

# Revisit What You See: Revealing Visual Semantics in Vision Tokens to Guide LVLm Decoding

Beomsik Cho Jaehyung Kim

Yonsei University

{bscho333, jaehyungk}@yonsei.ac.kr

## Abstract

Large Vision Language Models (LVLms) achieve strong performance across multimodal tasks by integrating visual perception with language understanding. However, how vision information contributes to the model’s decoding process remains under-explored, as reflected in frequent hallucinations. Through a series of analyses, we found that (i) vision tokens provide meaningful visual information even when hallucinations occur, and (ii) their semantics are encoded in the textual space and become explicit under appropriate vocabulary constraints. Building on these observations, we propose **ReVisiT**, a simple training-free decoding method that guides text generation in LVLms by **Referencing Vision Tokens**. Our approach leverages the semantic information embedded within vision tokens by projecting them into the text token distribution. Specifically, ReVisiT dynamically selects the most relevant vision token at each decoding step via context-aware constrained divergence minimization. Then, ReVisiT uses its constrained projection to refine the output distribution to better incorporate visual semantics. Across five benchmarks on recent LVLms, ReVisiT achieves competitive or superior results to state-of-the-art decoding baselines while reducing computational cost by up to  $2\times$ .<sup>1</sup>

## 1 Introduction

With the recent success of Large Language Models (LLMs) (Touvron et al., 2023; Achiam et al., 2023; Team et al., 2023), Large Vision-Language Models (LVLms) have emerged as powerful multimodal architectures that integrate visual perception with language understanding and text generation (Dai et al., 2023; Liu et al., 2024a; Zhu et al., 2024; Bai et al., 2025; Zhu et al., 2025). Typically, LVLms encode visual inputs into the LLM decoder’s text embedding space as *vision tokens*, which are then

<sup>1</sup>Code: <https://github.com/bscho333/ReVisiT>.

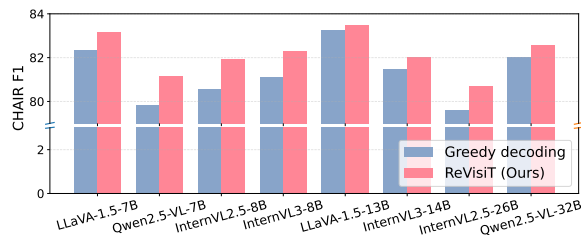


Figure 1: **ReVisiT across various model sizes and architectures.** We evaluate on the CHAIR benchmark and report object-level F1 computed over CHAIR object matches. Consistent improvements hold across different size buckets (7–8B / 13–14B / 26–32B) and architectures, demonstrating scalability and broad applicability.

processed alongside text tokens during decoding. This way of treating vision tokens as static auxiliary contexts, similar to retrieval-augmented generation (Lewis et al., 2020; Gao et al., 2023) in LLMs, enables the construction of complex multimodal systems in a relatively simple manner. However, this approach is often insufficient to capture the unique characteristic and role of vision tokens as sole carriers of visual information, as denoted by hallucinations frequently observed in LVLm’s text outputs (Leng et al., 2024; Favero et al., 2024; Huo et al., 2025; Rohrbach et al., 2018; Li et al., 2023b; Guan et al., 2024). To mitigate this, research has begun to expand the understanding of vision tokens along with their additional utilization (Jiang et al., 2025a,b). Yet, it is still under-explored *how vision tokens influence the decoding process of LVLms and what kinds of textual semantics they encode*.

To investigate this direction, we conduct a series of analyses on vision tokens, leading to two key insights. First, by examining the output distributions of LVLms particularly at hallucinated steps, we find that ground-truth objects often remain among the model’s high-probability candidates and this visual grounding signal in the output distribution comes from vision tokens. Second, we observe that the textual semantics of vision to-

kens is initially obscured, but the interpretable and well-aligned semantics could be discovered with the proper constraints to focus on specific subset. Here, we interpret the semantics of vision tokens by projecting them into the text token distribution using the LLM decoder’s language modeling head, as both are embedded in the same space. Overall, these findings suggest that vision tokens intrinsically encode object-level semantics, though such signals typically remain latent under conventional unconstrained decoding.

Building on this observation, we propose **ReVisiT**, a simple yet effective decoding strategy that guides text generation in LVLMs by **Referencing Vision Tokens**. At each step, ReVisiT selects the most relevant vision token based on *context-aware constrained divergence* from the current output distribution. Then, ReVisiT uses its constrained projection as reference to refine token-level logits. This activates the intrinsic semantics of vision tokens without training, architectural modifications, external modules, or multi-pass inference.

As highlighted in Fig. 1, ReVisiT consistently improves performance across six model sizes spanning three architectures, demonstrating scalability and broad applicability across LVLM families. To be specific, on three hallucination benchmarks: HallusionBench (Guan et al., 2024), CHAIR (Rohrbach et al., 2018), and POPE (Li et al., 2023b), ReVisiT reduces hallucinations robustly on three representative model architectures: LLaVA-1.5 (Liu et al., 2024a), Qwen2.5-VL (Bai et al., 2025), and InternVL3-8B (Zhu et al., 2025). Beyond hallucination, we further validate its effectiveness on general-purpose benchmarks, including VQAv2 (Antol et al., 2015) and MMMU (Yue et al., 2024), indicating that the semantics surfaced by our context-aware constrained projection generalize across tasks and architectures.

In summary, our contributions are as follows:

- We provide quantitative and qualitative evidence that vision tokens intrinsically encode interpretable semantics exposed when projected over *semantically coherent vocabulary subset*.
- We introduce **ReVisiT**, a training-free LVLM decoding method that references vision tokens via *context-aware constrained divergence*.
- We validate across *six model sizes (7B to 32B) spanning three architectures*, reducing hallucinations and yielding gains on five benchmarks, while maintaining inference cost.

## 2 Related Works

**Aligning visual inputs with language generation.** Prior attempts to align visual inputs with language generation in LVLMs can be broadly grouped into *pre-alignment*, *intra-alignment*, and *post-alignment*, depending on whether the alignment signal is injected before, during, or after decoding. Pre-alignment methods modify model architectures and training pipelines so that visual and textual modalities are better aligned before inference, for example by refining vision encoders and cross-modal projection modules (Bai et al., 2023; Chen et al., 2024c; Lu et al., 2024; You et al., 2024). Post-alignment approaches operate on the generated sequence and correct hallucinations or factual errors using auxiliary revisors or verification pipelines conditioned on the image and caption (Zhou et al., 2023; Yin et al., 2024). Intra-alignment methods intervene during decoding, for example by contrastively comparing logits under perturbed inputs to suppress language-prior-driven hallucinations (Leng et al., 2024; Favero et al., 2024; Kim et al., 2024), by recalibrating attention to visual tokens using vision-aware adjustments derived from attention patterns (Gong et al., 2024; Woo et al., 2024; Huo et al., 2025; Kang et al., 2025), or by introducing auxiliary visual-sensitive distributions or penalties and modifying decoding rules to better ground tokens in the visual input (Chen et al., 2024a; Huang et al., 2024; Wang et al., 2025). Across these approaches, visual information typically influences decoding implicitly through hidden states, logits, or attention scores.

**Interpreting vision representation via projection into language space.** Complementary to alignment frameworks, another line of work focuses on interpreting internal representation by projecting them into the vocabulary space using logit-lens-style probes (nostalgebraist, 2020; Chuang et al., 2024). Recently, this approach has been extended to vision representations in LVLMs to analyze vision-token semantics and support feature editing (Jiang et al., 2025a,b). However, prior work typically collapses them into scalar confidence or importance scores for diagnosing or reweighting internal model components. In contrast, we treat the projected distribution itself as a *context-aware constrained* signal that directly modulates next-token prediction. This positions our method at the intersection of projection-based interpretability and decoding-time visual grounding: we explicitly

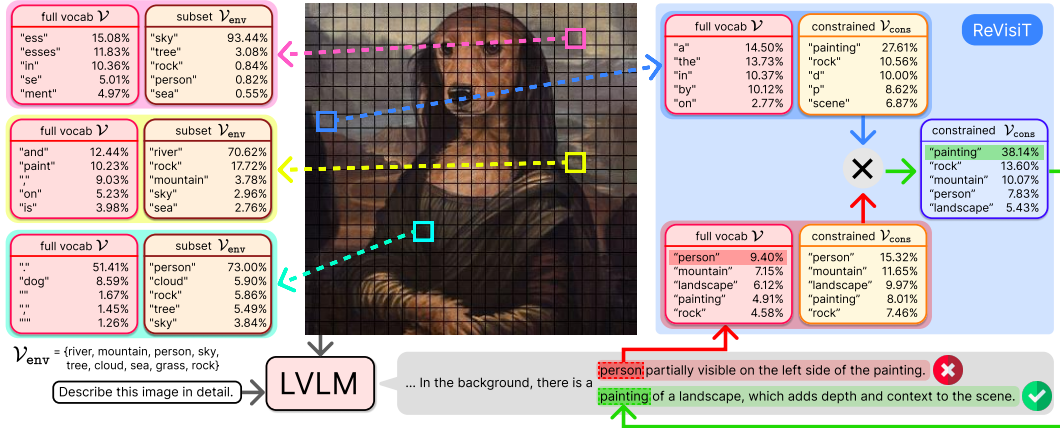


Figure 2: **Motivation.** We qualitatively analyzed various vision tokens with LLaVA-1.5-7B (Liu et al., 2024a). Dotted arrows represent vision token projection over specified vocabulary set. For each box, representing text token distribution, we annotated top-5 probable text tokens. Left part illustrates the effectiveness of vocabulary constraint, whereas right part shows the distribution shift during ReVisit. See Appendix F.1 for a detailed discussion.

expose the textual semantics of individual vision tokens in the language space and use them as an interpretable guidance signal for generation.

### 3 Empirical Analysis of How Vision Tokens Influence LVLM Decoding

In this section, we investigate whether vision tokens encode interpretable semantics in LVLM’s text token space that can directly inform its decoding process. To conduct systematic analyses, we adopt the CHAIR benchmark (Rohrbach et al., 2018), as it provides a principled definition of object existence based on 80 MS-COCO (Lin et al., 2014) object categories with 403 synonyms in total; for each sample, the set of synonyms of annotated object categories is regarded as ground-truth (GT) objects, while the remaining category synonyms are regarded as hallucinated (Hal) objects. We refer to a *hallucinated step* as a decoding step where generated token corresponds to an object in this MS-COCO synonym set but is not part of the ground-truth annotations for the given image.

Building on this definition, we sample 500 images from the MS-COCO validation set and generate captions with *Qwen2.5-VL-7B* (Bai et al., 2025) using vanilla greedy decoding with a maximum of 512 tokens and the standard CHAIR prompt, “Please describe this image in detail”.

#### 3.1 Evidence of visual grounding in output distributions with vision tokens

To examine how vision tokens affect LVLM decoding, we first ask whether ground-truth (GT) objects

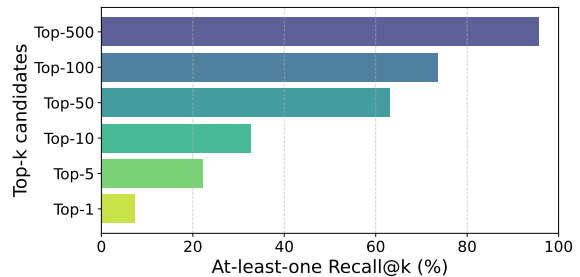


Figure 3: **Ground-truth objects frequently remain in top-probability predictions.** At 190 hallucinated steps, at least one GT object is recalled in 95.8% of cases within top-500 predictions, corresponding to only 0.33% of full vocabulary (151,665 tokens).

remain among the model’s high-probability predictions at hallucinated steps. We quantify this using *at-least-one recall@k*, which measures whether at least one GT object synonym appears among the top-*k* predictions of a given distribution (formal definition in Appendix B.1). Among the 500 generated captions, 176 captions contain at least one hallucinated step, yielding 190 hallucinated steps for step-level analysis. For each hallucinated step, we report *at-least-one recall@k* on the model’s output distribution at that step, excluding the greedy (top-1) prediction from the candidate set.

The results are summarized in Fig. 3. Here, GT objects frequently remain accessible in output distribution: in 63.2% of hallucinated steps, at least one GT object appears within top-50 candidates, and this recall rises to 95.8% within top-500. Given the vocabulary size of 151,665 tokens, top-500 corresponds to only 0.33% of the full vocabulary.<sup>2</sup>

<sup>2</sup>Step-selection details and 95% confidence intervals are

Table 1: **Comparison of Hal Max Avg and GT Max Avg.** The results for (i) all output tokens across all 500 CHAIR samples (88,706 tokens), (ii) all output tokens within hallucinated samples (176 samples, 35,116 tokens), and (iii) hallucinated decoding steps within hallucinated samples (176 samples, 190 steps). The higher value between Hal and GT is highlighted in **bold**.

Analysis Setting	Condition	GT Max Avg	Hal Max Avg
All tokens	w/ image	<b>24.49</b>	20.24
	w/o image	21.83	<b>22.66</b>
All tokens (in hallucinated samples)	w/ image	<b>24.42</b>	21.74
	w/o image	21.73	<b>23.88</b>
Hallucinated tokens	w/ image	32.00	<b>36.44</b>
	w/o image	30.26	<b>35.23</b>

A qualitative example is also presented in Fig. 2 (right), where the model incorrectly selects “person” as the greedy prediction, yet visually grounded alternatives such as “mountain,” “landscape,” and “painting” appear among top-5 candidates. This indicates that relevant visual candidates are assigned high probabilities in the output distribution, showing that LVLM indeed possesses the correct visual information even when hallucinations occur.

We further conduct a fine-grained comparison between the occurrences of GT and hallucinated object subsets during the decoding. Specifically, for each generated token, we track the maximum probability among objects in each subset. Then, we report the average of these per-step maxima as *GT Max Avg* for GT object subsets and *Hal Max Avg* for hallucinated object subsets, respectively (formal definition in Appendix B.3). To investigate the contribution of vision tokens by isolating them, we additionally report *w/o image* setting where we remove all vision tokens and perform teacher forcing with the previously generated caption as input while keeping decoding otherwise identical.

As shown in Table 1 (top), with vision tokens present, GT objects receive higher probability mass than hallucinated ones on average ( $GT > Hal$ ); removing the vision tokens reverses the ordering ( $Hal > GT$ ), confirming a visual grounding contribution from them. Interestingly, even for hallucinated samples, the similar results are observed (Table 1 (middle)) which indicates that grounding signal still exists in general. However, when focusing on the specific hallucinated steps, the reversed ordering ( $Hal > GT$ ) persists even with the presentation of image, indicating that *language priors* (i.e., textual co-occurrence biases from pre-training (Leng et al., 2024; Favero et al., 2024)) can dominate the prediction at specific positions and it leads to the

provided in Appendix B.2.

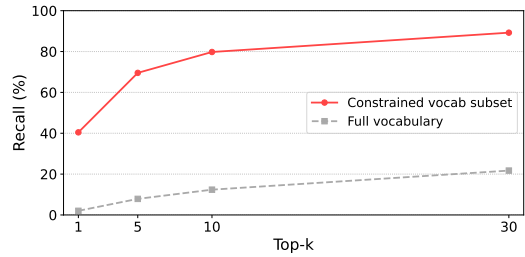


Figure 4: **Vision token projection.** Vision tokens reveal rich semantics when projected over semantically coherent subsets.

hallucination in LVLM decoding.<sup>3</sup>

In summary, these results indicate that, *even when hallucinations occur, LVLMs consistently retain strong visual grounding signals in their output distributions*, but not selected during decoding.

### 3.2 Object-level semantics in vision tokens

To identify the source of visual grounding signals, we focus on *vision tokens*, the sole carriers of visual information in LVLMs. Vision tokens share the same embedding space with text tokens and are autoregressively processed during decoding like them. This motivates probing their representations by projecting them through the model’s language-modeling head to obtain the text token distribution over the vocabulary, analogous to logit-lens and early-exit analyses for text token embeddings (nostalgebraist, 2020; Chuang et al., 2024; Wang et al., 2025). Specifically, for each of 500 MS-COCO validation images, we project each vision token’s final-layer hidden state to obtain the text token distribution, yielding 173,173 vision tokens and their projected distributions (details in Appendix B.4).

Then, to assess the presence of visual grounding signals in vision tokens, we compute token-level *at-least-one recall@k* for the projected distribution, similar to analyses in Fig. 3. The quantitative results are summarized in Fig. 4. Quantitatively, across 173,173 vision tokens, *at-least-one recall@k* is low under the full vocabulary: only **2.03%** of projections surface a GT synonym at top-1, increasing to **21.72%** at top-30. Qualitative inspection also indicates that tokens corresponding to background regions often yield generic function words, with object labels rarely appearing (red boxes on the left side of Fig. 2). These observations indicate that naive vision-token projections with the full vocabulary are diffuse and difficult to interpret.

<sup>3</sup>Wilcoxon signed-rank test results are provided in Appendix B.3.

In contrast, constraining the projection to *semantically coherent* vocabulary subsets reveals substantially clearer alignment with visual semantics. When restricting the prediction space to the CHAIR-derived object subset (176 single-token synonyms covering 62 categories), *at-least-one recall@k* on the same 173,173 projections increases sharply to **40.44%** at top-1—about  $20\times$  higher than the naive full-vocabulary projection (2.03%). At top-30, recall reaches **89.24%** while inspecting only  $30/176 \approx 17\%$  of the subset.<sup>4</sup> For comparability, both settings evaluate GT synonyms from the same 62 categories. Moreover, when we manually define domain-specific subsets, such as environment-related terms (e.g., “river,” “mountain,” and “tree”), the projected distribution aligns with visually salient regions and reveals object labels that remain hidden under the full-vocabulary setting (brown boxes on the left side of Fig. 2).

These findings show that *vision tokens encode rich object-level semantics and properly designed semantic constraints can reliably reveal these latent visual signals*.

For clarity, the subsets used so far are for controlled analysis and qualitative illustration, whereas the actual ReVisiT uses the adaptive context-aware candidate set introduced next in Eq. (3).

#### 4 ReVisiT: Referencing Vision Tokens to Guide Text Generation with LVLMS

Based on insights from Sec. 3, we propose ReVisiT, a decoding method that leverages these inherent visual semantics to enhance visual grounding during text generation. ReVisiT uses the vanilla output distribution as a context-aware constraint to surface the relevant visual semantics encoded in vision tokens, and integrates the revealed information to refine the next-token distribution. Specifically, ReVisiT consists of three main steps: (1) adaptively constraining the vocabulary subset based on the vanilla output distribution, (2) projecting the hidden states of vision tokens onto this subset and selecting the most relevant one, and (3) refining the output distribution through log-space fusion/addition and normalization. The algorithm and illustration of ReVisiT are presented in Appendix C.

**Preliminary.** Let  $\mathbf{x}$  and  $\mathbf{v}$  denote the input text tokens (e.g., query) and vision tokens, respectively. Specifically, the vision tokens  $\mathbf{v}$  are generated from

<sup>4</sup>Token-level *at-least-one recall@k* statistics, including 95% confidence intervals, are reported in Appendix B.4.

the given input image by forwarding it through the vision encoder and cross-modality projector sequentially (Liu et al., 2024a; Bai et al., 2025). Then, our goal is to generate output text tokens  $\mathbf{y}$  using LVLMS  $\mathcal{M}$ ; following the auto-regressive nature of LLMs, output tokens  $\mathbf{y}$  (e.g., response) are sequentially generated, i.e.,  $y_t \sim \mathcal{M}(\cdot | \mathbf{v}, \mathbf{x}, \mathbf{y}_{<t})$ . Let us assume that LVLMS  $\mathcal{M}$  has  $L$  decoding layers and text token vocabulary  $\mathcal{V}$ , and denote its hidden state of  $l$ -th layer as  $h^l$ . The output distribution is parameterized by model weights  $\theta$  and denoted as  $p_\theta$ . Then, at time step  $t$ , output probability over  $\mathcal{V}$  is calculated as:

$$p_\theta(h_{T+t-1}^L, \mathcal{V}) = \text{softmax}_{\mathcal{V}}(\phi(h_{T+t-1}^L)), \quad (1)$$

where  $T = |\mathbf{v}| + |\mathbf{x}|$ ,  $\phi$  is the language modeling head that maps the final hidden state of LVLMS into logits over vocabulary  $\mathcal{V}$ , and  $\text{softmax}_{\mathcal{V}}$  denotes a softmax operation over  $\mathcal{V}$  such that  $\sum_{w \in \mathcal{V}} p_\theta(w | h_{T+t-1}^L, \mathcal{V}) = 1$ . Then, the output token  $y_t$  is sampled from the obtained distribution:

$$y_t \sim p_\theta(h_{T+t-1}^L, \mathcal{V}). \quad (2)$$

#### 4.1 Context-aware vocabulary subset construction

At each decoding timestep  $t$ , the LVLMS  $\mathcal{M}$  generates an output distribution  $p_\theta(h_{T+t-1}^L, \mathcal{V})$  over the entire vocabulary space  $\mathcal{V}$ . However, as observed in Fig. 2 and discussed in Sec. 3.1, this distribution is often diffusely spread across many irrelevant tokens. Therefore, we dynamically constrain a vocabulary  $\mathcal{V}$  into subset  $\mathcal{V}_{\text{cons}}^t$  more suitable to capture the plausible semantic space of the output distribution. By restricting the output space to a smaller, contextually focused subset, this constraint reduces semantic distraction and facilitates more precise integration of visual information. To be specific, motivated by the Adaptive Plausibility Constraint (APC) (Li et al., 2023a), we adaptively define the vocabulary subset  $\mathcal{V}_{\text{cons}}^t$  as below:

$$\mathcal{V}_{\text{cons}}^t = \left\{ w \in \mathcal{V} : \begin{array}{l} p_\theta(w | h_{T+t-1}^L, \mathcal{V}) \\ \geq \alpha \cdot \max_{w'} p_\theta(w' | h_{T+t-1}^L, \mathcal{V}) \end{array} \right\}, \quad (3)$$

where  $\alpha \in (0, 1)$  controls the tightness of the selected candidate set. Smaller  $\alpha$  yields a looser subset, while larger  $\alpha$  yields a tighter subset. This subset  $\mathcal{V}_{\text{cons}}^t$  serves as a dynamically adapted output space, over which all subsequent vision token projections and refinements are restricted, thereby en-

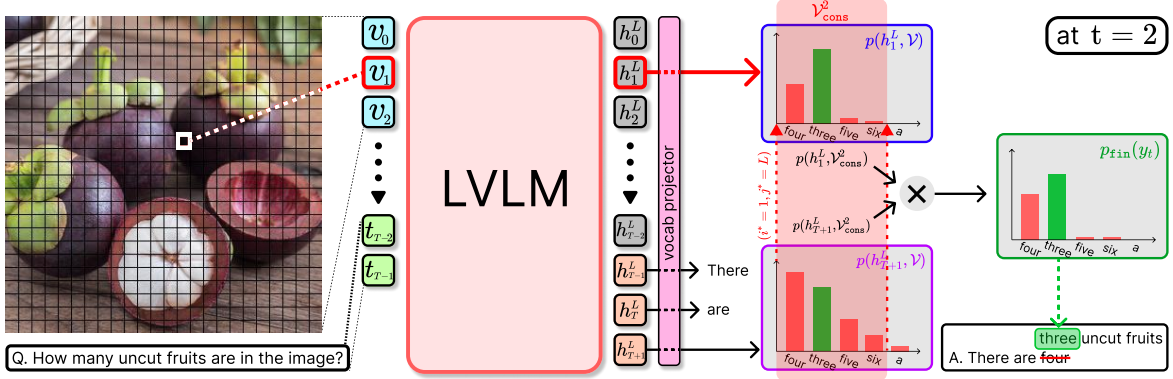


Figure 5: **An overview of ReVisiT.** At each decoding step, ReVisiT (1) constrains the vocabulary  $\mathcal{V}$  to  $\mathcal{V}_{\text{cons}}^t$ , (2) projects vision token embeddings into  $\mathcal{V}_{\text{cons}}^t$  and selects most relevant token, and (3) refines the final output distribution. ReVisiT leverages vision tokens to serve as reference signals for decoding, enhancing visual grounding without additional forward passes or external supervision.

hancing visual grounding without additional computation.

## 4.2 Vision token selection with projection

Given  $\mathcal{V}_{\text{cons}}^t$  at each decoding timestep  $t$ , we next describe how vision token hidden states  $h_i^j$  are projected and selected within this constrained space. Let  $\mathcal{J}$  denote the set of candidate decoder layers where vision token hidden states are selected. Specifically, for each  $h_i^j$ , where  $i$  denotes the index of the vision tokens and  $j$  denotes the candidate layer, we obtain the projected distribution over  $\mathcal{V}_{\text{cons}}^t$ :

$$p_{\theta}(h_i^j, \mathcal{V}_{\text{cons}}^t), \quad \text{for } i \in \{0, \dots, |\mathcal{V}| - 1\}, j \in \mathcal{J}. \quad (4)$$

Intuitively, this distribution represents the likelihood of each candidate token within  $\mathcal{V}_{\text{cons}}^t$  conditioned on the vision token embedding  $h_i^j$ . Next, to identify the vision token most relevant to the current decoding context, we compare each vision token distribution  $p_{\theta}(h_i^j, \mathcal{V}_{\text{cons}}^t)$  against the vanilla output distribution  $p_{\theta}(h_{T+t-1}^L, \mathcal{V}_{\text{cons}}^t)$  by computing the Jensen-Shannon Divergence (JSD):

$$(i^*, j^*) = \arg \min_{i,j} \text{JSD} \left( p_{\theta}(h_{T+t-1}^L, \mathcal{V}_{\text{cons}}^t) \parallel p_{\theta}(h_i^j, \mathcal{V}_{\text{cons}}^t) \right). \quad (5)$$

Here,  $i^*$  and  $j^*$  denote the indices of the selected vision token and decoder layer, respectively. We adopt JSD as the divergence metric due to its symmetric properties and bounded output range.

**Remark.** For the efficiency, the vision token projections over the full vocabulary,  $p_{\theta}(h_i^j, \mathcal{V})$ , are precomputed once before decoding begins. Then, at each timestep  $t$ , we apply slicing and masking to adapt the cached projections to the dynamically

constrained subset  $\mathcal{V}_{\text{cons}}^t$ , enabling efficient selection without additional forward computation.

## 4.3 Logit refinement with projected token distribution of selected vision token

After identifying the most contextually relevant vision token at each decoding step, we then refine the vanilla output distribution guided by the reference visual grounding signal from the selected vision token. This refinement admits a KL-regularized, Product-of-Experts (PoE)-style interpretation; we provide the derivation in Appendix C.2. Specifically, given the vanilla output distribution on  $\mathcal{V}_{\text{cons}}^t$ ,  $p_{\theta}(y_t | h_{T+t-1}^L, \mathcal{V}_{\text{cons}}^t)$ , and the selected vision-token distribution on the same constrained set,  $p_{\theta}(y_t | h_{i^*}^{j^*}, \mathcal{V}_{\text{cons}}^t)$ , we combine them via log-space addition:

$$l_{\text{ReVisiT}}(y_t) = \log p_{\theta}(y_t | h_{T+t-1}^L, \mathcal{V}_{\text{cons}}^t) + \log p_{\theta}(y_t | h_{i^*}^{j^*}, \mathcal{V}_{\text{cons}}^t). \quad (6)$$

The final distribution is then defined as

$$p_{\text{fin}}(y_t) = \begin{cases} \text{softmax}_{\mathcal{V}_{\text{cons}}^t}(l_{\text{ReVisiT}}(y_t)), & y_t \in \mathcal{V}_{\text{cons}}^t, \\ 0, & \text{otherwise.} \end{cases} \quad (7)$$

The next output token  $y_t$  is sampled from the resulting distribution  $p_{\text{fin}}$ . This refinement mechanism allows the decoding process to dynamically integrate visual grounding signals extracted from vision tokens, without introducing additional inference passes or architectural modifications.

## 5 Experiments

### 5.1 Setups

**Models and baselines.** We mainly evaluate three representative state-of-the-art LLMs: *LLaVA-1.5-*

Table 2: **HallusionBench**. Accuracies (%) on five metrics:  $qAcc$  (Question-pair accuracy),  $fAcc$  (Figure accuracy),  $aAcc$  (All accuracy),  $EaAcc$  (Easy question accuracy), and  $HaAcc$  (Hard question accuracy). Higher scores ( $\uparrow$ ) indicate better performance. The best results in each metric are **bolded**, and the second-best are underlined.

Method	LLaVA-1.5-7B					Qwen2.5-VL-7B					InternVL3-8B				
	$qAcc \uparrow$	$fAcc \uparrow$	$aAcc \uparrow$	$EaAcc \uparrow$	$HaAcc \uparrow$	$qAcc \uparrow$	$fAcc \uparrow$	$aAcc \uparrow$	$EaAcc \uparrow$	$HaAcc \uparrow$	$qAcc \uparrow$	$fAcc \uparrow$	$aAcc \uparrow$	$EaAcc \uparrow$	$HaAcc \uparrow$
Greedy	11.55	22.61	48.22	44.40	48.54	29.96	34.78	61.08	<b>65.70</b>	53.65	25.99	33.91	56.35	61.73	48.54
DoLa	13.36	22.61	<u>49.24</u>	<b>45.85</b>	<u>49.27</u>	27.08	30.87	58.04	62.09	51.46	<u>32.49</u>	<u>37.39</u>	<b>59.90</b>	<u>65.70</u>	<u>51.82</u>
VCD	<u>14.44</u>	22.61	47.55	44.77	47.45	29.60	<u>35.22</u>	61.25	<u>64.62</u>	54.74	28.88	35.65	57.36	62.82	49.64
M3ID	11.91	<u>23.04</u>	48.22	43.68	<u>49.27</u>	30.32	33.48	60.07	62.82	54.38	28.16	33.91	57.02	60.65	50.73
CODE	10.47	20.43	46.36	41.52	47.81	28.88	31.74	57.53	57.76	54.74	28.52	36.52	56.68	62.09	49.27
SID	12.27	21.30	48.22	44.77	48.18	<u>32.13</u>	<b>36.09</b>	61.59	<b>65.70</b>	55.11	31.05	36.09	58.04	61.73	<b>52.19</b>
<b>Ours</b>	<b>20.22</b>	<b>26.52</b>	<b>51.44</b>	<u>45.13</u>	<b>54.38</b>	<b>32.85</b>	<b>36.09</b>	<b>61.76</b>	64.26	<b>56.93</b>	<b>33.21</b>	<b>39.13</b>	<u>59.39</u>	<b>66.07</b>	<b>52.19</b>

Table 3: **CHAIR benchmark**. Lower scores ( $\downarrow$ ) on  $CHAIR_S$ ,  $CHAIR_I$  and higher ( $\uparrow$ ) F1 indicate better performance. The best results in each metric are **bolded**, and the second-best are underlined.

Method	LLaVA-1.5-7B			Qwen2.5-VL-7B			InternVL3-8B		
	$CHAIR_S \downarrow$	$CHAIR_I \downarrow$	F1 $\uparrow$	$CHAIR_S \downarrow$	$CHAIR_I \downarrow$	F1 $\uparrow$	$CHAIR_S \downarrow$	$CHAIR_I \downarrow$	F1 $\uparrow$
Greedy	53.8	14.66	82.33	35.2	8.43	79.85	29.4	<u>7.91</u>	81.11
DoLa	53.6	14.45	82.27	31.0	<u>7.63</u>	79.33	<u>28.8</u>	8.49	81.37
VCD	52.8	15.61	81.84	35.0	9.07	80.10	30.8	8.43	81.56
M3ID	57.0	16.57	82.06	33.8	9.80	78.80	29.2	8.02	81.81
CODE	<b>45.2</b>	<b>13.13</b>	81.48	<b>17.0</b>	9.35	48.50	29.0	8.25	81.32
SID	50.6	13.58	<b>83.34</b>	37.6	8.84	80.40	33.0	9.58	81.82
<b>Ours</b>	<u>50.6</u>	<u>13.43</u>	<u>83.17</u>	<u>29.8</u>	<b>7.04</b>	<b>81.16</b>	<b>27.6</b>	<b>7.27</b>	<b>82.29</b>

7B (Liu et al., 2024a) as a widely adopted conventional baseline, *Qwen2.5-VL-7B* (Bai et al., 2025), and *InternVL3-8B* (Zhu et al., 2025). As baselines, we adopt various intra-alignment decoding strategies to compare the effectiveness of our method in enhancing visual grounding. (a) *Greedy decoding* is a standard decoding baseline, and (b) *DoLa* (Chuang et al., 2024) improves factuality by contrasting output logits from earlier and later decoder layers at each timestep, and has been widely adopted for hallucination mitigation in LVLMS despite being originally proposed for LLMs. For intra-alignment baselines, (c) *VCD* (Leng et al., 2024) mitigates hallucination by perturbing visual inputs with Gaussian noise and enforcing consistency between the original and perturbed outputs; (d) *M3ID* (Favero et al., 2024) contrasts outputs with and without visual input to alleviate language prior bias and enhance visual grounding; (e) *CODE* (Kim et al., 2024) leverages the model’s self-generated image description as a contrasting reference to alleviate hallucination; (f) *SID* (Huo et al., 2025) applies contrastive decoding on low-attention vision patches to refine visual grounding.

**Implementation details.** In entire experiments, we apply deterministic **greedy decoding** (*i.e.*, temperature = 0) with a maximum output length of 512 tokens for *all decoding methods*. Regarding baselines, we follow the default hyperparameters provided by their original implementations.

For ReVisiT, we set the hyperparameters as follows: for LLaVA-1.5-7B and InternVL3-8B, we select vision tokens mostly from the last decoder layer (“last”), while for Qwen2.5-VL-7B, we select vision tokens mostly from all even-numbered decoder layers (“all”). The threshold  $\alpha$  is chosen per task type. More details are in Appendix D.2.

## 5.2 Main results

We evaluate our method on five benchmarks that jointly span three axes: (i) *generative vs. discriminative* settings, (ii) *manually designed vs. LLM-aided* evaluation, and (iii) *hallucination-focused vs. general VQA* tasks, allowing us to assess the robustness and generalizability of the model’s descriptive and decision-making abilities. Detailed benchmark and metric descriptions are in Appendix D.1, and full quantitative results are in Appendix E.

**HallusionBench** (Guan et al., 2024) probes image–context reasoning and entangled hallucinations with an LLM-based judge. In the visual-dependent setting (Table 2), ReVisiT attains the best score on four out of five metrics for each LVLm; on LLaVA-1.5-7B and InternVL3-8B it is second-best on the remaining metric, while on Qwen2.5-VL-7B its  $EaAcc$  stays within 1.5 points of the best method. In particular, ReVisiT improves  $qAcc$  over greedy decoding by approximately 75% on LLaVA-1.5-7B and 28% on InternVL3-8B, indicating that referencing vision tokens yields more reliable visual grounding even when image and

Table 4: **POPE benchmark**. Average for 3 datasets and 3 types. Higher scores ( $\uparrow$ ) on Accuracy, Precision, and F1 indicate better performance. The best results in each metric are **bolded**, and the second-best are underlined.

Method	LLaVA-1.5-7B			Qwen2.5-VL-7B			InternVL3-8B		
	Accuracy $\uparrow$	Precision $\uparrow$	F1 $\uparrow$	Accuracy $\uparrow$	Precision $\uparrow$	F1 $\uparrow$	Accuracy $\uparrow$	Precision $\uparrow$	F1 $\uparrow$
Greedy	79.47	74.48	82.36	84.23	<b>93.01</b>	82.53	<u>89.17</u>	<u>88.87</u>	<u>89.36</u>
DoLa	79.58	74.67	82.41	81.36	92.36	78.71	86.63	<b>92.08</b>	85.64
VCD	77.74	72.51	80.97	84.43	92.48	82.91	85.39	86.33	85.41
M3ID	79.48	74.48	82.37	83.56	91.52	82.00	89.05	<u>88.87</u>	89.22
CODE	78.58	73.98	81.35	83.55	90.20	82.26	86.70	85.40	87.22
SID	80.40	<u>75.94</u>	<u>82.91</u>	<b>85.78</b>	92.27	<b>84.72</b>	85.38	86.46	85.44
<b>Ours</b>	<b>81.80</b>	<b>78.03</b>	<b>83.45</b>	<u>84.53</u>	<u>93.00</u>	<u>82.93</u>	<b>89.21</b>	88.63	<b>89.44</b>

Table 5: **VQAv2 and MMMU**. We report *Accuracy* for both benchmarks. Higher scores ( $\uparrow$ ) indicate better performance. The best results in each setting are **bolded**, and the second-best are underlined.

Method	VQAv2 (Accuracy $\uparrow$ )			MMMU (Accuracy $\uparrow$ )		
	LLaVA-1.5-7B	Qwen2.5-VL-7B	InternVL3-8B	LLaVA-1.5-7B	Qwen2.5-VL-7B	InternVL3-8B
Greedy	69.47	65.80	<u>68.73</u>	34.39	51.11	<u>53.54</u>
DoLa	69.87	60.60	68.13	34.29	50.15	53.08
VCD	65.93	66.07	53.60	33.51	<u>51.13</u>	52.60
M3ID	<u>70.33</u>	59.87	<b>69.53</b>	<u>34.51</u>	46.63	53.43
CODE	66.00	46.60	56.33	34.48	42.79	52.41
SID	<u>70.33</u>	<u>67.13</u>	55.27	33.10	49.75	51.85
<b>Ours</b>	<b>70.40</b>	<b>67.60</b>	68.40	<b>35.13</b>	<b>51.18</b>	<b>54.14</b>

context provide conflicting cues.

**CHAIR** (Rohrbach et al., 2018) measures hallucination rates in image captioning by comparing generated captions to ground-truth object annotations from the MS-COCO (Lin et al., 2014) dataset. We report sentence-level ( $CHAIR_S$ ) and instance-level ( $CHAIR_I$ ) hallucination scores on 500 sampled images, and additionally include an object-level precision/recall *F1* computed over CHAIR object matches to reflect both hallucination suppression and grounded object coverage. As shown in Table 3, ReVisiT achieves the best performance on InternVL3-8B and consistently ranks within top-2 across all three metrics on LLaVA-1.5-7B and Qwen2.5-VL-7B. This indicates that ReVisiT reduces hallucinations without collapsing into overly conservative captions, achieving a more balanced trade-off between hallucination suppression and object coverage, as reflected by improvements in all three metrics.

**POPE** (Li et al., 2023b) evaluates object presence with binary questions across three datasets (MS-COCO, A-OKVQA (Schwenk et al., 2022), and GQA (Hudson and Manning, 2019)) and three query types (random, popular, adversarial). We report averaged *Accuracy*, *Precision*, and *F1* over all datasets and query types. As summarized in Table 4, ReVisiT achieves the best *Accuracy* and *F1* on both LLaVA-1.5-7B and InternVL3-8B, and ranks second on both metrics for Qwen2.5-VL-7B, yielding top-2 performance in every model-metric pair and indicating robust gains in discriminative

grounding across architectures.

**VQAv2** (Goyal et al., 2017) measures VQA correctness under inter-annotator variability via consensus accuracy over 10 annotations. As shown in Table 5 (left), ReVisiT improves over greedy decoding on both LLaVA-1.5-7B and Qwen2.5-VL-7B, achieving the highest accuracy among all decoding methods on these models. In contrast to baselines that exhibit substantial performance drops and ranking changes, ReVisiT maintains consistently competitive performance across all three models, indicating the robustness of ReVisiT across architectures without causing VQA failures.

**MMMU** (Yue et al., 2024) is a college-level benchmark spanning 30 disciplines of multimodal question answering, requiring substantial subject knowledge and reasoning in both multiple-choice and open-ended formats. As shown in Table 5 (right), ReVisiT achieves the highest accuracy for all three LVLMS, demonstrating its ability to generalize across diverse LVLMS and task setups.

**Overall.** Across these five benchmarks, ReVisiT remains consistently competitive across diverse evaluation settings, from object-level grounding to knowledge-intensive multimodal reasoning. This consistency suggests that the proposed context-aware constrained divergence captures a broadly useful decoding signal that improves visual grounding while reducing hallucinations, rather than a benchmark-specific heuristic. Such versatility, achieved without any additional training or multi-pass inference, underscores the efficiency

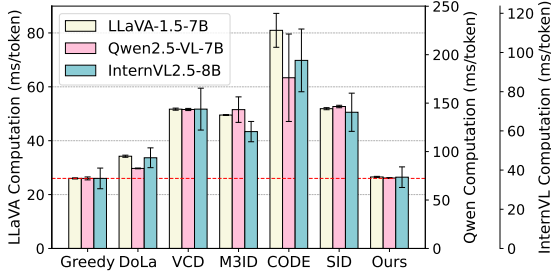


Figure 6: **Inference speed.** Comparison of per-token inference latency across different decoding strategies for LLaVA-1.5-7B (left y-axis), Qwen2.5-VL-7B (right y-axis), and InternVL3-8B (outer right y-axis), with standard deviations visualized as error bars.

and generalizability of leveraging intrinsic vision-token semantics for robust multimodal alignment.

### 5.3 Additional analyses

**Inference speed improvement.** To evaluate the inference efficiency of ReVisiT compared to baseline decoding strategies, we measure token-level inference latency (ms/token) for LLaVA-1.5-7B, Qwen2.5-VL-7B, and InternVL3-8B.

As shown in Figure 6, ReVisiT remains comparable to vanilla greedy decoding, increasing runtime by only 0.6% to 2.0% across three LLMs. In contrast, methods that require additional forward passes (e.g., VCD, M3ID, SID, CODE) *approach or exceed*  $2\times$  the greedy runtime, while even DoLa, which only adds a projection on intermediate layers at each step, incurs 14% to 30% overhead across all three models. These results highlight a key advantage of ReVisiT: it enhances visual grounding with negligible computational overhead, unlike prior methods that significantly compromise inference efficiency. See Appendix E.4 for measurement setup, hardware configuration, and full numerical results.

**Ablation study.** We ablate three factors on Qwen2.5-VL under the CHAIR benchmark (Figure 7): (1) vision-token selection criterion (min/random/max-JSD), (2) vocabulary constraint (subset vs. full), and (3) layer scope (all layers vs. last layer) with threshold  $\alpha$ .

(1) *Selection:* Under a constrained (subset) vocabulary, max-JSD collapses ( $F1 = 0.67$ ) and random underperforms ( $F1 = 17.63$ ), while our min-JSD attains the best performance ( $F1 = 81.16$ ), showing the effectiveness of a context-aware constrained divergence for token selection.

(2) *Vocabulary constraint:* Even with the same

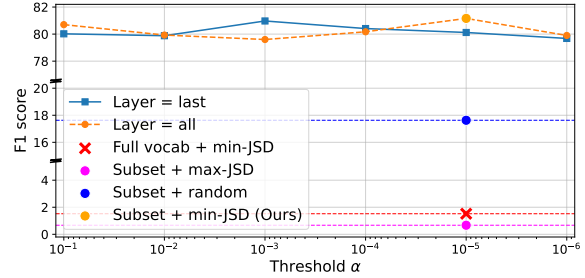


Figure 7: **Ablation study.** We evaluate the impact of vocabulary subset constraint, vision token selection, layer scope, and threshold  $\alpha$  variation on CHAIR benchmark.

min-JSD selection, projecting over the full vocabulary fails ( $F1 = 1.52$ ), exhibiting a degenerate behavior where vision-token projections are referenced off-target. This confirms that constraining the candidate set to context-relevant tokens is crucial; qualitative failures are provided in Appendix F.2.

(3) *Layer scope & threshold:* Across  $\alpha$ , the all-layers variant peaks at  $\alpha = 10^{-5}$  ( $F1 = 81.16$ ), while the last-layer variant peaks at  $\alpha = 10^{-3}$  ( $F1 = 80.97$ ). Overall sensitivity to  $\alpha$  is modest (within  $\approx 1.5$  F1 of each peak), indicating that *all-layers + vocabulary subset + min-JSD* is a robust setting on CHAIR. Additionally, on VQAv2, accuracy increases from 65.93 at  $\alpha = 10^{-1}$  to 67.60 at  $\alpha = 10^{-2}$ , but decreases to 64.00 at  $\alpha = 10^{-4}$ , again suggesting that overly loose candidate sets are detrimental.

Comprehensive ablation results, including the full tables and additional design-choice experiments, are reported in Appendix E.5.

## 6 Conclusion

In this paper, we revisited the role of vision tokens in LLM decoding and provided quantitative evidence that they encode *interpretable* semantics that become explicit under semantically coherent vocabulary constraints. Building on these findings, we introduced ReVisiT, a training-free, model-agnostic decoding strategy that references the most relevant vision token via a *context-aware constrained divergence* and refines token logits accordingly. As demonstrated by extensive experiments, ReVisiT consistently improves performance across various state-of-the-art LLMs, demonstrating scalability and broad applicability without architectural modification.

## Limitations

A potential limitation of our method is its strong reliance on the semantics encoded in vision tokens, which may lead to overfitting to visually salient cues. While this behavior improves visual grounding in most cases, it can result in an overemphasis on perceptually grounded elements at the expense of implicit, commonsense, or non-visual contextual information. Although we compare fixed interpolation schemes in Appendix E.5, more adaptive interpolation or confidence-calibration strategies between the reference and vanilla logits remain worth exploring, as a context-dependent fusion rule may better balance visual grounding against non-visual reasoning demands.

Another limitation is its exclusive reliance on the model’s internal representations without access to external knowledge sources. As a result, it cannot correct factual errors or hallucinations originating from pretraining data, particularly in cases where visual evidence alone is insufficient for disambiguation or grounding. While our method improves the alignment between visual inputs and generated text, it does not incorporate retrieval mechanisms that could supplement the model with up-to-date or context-specific information. Nevertheless, due to its modular and decoding-time nature, our method is compatible with retrieval-augmented generation (RAG) pipelines and could be integrated with external knowledge modules in future work.

Finally, our empirical analysis and experiments are conducted on a limited set of open-source LVLMs in the 7–32B parameter range. Due to computational resource constraints, we did not evaluate our method on 70B-scale or larger models, and we therefore do not claim that the observed gains directly transfer to substantially larger or proprietary LVLMs.

## Ethical Considerations

This work relies on publicly available multimodal benchmarks such as HallusionBench, CHAIR, POPE, VQAv2, and MMMU, and on released LVLM checkpoints including LLaVA-1.5-7B, Qwen2.5-VL-7B, and InternVL3-8B. We do not collect new data or interact with human subjects, and we follow the original licensing and usage conditions of all datasets and models.

As with any approach that operates on LVLMs, our method inherits any biases, uneven coverage, or other failure modes present in the underlying mod-

els and training corpora. Since our contribution targets the decoding procedure by improving visual grounding and reducing object hallucinations, it does not introduce new training data or objectives explicitly aimed at mitigating social biases. Potential deployments of LVLMs enhanced with our method, especially in high-stakes or user-facing applications, should therefore include appropriate human oversight and domain-specific safeguards.

## Acknowledgments

All authors are affiliated with the Department of Artificial Intelligence at Yonsei University. This research was supported in part by Institute for Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. RS-2020-II201361, Artificial Intelligence Graduate School Program (Yonsei University); No. RS-2025-02215344, Development of AI Technology with Robust and Flexible Resilience Against Risk Factors; No. RS-2025-25442405, Development of a Self-Learning World Model-Based AGI System for Hyperspectral Imaging).

## References

- Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altschmidt, Sam Altman, Shyamal Anadkat, and 1 others. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.
- Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C. Lawrence Zitnick, and Devi Parikh. 2015. Vqa: Visual question answering. In *International Conference on Computer Vision (ICCV)*.
- Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. 2023. Qwen-vl: A versatile vision-language model for understanding, localization, text reading, and beyond. *arXiv preprint arXiv:2308.12966*.
- Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibao Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, and 1 others. 2025. Qwen2.5-vl technical report. *arXiv preprint arXiv:2502.13923*.
- Zhaorun Chen, Zhuokai Zhao, Hongyin Luo, Huaxiu Yao, Bo Li, and Jiawei Zhou. 2024a. Halc: Object hallucination reduction via adaptive focal-contrast decoding. In *Proceedings of the International Conference on Machine Learning (ICML)*.

- Zhe Chen, Weiyun Wang, Yue Cao, Yangzhou Liu, Zhangwei Gao, Erfei Cui, Jinguo Zhu, Shenglong Ye, Hao Tian, Zhaoyang Liu, and 1 others. 2024b. Expanding performance boundaries of open-source multimodal models with model, data, and test-time scaling. *arXiv preprint arXiv:2412.05271*.
- Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, and 1 others. 2024c. Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. In *Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Yung-Sung Chuang, Yujia Xie, Hongyin Luo, Yoon Kim, James Glass, and Pengcheng He. 2024. Dola: Decoding by contrasting layers improves factuality in large language models. In *International Conference on Learning Representations (ICLR)*.
- Wenliang Dai, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale Fung, and Steven Hoi. 2023. Instructblip: Towards general-purpose vision-language models with instruction tuning. In *Advances in Neural Information Processing Systems (NeurIPS)*.
- Alessandro Favero, Luca Zancato, Matthew Trager, Sidharth Choudhary, Pramuditha Perera, Alessandro Achille, Ashwin Swaminathan, and Stefano Soatto. 2024. Multi-modal hallucination control by visual information grounding. In *Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yixin Dai, Jiawei Sun, Haofen Wang, and Haofen Wang. 2023. Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997*, 2(1).
- Xuan Gong, Tianshi Ming, Xinpeng Wang, and Zhihua Wei. 2024. Damro: Dive into the attention mechanism of llm to reduce object hallucination. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*.
- Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. 2017. Making the v in vqa matter: Elevating the role of image understanding in visual question answering. In *Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Tianrui Guan, Fuxiao Liu, Xiyang Wu, Ruiqi Xian, Zongxia Li, Xiaoyu Liu, Xijun Wang, Lichang Chen, Furong Huang, Yaser Yacoob, and 1 others. 2024. Hallusionbench: an advanced diagnostic suite for entangled language hallucination and visual illusion in large vision-language models. In *Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Qidong Huang, Xiaoyi Dong, Pan Zhang, Bin Wang, Conghui He, Jiaqi Wang, Dahua Lin, Weiming Zhang, and Nenghai Yu. 2024. Opera: Alleviating hallucination in multi-modal large language models via over-trust penalty and retrospection-allocation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Drew A Hudson and Christopher D Manning. 2019. Gqa: A new dataset for real-world visual reasoning and compositional question answering. In *Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Fushuo Huo, Wenchao Xu, Zhong Zhang, Haozhao Wang, Zhicheng Chen, and Peilin Zhao. 2025. Self-introspective decoding: Alleviating hallucinations for large vision-language models. In *International Conference on Learning Representations (ICLR)*.
- Nick Jiang, Anish Kachinthaya, Suzie Petryk, and Yossi Gandelsman. 2025a. Interpreting and editing vision-language representations to mitigate hallucinations. In *International Conference on Learning Representations (ICLR)*.
- Zhangqi Jiang, Junkai Chen, Beier Zhu, Tingjin Luo, Yankun Shen, and Xu Yang. 2025b. Devils in middle layers of large vision-language models: Interpreting, detecting and mitigating object hallucinations via attention lens. In *Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Seil Kang, Jinyeong Kim, Junhyeok Kim, and Seong Jae Hwang. 2025. See what you are told: Visual attention sink in large multimodal models. In *International Conference on Learning Representations (ICLR)*.
- Junho Kim, Hyunjun Kim, Kim Yeonju, and Yong Man Ro. 2024. Code: Contrasting self-generated description to combat hallucination in large multi-modal models. In *Advances in Neural Information Processing Systems (NeurIPS)*.
- Sicong Leng, Hang Zhang, Guanzheng Chen, Xin Li, Shijian Lu, Chunyan Miao, and Lidong Bing. 2024. Mitigating object hallucinations in large vision-language models through visual contrastive decoding. In *Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, and 1 others. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Advances in Neural Information Processing Systems (NeurIPS)*.
- Xiang Lisa Li, Ari Holtzman, Daniel Fried, Percy Liang, Jason Eisner, Tatsunori Hashimoto, Luke Zettlemoyer, and Mike Lewis. 2023a. Contrastive decoding: Open-ended text generation as optimization. In *Annual Meeting of the Association for Computational Linguistics (ACL)*.
- Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Wayne Xin Zhao, and Ji-Rong Wen. 2023b. Evaluating object hallucination in large vision-language models. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*.
- Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár,

- and C Lawrence Zitnick. 2014. Microsoft coco: Common objects in context. In *European Conference on Computer Vision (ECCV)*.
- Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. 2024a. Improved baselines with visual instruction tuning. In *Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023. Visual instruction tuning. In *Advances in Neural Information Processing Systems (NeurIPS)*.
- Shi Liu, Kecheng Zheng, and Wei Chen. 2024b. Paying more attention to image: A training-free method for alleviating hallucination in lvlms. In *European Conference on Computer Vision (ECCV)*.
- Shiyin Lu, Yang Li, Qing-Guo Chen, Zhao Xu, Weihua Luo, Kaifu Zhang, and Han-Jia Ye. 2024. Ovis: Structural embedding alignment for multimodal large language model. *arXiv preprint arXiv:2405.20797*.
- nostalgebraist. 2020. interpreting gpt: the logit lens. <https://www.lesswrong.com/posts/AckRB8wDpdaN6v6ru/interpreting-gpt-the-logit-lens>.
- Anna Rohrbach, Lisa Anne Hendricks, Kaylee Burns, Trevor Darrell, and Kate Saenko. 2018. Object hallucination in image captioning. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*.
- Dustin Schwenk, Apoorv Khandelwal, Christopher Clark, Kenneth Marino, and Roozbeh Mottaghi. 2022. A-okvqa: A benchmark for visual question answering using world knowledge. In *European Conference on Computer Vision (ECCV)*.
- Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, Katie Millican, and 1 others. 2023. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*.
- Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, and 1 others. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.
- Chenxi Wang, Xiang Chen, Ningyu Zhang, Bozhong Tian, Haoming Xu, Shumin Deng, and Huajun Chen. 2025. Mllm can see? dynamic correction decoding for hallucination mitigation. In *International Conference on Learning Representations (ICLR)*.
- Xintong Wang, Jingheng Pan, Liang Ding, and Chris Biemann. 2024. Mitigating hallucinations in large vision-language models with instruction contrastive decoding. *arXiv preprint arXiv:2403.18715*.
- Sangmin Woo, Donguk Kim, Jaehyuk Jang, Yubin Choi, and Changick Kim. 2024. Don't miss the forest for the trees: Attentional vision calibration for large vision language models. *arXiv preprint arXiv:2405.17820*.
- Shukang Yin, Chaoyou Fu, Sirui Zhao, Tong Xu, Hao Wang, Dianbo Sui, Yunhang Shen, Ke Li, Xing Sun, and Enhong Chen. 2024. Woodpecker: Hallucination correction for multimodal large language models. In *Science China Information Sciences (SCIS)*.
- Haoxuan You, Haotian Zhang, Zhe Gan, Xianzhi Du, Bowen Zhang, Zirui Wang, Liangliang Cao, Shih-Fu Chang, and Yinfei Yang. 2024. Ferret: Refer and ground anything anywhere at any granularity. In *International Conference on Learning Representations (ICLR)*.
- Xiang Yue, Yuansheng Ni, Kai Zhang, Tianyu Zheng, Ruoqi Liu, Ge Zhang, Samuel Stevens, Dongfu Jiang, Weiming Ren, Yuxuan Sun, and 1 others. 2024. Mmmu: A massive multi-discipline multimodal understanding and reasoning benchmark for expert agi. In *Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Yiyang Zhou, Chenhang Cui, Jaehong Yoon, Linjun Zhang, Zhun Deng, Chelsea Finn, Mohit Bansal, and Huaxiu Yao. 2023. Analyzing and mitigating object hallucination in large vision-language models. *arXiv preprint arXiv:2310.00754*.
- Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. 2024. Minigt-4: Enhancing vision-language understanding with advanced large language models. In *International Conference on Learning Representations (ICLR)*.
- Jinguo Zhu, Weiyun Wang, Zhe Chen, Zhaoyang Liu, Shenglong Ye, Lixin Gu, Hao Tian, Yuchen Duan, Weijie Su, Jie Shao, and 1 others. 2025. Internv13: Exploring advanced training and test-time recipes for open-source multimodal models. *arXiv preprint arXiv:2504.10479*.

## A Additional Related Works

In this section, we provide an extended discussion of the literature on aligning visual inputs with language generation in LVLMs, expanding upon the taxonomy introduced in Sec. 2. While our primary classification follows the temporal stage of intervention—*pre-alignment*, *intra-alignment*, and *post-alignment*—we also highlight how interpretability-driven projection methods have recently been integrated into these frameworks.

**Pre-alignment.** This line of research focuses on establishing foundational cross-modal mapping during the model’s development phase. Recent state-of-the-art LVLMs refine these mappings by coupling decoder-only backbones with high-resolution vision encoders (Bai et al., 2023; Chen et al., 2024c) or introducing dynamic resolution processing to accommodate diverse aspect ratios (Bai et al., 2025). Other efforts improve fine-grained spatial grounding via hybrid region representations (You et al., 2024) or probabilistic vision tokenization schemes designed to better match language embeddings (Lu et al., 2024). While these methods enhance the model’s inherent capacity, they are static and cannot dynamically correct errors that emerge during the inference process.

**Intra-alignment.** Methods in this category intervene during the decoding process itself to guide the model toward visually grounded outputs. Perturbation-based approaches contrastively compare logits under original and perturbed inputs (e.g., noisy images or instruction disturbances) to suppress hallucinations caused by language priors (Leng et al., 2024; Wang et al., 2024; Favero et al., 2024; Kim et al., 2024). Calibration-based methods instead focus on the internal attention mechanism; they recalibrate how the decoder attends to or weights visual tokens by deriving vision-aware adjustments from learned attention patterns (Gong et al., 2024; Liu et al., 2024b; Woo et al., 2024; Huo et al., 2025; Kang et al., 2025). Notably, Jiang et al. (2025b) further explore this by diagnosing and editing attention patterns in middle layers to mitigate unwanted artifacts during inference. A third family introduces auxiliary visual-sensitive distributions or penalty terms to directly modify decoding rules so that tokens are more strongly grounded in the visual input (Chen et al., 2024a; Huang et al., 2024; Wang et al., 2025).

Across these lines of work, visual information is typically incorporated implicitly through hidden states, logits, or attention scores, leaving the semantic role of individual vision tokens latent. In contrast, ReVisiT makes the textual semantics of individual vision tokens explicit in the language space and leverages their induced distributions as a context-aware constraint, providing an interpretable and effective handle for steering LVLM decoding.

**Post-alignment.** This stage operates on the generated sequence to identify and correct hallucinations after the decoding is complete. Traditional methods use auxiliary revisors or multi-stage verification pipelines conditioned on external VQA models (Zhou et al., 2023; Yin et al., 2024). Recently, interpretability-based techniques have been adopted for diagnostic purpose; for instance, Jiang et al. (2025a) project vision tokens into the language space to diagnose discrepancies between visual perception and generated text as a post-hoc verification signal. While such methods provide a diagnostic signal for post-hoc verification, ReVisiT utilizes similar projection techniques as a *predictive* signal during decoding to prevent hallucinations before they are fully generated.

## B Additional Motivation and Analyses

### B.1 Metric formulation: *at-least-one recall@k*

**Definition.** Given a hidden state  $h$  and the corresponding text token distribution  $p_\theta(h, \mathcal{V})$  over the vocabulary  $\mathcal{V}$  as defined in Eq. (1), we evaluate whether at least one ground-truth (GT) object synonym is ranked within the top- $k$  candidates.

Let  $\mathcal{U}$  denote a collection of evaluation units (e.g., hallucinated decoding steps or vision tokens, depending on the analysis). For each  $u \in \mathcal{U}$ , let  $h^{(u)}$  be the associated hidden state, and let  $G_u \subseteq \mathcal{V}$  be the set of GT object synonyms relevant to  $u$ .

We define the top- $k$  set

$$\text{TopK}_k(p_\theta(h^{(u)}, \mathcal{V})) \subseteq \mathcal{V}$$

as the set of  $k$  tokens with the highest probabilities under  $p_\theta(h^{(u)}, \mathcal{V})$ .

Then, *at-least-one recall@k* over  $\mathcal{U}$  with evaluation vocabulary  $\mathcal{V}$  is defined as

$$\begin{aligned} & \text{at-least-one recall@k}(\mathcal{U}; \mathcal{V}) \\ &= \frac{1}{|\mathcal{U}|} \sum_{u \in \mathcal{U}} \mathbf{1}\{\text{TopK}_k(p_\theta(h^{(u)}, \mathcal{V})) \cap G_u \neq \emptyset\}. \end{aligned} \quad (8)$$

### B.2 Step-level hallucination analysis for

#### Figure 3

**Setup formulation.** CHAIR (Rohrbach et al., 2018) evaluates captions against MS-COCO object annotations under an object-existence criterion. On the 500 captions generated in default greedy setup, CHAIR identifies 275 hallucinated words in total. We then move to step-level analysis, where each hallucinated object mention defines a decoding step (i.e., an evaluation unit).

Among these 275 hallucinated steps, we retain only those in which (i) the hallucinated word and (ii) at least one ground-truth (GT) object synonym for the same image are both tokenized as a single token by the tokenizer. This restriction allows each retained object label to be aligned directly with a specific token index, yielding  $n = 190$  hallucinated decoding steps for step-level analysis.

In the notation of Appendix B.1, we instantiate  $\mathcal{U}$  as this set of hallucinated decoding steps and  $\mathcal{V}$  as the full vocabulary, and compute *at-least-one recall@k*( $\mathcal{U}; \mathcal{V}$ ) with  $|\mathcal{U}| = 190$ .

**Reporting.** Table 6 reports the resulting recall values with 95% Wilson score intervals.

Table 6: **Step-level GT recall at hallucinated positions.** At-least-one GT recall at hallucinated steps ( $n = 190$ ) with 95% Wilson intervals.

Top- $k$	% of vocab	<i>at-least-one recall@k</i> [95% CI]
Top-50	0.033%	63.2% [56.0, 70.0]
Top-100	0.066%	73.7% [66.9, 79.6]
Top-500	0.33%	95.8% [91.6, 98.1]

### B.3 Paired GT vs. hallucinated probabilities for Table 1

**Setup formulation.** Given an output hidden state  $h_{T+t-1}^L$  from the last decoder layer  $L$  at decoding step  $t$  and the corresponding text token distribution  $p_\theta(h_{T+t-1}^L, \mathcal{V})$  over the vocabulary  $\mathcal{V}$  as defined in Eq. (1), we compare how much probability mass is assigned to ground-truth versus hallucinated objects.

For CHAIR, let  $\mathcal{V}_{\text{coco}}$  denote the MS-COCO synonym set ( $|\mathcal{V}_{\text{coco}}| = 403$ ). To obtain unambiguous token-object correspondence and avoid multi-token variants, we construct the single-token subset

$$\mathcal{V}_{\text{obj}} = \{w \in \mathcal{V}_{\text{coco}} : \text{len}(\text{tokenize}("_w")) = 1\}, \quad (9)$$

which contains 176 word types covering 62 object categories.

At decoding position  $t$ , we define the CHAIR-based ground-truth and hallucinated synonym subsets over  $\mathcal{V}_{\text{obj}}$  as  $\mathcal{G}_t \subseteq \mathcal{V}_{\text{obj}}$  and  $\mathcal{H}_t = \mathcal{V}_{\text{obj}} \setminus \mathcal{G}_t$ , respectively. For step-level analyses, we further require that (i) the hallucinated word at  $t$  is single-tokenizable and (ii)  $\mathcal{G}_t \neq \emptyset$ , yielding  $n = 190$  hallucinated decoding steps (selection details match Appendix B.2). We evaluate two conditions: *w/ image* (vision tokens present) and *w/o image* (all vision tokens removed); in the latter, decoding positions are aligned by teacher forcing with the previously generated caption while keeping all other decoding details identical.

For each decoding position  $t$ , we operate directly on the object-only distribution  $p_\theta(h_{T+t-1}^L, \mathcal{V}_{\text{obj}})$ . Let

$$m_t^{\text{GT}} = \max_{w \in \mathcal{G}_t} p_\theta(w | h_{T+t-1}^L, \mathcal{V}_{\text{obj}}),$$

$$m_t^{\text{Hal}} = \max_{w \in \mathcal{H}_t} p_\theta(w | h_{T+t-1}^L, \mathcal{V}_{\text{obj}}),$$

and define the corresponding quantities  $m_t^{\text{GT}, w/o}$  and  $m_t^{\text{Hal}, w/o}$  under the w/o-image condition. Table 1 reports the averages of these per-step maxima of object probabilities.

Table 7: **Wilcoxon signed-rank comparison of GT vs. hallucinated scores.**  $\Delta_t = (\text{GT} - \text{Hal})_{\text{with}} - (\text{GT} - \text{Hal})_{\text{w/o}}$ .  $N$  is the number of paired decoding positions, and Holm-adjusted  $p$ -values are reported for  $\Delta_t$ .

Setting	$N$	median( $d^{\text{with}}$ )	median( $d^{\text{w/o}}$ )	median( $\Delta$ )	$p_{\text{Holm}}$ on $\Delta$
All tokens	88,706	-3.289	-4.826	0.715	$< 10^{-300}$
All tokens (in hallucinated set)	35,116	-3.720	-5.273	0.774	$1.09 \times 10^{-116}$
Hallucinated steps	190	-8.281	-9.435	-0.180	0.95

**Statistical test and reporting.** For paired comparisons, we form the step-level differences

$$\begin{aligned} d_t^{\text{with}} &= m_t^{\text{GT}} - m_t^{\text{Hal}}, \\ d_t^{\text{w/o}} &= m_t^{\text{GT,w/o}} - m_t^{\text{Hal,w/o}}, \\ \Delta_t &= d_t^{\text{with}} - d_t^{\text{w/o}}. \end{aligned}$$

We apply the Wilcoxon signed-rank test (two-sided; Pratt) to  $d_t^{\text{with}}$ ,  $d_t^{\text{w/o}}$ , and  $\Delta_t$  on the intersection of decoding positions available in both conditions. We report the paired sample size  $N$ , the median of each quantity, and Holm-adjusted  $p$ -values for  $\Delta_t$ .

In the full-token analyses, the median  $\Delta$  is positive and highly significant, indicating a systematic shift in the step-level difference  $m_t^{\text{GT}} - m_t^{\text{Hal}}$  toward GT when vision tokens are present. For the hallucinated-step subset ( $n=190$ ), the median  $\Delta$  is small and not significant, i.e., the reversed ordering (Hal > GT) persists at those positions despite the image.

#### B.4 Vision-token projection analysis for Figure 4

**Setup formulation.** We focus on the final decoder layer ( $L$ ) and treat each vision token as an evaluation unit. Let the 500 MS-COCO validation images be indexed by  $n \in \{1, \dots, 500\}$ . For image  $n$ , let  $\mathbf{v}^{(n)} = (v_1^{(n)}, \dots, v_{|\mathbf{v}^{(n)}|}^{(n)})$  denote the sequence of vision tokens, and let  $h_i^{L,(n)}$  be the corresponding final-layer hidden state of  $v_i^{(n)}$ .

We define the evaluation set  $\mathcal{U}$  as the collection of all vision tokens across the 500 images,

$$\mathcal{U} = \{(n, i) : n \in \{1, \dots, 500\}, i \in \{1, \dots, |\mathbf{v}^{(n)}|\}\},$$

and denote its size by

$$|\mathcal{U}| = N = \sum_{n=1}^{500} |\mathbf{v}^{(n)}| = 173,173.$$

For each unit  $u = (n, i) \in \mathcal{U}$ , we write the associated hidden state as  $h^{(u)} = h_i^{L,(n)}$ .

Passing  $h^{(u)}$  through the language-modeling head yields a text token distribution over the full vocabulary  $\mathcal{V}$ ,  $p_\theta(h^{(u)}, \mathcal{V})$ . Additionally, we reuse  $\mathcal{V}_{\text{obj}}$ , the single-token MS-COCO synonym subset defined in Eq. (9) (176 word types covering 62 object categories), and obtain  $p_\theta(h^{(u)}, \mathcal{V}_{\text{obj}})$  by applying softmax over  $\mathcal{V}_{\text{obj}}$ .

For each vision-token unit  $u \in \mathcal{U}$ , we construct a GT synonym set  $G_u$  from the MS-COCO categories annotated in the same image as  $h^{(u)}$ , restricted to the 62 categories covered by  $\mathcal{V}_{\text{obj}}$ . In the notation of Appendix B.1, we thus instantiate  $\mathcal{U}$  as the set of all  $N$  vision tokens and consider two evaluation vocabularies,  $\mathcal{V}$  and  $\mathcal{V}_{\text{obj}}$ . We then compute at-least-one recall@ $k(\mathcal{U}; \mathcal{V})$  and at-least-one recall@ $k(\mathcal{U}; \mathcal{V}_{\text{obj}})$  according to Eq. (8).

**Reporting.** Table 8 summarizes token-level at-least-one recall@ $k(\mathcal{U}; \mathcal{V})$  and at-least-one recall@ $k(\mathcal{U}; \mathcal{V}_{\text{obj}})$  over  $|\mathcal{U}| = N = 173,173$  vision-token projections, together with 95% Wilson score intervals.

Table 8: **Vision-token at-least-one GT recall.** At-least-one GT recall over  $N=173,173$  vision-token projections with 95% Wilson score intervals. GT targets are restricted to the same 62 categories in both settings.

Top- $k$	at-least-one recall@ $k(\mathcal{V})$	at-least-one recall@ $k(\mathcal{V}_{\text{obj}})$
Top-1	2.03% [1.96, 2.10]	40.44% [40.21, 40.67]
Top-5	7.87% [7.74, 8.00]	69.56% [69.34, 69.78]
Top-10	12.38% [12.23, 12.54]	79.78% [79.59, 79.97]
Top-30	21.72% [21.53, 21.91]	89.24% [89.09, 89.39]

## C Additional Method Details: ReVisiT Decoding

### C.1 Step-by-step decoding overview with Figure 5

To complement the formal description in Sec. 4, we present a step-by-step overview of ReVisiT and its pseudo-code implementation. Fig. 5 illustrates a single decoding step on a representative example, while Alg. 1 summarizes the full procedure over the entire decoding trajectory. ReVisiT operates purely at decoding time: the LVLm architecture and parameters are kept fixed, and only the decoding function is modified. By caching vision-token logits once and manipulating them through slicing and masking, our implementation is memory-efficient, model-agnostic, and requires no additional forward passes or training.

At a high level, ReVisiT modifies the vanilla next-token distribution only when vision tokens provide a more semantically consistent signal. Given the fruit image and the textual query “How many uncut fruits are in the image?” from the LLaVA-Bench-In-the-Wild dataset (Liu et al., 2023), the LVLm produces at each decoding step  $t$  a vanilla distribution  $p_\theta(h_{T+t-1}^L, \mathcal{V})$  over the full vocabulary (Eq. (1)). At  $t = 0$  and  $t = 1$ , the model generates “There” and “are”, and ReVisiT leaves these predictions unchanged because the vanilla distribution is already highly confident. At  $t = 2$ , however, the vanilla distribution erroneously favors “four” over the correct answer “three”.

Concretely, in step (1), the vanilla distribution is restricted to a context-aware constrained subset  $\mathcal{V}_{\text{cons}}^t$  via the adaptive plausibility constraint in Eq. (3). All subsequent operations are performed on this subset, and tokens outside  $\mathcal{V}_{\text{cons}}^t$  are effectively removed from consideration.

In step (2), for each candidate vision token hidden state  $h_i^j$  from layer  $j \in \mathcal{J}$ , we obtain its projected distribution over  $\mathcal{V}_{\text{cons}}^t$ ,  $p_\theta(h_i^j, \mathcal{V}_{\text{cons}}^t)$ , as defined in Eq. (4). We then measure the Jensen–Shannon divergence between  $p_\theta(h_{T+t-1}^L, \mathcal{V}_{\text{cons}}^t)$  and each  $p_\theta(h_i^j, \mathcal{V}_{\text{cons}}^t)$ , and select the pair  $(i^*, j^*)$  with the smallest divergence as the most relevant vision token for the current context.

In step (3), the vanilla distribution and the selected vision-token distribution are fused. For tokens in  $\mathcal{V}_{\text{cons}}^t$ , the final distribution  $p_{\text{fin}}$  is obtained via log-space addition (Eq. (6)) followed by normalization over the constrained set (Eq. (7)); for

tokens outside  $\mathcal{V}_{\text{cons}}^t$ , the probability mass is set to zero. The next token  $y_t$  is then sampled from  $p_{\text{fin}}$ . In the example of Fig. 5, this refinement step shifts the decision from “four” to the visually grounded answer “three”. Appendix F.1 provides the exact token-level probabilities underlying this example, together with additional case studies.

### C.2 Theoretical view of logit refinement

The refinement rule in Sec. 4.3 (Eq. (6)) admits a simple KL-regularized interpretation. Let  $p_{\text{output}}(y)$  denote the vanilla next-token distribution on the constrained candidate set  $\mathcal{V}_{\text{cons}}^t$ , and let  $p_{\text{vision}}(y)$  denote the selected vision-token distribution on the same set. We consider the objective

$$q^* = \arg \max_{q \in \Delta} \mathbb{E}_{y \sim q} [\log p_{\text{vision}}(y)] - \gamma \text{KL}(q \parallel p_{\text{output}}),$$

which encourages the refined distribution to assign higher mass to tokens preferred by the vision-induced reference while remaining close to the model’s original distribution. This objective yields the closed-form solution

$$q^*(y) \propto p_{\text{output}}(y) p_{\text{vision}}(y)^{1/\gamma}.$$

Hence, the proposed refinement is equivalent to a Product-of-Experts (PoE)-style combination in probability space, or additive fusion in log space. The default rule in the main text corresponds to the equal-strength case (i.e.,  $\gamma = 1$ ).

---

**Algorithm 1** ReVisiT decoding algorithm

---

**Input:** LVLM  $\mathcal{M}$  with final decoder layer  $L$ , projection (language modeling) head  $W_{\text{proj}}$ , and vocabulary  $\mathcal{V}$ , input image  $I$ , set of candidate decoder layers  $\mathcal{J}$ , input text prompt  $x_{\text{raw}}$

---

Tokenize input text:  $\mathbf{x} = \{x_0, \dots, x_{|\mathbf{x}|-1}\} \leftarrow \text{Tokenizer}(x_{\text{raw}})$

Encode image:  $\mathbf{v} = \{v_0, \dots, v_{|\mathbf{v}|-1}\} \leftarrow \text{VisionEncoder}(I)$

Construct input sequence:  $\mathbf{z} = \mathbf{v} \parallel \mathbf{x} \quad \triangleright \mathbf{z} = \{z_0, \dots, z_{T-1}\}$  where  $T = |\mathbf{v}| + |\mathbf{x}|$

Initialize output sequence  $\mathbf{y} \leftarrow \emptyset \quad \triangleright t \leftarrow 0$

**while** not end-of-sequence **do**

    Get decoder hidden states  $h_{T+t-1}^L$  from  $\mathcal{M}$  with context  $(\mathbf{z}, \mathbf{y})$

**if** first-timestep **then**

        Given  $\{h_i^j\}$  for  $i = 0, \dots, T-1$  and  $j \in \mathcal{J}$  from initial forward pass

        Initialize cache  $\mathbf{l}_{\text{vision}} \leftarrow \emptyset$

**for** each layer  $j \in \mathcal{J}$  **do**

**for**  $i = 0$  to  $|\mathbf{v}| - 1$  **do**

$l_i^j = W_{\text{proj}}^\top h_i^j$

                Append  $l_i^j$  to  $\mathbf{l}_{\text{vision}}$

**end for**

**end for**

        Cache matrix  $\mathbf{l}_{\text{vision}} \in \mathbb{R}^{|\mathcal{J}| \cdot |\mathbf{v}| \times |\mathcal{V}|}$

        first-timestep  $\leftarrow$  False

**end if**

    Compute vanilla distribution  $p_{\text{base}} = \text{softmax}_{\mathcal{V}}(W_{\text{proj}}^\top h_{T+t-1}^L)$  (Eq. (1))

    Select constrained vocabulary  $\mathcal{V}_{\text{cons}}^t \subseteq \mathcal{V}$  based on  $p_{\text{base}}$  (Eq. (3))

    Let  $p_{\text{base}}^{\text{cons}} = \text{softmax}_{\mathcal{V}_{\text{cons}}^t}(W_{\text{proj}}^\top h_{T+t-1}^L)$

$(j^*, i^*) \leftarrow \arg \min_{(j,i)} \text{JSD}(p_{\text{base}}^{\text{cons}} \parallel \text{softmax}_{\mathcal{V}_{\text{cons}}^t}(\mathbf{l}_{\text{vision}}[j, i]))$

    Let  $p_{\text{ref}}^{\text{cons}} = \text{softmax}_{\mathcal{V}_{\text{cons}}^t}(\mathbf{l}_{\text{vision}}[j^*, i^*])$

    Logit adjustment via log-space addition:  $l_{\text{ReVisiT}} = \log(p_{\text{base}}^{\text{cons}}) + \log(p_{\text{ref}}^{\text{cons}})$

    Normalize over constrained vocabulary:  $p_{\text{ReVisiT}} = \text{softmax}_{\mathcal{V}_{\text{cons}}^t}(l_{\text{ReVisiT}})$

    Sample next token:  $y_t \sim p_{\text{ReVisiT}}$

    Append  $y_t$  to  $\mathbf{y}$

$t \leftarrow t + 1$

**end while**

**return**  $\mathbf{y}$

---

## D Additional Experimental Details

### D.1 Dataset and evaluation setup

**HallusionBench evaluation.** HallusionBench (Guan et al., 2024) is a diagnostic benchmark for image-context reasoning that disentangles *Language Hallucination* (over-reliance on parametric priors) and *Visual Illusion* (misinterpretation of visual inputs). All questions are binary (Yes/No) and organized into *control pairs* to test consistency across visual contexts. We focus on the *Visual Dependent (VD)* split, which comprises 591 Yes/No questions that *require* the image to answer; each figure consists of an original image and one or more human-edited variants (e.g., flipping, order reversing, masking, optical-character editing, object editing, color editing) to stress visual robustness. We use the official data and evaluation scripts.<sup>5</sup>

*Protocol.* Following the official setup, we adopt the *text-only GPT-4-assisted* judging scheme. Let figures be indexed by  $i \in \mathbb{I}$ . For each figure  $i$ , let  $\{I_{i,j}\}_{j \in \mathcal{J}_i}$  denote its image variants with  $j = 0$  the original and  $j \geq 1$  edited versions, and let  $\{q_{i,k}\}_{k \in \mathcal{K}_i}$  be the associated Yes/No questions (shared across variants). Given a model’s raw response  $\mathcal{M}(I_{i,j}, q_{i,k})$ , the judge (GPT-4, temperature 0) receives a textual prompt comprising  $\{q_{i,k}$  (question),  $y_{i,j,k}$  (reference answer),  $\mathcal{M}(I_{i,j}, q_{i,k})$  (model response) $\}$  and returns a label in  $\{\text{Correct, Incorrect, Uncertain}\}$ . We binarize correctness as

$$b_{\mathcal{M}}(I_{i,j}, q_{i,k}) = \begin{cases} 1, & \text{if the judge returns Correct,} \\ 0, & \text{if the judge returns Incorrect or Uncertain.} \end{cases}$$

*Metrics.* Let  $\mathcal{T} = \{(i, j, k) : i \in \mathbb{I}, j \in \mathcal{J}_i, k \in \mathcal{K}_i\}$  be all VD image–question triples, and let  $\mathcal{Q} = \{(i, k) : i \in \mathbb{I}, k \in \mathcal{K}_i\}$  be the set of control *question pairs*. We report the standard HallusionBench accuracies:

$$\begin{aligned} \text{aAcc} &= \frac{1}{|\mathcal{T}|} \sum_{(i,j,k) \in \mathcal{T}} b_{\mathcal{M}}(I_{i,j}, q_{i,k}), \\ \text{qAcc} &= \frac{1}{|\mathcal{Q}|} \sum_{(i,k) \in \mathcal{Q}} \mathbf{1} \left\{ \bigwedge_{j \in \mathcal{J}_i} b_{\mathcal{M}}(I_{i,j}, q_{i,k}) = 1 \right\}, \\ \text{fAcc} &= \frac{1}{\sum_{i \in \mathbb{I}} |\mathcal{J}_i|} \sum_{i \in \mathbb{I}} \sum_{j \in \mathcal{J}_i} \mathbf{1} \left\{ \bigwedge_{k \in \mathcal{K}_i} b_{\mathcal{M}}(I_{i,j}, q_{i,k}) = 1 \right\}. \end{aligned}$$

Here, **aAcc** is *accuracy per question* (overall), **qAcc** is *accuracy per question pair* (the same

<sup>5</sup><https://github.com/tianyi-lab/HallusionBench>

question must be correct across the figure’s image variants), and **fAcc** is *accuracy per image* (all questions assigned to a given image—original or edited—must be correct). Following the leaderboard convention, we additionally report **Easy aAcc** (restricted to original images,  $j=0$ ) and **Hard aAcc** (restricted to edited images,  $j \geq 1$ ).

**CHAIR evaluation.** CHAIR (Rohrbach et al., 2018) is a generative benchmark designed to measure object hallucinations in image captioning. A hallucinated object refers to any entity mentioned in the generated caption that is not present in the corresponding image. This metric is widely adopted to evaluate the visual grounding capability of vision-language models.

*Protocol.* We follow the standard CHAIR evaluation procedure based on the MS-COCO annotations and the official implementation.<sup>6</sup> Each image is associated with a set of ground-truth objects defined by the 80 MS-COCO detection categories. During evaluation, noun phrases extracted from generated captions are matched against this predefined object list; mentions of out-of-vocabulary objects are ignored, while in-vocabulary objects absent from the ground-truth annotations are considered hallucinated. We evaluate on 500 images randomly sampled from the COCO validation set with a fixed seed (seed= 42) for reproducibility, prompting each image with “Please describe this image in detail.”

*Metrics.* We report the two standard CHAIR metrics and an additional **F1**:

$$\begin{aligned} \text{CHAIR}_S &= \frac{|\{\text{hallucinated captions}\}|}{|\{\text{total captions}\}|}, \\ \text{CHAIR}_I &= \frac{|\{\text{hallucinated object mentions}\}|}{|\{\text{total object mentions}\}|}, \\ \text{F1} &= \frac{\sum_{\text{images}} \text{F1}_{\text{sample}}}{|\{\text{total captions}\}|}, \\ \text{where } \text{F1}_{\text{sample}} &= \begin{cases} \frac{2 \text{Prec} \cdot \text{Rec}}{\text{Prec} + \text{Rec}}, & \text{if Prec} + \text{Rec} > 0, \\ 0, & \text{otherwise,} \end{cases} \end{aligned}$$

$$\begin{aligned} \text{with } \text{Prec} &= \frac{|\{\text{predicted objects that are in GT}\}|}{|\{\text{predicted objects}\}|} \\ &\quad (\text{duplicates in predictions counted}), \\ \text{Rec} &= \frac{|\{\text{GT objects mentioned at least once}\}|}{|\{\text{GT objects}\}|} \\ &\quad (\text{GT is a set; no duplicates}). \end{aligned}$$

**POPE evaluation.** POPE (Li et al., 2023b) is a discriminative benchmark designed to evaluate

<sup>6</sup><https://github.com/LisaAnne/Hallucination>

the object-level visual grounding capabilities of LVLMs.

*Protocol.* It formulates the task as a binary object presence question-answering problem, where the model is asked to determine whether a specific object is present in a given image. The benchmark covers three datasets: MS-COCO (Lin et al., 2014), A-OKVQA (Schwenk et al., 2022), and GQA (Hudson and Manning, 2019). For each dataset, POPE defines three types of query strategies—*random*, *popular*, and *adversarial*—yielding a total of nine evaluation scenarios. Binary questions are constructed in the form “Is there a <object> in the image?”, with the queried object selected according to the designated strategy. Random queries are uniformly sampled from the dataset’s object vocabulary. Popular queries focus on frequently occurring objects, while adversarial queries target objects that are semantically plausible in the given context but are absent from the image, thus probing the model’s reliance on prior co-occurrence statistics rather than visual evidence. Ground-truth labels indicating object presence are provided by the official POPE benchmark.<sup>7</sup> For MS-COCO, labels are derived from object detection annotations, whereas for A-OKVQA and GQA, they are obtained via SEEM-based segmentation annotations. Each query type consists of 3,000 binary QA instances, evenly split between positive and negative samples, resulting in a total of 27,000 QA examples.

*Metrics.* Evaluation is conducted using standard classification metrics derived from the confusion matrix, including accuracy, precision, and F1 score. Accuracy measures the overall proportion of correct predictions, while precision captures the fraction of predicted positive instances that are indeed correct. F1 score, defined as the harmonic mean of precision and recall, provides a balanced assessment of model performance on this binary classification task.

**VQAv2 evaluation.** VQAv2 (Goyal et al., 2017) is a large-scale open-ended visual question answering benchmark where each question is annotated with 10 human answers.

*Protocol.* We use the official evaluation protocol and code for computing agreement with the 10 human annotations.<sup>8</sup> Since our models are prompted in a free-form manner and may gener-

ate short phrases or sentences rather than a single word, we first tokenize each generated answer into a sequence of whitespace-separated tokens  $(w_1, \dots, w_T)$ . For each token  $w_t$ , we query the official VQAv2 scoring function  $\text{Acc}(w_t)$ , and define the per-question score as the maximum token-level score  $\max_t \text{Acc}(w_t)$ . This procedure allows us to fairly evaluate free-form answers while remaining compatible with the official scoring rule.

*Metrics.* Concretely, for any candidate answer string  $ans$ , the VQAv2 accuracy is

$$\text{Acc}(ans) = \min \left\{ \frac{\# \text{ of humans that said } ans}{3}, 1 \right\},$$

where the numerator counts how many of the 10 annotators produced  $ans$ . Given a model prediction tokenized as  $(w_1, \dots, w_T)$ , the per-question accuracy used in our experiments is

$$\text{Acc}_{\text{question}} = \max_{t \in \{1, \dots, T\}} \text{Acc}(w_t),$$

and the dataset accuracy is the mean of  $\text{Acc}_{\text{question}}$  over all questions.

**MMMU evaluation.** MMMU (Yue et al., 2024) is a large-scale, multi-discipline VQA benchmark that evaluates college-level multimodal understanding and reasoning across six disciplines, 30 subjects, and heterogeneous image types (charts, diagrams, tables, maps, music sheets, chemical structures, etc.). We use the validation split for controlled comparisons.

*Protocol.* The validation split includes open-ended and multiple-choice (MC) questions. Since our LLaVA-1.5 configuration supports only single-image inputs, we exclude the 43 multi-image items from the 900 validation questions and report results on the remaining 857 items. For fairness, all methods are evaluated on this filtered subset. Following the official setup,<sup>9</sup> we append instruction postfixes: for open-ended questions we add “Answer the question using a single word or phrase.”; for MC questions we add “Answer with the option’s letter from the given choices directly.” to adopt official parsing tool.

*Metrics.* Let  $\mathcal{S}$  be the set of 30 subjects and let  $\mathcal{D}_s$  denote the items belonging to subject  $s \in \mathcal{S}$  after filtering. For each subject  $s$ , we compute the subject-wise accuracy

$$\text{Acc}_{\mathcal{M}}(s) = \frac{1}{|\mathcal{D}_s|} \sum_{x \in \mathcal{D}_s} \mathbf{1}\{\mathcal{M}(x) = y(x)\},$$

<sup>7</sup><https://github.com/AoiDragon/POPE>

<sup>8</sup><https://visualqa.org/evaluation.html>

<sup>9</sup><https://github.com/MMMU-Benchmark/MMMU>

where  $y(x)$  is the ground-truth answer and  $\mathcal{M}(x)$  is the model prediction. We then report the macro-averaged accuracy over subjects,

$$\text{Accuracy} = \frac{1}{|\mathcal{S}|} \sum_{s \in \mathcal{S}} \text{Acc}_{\mathcal{M}}(s),$$

which follows the official subject-level evaluation protocol.

**LLaVA-Bench-In-the-Wild dataset.** We additionally employ the LLaVA-Bench-In-the-Wild (Liu et al., 2023) dataset to qualitatively evaluate the visual grounding ability of LVLMs in open-ended image understanding. The dataset consists of 24 diverse images, including both real-world photographs and abstract illustrations, and 60 textual prompts designed to elicit complex visual reasoning and language generation. Following prior work (Leng et al., 2024), we present representative qualitative examples to illustrate how different decoding strategies affect grounding fidelity and response quality in challenging scenarios.

## D.2 Implementation details

**Model setup.** All LVLMs are used with their official checkpoints and default inference configurations, including precision (e.g., fp16/bf16), attention mechanism, and tokenizer settings, unless otherwise specified.

**Baseline decoding methods.** For all decoding baselines, we primarily adopt the official inference code and default hyperparameters released by the authors. When an official implementation for a given LVLM backbone is unavailable or cannot be directly applied, we instead follow the configuration described in the original paper. In particular, for SID (Huo et al., 2025) we follow the official LLaVA implementation, which uses a fixed budget of visual tokens, and for LVLMs whose number of visual tokens varies with the input resolution we select approximately 10% of the visual tokens per image, consistent with the paper description.

**ReVisiT hyperparameters.** ReVisiT involves two hyperparameters: (1) the pool of decoder layer(s) from which vision tokens are selected and (2) the alignment weight  $\alpha$  that determines the size of the constrained vocabulary subset. For LLaVA-1.5-7B and InternVL3-8B, we select vision tokens from the final decoder layer (last) and

fix this choice across all benchmarks. For Qwen2.5-VL-7B, aggregating vision tokens from all even-numbered decoder layers (all) consistently performs better, and we use this configuration for all tasks. The threshold  $\alpha$  controls the tightness of the constrained vocabulary subset  $\mathcal{V}_{\text{cons}}^t$ : smaller  $\alpha$  produces a looser subset with higher candidate diversity, whereas larger  $\alpha$  produces a tighter subset that suppresses distractors more aggressively. Empirically, generative captioning tends to benefit from relatively looser subsets to preserve grounded coverage, while short-answer or discriminative settings often benefit from tighter subsets to improve precision. We select  $\alpha$  from a logarithmic grid between  $10^{-1}$  and  $10^{-6}$  on a small validation split; when task-specific tuning is not required, we use  $\alpha = 10^{-5}$  as the default.

## E Detailed Quantitative Results

### E.1 Comprehensive results across eight models

Table 9: **Comprehensive CHAIR results across various model sizes and architectures.** Lower scores ( $\downarrow$ ) on CHAIR<sub>S</sub>, CHAIR<sub>I</sub> and higher ( $\uparrow$ ) F1 indicate better performance. The best results in each model are **bolded**.

Model	Method	CHAIR <sub>S</sub> $\downarrow$	CHAIR <sub>I</sub> $\downarrow$	F1 $\uparrow$
LLaVA-1.5-7B (Liu et al., 2024a)	Greedy	53.8	14.66	82.33
	<b>Ours</b>	<b>50.6</b>	<b>13.43</b>	<b>83.17</b>
Qwen2.5-VL-7B (Bai et al., 2025)	Greedy	35.2	8.43	79.85
	<b>Ours</b>	<b>29.8</b>	<b>7.04</b>	<b>81.16</b>
InternVL2.5-8B (Chen et al., 2024b)	Greedy	33.4	8.47	80.58
	<b>Ours</b>	<b>29.8</b>	<b>7.81</b>	<b>81.93</b>
InternVL3-8B (Zhu et al., 2025)	Greedy	29.4	7.91	81.11
	<b>Ours</b>	<b>27.6</b>	<b>7.27</b>	<b>82.29</b>
LLaVA-1.5-13B (Liu et al., 2024a)	Greedy	<b>48.0</b>	13.29	83.27
	<b>Ours</b>	52.6	<b>13.25</b>	<b>83.50</b>
InternVL3-14B (Zhu et al., 2025)	Greedy	33.2	9.34	81.49
	<b>Ours</b>	<b>33.0</b>	<b>8.66</b>	<b>82.03</b>
InternVL2.5-26B (Chen et al., 2024b)	Greedy	31.8	8.39	79.60
	<b>Ours</b>	<b>30.4</b>	<b>7.42</b>	<b>80.71</b>
Qwen2.5-VL-32B (Bai et al., 2025)	Greedy	53.6	8.85	82.00
	<b>Ours</b>	<b>53.4</b>	<b>8.5</b>	<b>82.55</b>

## E.2 Comprehensive POPE results

Table 10: **Comprehensive results on POPE benchmark** across datasets (MS-COCO, A-OKVQA, GQA) and setups (Random / Popular / Adversarial). Higher scores ( $\uparrow$ ) on Accuracy, Precision, and F1 indicate better performance.

Dataset	Setup	Method	LLaVA-1.5-7B			Qwen2.5-VL-7B			InternVL3-8B			
			Acc $\uparrow$	Prec $\uparrow$	F1 $\uparrow$	Acc $\uparrow$	Prec $\uparrow$	F1 $\uparrow$	Acc $\uparrow$	Prec $\uparrow$	F1 $\uparrow$	
MS-COCO	Random	Greedy	89.07	89.54	89.00	84.07	99.71	81.09	94.53	97.31	94.37	
		DoLa	89.00	89.58	88.92	81.53	<b>99.89</b>	77.37	91.17	<b>98.43</b>	90.45	
		VCD	87.23	86.62	87.34	<u>84.57</u>	99.15	<u>81.88</u>	93.00	95.62	92.79	
		M3ID	89.07	89.54	<u>89.00</u>	83.93	99.32	80.96	94.33	97.30	94.15	
		CODE	86.53	87.23	86.41	84.00	98.02	81.26	93.70	94.50	93.64	
		SID	<b>89.67</b>	<b>91.49</b>	<b>89.43</b>	<b>86.93</b>	99.03	<b>85.10</b>	92.97	95.48	92.77	
		<b>Ours</b>	<u>89.10</u>	<u>89.82</u>	<u>89.00</u>	<u>84.30</u>	<u>99.71</u>	<u>81.42</u>	<b>94.57</b>	<u>97.18</u>	<b>94.41</b>	
	Popular	Greedy	85.63	83.72	86.03	83.47	97.99	80.52	<u>90.57</u>	89.75	90.66	
		DoLa	85.63	83.85	86.00	81.10	<b>98.54</b>	76.96	89.20	<b>94.08</b>	88.57	
		VCD	83.53	80.75	84.25	<u>83.93</u>	97.39	<u>81.27</u>	88.57	87.39	88.74	
		M3ID	85.63	83.72	86.03	83.40	97.81	80.46	90.43	89.82	90.51	
		CODE	84.73	84.14	84.86	83.63	97.02	80.92	88.47	85.40	88.95	
		SID	<b>86.13</b>	<b>85.19</b>	<b>86.32</b>	<b>86.23</b>	97.22	<b>84.42</b>	88.53	87.29	88.72	
		<b>Ours</b>	<u>85.80</u>	<u>84.16</u>	<u>86.13</u>	<u>83.70</u>	<u>98.01</u>	<u>80.85</u>	<b>90.67</b>	<u>89.87</u>	<b>90.76</b>	
	Adversarial	Greedy	79.27	74.72	81.01	83.10	97.06	80.16	87.10	84.16	<u>87.63</u>	
		DoLa	79.40	74.97	81.08	80.83	<b>97.73</b>	76.71	<b>87.30</b>	<b>90.34</b>	86.80	
		VCD	77.67	72.90	79.77	<u>83.37</u>	95.96	<u>80.73</u>	86.13	83.33	86.69	
		M3ID	79.27	74.72	81.01	83.00	96.79	80.06	87.03	<u>84.31</u>	87.53	
		CODE	78.67	75.23	80.02	83.00	95.41	80.31	85.00	80.24	86.09	
		SID	<b>80.30</b>	<b>76.56</b>	<b>81.59</b>	<b>85.17</b>	94.74	<b>83.39</b>	85.63	82.61	86.27	
		<b>Ours</b>	<u>79.90</u>	<u>76.21</u>	<u>81.22</u>	<u>83.37</u>	<u>97.08</u>	<u>80.53</u>	<u>87.13</u>	<u>84.05</u>	<b>87.69</b>	
	A-OKVQA	Random	Greedy	86.33	81.00	87.42	87.80	96.86	86.49	94.00	<u>93.25</u>	<u>94.05</u>
			DoLa	86.43	81.25	87.47	85.33	<b>96.99</b>	83.26	91.10	<b>95.50</b>	90.65
			VCD	83.70	77.91	85.23	87.87	96.56	86.62	90.17	88.65	90.36
			M3ID	86.33	81.00	87.42	87.40	95.91	86.11	94.00	<u>93.25</u>	<u>94.05</u>
			CODE	83.80	78.48	85.18	87.50	93.64	86.55	91.20	87.50	91.61
			SID	<b>87.57</b>	<u>83.48</u>	<b>88.28</b>	<b>88.83</b>	96.12	<b>87.88</b>	89.70	87.81	89.95
<b>Ours</b>			<u>87.37</u>	<u>84.75</u>	<u>87.83</u>	<u>88.27</u>	<u>96.90</u>	<u>87.08</u>	<b>94.17</b>	<u>93.16</u>	<b>94.23</b>	
Popular		Greedy	78.77	71.74	81.72	85.93	<b>92.58</b>	84.74	<u>89.67</u>	85.93	90.18	
		DoLa	79.00	72.06	81.85	83.37	92.17	81.43	89.23	<b>91.71</b>	88.90	
		VCD	76.47	69.58	79.99	85.90	92.10	84.78	81.90	76.48	83.58	
		M3ID	78.77	71.74	81.72	85.77	92.21	84.59	<b>89.70</b>	<u>85.98</u>	<u>90.21</u>	
		CODE	77.70	71.17	80.68	86.07	90.62	<u>85.24</u>	83.57	76.82	85.40	
		SID	80.63	74.30	<u>82.87</u>	<b>87.00</b>	92.11	<b>86.16</b>	81.60	76.07	83.36	
		<b>Ours</b>	<b>81.97</b>	<b>77.54</b>	<b>83.31</b>	86.30	<u>92.44</u>	85.23	<u>89.67</u>	85.63	<b>90.22</b>	
Adversarial		Greedy	68.20	61.86	74.91	80.83	82.59	80.30	82.83	76.46	84.68	
		DoLa	68.33	62.00	74.95	79.63	<b>84.22</b>	78.17	<b>84.43</b>	<b>83.22</b>	84.71	
		VCD	67.80	61.69	74.47	81.10	82.69	80.63	78.07	71.84	80.81	
		M3ID	68.20	61.86	74.91	79.90	81.00	79.54	82.97	76.63	84.78	
		CODE	68.30	62.23	74.61	79.37	78.73	79.59	79.00	71.60	82.07	
		SID	<u>71.33</u>	<u>64.75</u>	<u>76.57</u>	<b>82.40</b>	<u>83.38</u>	<b>82.14</b>	77.43	71.18	80.34	
		<b>Ours</b>	<b>72.83</b>	<b>67.00</b>	<b>76.81</b>	<u>81.33</u>	82.82	<u>80.90</u>	<u>83.07</u>	<u>76.70</u>	<b>84.87</b>	
GQA		Random	Greedy	85.97	79.72	87.30	87.53	<b>97.15</b>	86.12	<u>90.47</u>	95.98	89.86
			DoLa	<u>86.07</u>	<u>79.92</u>	<u>87.36</u>	83.47	94.66	81.10	84.03	95.70	81.70
			VCD	83.77	77.16	85.53	<u>87.83</u>	96.78	<u>86.55</u>	85.50	95.55	83.70
			M3ID	86.00	79.74	87.33	87.43	96.14	86.12	90.20	95.89	89.55
			CODE	84.87	78.58	86.37	87.07	94.48	85.89	88.80	95.68	87.89
			SID	85.97	79.72	87.30	<b>88.80</b>	96.78	<b>87.76</b>	86.07	<b>96.96</b>	84.24
	<b>Ours</b>		<b>87.13</b>	<b>83.47</b>	<b>87.80</b>	87.80	<u>97.09</u>	86.46	<b>90.80</b>	<u>96.08</u>	<b>90.24</b>	
	Popular	Greedy	73.80	66.38	78.64	<u>83.80</u>	<b>88.82</b>	82.68	<b>88.27</b>	91.41	<b>87.80</b>	
		DoLa	<u>73.97</u>	66.56	<u>78.73</u>	<u>78.57</u>	83.71	76.80	82.57	<u>92.08</u>	80.35	
		VCD	71.87	64.76	77.32	83.33	87.09	82.44	83.80	91.56	82.13	
		M3ID	73.83	66.39	78.67	80.67	82.39	80.14	87.83	90.97	87.35	
		CODE	73.90	<u>66.60</u>	78.60	81.47	83.29	80.95	86.60	90.97	85.85	
		SID	73.80	66.38	78.64	<b>84.27</b>	87.25	<b>83.61</b>	84.40	<b>92.93</b>	82.68	
		<b>Ours</b>	<b>78.67</b>	<b>72.42</b>	<b>81.28</b>	83.77	<u>88.28</u>	<u>82.75</u>	<u>87.93</u>	90.24	<u>87.58</u>	
	Adversarial	Greedy	68.17	61.60	75.19	81.50	84.36	80.70	<b>85.13</b>	85.61	<b>85.03</b>	
		DoLa	68.40	61.80	75.30	78.37	83.32	76.63	80.63	87.69	78.63	
		VCD	67.63	61.22	74.82	<u>82.00</u>	<u>84.63</u>	<u>81.29</u>	81.33	86.55	79.90	
		M3ID	68.20	61.62	75.22	80.50	82.11	80.00	<u>85.00</u>	85.71	84.85	
		CODE	<u>68.70</u>	<u>62.12</u>	<u>75.39</u>	79.87	80.56	79.64	83.93	85.85	83.49	
		SID	68.17	61.60	75.19	<b>82.37</b>	83.79	<b>81.99</b>	82.07	<b>87.81</b>	80.59	
		<b>Ours</b>	<b>73.40</b>	<b>66.91</b>	<b>77.68</b>	81.93	<b>84.71</b>	81.18	84.90	84.78	<u>84.93</u>	

### E.3 Comprehensive MMMU results

Table 11: **MMMU per-subject accuracy.** We report *Accuracy (%)* for each subject, model, and decoding method. Higher scores ( $\uparrow$ ) indicate better performance.

Subject	LLaVA-1.5-7B (Accuracy $\uparrow$ )							Qwen2.5-VL-7B (Accuracy $\uparrow$ )							InternVL3-8B (Accuracy $\uparrow$ )						
	Greedy	DoLa	VCD	M3ID	CODE	SID	Ours	Greedy	DoLa	VCD	M3ID	CODE	SID	Ours	Greedy	DoLa	VCD	M3ID	CODE	SID	Ours
Accounting	16.67	16.67	16.67	16.67	<b>20.00</b>	13.33	<b>16.67</b>	<b>46.67</b>	36.67	40.00	<b>46.67</b>	26.67	36.67	<b>36.67</b>	26.67	23.33	26.67	26.67	26.67	<b>33.33</b>	<b>30.00</b>
Agriculture	<b>41.38</b>	<b>41.38</b>	37.93	<b>41.38</b>	<b>41.38</b>	37.93	37.93	<b>55.17</b>	48.28	<b>55.17</b>	51.72	41.38	44.83	<b>55.17</b>	<b>55.17</b>	51.72	51.72	<b>55.17</b>	<b>55.17</b>	<b>55.17</b>	<b>55.17</b>
Architecture and Engineering	31.03	31.03	34.48	31.03	<b>41.38</b>	37.93	31.03	<b>58.62</b>	51.72	<b>58.62</b>	51.72	51.72	44.83	48.28	37.93	37.93	37.93	37.93	<b>41.38</b>	27.59	34.48
Art	<b>56.67</b>	53.33	53.33	<b>56.67</b>	53.33	50.00	53.33	<b>73.33</b>	<b>73.33</b>	<b>73.33</b>	<b>73.33</b>	60.00	<b>73.33</b>	70.00	<b>76.67</b>	70.00	<b>76.67</b>	<b>76.67</b>	73.33	<b>76.67</b>	73.33
Art Theory	<b>64.00</b>	<b>64.00</b>	56.00	<b>64.00</b>	56.00	60.00	<b>64.00</b>	<b>88.00</b>	<b>88.00</b>	<b>88.00</b>	80.00	76.00	<b>88.00</b>	<b>88.00</b>	92.00	92.00	92.00	92.00	<b>96.00</b>	92.00	92.00
Basic Medical Science	43.33	43.33	<b>53.33</b>	46.67	46.67	<b>50.00</b>	46.67	60.00	<b>63.33</b>	56.67	53.33	56.67	56.67	60.00	<b>66.67</b>	<b>66.67</b>	60.00	<b>66.67</b>	60.00	60.00	60.00
Biology	<b>17.86</b>	<b>17.86</b>	14.29	<b>17.86</b>	10.71	10.71	<b>17.86</b>	<b>42.86</b>	39.29	<b>42.86</b>	<b>42.86</b>	39.29	<b>42.86</b>	<b>42.86</b>	<b>53.57</b>	<b>53.57</b>	<b>53.57</b>	<b>53.57</b>	46.43	<b>53.57</b>	<b>53.57</b>
Chemistry	<b>20.00</b>	<b>20.00</b>	<b>20.00</b>	<b>20.00</b>	<b>20.00</b>	<b>20.00</b>	<b>24.00</b>	28.00	28.00	<b>32.00</b>	28.00	20.00	<b>32.00</b>	<b>36.00</b>	28.00	28.00	28.00	28.00	24.00	<b>36.00</b>	<b>32.00</b>
Clinical Medicine	34.48	34.48	24.14	34.48	<b>37.93</b>	27.59	34.48	<b>62.07</b>	55.17	<b>62.07</b>	48.28	51.72	58.62	58.62	68.97	68.97	<b>72.41</b>	68.97	68.97	<b>72.41</b>	68.97
Computer Science	<b>32.14</b>	<b>32.14</b>	28.57	<b>32.14</b>	25.00	28.57	<b>32.14</b>	<b>50.00</b>	<b>53.57</b>	<b>53.57</b>	50.00	39.29	50.00	<b>53.57</b>	53.57	50.00	53.57	53.57	50.00	50.00	<b>57.14</b>
Design	50.00	50.00	50.00	50.00	50.00	53.33	46.67	<b>73.33</b>	<b>73.33</b>	<b>73.33</b>	<b>73.33</b>	56.67	<b>73.33</b>	<b>73.33</b>	<b>73.33</b>	<b>73.33</b>	<b>73.33</b>	<b>73.33</b>	<b>73.33</b>	<b>73.33</b>	<b>73.33</b>
Diagnostics and Laboratory Medicine	<b>34.48</b>	<b>34.48</b>	<b>34.48</b>	<b>34.48</b>	31.03	<b>34.48</b>	<b>34.48</b>	41.38	27.59	41.38	37.93	34.48	37.93	<b>44.83</b>	51.72	51.72	55.17	51.72	55.17	44.83	<b>58.62</b>
Economics	<b>32.14</b>	28.57	25.00	<b>32.14</b>	<b>32.14</b>	<b>32.14</b>	<b>32.14</b>	<b>42.86</b>	<b>42.86</b>	35.71	39.29	28.57	<b>46.43</b>	<b>42.86</b>	<b>57.14</b>	<b>53.57</b>	<b>57.14</b>	<b>57.14</b>	<b>57.14</b>	<b>57.14</b>	<b>57.14</b>
Electronics	26.67	26.67	26.67	26.67	26.67	23.33	<b>30.00</b>	30.00	<b>36.67</b>	36.67	30.00	26.67	<b>33.33</b>	<b>36.67</b>	26.67	<b>30.00</b>	20.00	26.67	20.00	20.00	26.67
Energy and Power	43.33	43.33	<b>46.67</b>	43.33	<b>46.67</b>	43.33	<b>46.67</b>	<b>33.33</b>	<b>33.33</b>	<b>33.33</b>	<b>33.33</b>	30.00	30.00	30.00	40.00	40.00	40.00	40.00	40.00	36.67	<b>43.33</b>
Finance	<b>16.67</b>	<b>16.67</b>	<b>16.67</b>	<b>16.67</b>	<b>16.67</b>	13.33	<b>16.67</b>	23.33	<b>33.33</b>	26.67	23.33	23.33	<b>26.67</b>	<b>26.67</b>	26.67	23.33	26.67	23.33	23.33	23.33	<b>30.00</b>
Geography	34.48	34.48	37.93	34.48	<b>44.83</b>	34.48	34.48	37.93	37.93	<b>41.38</b>	<b>41.38</b>	<b>41.38</b>	31.03	37.93	<b>48.28</b>	<b>48.28</b>	<b>48.28</b>	<b>48.28</b>	<b>48.28</b>	37.93	<b>48.28</b>
History	42.86	42.86	42.86	42.86	42.86	<b>46.43</b>	<b>46.43</b>	67.86	67.86	64.29	64.29	<b>75.00</b>	67.86	67.86	<b>78.57</b>	<b>78.57</b>	75.00	<b>78.57</b>	67.86	75.00	75.00
Literature	72.41	72.41	<b>75.86</b>	72.41	72.41	72.41	72.41	<b>82.76</b>	<b>82.76</b>	79.31	41.38	44.83	75.86	<b>82.76</b>	82.76	<b>86.21</b>	82.76	82.76	82.76	82.76	<b>86.21</b>
Manage	24.14	24.14	<b>27.59</b>	24.14	20.69	20.69	24.14	<b>44.83</b>	41.38	41.38	31.03	37.93	41.38	<b>44.83</b>	55.17	55.17	55.17	55.17	55.17	51.72	<b>58.62</b>
Marketing	<b>26.67</b>	<b>26.67</b>	20.00	<b>26.67</b>	23.33	<b>26.67</b>	<b>26.67</b>	56.67	53.33	<b>63.33</b>	56.67	46.67	56.67	<b>63.33</b>	56.67	<b>63.33</b>	56.67	56.67	56.67	56.67	53.33
Materials	<b>25.93</b>	<b>29.63</b>	<b>29.63</b>	<b>29.63</b>	<b>29.63</b>	<b>25.93</b>	<b>25.93</b>	40.74	29.63	<b>48.15</b>	40.74	37.04	44.44	40.74	40.74	40.74	40.74	40.74	40.74	40.74	<b>44.44</b>
Math	<b>35.71</b>	<b>35.71</b>	32.14	32.14	21.43	28.57	<b>35.71</b>	42.86	42.86	32.14	39.29	46.43	<b>50.00</b>	46.43	<b>39.29</b>	<b>39.29</b>	35.71	<b>39.29</b>	<b>39.29</b>	35.71	<b>39.29</b>
Mechanical Engineering	<b>20.69</b>	<b>20.69</b>	<b>20.69</b>	<b>20.69</b>	20.69	13.79	<b>20.69</b>	37.93	41.38	37.93	37.93	<b>48.28</b>	44.83	41.38	34.48	34.48	34.48	34.48	31.03	34.48	<b>41.38</b>
Music	36.00	36.00	36.00	36.00	<b>44.00</b>	<b>44.00</b>	<b>44.00</b>	44.00	40.00	44.00	44.00	36.00	36.00	<b>48.00</b>	40.00	44.00	36.00	40.00	<b>48.00</b>	36.00	36.00
Pharmacy	40.74	40.74	40.74	40.74	<b>44.44</b>	37.04	40.74	55.56	<b>70.37</b>	62.96	48.15	44.44	62.96	55.56	51.85	51.85	48.15	51.85	48.15	51.85	<b>55.56</b>
Physics	23.33	23.33	26.67	23.33	23.33	23.33	<b>30.00</b>	<b>36.67</b>	<b>36.67</b>	<b>33.33</b>	<b>33.33</b>	<b>33.33</b>	<b>33.33</b>	30.00	<b>40.00</b>	<b>40.00</b>	<b>40.00</b>	<b>40.00</b>	<b>40.00</b>	<b>40.00</b>	<b>40.00</b>
Psychology	<b>34.62</b>	<b>34.62</b>	26.92	<b>34.62</b>	<b>34.62</b>	<b>30.77</b>	<b>34.62</b>	<b>73.08</b>	69.23	<b>73.08</b>	57.69	50.00	69.23	<b>73.08</b>	76.92	73.08	76.92	76.92	76.92	<b>80.77</b>	76.92
Public Health	16.67	16.67	13.33	16.67	<b>20.00</b>	16.67	16.67	50.00	<b>56.67</b>	53.33	50.00	46.67	53.33	50.00	<b>63.33</b>	<b>63.33</b>	<b>63.33</b>	<b>63.33</b>	<b>63.33</b>	<b>63.33</b>	<b>63.33</b>
Sociology	<b>36.67</b>	<b>36.67</b>	<b>36.67</b>	<b>36.67</b>	<b>36.67</b>	<b>36.67</b>	<b>36.67</b>	<b>53.33</b>	50.00	50.00	50.00	33.33	50.00	50.00	<b>63.33</b>	60.00	60.00	<b>63.33</b>	<b>63.33</b>	60.00	60.00
Overall	34.39	34.29	33.51	34.51	34.48	33.10	<b>35.13</b>	51.11	50.15	51.13	46.63	42.79	49.75	<b>51.18</b>	53.54	53.08	52.60	53.43	52.41	51.85	<b>54.14</b>

#### E.4 Token-level computation speed

All measurements are conducted on an NVIDIA RTX A6000 GPU with 48GB memory and an Intel(R) Xeon(R) Gold 5320 CPU running at 2.20 GHz. We perform 6 warmup iterations before measurement to eliminate initial overhead, and each of the 300 evaluation samples is decoded three times per method. We report the average decoding time per generated token, measured in milliseconds (ms/token). Other evaluation configurations, including preprocessing pipelines and evaluation scripts, follow the official CHAIR benchmark implementation.

Table 12: **Per-token inference latency** (ms/token) of different decoding strategies. We report the mean  $\pm$  standard deviation over 300 samples.

Method	LLaVA-1.5-7B	Qwen2.5-VL-7B
Greedy	26.0 $\pm$ 0.2	72.3 $\pm$ 1.5
DoLa	34.2 $\pm$ 0.4	82.6 $\pm$ 0.4
VCD	51.7 $\pm$ 0.4	143.4 $\pm$ 1.1
M3ID	49.5 $\pm$ 0.2	143.3 $\pm$ 13.2
CODE	81.0 $\pm$ 6.3	176.2 $\pm$ 45.1
SID	51.9 $\pm$ 0.4	146.5 $\pm$ 1.3
<b>Ours</b>	<b>26.5 <math>\pm</math> 0.2</b>	<b>72.7 <math>\pm</math> 0.4</b>

#### E.5 Comprehensive ablation results

This section provides comprehensive ablation results complementing Sec. 5.3.

We first report CHAIR ablations on vocabulary constraint, token selection, layer scope, and threshold  $\alpha$ , together with additional design-choice comparisons on selection metrics and fusion strategies (Table 14).

For selection, we compare alternative distance measures, including KL divergence and cosine similarity, to test whether the effectiveness of the token-selection step depends strongly on the exact metric choice or whether JSD mainly serves as a stable default.

For refinement, let  $p_{\text{output}}$  denote the vanilla next-token distribution on  $\mathcal{V}_{\text{cons}}^t$ , and let  $p_{\text{vision}}$  denote the selected vision-token distribution on the same constrained set. We then compare interpolation and Product-of-Experts (PoE) fusion under different coefficients. Specifically, we consider

$$(1 - \lambda) p_{\text{output}} + \lambda p_{\text{vision}}$$

for interpolation, and

$$\log p_{\text{output}} + \lambda \log p_{\text{vision}}$$

for PoE fusion. This PoE-style form is also consistent with the KL-regularized interpretation of the refinement rule in Eq. (6); a detailed derivation is provided in Appendix C.2. In interpolation,  $\lambda = 0.5$  corresponds to balanced mixing, while  $\lambda = 0.25$  and  $\lambda = 0.75$  place relatively more weight on the output and vision distributions, respectively. In PoE fusion,  $\lambda = 1.0$  corresponds to the default equal-strength combination, while smaller values weaken and larger values strengthen the influence of the vision-induced distribution.

As shown in Table 14, JSD provides the strongest overall trade-off among the tested selection metrics, while KL divergence and cosine similarity remain competitive but consistently yield weaker hallucination reduction and lower object-level F1. For refinement, the default PoE fusion performs best overall; interpolation remains competitive near balanced mixing, but is generally weaker, and both overly weak and overly strong PoE coefficients degrade performance.

We then present additional VQAv2 ablation results on layer-scope and  $\alpha$  choices to show that the same trend persists on a discriminative benchmark (Table 13).

Table 13: **VQAv2 ablation results.** Accuracy (%) under different choices of layer scope and threshold  $\alpha$  for Qwen2.5-VL-7B and InternVL3-8B. Higher scores ( $\uparrow$ ) indicate better performance.

Layer Scope	$\alpha$	Qwen2.5-VL-7B	InternVL3-8B
last	$10^{-1}$	65.73	67.60
last	$10^{-2}$	65.13	66.93
last	$10^{-3}$	65.53	64.53
last	$10^{-4}$	62.60	63.40
all	$10^{-1}$	65.93	68.40
all	$10^{-2}$	67.60	68.13
all	$10^{-3}$	66.27	64.87
all	$10^{-4}$	64.00	62.27

Table 14: **Ablation study.** Panel (a) varies the vision token selection criterion (*min/max* = min-/max-JSD, *random* = uniform) and use of a vocabulary subset (*full* = no constraint, *subset* = vocabulary subset constraint). Panel (b) varies the layer scope (*last* vs *all*) and the threshold  $\alpha$ . Panel (c) compares alternative selection metrics while fixing all other components. Panel (d) compares fusion design choices under the same constrained candidate set. Metrics are CHAIR<sub>S</sub> (↓), CHAIR<sub>I</sub> (↓), and F1 (↑). Highlighted rows mark the main configuration.

<b>(a) Vocabulary Constraint × Token Selection</b>				
Vocab	Selection	CHAIR <sub>S</sub> ↓	CHAIR <sub>I</sub> ↓	F1 ↑
subset	max	0.0	0.00	0.67
subset	random	2.6	7.94	17.63
subset	min	29.8	7.04	81.16
full	min	0.2	7.14	1.52
<b>(b) Layer Scope × Threshold <math>\alpha</math></b>				
Layer Scope	$\alpha$	CHAIR <sub>S</sub> ↓	CHAIR <sub>I</sub> ↓	F1 ↑
last	1e−1	33.8	8.32	80.02
last	1e−2	37.0	9.13	79.88
last	1e−3	34.0	7.88	80.97
last	1e−4	34.2	8.21	80.41
last	1e−5	31.4	8.04	80.12
last	1e−6	32.8	7.21	79.68
all	1e−1	33.8	8.82	80.70
all	1e−2	36.8	8.91	79.93
all	1e−3	38.8	9.36	79.60
all	1e−4	32.8	8.17	80.17
all	1e−5	29.8	7.04	81.16
all	1e−6	32.4	9.38	79.90
<b>(c) Selection Metric Comparison</b>				
Selection Metric		CHAIR <sub>S</sub> ↓	CHAIR <sub>I</sub> ↓	F1 ↑
KL		32.0	7.62	79.09
Cosine		32.4	8.07	80.16
JSD		29.8	7.04	81.16
<b>(d) Fusion Design Comparison</b>				
Fusion	Weight	CHAIR <sub>S</sub> ↓	CHAIR <sub>I</sub> ↓	F1 ↑
Interpolation	0.25	35.6	9.13	79.72
Interpolation	0.5	32.2	8.75	81.00
Interpolation	0.75	31.2	9.68	80.22
Product-of-Experts	0.25	33.4	8.07	80.25
Product-of-Experts	0.5	32.0	7.73	80.35
Product-of-Experts	1.0	29.8	7.04	81.16
Product-of-Experts	2.0	33.2	9.11	79.86

## F Detailed Qualitative Results

### F.1 Vision token analysis case studies

Building on the decoding procedure in Appendix C, we now examine two representative case studies that instantiate our vision token analysis: the fruit-counting example in Fig. 5 and the landscape description in Fig. 2. For both cases, we report token-level distributions and positional metadata to make the contribution of individual vision tokens explicit.

All examples are drawn from the LLaVA-Bench-In-the-Wild dataset (Liu et al., 2023) using LLaVA-1.5-7B. Vision token embeddings are extracted from the final decoder layer ( $L = 32$ ), and the threshold for constrained vocabulary selection is set to  $\alpha = 10^{-5}$ , consistent with our default generative configuration.

**Fruit-counting example (Fig. 5).** This example corresponds to the decoding step  $t = 2$  described in Appendix C. At this step, ReVisiT selects vision token index 229 as the most relevant reference based on the context-aware constrained distribution over  $\mathcal{V}_{\text{cons}}^2$ . Table 15 reports the log-probability values from three sources: (1) the base model output (Eq. (1)), (2) the vision-token projection at index 229 (Eq. (4)), and (3) the final combined distribution after fusion (Eq. (7)). Within the constrained subset, the vision token sharpens the distribution over numerals: it boosts the probability of “three” while suppressing “four” and other alternatives, thereby aligning the next-token prediction with the visual evidence.

**Landscape description example (Fig. 2).** Figure 2 visualizes a complementary case from the same dataset (image ID: 003.jpg, prompt: “Describe this photo in detail.”). Here we focus on how the semantics of individual vision tokens become more interpretable under vocabulary constraints.

The three tokens shown on the **left** of the figure (pink, yellow, cyan) are used to examine the semantic alignment of vision-token projections. For each token, we compute the projected distribution over (i) the full vocabulary  $\mathcal{V}$  and (ii) a manually defined constrained subset  $\mathcal{V}_{\text{env}} = \{\text{river, mountain, person, sky, tree, cloud, sea, grass, rock}\}$ . Table 16 lists the corresponding log-probabilities and probabilities. When restricted to  $\mathcal{V}_{\text{env}}$ , each vision token assigns most of its mass to a small number of semantically coherent labels (e.g., “sky” for the pink token,

“river”/“rock” for the yellow token, “person” for the cyan token), making their textual semantics explicit and interpretable.

The token on the **right** (vision token index 193) demonstrates how ReVisiT uses a single vision token as a reference signal during decoding. At generation step  $t = 72$ , this token acts as the selected reference when refining the output distribution over  $\mathcal{V}_{\text{cons}}^{72}$ . Table 17 provides the detailed scores. The probability of “painting” increases substantially after fusion, while the hallucinatory alternative “person” is suppressed, showing how context-aware constrained divergence encourages visually grounded descriptions without additional training.

### F.2 Case study without vocabulary subset

This case study illustrates how crucial the vocabulary subset constraint is for referencing vision tokens in logit lens manner. Figure 8 shows three qualitative comparison between with and without vocabulary subset.

When the subset is removed (*w/o subset*), the model quickly diverges into degenerate behavior: it produces partially fluent prefixes but then collapses into long, repetitive sequences of a single token or meaningless character streams until the maximum length is reached. In contrast, with the subset enabled (*w/ subset*), the model generates coherent, visually grounded descriptions that correctly capture the scene content.

These examples highlight a failure mode of directly projecting vision tokens onto the full vocabulary space. Without the subset, the projected logits can over-amplify a few high-frequency or locally consistent tokens that are unrelated to the image, causing the fused distribution to collapse and reinforcing the same token at every subsequent step. By constraining the output space to a contextually plausible vocabulary subset, ReVisiT prevents this collapse, keeps probability mass concentrated on semantically relevant candidates, and preserves both fluency and visual faithfulness during generation.

### F.3 Additional qualitative examples

To further illustrate the qualitative behavior of ReVisiT, we present two representative examples, including a sample from the LLaVA-Bench-In-The-Wild dataset, which contains diverse real-world images requiring complex reasoning beyond conventional benchmarks.

In the first example (Fig. 9), the input is a cartoon-style illustration that contrasts classical statistical learning with neural networks via a visual metaphor. Baseline methods fail to capture this high-level analogy: they either focus on local, literal details of the scene or fall into repetitive, generic phrasing, thereby missing the notion of “stacking more layers.” In contrast, ReVisiT correctly identifies that the comic is about neural network layers and produces a coherent description that explicitly aligns the visual elements (e.g., the whiteboard and stacked layers) with the intended concept.

The second example (Fig. 10) is an illustration of a bear, a cat, and a rabbit seated around a table with a plate of donuts. Here, baselines introduce object- and attribute-level hallucinations, such as mentioning cookies or pink cups that are not present in the image. ReVisiT, on the other hand, provides a faithful caption that correctly enumerates the animals and their activities without introducing unsupported details, demonstrating improved visual grounding in everyday scenes as well.

## G Usage of AI assistants

In preparing this work, we utilized AI-based writing assistants in a limited way to suggest alternative phrasings, correct grammatical errors, and improve the readability. All research ideas, experimental designs, implementation details, and reported results were determined, implemented, and verified by the authors, and all model outputs and quantitative results in the paper were generated and checked through our own code and experiments. AI assistants did not generate any factual content reported in the paper (e.g., experimental results, dataset statistics, or citations), ensuring the originality, scientific contributions, technical content, methodology, and experimental findings are entirely attributable to the authors.

Table 15: **Detailed numerical values for Figure 5.** Token scores over the constrained vocabulary  $\mathcal{V}_{\text{cons}}^2$  at decoding step  $t = 2$  with ReVisiT are presented. The selected vision token (index 229) amplifies the relevance of “three”, increasing its probability from 35.45% (base) to 58.27% (final), while suppressing “four” from 50.78% to 38.21%.

Rank	Token	Base Logit		(229) Vision Token Logit		Final Logit	
		Log-Prob	Prob (%)	Log-Prob	Prob (%)	Log-Prob	Prob (%)
1	three	-1.04	35.45	-1.47	22.98	-2.51	58.27
2	four	-0.68	50.78	-2.25	10.52	-2.93	38.21
3	two	-4.57	1.04	-1.19	30.44	-5.76	2.26
4	five	-2.54	7.91	-4.45	1.16	-6.99	0.66
5	a	-4.33	1.31	-3.08	4.58	-7.42	0.43
6	six	-3.94	1.94	-5.15	0.58	-9.10	0.08
7		-5.10	0.61	-4.22	1.47	-9.32	0.06
8	several	-5.75	0.32	-5.39	0.45	-11.15	0.01
9	seven	-5.79	0.31	-6.12	0.22	-11.90	0.00
10	un	-8.65	0.02	-3.80	2.24	-12.45	0.00
11	at	-8.28	0.03	-4.77	0.85	-13.05	0.00
12	one	-10.05	0.00	-3.69	2.50	-13.74	0.00
13	in	-11.15	0.00	-2.99	5.01	-14.14	0.00
14	eight	-7.38	0.06	-6.78	0.11	-14.16	0.00
15	f	-9.52	0.01	-4.76	0.86	-14.28	0.00
16	multiple	-7.66	0.05	-6.66	0.13	-14.32	0.00
17	only	-8.68	0.02	-5.89	0.28	-14.57	0.00
18	nine	-7.71	0.04	-6.97	0.09	-14.68	0.00
19	half	-8.62	0.02	-6.22	0.20	-14.85	0.00
20	many	-8.70	0.02	-6.21	0.20	-14.91	0.00
21	all	-10.38	0.00	-4.85	0.78	-15.23	0.00
22	still	-11.11	0.00	-4.15	1.57	-15.26	0.00
23	an	-10.33	0.00	-5.10	0.61	-15.43	0.00
24	some	-10.05	0.00	-5.40	0.45	-15.46	0.00
25	more	-11.15	0.00	-4.68	0.92	-15.83	0.00
26	about	-10.05	0.00	-5.90	0.27	-15.95	0.00
27	both	-10.74	0.00	-5.23	0.53	-15.98	0.00
28	the	-11.98	0.00	-4.07	1.71	-16.05	0.00
29	not	-11.58	0.00	-4.59	1.02	-16.16	0.00
30	over	-11.99	0.00	-4.25	1.42	-16.25	0.00
31	different	-11.93	0.00	-4.73	0.88	-16.66	0.00
32	ten	-8.56	0.02	-8.15	0.03	-16.71	0.00
33	just	-10.97	0.00	-5.78	0.31	-16.75	0.00
34	fruit	-11.81	0.00	-5.04	0.65	-16.85	0.00
35	as	-12.06	0.00	-4.82	0.81	-16.88	0.00
36	small	-12.11	0.00	-4.86	0.78	-16.97	0.00
37	various	-10.39	0.00	-6.71	0.12	-17.10	0.00
38	also	-11.94	0.00	-5.21	0.55	-17.15	0.00
39	very	-11.80	0.00	-5.54	0.39	-17.35	0.00
40	around	-11.28	0.00	-6.07	0.23	-17.35	0.00
41	numerous	-10.02	0.00	-7.51	0.05	-17.54	0.00
42	no	-10.70	0.00	-6.93	0.10	-17.64	0.00
43	total	-11.17	0.00	-6.80	0.11	-17.97	0.00
44	currently	-11.17	0.00	-6.89	0.10	-18.06	0.00
45	cut	-11.78	0.00	-6.46	0.16	-18.24	0.00
46	quite	-11.62	0.00	-6.71	0.12	-18.33	0.00
47	few	-11.91	0.00	-6.64	0.13	-18.55	0.00
48	twelve	-10.07	0.00	-8.53	0.02	-18.60	0.00
49	lots	-11.52	0.00	-7.37	0.06	-18.89	0.00
50	th	-11.34	0.00	-7.63	0.05	-18.98	0.00
51	actually	-11.84	0.00	-7.40	0.06	-19.24	0.00
52	approximately	-11.05	0.00	-8.45	0.02	-19.50	0.00
53	exactly	-12.14	0.00	-7.48	0.06	-19.62	0.00
54	plenty	-11.78	0.00	-7.95	0.04	-19.73	0.00
55	eleven	-10.82	0.00	-9.25	0.01	-20.07	0.00

Table 16: **Detailed numerical values for Figure 2 (left)**. Log-probabilities and probabilities (%) of selected constrained vocabulary tokens as projected by three vision tokens (pink, yellow, cyan) under the full vocabulary  $\mathcal{V}$  and the constrained vocabulary  $\mathcal{V}_{\text{env}}$ .

Vision Token	Top (Pink)				Middle (Yellow)				Bottom (Cyan)			
Vocab	$\mathcal{V}$		$\mathcal{V}_{\text{env}}$		$\mathcal{V}$		$\mathcal{V}_{\text{env}}$		$\mathcal{V}$		$\mathcal{V}_{\text{env}}$	
Token	Log-Prob	%	Log-Prob	%	Log-Prob	%	Log-Prob	%	Log-Prob	%	Log-Prob	%
river	-14.74	0.00	-6.16	0.21	-3.61	2.70	-0.36	69.75	-14.58	0.00	-5.02	0.66
mountain	-14.07	0.00	-5.48	0.42	-6.47	0.15	-3.22	3.99	-15.02	0.00	-5.46	0.43
person	-13.39	0.00	-4.80	0.82	-12.27	0.00	-9.01	0.01	-9.86	0.01	-0.30	73.81
sky	-8.66	0.02	-0.07	93.44	-6.76	0.12	-3.51	2.99	-12.82	0.00	-3.26	3.83
tree	-12.06	0.00	-3.48	3.09	-11.71	0.00	-8.46	0.02	-12.47	0.00	-2.92	5.42
cloud	-13.88	0.00	-5.30	0.50	-9.52	0.01	-6.26	0.19	-12.46	0.00	-2.91	5.47
sea	-13.80	0.00	-5.21	0.55	-6.82	0.11	-3.57	2.81	-13.24	0.00	-3.68	2.52
grass	-15.41	0.00	-6.82	0.11	-7.19	0.08	-3.94	1.95	-13.41	0.00	-3.86	2.11
rock	-13.35	0.00	-4.77	0.85	-4.95	0.71	-1.70	18.33	-12.41	0.00	-2.85	5.77

Table 17: **Detailed numerical values for Figure 2 (right)**. Token scores over the constrained vocabulary  $\mathcal{V}_{\text{cons}}^{72}$  at decoding step  $t = 72$  with ReVisiT. The selected vision token (index 193) amplifies the relevance of “painting”, increasing its probability from 8.08% (base) to 39.15% (final) while suppressing hallucinatory token “person” from 15.10% to 7.19%.

Rank	Token	Base Logit		(193) Vision Token Logit		Final Logit	
		Log-Prob	Prob (%)	Log-Prob	Prob (%)	Log-Prob	Prob (%)
1	painting	-2.52	8.08	-1.27	27.99	-3.79	39.15
2	rock	-2.60	7.42	-2.28	10.22	-4.88	13.11
3	mountain	-2.14	11.76	-3.03	4.83	-5.17	9.82
4	person	-1.89	15.10	-3.59	2.75	-5.48	7.19
5	landscape	-2.27	10.30	-3.42	3.27	-5.70	5.82
6	d	-3.45	3.16	-2.29	10.14	-5.74	5.55
7	p	-3.91	2.01	-2.45	8.60	-6.36	3.00
8	scene	-3.80	2.23	-2.70	6.75	-6.50	2.60
9	small	-2.83	5.91	-3.70	2.48	-6.52	2.54
10	beautiful	-3.67	2.54	-3.03	4.84	-6.70	2.13
11	large	-2.93	5.34	-3.78	2.28	-6.71	2.11
12	scen	-3.57	2.81	-3.17	4.19	-6.74	2.04
13	hill	-4.01	1.82	-2.83	5.91	-6.84	1.86
14	chair	-2.85	5.78	-4.38	1.26	-7.23	1.26
15	bow	-3.87	2.09	-3.68	2.53	-7.54	0.92
16	second	-3.80	2.23	-4.52	1.09	-8.33	0.42
17	smaller	-3.02	4.90	-6.04	0.24	-9.06	0.20
18	boat	-3.00	4.98	-6.34	0.18	-9.34	0.15
19	distant	-4.14	1.59	-5.43	0.44	-9.57	0.12



Q. Describe this photo in detail.

**Greedy** The image features a comic strip with a man standing in front of a whiteboard, pointing at a graph. The man appears to be explaining something to the audience, possibly related to the graph. The graph is showing a downward trend, indicating a decrease in value. There are several people in the scene, with one person standing close to the man and the others scattered around the room. The man is wearing a funny hat, adding a humorous touch to the scene. The overall atmosphere suggests a learning environment where the man is teaching others about the graph and its implications.

**M3ID** The image consists of two comic strips, each featuring a person drawing a graph on a whiteboard while talking about stack layers. In the first strip, the person is smiling, while in the second strip, they are angry. In both strips, the person is pointing towards a graph and making a statement. In the second strip, the angry figure is drawing the graph, while the smiling figure is holding a VC dimension of kernel expert and minimizing the structural risk in new one off margin considered using next kernel stable make the next kernel table off margins considered using next kernel stable make the next kernel stable table off margin.

**ReVisiT** The image is a comic strip that is a visual representation of a person learning about neural networks. In the comic, the person is pointing to a graph on a whiteboard, explaining the concept of layers in a neural network. The comic is a cartoonish representation of a person with a head and a large smile, pointing to the whiteboard. The whiteboard is the main focus of the comic, and it is a visual representation of a neural network. The person is explaining the layers of the network, with the layers being drawn on the whiteboard. The comic is a fun and engaging way to learn about neural networks and the layers of a network.

Figure 9: **Qualitative example.** The input image is a cartoon-style illustration contrasting classical statistical learning and neural network reasoning via a visual metaphor, emphasizing the shift from theoretical rigor to the heuristic of “stacking more layers” We compare the generated responses of vanilla greedy decoding, M3ID, and ReVisiT, highlighting how ReVisiT better captures the intended visual analogy compared to the baseline methods.

Q. What are the animals in the painting and what are they doing?

**Greedy** The painting features a bear, a cat, and a rabbit sitting around a table, eating food. They are enjoying a meal together, with the bear eating a donut, the cat drinking from a cup, and the rabbit eating a cookie.

**DoLa** The painting features a bear, a cat, and a rabbit sitting around a table, eating food. They are enjoying a meal together, with the bear eating a donut, the cat drinking from a cup, and the rabbit eating a cookie.

**VCD** The painting features a bear, a bunny, and a cat sitting around a table, eating cookies. They are all gathered around the table, enjoying their treats together.

**M3ID** The painting features a bear, a cat, and a rabbit sitting around a wooden table, enjoying a meal together. They are eating cookies and drinking pink cups.

**SID** The painting features a bear, a cat, and a rabbit sitting around a table, eating cookies and drinking from cups.

**ReVisiT** The painting features a bear, a cat, and a rabbit. The bear is eating a donut, while the cat and the rabbit are sitting at a table with a plate of donuts.

Figure 10: **Qualitative comparison with baselines.** The input image is a illustration showing a bear, a cat, and a rabbit seated around a table with a plate of donuts. We compare the responses of vanilla greedy decoding and ReVisiT to the question, “What are the animals in the painting and what are they doing?” While the greedy output introduces a hallucinated detail (“cookie”) and assigns actions not visually supported (e.g., “the cat drinking from a cup”), ReVisiT provides a more faithful description aligned with the visual content, accurately identifying the animals and their activities.