.09	0.22	0.06	0.32	0.01	0.54	0.01	0.27	0.11
middle_amplify_visual_mask_bg_suppress	0.1	0.38	0.17	0.1	0.26	0.01	0.36	0.01	0.6	0.01	0.3	0.1
early_visual_only	0.7	0	0.26	0.19	0.62	0	0.72	0	0.78	0	0.61	0.04
extreme_visual_early	NaN	0	NaN	0.08	NaN	0	NaN	0	NaN	0	NaN	0.02
uniform_focus	NaN	0	NaN	0	NaN	0	NaN	0	NaN	0	NaN	0

Table 8. Qwen 2.5 VL-7B-Instruct MRCE and Accuracy results for different object count buckets averaged over the `background texture` dataset. Results sorted by Mean MRCE over all buckets. We see that object mask guided attention reweighting methods and methods like `alternating_amp_sup` beat `baseline` in MRCE. As object counts increase, attention reweighting methods lead to modest gains from baseline performance.

Attn. Strat.	0-10		10-20		20-30		30-40		40-50		Mean	
	MRCE	Acc.	MRCE	Acc.	MRCE	Acc.	MRCE	Acc.	MRCE	Acc.	MRCE	Acc.
early_amplify_visual_mask	0.08	0.61	0.13	0.12	0.11	0.12	0.15	0.12	0.21	0.03	0.14	0.2
early_amplify_visual_mask_bg_suppress	0.09	0.57	0.13	0.11	0.1	0.15	0.15	0.12	0.22	0.01	0.14	0.19
late_amplify_visual_mask_bg_suppress	0.08	0.61	0.12	0.15	0.1	0.12	0.17	0.08	0.22	0.01	0.14	0.19
alternating_amp_sup	0.09	0.57	0.12	0.15	0.1	0.13	0.17	0.08	0.22	0.02	0.14	0.19
progressive_visual_grow	0.09	0.59	0.12	0.08	0.1	0.15	0.16	0.11	0.23	0.01	0.14	0.19
late_amplify_visual_mask	0.08	0.61	0.12	0.15	0.1	0.12	0.17	0.09	0.22	0	0.14	0.19
middle_amplify_visual_mask	0.09	0.59	0.14	0.11	0.11	0.09	0.15	0.11	0.2	0.05	0.14	0.19
baseline	0.08	0.61	0.12	0.14	0.1	0.12	0.17	0.09	0.22	0	0.14	0.19
middle_amplify_visual_mask_bg_suppress	0.09	0.57	0.14	0.05	0.11	0.13	0.16	0.09	0.2	0.05	0.14	0.18
progressive_visual_fade	0.13	0.58	0.15	0.1	0.11	0.1	0.14	0.15	0.19	0.06	0.14	0.2
uniform_balance	0.12	0.57	0.13	0.14	0.1	0.15	0.16	0.09	0.21	0.04	0.14	0.2
late_visual_retention	0.12	0.57	0.13	0.14	0.1	0.14	0.16	0.09	0.21	0.04	0.14	0.19
extreme_text_late	0.13	0.54	0.13	0.16	0.1	0.14	0.16	0.08	0.22	0.02	0.15	0.19
middle_visual_boost	0.11	0.53	0.21	0.05	0.15	0.06	0.12	0.2	0.16	0.11	0.15	0.19
uniform_amplify	0.12	0.52	0.23	0.07	0.19	0.05	0.11	0.22	0.15	0.1	0.16	0.19
uniform_suppress	0.1	0.54	0.17	0.02	0.14	0.09	0.22	0.03	0.27	0.01	0.18	0.14
early_visual_only	NaN	0	NaN	0	NaN	0	NaN	0	NaN	0	NaN	0
extreme_visual_early	NaN	0	NaN	0	NaN	0	NaN	0	NaN	0	NaN	0
uniform_focus	NaN	0	NaN	0	NaN	0	NaN	0	NaN	0	NaN	0

MRCE with 0.19-0.20 accuracy, suggesting that Qwen 2.5's architecture may already incorporate more effective visual attention mechanisms that limit the potential gains from our reweighting interventions. The compressed performance distribution—with nearly all viable strategies clustering tightly around baseline—indicates that this model family exhibits greater robustness to attention modifications, though it still demonstrates the characteristic degradation pattern across count buckets (0-10 range: MRCE $\sim$0.08-0.13, Acc $\sim$0.57-0.61; 40-50 range: MRCE $\sim$0.19-0.23, Acc $\sim$0.0-0.06). Notably, the same catastrophic failure modes persist for aggressive early-layer suppression strategies (`early_visual_only`, `extreme_visual_early`, `uniform_focus`), reinforcing our earlier finding that preserving text-visual alignment throughout the model is critical. These re-

Table 9. KimiVL-A3B-Instruct MRCE and Accuracy results for different object count buckets averaged over the `background texture` dataset. Results sorted by Mean MRCE over all buckets. We see that attention reweighting methods like `uniform_suppress` and `alternating_amp_sup` beat `baseline` in both MRCE, but do not match accuracy.

Attn. Strat.	0-10		10-20		20-30		30-40		40-50		Mean	
	MRCE	Acc.	MRCE	Acc.	MRCE	Acc.	MRCE	Acc.	MRCE	Acc.	MRCE	Acc.
alternating_amp_sup	0.05	0.58	0.10	0.14	0.09	0.15	0.07	0.10	0.14	0.10	0.09	0.24
uniform_suppress	0.07	0.46	0.10	0.14	0.06	0.20	0.11	0.00	0.10	0.10	0.09	0.20
early_visual_only	0.07	0.54	0.10	0.07	0.09	0.10	0.15	0.10	0.11	0.10	0.10	0.21
baseline	0.04	0.73	0.07	0.43	0.09	0.15	0.27	0.10	0.25	0.10	0.14	0.32
early_amplify_visual_mask	0.05	0.69	0.07	0.43	0.12	0.10	0.32	0.05	0.20	0.10	0.15	0.29
early_amplify_visual_mask_bg_suppress	0.05	0.69	0.07	0.43	0.12	0.10	0.32	0.05	0.20	0.10	0.15	0.29
late_amplify_visual_mask	0.04	0.73	0.08	0.36	0.09	0.15	0.32	0.05	0.30	0.10	0.16	0.30
late_amplify_visual_mask_bg_suppress	0.04	0.73	0.08	0.36	0.09	0.15	0.32	0.05	0.30	0.10	0.16	0.30
progressive_visual_grow	0.04	0.73	0.14	0.21	0.12	0.10	0.23	0.00	0.30	0.10	0.16	0.26
extreme_text_late	0.05	0.69	0.14	0.21	0.17	0.10	0.34	0.00	0.40	0.05	0.21	0.24
middle_amplify_visual_mask	0.05	0.65	0.16	0.36	0.20	0.05	0.39	0.00	0.35	0.05	0.22	0.24
middle_amplify_visual_mask_bg_suppress	0.05	0.65	0.16	0.36	0.20	0.05	0.39	0.00	0.35	0.05	0.22	0.24
late_visual_retention	0.06	0.65	0.13	0.21	0.24	0.00	0.44	0.00	0.56	0.00	0.28	0.20
uniform_balance	0.06	0.65	0.13	0.29	0.25	0.00	0.44	0.00	0.72	0.00	0.32	0.21
progressive_visual_fade	0.06	0.62	0.17	0.29	0.29	0.00	0.44	0.00	0.66	0.00	0.32	0.20
uniform_amplify	0.06	0.62	0.23	0.00	0.29	0.00	0.46	0.00	0.78	0.00	0.35	0.16
middle_visual_boost	0.07	0.58	0.20	0.14	0.34	0.00	0.60	0.00	0.96	0.00	0.42	0.17
extreme_visual_early	0.17	0.15	0.36	0.00	0.80	0.00	1.16	0.00	1.03	0.00	0.69	0.04
uniform_focus	0.29	0.08	0.64	0.00	1.24	0.00	1.10	0.00	1.13	0.00	0.86	0.02

sults suggest that while attention reweighting remains a viable intervention strategy, its effectiveness is highly architecture-dependent, with newer model families potentially incorporating design elements that already optimize attention patterns for visual reasoning tasks, thereby reducing the headroom for external manipulation.

Tab. 9 shows result from Kimi-VL-A3B. The results on the background texture dataset reveal a striking divergence from the Qwen model families, with attention reweighting strategies exhibiting counterintuitive behavior where improvements in MRCE metrics fail to translate to accuracy gains and often cause catastrophic performance collapse. While `uniform_suppress` and `alternating_amp_sup` achieve the lowest mean MRCE (0.09) and nominally outperform baseline (0.14 MRCE), they paradoxically underperform in accuracy (0.20-0.24 vs. 0.32 baseline), suggesting these interventions induce a counting bias that reduces error magnitude but compromises exact match performance. More concerning, many ostensibly moderate strategies—including previously successful approaches like `progressive_visual_grow`, `late_amplify_visual_mask`, and various middle-layer interventions—cause complete accuracy collapse to 0.00 in higher count buckets (30-40 and 40-50 ranges), despite maintaining reasonable MRCE values. This pattern indicates that KimiVL's architecture is fundamentally more brittle to attention manipulation, with interventions that successfully enhance visual attention in other models disrupting critical computational pathways in this smaller (3B parameter) architecture. The extreme sensitivity extends to aggressive strategies, where `uniform_focus` and `extreme_visual_early` produce catastrophic failures (0.86 and 0.69 mean MRCE respectively), reinforcing that smaller VLMs may lack the representational capacity to tolerate significant attention reweighting without losing essential cross-modal reasoning capabilities that enable accurate counting.

8. Sample Images

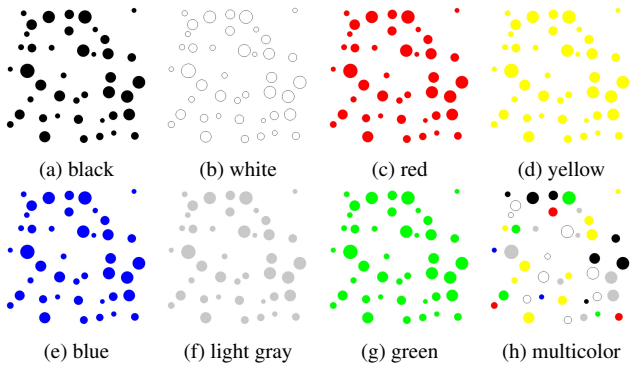

Figure 4. Example images for the **Object** category, **Color** pattern, showing different object colors.

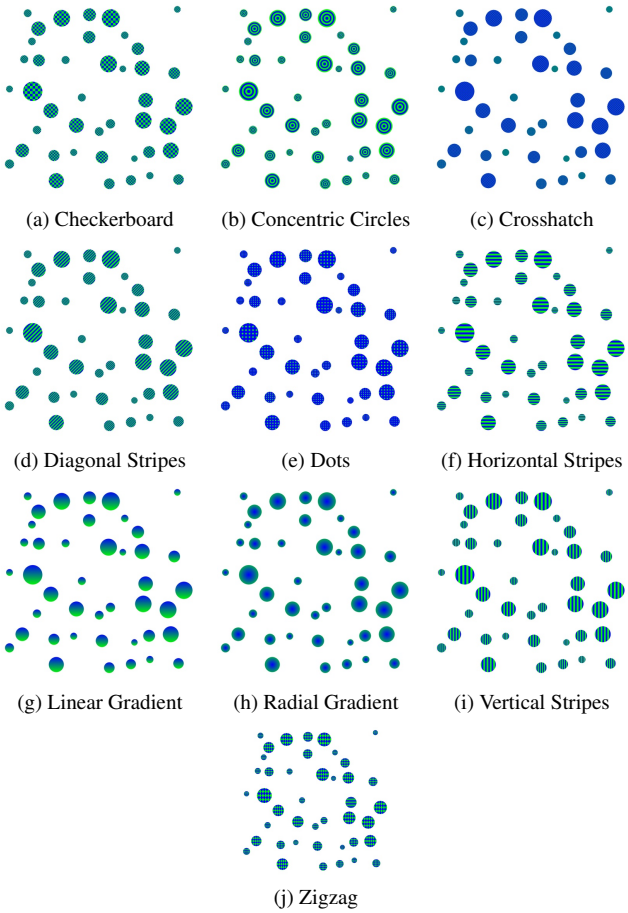

Figure 5. Example images for the **Object** category, **Texture** pattern, showing various texture types.

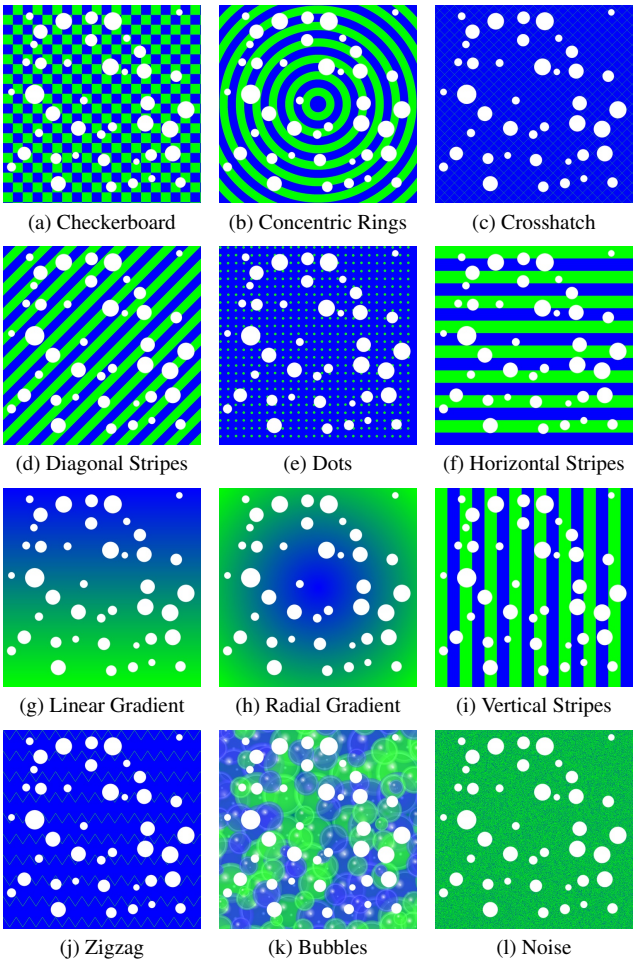

Figure 7. Example images for the **Background** category, **Texture** pattern, showing various background texture types.

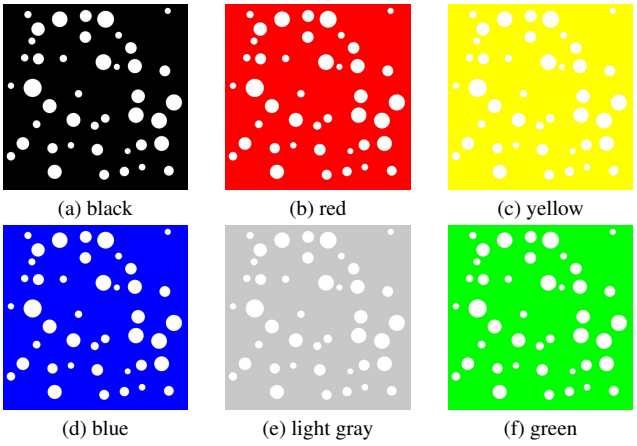

Figure 6. Example images for the **Background** category, **Color** pattern, showing different background colors.

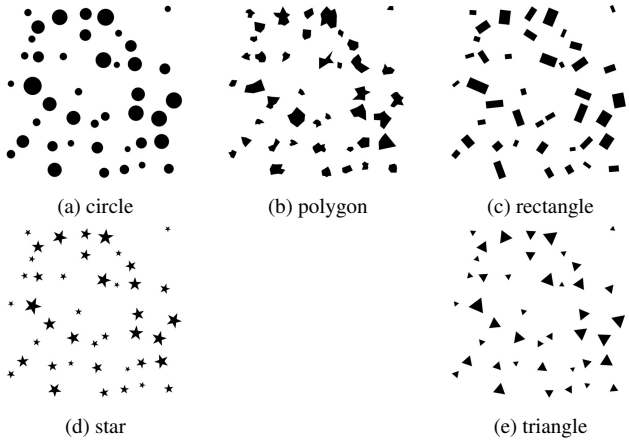

Figure 8. Example images for the **Object** category, **Shape** pattern, showing different object shapes.

9. Prompts

Table 10. Prompts used when image has different Object Color or Shape.

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 target Cue (Color) - Replace {color} with object colors (blue, green, yellow, gray, multicolor, e.g.).
P3	`Count the number of {color} color {shape} in this image...`	**Compositional (Simple) Attribute:** Bind target cues (color and shape). Replace {shape} with "circles"(as default in color experiment), "squares", "triangles","stars", etc. for shape experiment. Replace {color} with "black"(as default in shape experiment), "yellow", "red","blue", etc. for color experiment.

Table 11. Prompts used when image has different Background 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 Object Cue (Color) - Replace {color} with object colors ("white" for default).
P3	`Count the number of {color} color {shape} in this image...`	**Compositional (Target):** Binding (Simple+ Simple). Tests binding between two independent object attributes - Replace {shape} with object shape ("circles" for default).
P4	`Count the number of {color} color objects in this image with {pattern} background...`	**Compositional (Target+):** Binding (Complex + Simple). Tests binding a simple cue with a complex one. Tests whether the model can integrate object-level and background-level features.
P5	`Count the number of {color} color shape in this image with {color}{pattern} background...`	**Compositional (High Load):** Multi-attribute binding under high cognitive load. object color + background color ("blue-green" for default) + background pattern

10. Effects of Visual Complexity

Tables 13, Table 14, Table 15, and Table 16 present the Mean Relative Count Error (MRCE) for prompts 1, 3, 4, and 5, respectively.

11. Attention on Visual Tokens

Figures 9-10 shows the distribution of attention over vision as well as counting error across prompts for the Qwen2.5-32B-Instruct and InternVL3-9B. Across models (i.e. Qwen2.5-32B-Instruct, InternVL3-9B, Qwen2.5-7B, and Kimi-VL-A3B), we observe a consistent divide in how architectural scale influences the effect of prompt specificity. The smaller models—Qwen2.5-7B and Kimi-VL-A3B (3B active parameters)—in most cases show an initial reduction in relative count error as the prompts become more explicit, but this improvement saturates and eventually plateaus or even reverses. In contrast, the larger or more vision-specialized models—Qwen2.5-32B-Instruct and InternVL3-9B—do not exhibit this behavior. For these models, in most cases, increasing linguistic specificity does not reliably improve performance; their error remains relatively stable or fluctuates despite more detailed instructions. These findings suggest that, unlike smaller models that benefit from increased prompt granularity, higher-capacity or vision-specialized architectures may not gain additional advantage from more explicit linguistic guidance for this counting task.

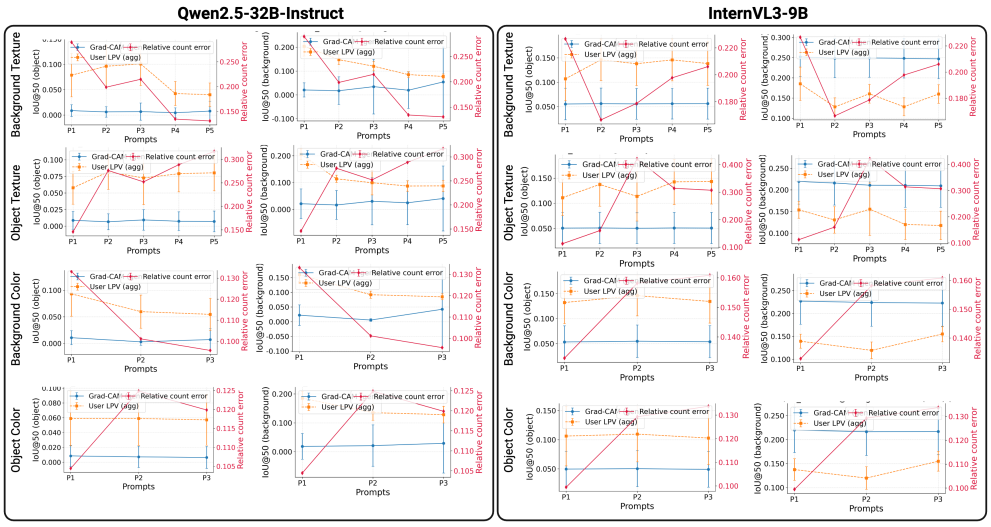

Figure 9. Visualization of the model's attention for models the Qwen2.5-32B-Instruct and InternVL3-9B

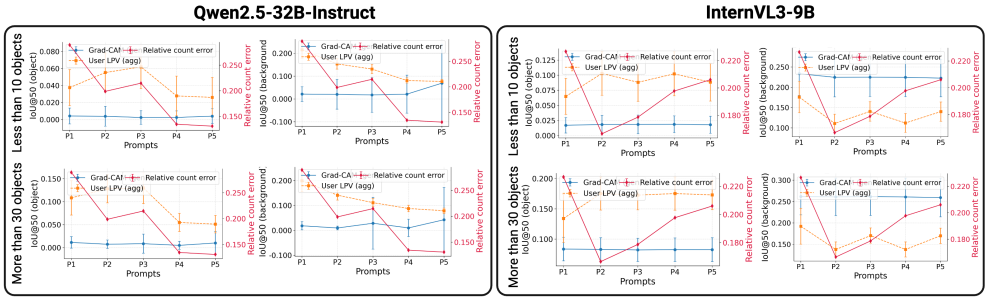

Figure 10. Visualization of the model's attention across different background-texture patterns for images containing fewer than 10 objects or more than 30 objects for models the Qwen2.5-32B-Instruct and InternVL3-9B.

Table 12. Prompts used when image has different Background Color.

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} objects in this image with {color} background...`	**Compositional (Simple) Attribute:** Bind target cues and contextual cue: Tests selective filtering based on a single attribute (background color), with object color("white" for default).
P3	`Count the number of {color} {shape} in this image with {color} background...`	**Compositional (Complex) Attribute:** Bind target and contextual cue: a single background attribute (background color), with two object attributes("white circle" for default).

Table 13. Mean Relative Count Error (lower is better) for Prompt 1 across all patterns. "Bg" denotes *Background*, "Obj" denotes *Object* and "diag. str." denotes diagonal stripes, "ver. str." denotes vertical stripes, "hor. str." denotes horizontal stripes, "con. cir." denotes concentric circles, "lin. grad." linear gradient, "rad. grad." denotes radial gradient, "con. rgs." denotes concentric rings, "cr. hatch" denotes cross hatch categories.

Cat.	Feat.	Pattern	Qwen7b	Qwen32b	Intern	Kimi
Bg	Color	blue	0.154	0.142	0.103	0.079
Bg	Color	black	0.170	0.143	0.115	0.080
Bg	Color	green	0.235	0.112	0.112	0.145
Bg	Color	gray	0.254	0.135	0.138	0.078
Bg	Color	red	0.311	0.129	0.131	0.247
Bg	Color	yellow	0.318	0.138	0.198	0.341
Bg	Texture	noise	0.234	0.121	0.128	0.086
Bg	Texture	cr. hatch	0.355	0.141	0.122	0.078
Bg	Texture	lin. grad.	0.504	0.115	0.078	0.084
Bg	Texture	rad. grad.	0.572	0.120	0.100	0.139
Bg	Texture	checkerboa	0.832	0.141	0.160	0.432
Bg	Texture	dots	0.803	0.184	0.144	0.453
Bg	Texture	diag. str.	0.751	0.163	0.098	0.687
Bg	Texture	con. rgs	0.723	0.270	0.198	0.627
Bg	Texture	hor. str.	0.734	0.306	0.109	0.757
Bg	Texture	ver. str.	0.695	0.413	0.153	0.773
Bg	Texture	bubbles	0.888	1.209	1.203	0.845
Obj	Color	white	0.210	0.066	0.102	0.098
Obj	Color	red	0.187	0.130	0.084	0.084
Obj	Color	yellow	0.201	0.113	0.082	0.116
Obj	Color	blue	0.217	0.130	0.108	0.068
Obj	Color	light gray	0.215	0.088	0.158	0.097
Obj	Color	green	0.335	0.075	0.081	0.078
Obj	Color	multicolor	0.553	0.129	0.081	0.350
Obj	Shape	star	0.216	0.143	0.083	0.077
Obj	Shape	circle	0.178	0.137	0.154	0.072
Obj	Shape	rectangle	0.390	0.123	0.195	0.085
Obj	Shape	polygon	0.397	0.118	0.140	0.138
Obj	Shape	triangle	0.493	0.154	0.069	0.320
Obj	Texture	rad. grad.	0.200	0.124	0.085	0.074
Obj	Texture	dots	0.462	0.143	0.132	0.071
Obj	Texture	con. cir.	0.576	0.075	0.106	0.073
Obj	Texture	lin. grad.	0.576	0.113	0.077	0.067
Obj	Texture	cr. hatch	0.617	0.076	0.101	0.079
Obj	Texture	checkerboa	0.644	0.110	0.101	0.062
Obj	Texture	ver. str.	0.617	0.128	0.089	0.092
Obj	Texture	zigzag	0.551	0.148	0.153	0.079
Obj	Texture	diag. str.	0.548	0.198	0.125	0.068
Obj	Texture	hor. str.	0.605	0.342	0.156	0.089

Table 14. Mean Relative Count Error (lower is better) for Prompt 3 across all patterns. "Bg" denotes *Background*, "Obj" denotes *Object* and "diag. str." denotes diagonal stripes, "ver. str." denotes vertical stripes, "hor. str." denotes horizontal stripes, "con. cir." denotes concentric circles, "lin. grad." linear gradient, "rad. grad." denotes radial gradient, "con. rgs." denotes concentric rings, "cr. hatch" denotes cross hatch categories.

Cat.	Feat.	Pattern	Qwen7b	Qwen32b	Intern	Kimi
Bg	Color	blue	0.142	0.096	0.130	0.064
Bg	Color	green	0.164	0.092	0.143	0.070
Bg	Color	black	0.168	0.085	0.141	0.079
Bg	Color	red	0.166	0.099	0.176	0.067
Bg	Color	gray	0.186	0.106	0.159	0.076
Bg	Color	yellow	0.190	0.096	0.216	0.094
Bg	Texture	lin. grad.	0.164	0.093	0.072	0.098
Bg	Texture	noise	0.144	0.092	0.161	0.073
Bg	Texture	rad. grad.	0.179	0.138	0.137	0.078
Bg	Texture	ver. str.	0.224	0.183	0.203	0.077
Bg	Texture	checkerboa	0.243	0.166	0.187	0.121
Bg	Texture	dots	0.280	0.183	0.173	0.107
Bg	Texture	hor. str.	0.246	0.219	0.126	0.167
Bg	Texture	con. rgs	0.254	0.186	0.261	0.070
Bg	Texture	cr. hatch	0.201	0.495	0.140	0.067
Bg	Texture	diag. str.	0.260	0.411	0.162	0.082
Bg	Texture	bubbles	0.271	0.196	0.344	0.205
Obj	Color	green	0.102	0.064	0.074	0.077
Obj	Color	blue	0.105	0.092	0.106	0.062
Obj	Color	red	0.126	0.108	0.088	0.067
Obj	Color	yellow	0.142	0.058	0.093	0.108
Obj	Color	white	0.111	0.125	0.128	0.130
Obj	Color	light gray	0.176	0.113	0.188	0.085
Obj	Color	multicolor	0.319	0.279	0.259	0.313
Obj	Shape	polygon	0.112	0.084	0.148	0.082
Obj	Shape	star	0.148	0.144	0.090	0.085
Obj	Shape	circle	0.182	0.134	0.182	0.076
Obj	Shape	triangle	0.101	0.148	0.088	0.281
Obj	Shape	rectangle	0.169	0.176	0.230	0.082
Obj	Texture	con. cir.	0.124	0.104	0.170	0.070
Obj	Texture	ver. str.	0.179	0.183	0.164	0.084
Obj	Texture	hor. str.	0.194	0.219	0.188	0.083
Obj	Texture	lin. grad.	0.176	0.093	0.421	0.077
Obj	Texture	dots	0.270	0.183	0.310	0.065
Obj	Texture	rad. grad.	0.217	0.138	0.432	0.085
Obj	Texture	checkerboa	0.355	0.326	0.492	0.078
Obj	Texture	zigzag	0.261	0.369	0.677	0.088
Obj	Texture	diag. str.	0.504	0.411	0.535	0.091
Obj	Texture	cr. hatch	0.490	0.495	0.819	0.301

Table 15. Mean Relative Count Error (lower is better) for Prompt 4 across all patterns. "Bg" denotes *Background*, "Obj" denotes *Object* and "diag. str." denotes diagonal stripes, "ver. str." denotes vertical stripes, "hor. str." denotes horizontal stripes, "con. cir." denotes concentric circles, "lin. grad." linear gradient, "rad. grad." denotes radial gradient, "con. rgs." denotes concentric rings, "cr. hatch" denotes cross hatch categories.

Cat.	Feat.	Pattern	Qwen7b	Qwen32b	Intern	Kimi
Bg	Texture	lin. grad.	0.183	0.103	0.112	0.079
Bg	Texture	noise	0.137	0.089	0.197	0.070
Bg	Texture	rad. grad.	0.187	0.138	0.135	0.063
Bg	Texture	cr. hatch	0.192	0.138	0.172	0.105
Bg	Texture	checkerboa	0.245	0.122	0.194	0.083
Bg	Texture	hor. str.	0.272	0.133	0.144	0.108
Bg	Texture	ver. str.	0.232	0.139	0.216	0.087
Bg	Texture	dots	0.271	0.143	0.200	0.106
Bg	Texture	diag. str.	0.273	0.154	0.191	0.131
Bg	Texture	con. rgs	0.287	0.158	0.268	0.091
Bg	Texture	bubbles	0.261	0.163	0.346	0.230
Obj	Texture	con. cir.	0.185	0.104	0.177	0.075
Obj	Texture	rad. grad.	0.194	0.129	0.218	0.117
Obj	Texture	lin. grad.	0.190	0.108	0.268	0.101
Obj	Texture	ver. str.	0.187	0.181	0.258	0.072
Obj	Texture	hor. str.	0.243	0.199	0.266	0.094
Obj	Texture	dots	0.219	0.298	0.166	0.166
Obj	Texture	zigzag	0.239	0.228	0.341	0.098
Obj	Texture	checkerboa	0.387	0.311	0.433	0.087
Obj	Texture	diag. str.	0.516	0.457	0.594	0.056
Obj	Texture	cr. hatch	0.679	0.873	0.419	0.798

Table 16. Mean Relative Count Error (lower is better) for Prompt 5 across all patterns. "Bg" denotes *Background*, "Obj" denotes *Object* and "diag. str." denotes diagonal stripes, "ver. str." denotes vertical stripes, "hor. str." denotes horizontal stripes, "con. cir." denotes concentric circles, "lin. grad." linear gradient, "rad. grad." denotes radial gradient, "con. rgs." denotes concentric rings, "cr. hatch" denotes cross hatch categories.

Cat.	Feat.	Pattern	Qwen7b	Qwen32b	Intern	Kimi
Bg	Texture	noise	0.152	0.069	0.203	0.067
Bg	Texture	lin. grad.	0.185	0.109	0.126	0.081
Bg	Texture	rad. grad.	0.192	0.121	0.153	0.065
Bg	Texture	cr. hatch	0.211	0.111	0.175	0.071
Bg	Texture	hor. str.	0.270	0.114	0.143	0.120
Bg	Texture	checkerboa	0.252	0.123	0.188	0.098
Bg	Texture	ver. str.	0.234	0.146	0.222	0.087
Bg	Texture	dots	0.293	0.106	0.211	0.114
Bg	Texture	diag. str.	0.282	0.163	0.202	0.110
Bg	Texture	con. rgs	0.295	0.189	0.283	0.103
Bg	Texture	bubbles	0.269	0.187	0.360	0.191
Obj	Texture	con. cir.	0.198	0.082	0.190	0.081
Obj	Texture	lin. grad.	0.175	0.114	0.241	0.103
Obj	Texture	rad. grad.	0.206	0.126	0.199	0.117
Obj	Texture	ver. str.	0.219	0.205	0.245	0.083
Obj	Texture	hor. str.	0.284	0.205	0.269	0.087
Obj	Texture	zigzag	0.240	0.307	0.318	0.110
Obj	Texture	dots	0.245	0.416	0.186	0.187
Obj	Texture	checkerboa	0.397	0.338	0.414	0.094
Obj	Texture	diag. str.	0.532	0.482	0.603	0.088
Obj	Texture	cr. hatch	0.745	0.893	0.404	0.786