CAVCD: CANDIDATE-AWARE VISUAL COUNTERFACTUAL DECODING FOR MITIGATING OBJECT HALLUCINATIONS IN LARGE VISION-LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Large vision-language models (LVLMs) suffer from object hallucination, where generated responses mention objects absent from the input image. Contrastive decoding alleviates this problem by contrasting original and perturbed predictions, but offers limited control over which candidate tokens to correct and how strongly, potentially disrupting otherwise correct predictions. To address these limitations, we propose Candidate-Aware Visual Counterfactual Decoding (CAVCD), a training-free method built on the idea that intervention-induced prediction changes should be assessed for individual candidates before guiding token selection. CAVCD first adapts visual interventions to the generation process, restricting visual access at the initial step and applying softer attention attenuation during subsequent generation. During continuation, it evaluates candidate responses across decoder layers and combines selected responses into a target distribution that guides logit correction. An adaptive gate then regulates correction strength, limiting excessive updates that could override image-supported token choices. Together, these designs make correction more selective and adaptive, helping suppress hallucinations while preserving plausible predictions. Extensive experiments demonstrate that CAVCD effectively mitigates object hallucination across multiple evaluation metrics.

1 INTRODUCTION

Large vision-language models (LVLMs) have demonstrated strong capabilities in image description, visual question answering, and multimodal instruction following (Liu et al., 2023; 2024; Dai et al., 2023). Despite these capabilities, LVLMs can produce fluent responses that are not supported by the input image, a phenomenon known as hallucination. For example, they may mention an object absent from an image or claim that it is present when answering a visual question (Rohrbach et al., 2018; Li et al., 2023b). These hallucinations undermine the reliability of LVLMs in applications that require accurate visual understanding.

A variety of methods have been proposed to mitigate object hallucination, with training-based methods and training-free decoding strategies forming two prominent lines of work. Training-based methods improve visual instruction data or align models with human feedback (Yu et al., 2024a;b; Sun et al., 2024). However, they require additional data preparation and model training, which can be costly. Training-free decoding strategies instead operate during inference without updating model parameters(Huang et al., 2024; Leng et al., 2024; Wang et al., 2024; Chen et al., 2024). Among them, contrastive decoding (CD) compares clean and negative predictions to adjust next-token logits. For instance, Visual Contrastive Decoding (VCD) obtains the negative prediction by perturbing the input image (Leng et al., 2024), whereas Self-Introspective Decoding (SID) retains low-attention visual tokens selected according to the current context (Huo et al., 2025). These CD methods share a common strategy: they use the difference between clean and negative logits to adjust token scores, which we refer to as **direct logit contrast**.

However, this direct logit contrast approach offers limited ability to distinguish helpful corrections from harmful ones or adapt correction strength to the current prediction. As a result, the correction may suppress an image-supported token and cause the model to select an unsupported alternative,

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

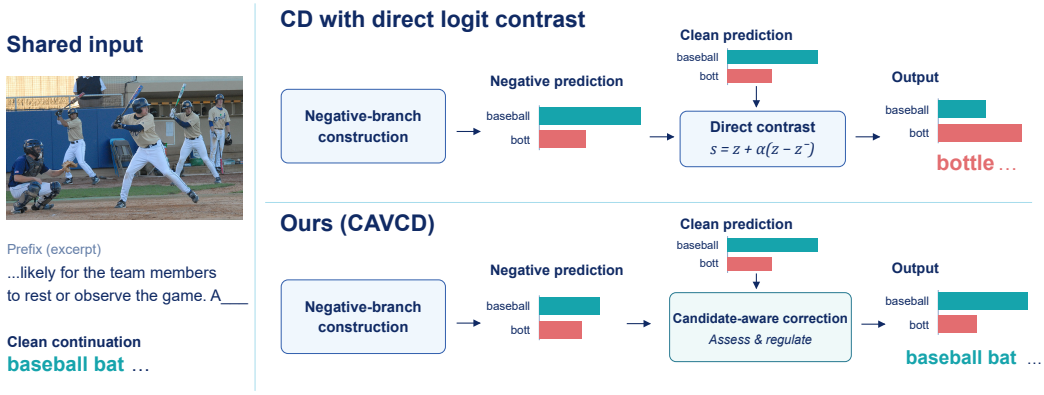


Figure 1: Conceptual comparison of direct logit contrast and CAVCD. Direct contrast adjusts clean logits using a negative prediction and can overturn an image-supported object choice. CAVCD assesses candidate responses and controls correction strength to preserve plausible predictions.

even when the clean prediction favored the correct choice, as illustrated in Figure 1. In addition, using the same visual intervention rule throughout generation may overlook the different intervention requirements at different generation steps. At the initial step, the model can be prone to focusing excessively on a few highly attended regions and overlooking potentially important visual cues elsewhere. If this produces an initial prediction that lacks visual support, the selected token becomes part of the text context for subsequent generation, allowing the error to persist or even compound.

To address these limitations, we propose Candidate-Aware Visual Counterfactual Decoding (CAVCD), a training-free framework that constructs the negative branch according to the generation stage and selectively uses the resulting intervention-induced responses to correct logits. As outlined in Figure 1, CAVCD assesses candidate responses and regulates their influence on the clean prediction during continuation. Here, a visual counterfactual changes the model's access to visual information while keeping the input image, instruction, and generated prefix unchanged. We call the changes between the clean and negative predictions **intervention-induced responses**. To distinguish the initial step from continuation steps, CAVCD uses a hard intervention that restricts visual access in the negative branch at the initial step, followed by direct logit contrast. During continuation steps, it instead uses a soft intervention that attenuates attention to highly attended visual tokens while keeping all visual tokens accessible in the negative branch. To limit inappropriate corrections during continuation, CAVCD assesses intervention-induced responses for plausible next-token candidates across multiple decoder layers. The combined assessments yield a target distribution over candidates, whose difference from the clean distribution determines the correction direction. An adaptive gate controls the correction strength at each continuation step.

In summary, our main contributions are as follows:

- We study how different uses of intervention-induced responses affect hallucination mitigation under the same visual intervention. Controlled experiments reveal that directly applying these responses can introduce inappropriate corrections, demonstrating that how responses are used is as important as how they are produced.

- We propose CAVCD, which adapts visual interventions to the generation stage and current attention, assesses candidates across decoder layers during continuation, and controls the direction and strength of logit corrections without updating model parameters.

- Experiments demonstrate that CAVCD reduces sentence-level object hallucination in image descriptions and improves accuracy in object-existence question answering. Ablation studies further support the effectiveness of selectively using intervention-induced responses and controlling the correction strength.

2 Related Work

2.1 Hallucination Mitigation in LVLMs

Existing approaches mitigate hallucination through training, post-generation correction, or decoding-time intervention. Training-based methods improve visual instruction data or align model behavior with factual supervision. HalluciDoctor identifies and repairs hallucinated content in instruction data (Yu et al., 2024a), while RLHF-V uses fine-grained correctional human feedback for multimodal alignment (Yu et al., 2024b). Post-generation methods instead examine and revise generated text: Woodpecker employs a verification pipeline (Yin et al., 2024), and LURE learns to rectify object hallucinations (Zhou et al., 2024). Training-free decoding intervenes directly in token selection; for example, OPERA combines an over-trust penalty with retrospection-based search (Huang et al., 2024). We focus on this inference-time setting and use intervention-induced responses to correct candidate logits without updating model parameters.

2.2 Contrastive and Adaptive Decoding

Contrastive decoding adjusts token scores using differences between predictive distributions, with plausibility constraints excluding low-probability candidates (Li et al., 2023a). LVLM methods differ in how they construct the contrasting condition: VCD perturbs the image (Leng et al., 2024), ICD modifies the instruction (Wang et al., 2024), and HALC uses local image views for focal contrast (Chen et al., 2024). Attention-guided methods make the visual intervention responsive to the current context. SID retains low-attention visual tokens selected at an early decoder layer to induce vision-and-text association hallucinations for logit contrast (Huo et al., 2025), while AvisC uses attention analysis for visual calibration (Woo et al., 2025). These studies show that the visual condition shapes the predictions available for decoding. Our hard intervention restricts visual access at the initial step; during continuation steps, our soft intervention attenuates highly attended visual keys. We then assess the intervention-induced responses before correcting candidate logits.

3 Methodology

3.1 Preliminaries

We consider an LVLM with frozen parameters θ that generates text given an image x and instruction u. At decoding step t, the model predicts the next token from vocabulary $\mathcal{V}$, conditioned on previously selected tokens $y_{<t}$. The clean forward pass produces logits $z_t \in \mathbb{R}^{|\mathcal{V}|}$, yielding the clean distribution

$$p_\theta(\cdot \mid x, u, y_{<t}) = \mathrm{softmax}(z_t). \tag{1}$$

To restrict correction to plausible alternatives, we define a **candidate set** from the clean prediction (Li et al., 2023a; Leng et al., 2024):

$$\begin{aligned} \mathcal{P}_t &= \left\{ i \in \mathcal{V} : z_{t,i} \geq \max_{j \in \mathcal{V}} z_{t,j} + \log \beta \right\}, \\ \mathcal{C}_t &= \mathrm{TopI}_{i \in \mathcal{P}_t}(z_{t,i}), \end{aligned} \tag{2}$$

where $0 < \beta \leq 1$ specifies the minimum probability relative to the most probable clean token, and I is the candidate budget. Since the clean top token always qualifies, $\mathcal{C}_t$ is nonempty.

As illustrated in Figure 2, CAVCD consists of three components: **Hard-to-Soft Visual Intervention**, **Candidate-Aware Target Construction**, and **Gated Token Correction**. The first component constructs the negative branch by restricting visual access at the initial step and attenuating attention to selected visual keys during continuation. The initial prediction is corrected through direct logit contrast. During continuation, the second component assesses intervention-induced responses for individual candidates across decoder layers and constructs a target distribution from the selected responses. The third component uses this distribution to guide logit correction and adaptively controls its strength, limiting interference with otherwise plausible predictions. We detail these components below.

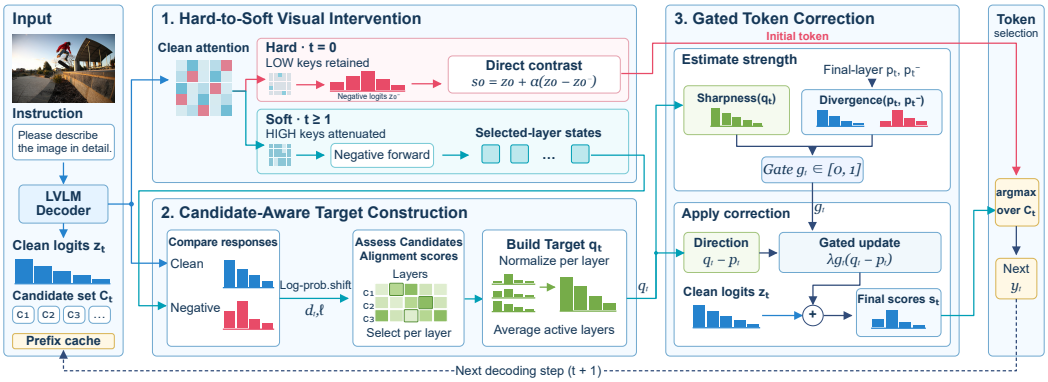


Figure 2: Overview of CAVCD. (1) **Hard-to-Soft Visual Intervention** retains low-attention visual keys for initial-step contrast, then attenuates high-attention keys during continuation while keeping all keys accessible. (2) **Candidate-Aware Target Construction** computes the log-probability shift between clean and negative predictions at selected decoder layers. It compares this shift with each candidate's loss-gradient direction to obtain alignment scores. Selected scores are normalized within each layer and averaged across active layers to form q_t. Heatmap rows and columns represent candidates and layers; outlined cells indicate selected responses. (3) **Gated Token Correction** combines target concentration and clean–negative divergence to control the strength of the update $\lambda g_t(q_t - p_t)$ added to the clean logits. Both paths select the next token from $\mathcal{C}_t$.

3.2 Hard and Soft Visual Interventions

We construct the negative branch by manipulating the model's access to visual information through attention-based interventions. We apply a hard intervention that restricts visual access at the initial step, and a soft intervention that attenuates attention to highly attended visual keys during continuation steps. Both interventions rely on attention-guided selection of visual tokens. At an early selector layer ℓ_s, we extract the last query's attention to each of the N visual tokens and average across heads, denoting the resulting score for visual token j by $a_{t,j}$. Selection is performed on the clean branch, so these scores reflect the current image, instruction, and generated prefix. The selected indices determine an additive bias to visual-key attention logits in decoder layers $\mathcal{L}_{\text{int}}$ following ℓ_s, while text-key logits and causal constraints remain unchanged.

Hard intervention at the initial step. At $t = 0$, we aim to prevent the model from committing to a prediction based on incomplete visual evidence. To achieve this, we identify and retain only the K least-attended visual tokens, $\mathcal{R}_0 = \text{BottomK}(a_0, K)$, in the negative branch. We then restrict the negative branch's visual attention to $\mathcal{R}_0$ by setting attention logits for all other visual keys to $-\infty$ across heads and query positions in the intervention layers $\mathcal{L}_{\text{int}}$ (Huo et al., 2025). Earlier layers retain full visual access, so information incorporated before the mask remains in the representations. The resulting negative logits z_0^- provide a counterfactual prediction that reflects limited visual grounding, which we use for initial-step logit correction through direct contrast in Section 3.4.

Soft intervention at continuation steps. At $t \geq 1$, the model has already selected an initial token and built up context. Here, we switch to a softer intervention strategy that attenuates rather than eliminates visual information. At each step, we identify the K most-attended visual tokens, $\mathcal{S}_t = \text{TopK}(a_t, K)$, recomputing this subset as the generated prefix grows. Instead of masking these tokens completely, we attenuate attention to them by adding a finite negative bias to their attention logits before softmax, across all heads and query positions in the intervention layers $\mathcal{L}_{\text{int}}$:

$$b_{t,j}^{\text{soft}} = \begin{cases} \log \rho, & j \in \mathcal{S}_t, \\ 0, & j \notin \mathcal{S}_t, \end{cases} \qquad 0 < \rho < 1. \tag{3}$$

This bias multiplies the selected keys' unnormalized attention weights by ρ, after which softmax renormalizes attention over all visual and text keys allowed by the attention mask. Crucially, all

visual keys remain accessible to the negative branch, preserving the model's ability to ground predictions in the image while reducing its reliance on the most salient regions.

To enable candidate-aware assessment of the intervention's effect, we extract paired hidden states from both branches at continuation steps. The clean and negative forward passes process the same token from the same incoming clean cache, with the soft intervention applied only to the negative branch. We collect their last-query hidden states from multiple decoder layers, denoted $h_{t,\ell}$ and $h^-_{t,\ell}$ respectively. These paired hidden states allow us to compare clean and negative predictions under identical generated prefixes, isolating the intervention's impact on different candidate tokens. The next section describes how we use these paired predictions to construct a target distribution for selective logit correction.

3.3 Candidate-Aware Target Distribution Construction

The soft intervention changes predictions across the vocabulary at each continuation step. Rather than applying these changes uniformly through direct logit contrast, we assess the intervention-induced response for each plausible candidate and select candidates with relatively high alignment scores to guide the correction. Throughout, we consider continuation steps $t \geq 1$ and candidate indices $i \in \mathcal{C}_t$, where $\mathcal{C}_t$ is the candidate set defined in Section 3.1.

We begin by extracting predictions at multiple decoder layers to capture how different layers respond to the intervention. We apply intermediate vocabulary readouts to paired hidden states from the clean and negative branches (Chuang et al., 2024; Belrose et al., 2023). For each layer ℓ in a readout set $\mathcal{L}$, let $h_{t,\ell}$ and $h^-_{t,\ell}$ denote the last-query hidden states of the clean and negative branches. The shared mapping $\phi = \mathrm{LMHead} \circ \mathrm{FinalNorm}$ yields clean logits $c_{t,\ell} = \phi(h_{t,\ell})$ and negative logits $n_{t,\ell} = \phi(h^-_{t,\ell})$. By comparing these paired logits across layers, we can measure how the intervention shifts the model's predictions and identify which candidates are most sensitive to the reduction in visual attention.

To identify which candidates are most affected by the intervention, we measure whether the intervention-induced probability shift aligns with the direction that would increase each candidate's loss. When the intervention reduces attention to salient visual regions and causes a candidate's probability to drop, we measure whether this shift aligns with the direction that would increase the candidate's loss. We express the intervention-induced response as a log-probability shift: defining the negative distribution as $\pi^-_{t,\ell} = \mathrm{softmax}(n_{t,\ell})$, the shift from the clean to the negative prediction is $d_{t,\ell} = \log \pi^-_{t,\ell} - \log \mathrm{softmax}(c_{t,\ell})$. For each candidate i, the gradient of its cross-entropy loss with respect to the negative logits is $\pi^-_{t,\ell} - e_i$, where e_i is the one-hot vector for token i. This gradient points in the direction that increases the candidate's loss. We then compute the alignment between the log-probability shift and this gradient direction using cosine similarity:

$$u_{t,\ell,i} = \cos\Big(\pi^-_{t,\ell} - e_i,\, d_{t,\ell}\Big), \qquad i \in \mathcal{C}_t. \tag{4}$$

A higher alignment score means that the intervention-induced shift more closely follows a local direction that increases the candidate's loss. We use this score to compare candidates within a layer, rather than as proof that a candidate is visually grounded. Since the loss-gradient direction depends on i, the same shift can yield different alignment scores for competing tokens.

We next construct a layer-specific target distribution by selecting candidates with relatively high alignment scores within each layer. We set $\tau_{t,\ell}$ to the 75th percentile of the scores $\{u_{t,\ell,i} : i \in \mathcal{C}_t\}$. This selection is relative to the other candidates in the same layer; a selected score need not be positive. We square the positive excess above the threshold before normalizing:

$$q_{t,\ell,i} = \frac{[\max(u_{t,\ell,i} - \tau_{t,\ell}, 0)]^2}{\sum_{j \in \mathcal{C}_t} [\max(u_{t,\ell,j} - \tau_{t,\ell}, 0)]^2}. \tag{5}$$

A layer is active only when the denominator exceeds 10^{-12}; numerical stability and fallback handling are detailed in Appendix A. Squaring emphasizes candidates whose alignment scores exceed the layer-specific threshold by a larger margin and suppresses those with marginal excess. After normalization, the layer-specific target distribution assigns probability according to this excess. We aggregate the distributions from active layers in the next step.

Finally, we aggregate the layer-specific target distributions from the active layers. We use a simple average to give each active layer equal influence, reflecting the assumption that no single layer's assessment should dominate:

$$q_{t,i} = \frac{1}{|\mathcal{L}_{\mathrm{act},t}|} \sum_{\ell \in \mathcal{L}_{\mathrm{act},t}} q_{t,\ell,i}. \tag{6}$$

Because each layer-specific target distribution is normalized before averaging, every active layer contributes equal total probability mass. The resulting target distribution q_t combines the candidate assessments across layers, assigning higher probability to candidates that multiple layers identify as intervention-sensitive. The next section describes how we use this target distribution to guide logit correction with adaptive strength.

3.4 Gated Token-Level Decoding

The target distribution q_t from Section 3.3 assigns probabilities to candidates based on their intervention sensitivity. We now describe how to use this distribution to correct the clean logits with adaptive strength, and present the unified token selection rule that applies stage-specific corrections at both initial and continuation steps.

To use the target distribution for token selection, we compare it with the clean distribution over the candidate set, $p_t = \mathrm{softmax}(z_t[\mathcal{C}_t])$. The residual $q_t - p_t$ specifies the direction of the logit correction: candidates with higher probabilities in the target distribution than in the clean distribution are promoted, and those with lower probabilities are demoted. Treating q_t as a fixed target, the cross-entropy $\mathcal{L}_t(z) = -\sum_{i \in \mathcal{C}_t} q_{t,i} \log \mathrm{softmax}(z[\mathcal{C}_t])_i$ has gradient $p_t - q_t$ at the clean logits. Thus, the continuation-step update corresponds to a single gradient descent step in logit space with step size λg_t. However, applying this correction with uniform strength at every continuation step risks overcorrecting when the intervention produces weak or ambiguous signals. We regulate correction strength using the average of target concentration and clean–negative divergence. These measures describe how concentrated the target distribution is and how much the intervention changes the prediction.

We quantify these two measures and combine them into a single gate value. For $|\mathcal{C}_t| > 1$, we measure concentration using normalized entropy, $\mathrm{Sharp}(q_t) = 1 - \mathcal{H}(q_t)/\log|\mathcal{C}_t|$, where $\mathcal{H}$ denotes entropy. A sharp target distribution (high $\mathrm{Sharp}(q_t)$) indicates that the alignment scores strongly favor a few candidates. We measure the intervention's impact using normalized Jensen–Shannon divergence between the clean and negative distributions over the candidate set, $J_t = \mathrm{JS}(p_t \| p_t^-)/\log 2$, where $p_t^- = \mathrm{softmax}(n_{t,\ell_f}[\mathcal{C}_t])$ and ℓ_f is the final readout layer. A large divergence indicates that the soft intervention substantially changes the model's prediction. Combining these two complementary signals through a simple average yields the gate value:

$$g_t = \mathrm{clip}\left(\frac{\mathrm{Sharp}(q_t) + J_t}{2}, 0, 1\right). \tag{7}$$

Here, $\mathrm{clip}(\cdot, 0, 1)$ clamps the result to $[0, 1]$, and all logarithms are natural. The gate scales the logit correction without changing the candidate preferences encoded by $q_t - p_t$. When no layer is active, we set $q_t = p_t$ and $g_t = 0$, retaining the clean prediction. For $|\mathcal{C}_t| = 1$, we select the sole candidate without computing the gate value.

We now present the unified token selection rule that integrates both initial-step and continuation-step corrections. At the initial step, we apply direct logit contrast using the hard intervention's negative logits z_0^- to strongly discourage predictions with weak visual support. During continuation steps, we instead use the target distribution and adaptive gate to apply selective, strength-controlled corrections. We use the same candidate set $\mathcal{C}_t$ derived from the clean prediction in both cases:

$$s_{t,i} = \begin{cases} z_{t,i} + \alpha(z_{t,i} - z_{t,i}^-), & i \in \mathcal{C}_t,\ t = 0, \\ z_{t,i} + \lambda g_t(q_{t,i} - p_{t,i}), & i \in \mathcal{C}_t,\ t \geq 1, \\ -\infty, & i \notin \mathcal{C}_t. \end{cases} \tag{8}$$

Here, α scales the initial-step logit contrast, while λ scales the gated correction at continuation steps. The correction is added to the clean logits $z_{t,i}$, which serve as the base scores. At the initial step, the correction term $\alpha(z_{t,i} - z_{t,i}^-)$ amplifies the difference between clean and negative logits. During

continuation, the correction term $\lambda g_t(q_{t,i} - p_{t,i})$ shifts the clean logits toward the target distribution, with strength modulated by the adaptive gate. Candidates outside $\mathcal{C}_t$ are excluded from selection by assigning them $-\infty$ scores. We greedily select $y_t = \arg\max_{i \in \mathcal{V}} s_{t,i}$.

Finally, we describe the execution strategy that enables efficient computation of both clean and negative predictions. We maintain a single persistent clean cache along the generated prefix. At the initial step, we execute the clean prefill and the negative prefill with the hard intervention separately, retaining only the clean cache. At each continuation step, the clean and negative forward passes process the same most recently selected token from the same incoming clean cache, with the soft intervention applied only to the negative branch. Only the updated clean cache persists; the negative cache is discarded after token selection. This shared-prefix execution ensures that the clean and negative predictions differ only in their access to visual information at the current step, isolating the intervention's effect. Decoding requires one clean and one negative forward pass per step with KV caching after prefill, and model parameters remain frozen throughout.

4 Experiments

4.1 Experimental Setup

Models and benchmarks. We evaluate our method on LLaVA-1.5-7B (Liu et al., 2023; 2024) and InstructBLIP (Dai et al., 2023). We consider both open-ended generation and discriminative object-existence evaluation. For open-ended generation, we evaluate object hallucination using CHAIR (Rohrbach et al., 2018). Following prior evaluation protocols (Wang et al., 2024; Huang et al., 2024; Yue et al., 2024), we randomly sample 500 images from the MS COCO validation set (Lin et al., 2014) and use the same images for all compared methods. For discriminative evaluation, we use POPE (Li et al., 2023b) on COCO, GQA, and A-OKVQA. Each dataset contains Random, Popular, and Adversarial subsets of 3,000 questions. We first compute each metric independently on every dataset and then report the macro-average across the three datasets for each sampling category.

Baselines and implementation. We compare with the unmodified greedy decoder (Clean) and four representative training-free decoding methods. VCD obtains a negative prediction by perturbing the input image (Leng et al., 2024), ICD derives it from a contrasting instruction (Wang et al., 2024), and SID uses attention-guided visual-token restriction to construct the negative branch (Huo et al., 2025). AvisC uses attention analysis for visual calibration (Woo et al., 2025). Together, these baselines cover image-level perturbation, instruction-level contrast, and attention-guided visual intervention. All methods are evaluated on the same examples under the corresponding backbone's generation and evaluation protocol. All experiments are performed at inference time with frozen model parameters and greedy decoding. For LLaVA-1.5-7B, we use $K = 100$ visual tokens, an initial-step correction scale of $\alpha = 1.5$, a soft-intervention attenuation factor of $\rho = 0.5$, and a continuation-step correction scale of $\lambda = 3$. For InstructBLIP, we set $K = 3$, $\alpha = 0.5$, and $\lambda = 1$. Candidate-set construction uses $\beta = 0.1$ and retains at most $I = 10$ tokens. Once selected for a backbone, each configuration is held fixed across its CHAIR and POPE evaluations.

4.2 Main Results

Tables 1 and 2 compare the complete decoder with the clean model and four training-free baselines on object-existence evaluation and open-ended captioning. Our method performs strongly across POPE sampling regimes, with the clearest object-existence gains on LLaVA under Popular and Adversarial sampling, and obtains the lowest sentence-level hallucination rate on both backbones.

Object-existence evaluation. Table 1 summarizes Accuracy and F1 on POPE, macro-averaged over COCO, GQA, and A-OKVQA. On LLaVA-1.5-7B, our method achieves the highest Accuracy in all three sampling regimes. Compared with SID, the gains increase from 0.19 percentage points on Random to 1.31 on Popular and 2.51 on Adversarial. Our method also obtains the highest F1 on Popular and Adversarial, while remaining close to SID on Random. In particular, it raises Adversarial Accuracy from 72.56% with the clean decoder to 78.10%, demonstrating stronger discrimination when absent objects are selected to align with common or image-related priors.

Table 1: POPE accuracy and F1, macro-averaged across the three datasets.

Backbone	Method	Random Acc.	Random F1	Popular Acc.	Popular F1	Adversarial Acc.	Adversarial F1
LLaVA-1.5-7B	Clean	86.97	87.79	80.20	82.68	72.56	77.42
	VCD	84.26	85.46	77.59	80.59	71.22	76.36
	ICD	85.81	86.79	80.66	82.91	72.78	77.46
	SID	88.08	**88.45**	81.89	83.54	75.59	78.97
	AvisC	86.69	87.49	79.34	81.99	71.96	76.97
	Ours	**88.27**	88.27	**83.20**	**84.07**	**78.10**	**80.19**
InstructBLIP	Clean	88.72	88.88	79.80	81.73	74.54	78.09
	VCD	87.28	87.31	79.48	81.02	75.07	77.92
	ICD	81.24	83.60	70.48	76.30	65.56	73.47
	SID	88.77	88.34	82.21	82.76	77.53	79.24
	AvisC	**89.12**	**89.01**	81.88	82.98	76.51	79.09
	Ours	88.86	88.35	**82.83**	**83.15**	**78.04**	**79.49**

On InstructBLIP, CAVCD achieves the highest Accuracy and F1 on Popular and Adversarial subsets, while AvisC leads on Random. It improves Accuracy over clean decoding by 3.03 and 3.50 percentage points on Popular and Adversarial, respectively, and also outperforms SID on both metrics in these settings. The gains demonstrate that hard visual restriction at the initial step reduces hallucination from the outset, while adaptive candidate-aware correction during continuation maintains this improvement without compromising prediction quality.

Open-ended object hallucination. Table 2 reports the CHAIR results. On LLaVA-1.5-7B, CAVCD achieves the lowest CHAIRs and CHAIRi among the compared decoders. Relative to the clean decoder, it reduces CHAIRs from 46.0% to 38.4% and CHAIRi from 12.2% to 10.1%. Compared with SID, CHAIRs decreases by 7.2 percentage points and CHAIRi by 0.9 points; compared with AvisC, CHAIRs decreases by 16.8 points. On InstructBLIP, CAVCD achieves the lowest CHAIRs and CHAIRi among the compared methods, reducing them from 56.6% and 22.6% with clean decoding to 44.4% and 11.5%, respectively. Compared with ICD, the strongest baseline on both metrics, CAVCD lowers CHAIRs by 9.6 percentage points and CHAIRi by 3.1 points. The consistent gains across both models stem from CAVCD's candidate-aware assessment: by evaluating how each candidate responds to visual intervention relative to its loss gradient, CAVCD distinguishes tokens that over-rely on spurious visual correlations from those grounded in actual image content. The adaptive gate further prevents overcorrection by selectively adjusting only hallucination-indicative tokens, rather than uniformly penalizing all changes in the negative prediction.

Table 2: CHAIR results (%). Lower is better.

	LLaVA-1.5-7B		InstructBLIP	
Method	$C_S \downarrow$	$C_I \downarrow$	$C_S \downarrow$	$C_I \downarrow$
Clean	46.0	12.2	56.6	22.6
VCD	45.4	12.2	63.6	19.3
ICD	50.8	13.1	54.0	14.6
SID	45.6	11.0	60.8	16.3
AvisC	55.2	14.1	64.0	17.0
Ours	**38.4**	**10.1**	**44.4**	**11.5**

4.3 Ablation Studies

We conduct CHAIR ablations on LLaVA-1.5-7B to examine candidate-aware response assessment, visual-token selection, and stage assignment. Unless otherwise specified, variants retain the hard intervention at the initial step and the soft intervention at continuation steps, and restrict token selection to the candidate set derived from the clean prediction.

Response assessment and gated logit correction. Table 3 compares logit corrections at continuation steps under the same visual conditions. Final-layer logit contrast replaces the correction based on the target distribution and adaptive gate with $z_t + 0.5(z_t - z_t^-)$; the candidate log-probability difference variant retains the target distribution construction and adaptive gate but replaces the alignment score with $\log \mathrm{softmax}(c_{t,\ell})_i - \log \mathrm{softmax}(n_{t,\ell})_i$. The complete method reaches 38.4% CHAIRs, compared with 45.6% for final-layer logit contrast and 38.8% for the candidate log-probability difference variant. Using only the final-layer target distribution raises CHAIRs to 53.6%,

Table 3: Ablations of response assessment and gated logit correction.

Continuation-step correction	CHAIRs (%) ↓	CHAIRi (%) ↓
CAVCD	**38.4**	**10.1**
Direct logit contrast ($\alpha = 0.5$)	45.6	12.2
Log-probability difference	38.8	10.6
Final-layer target only	53.6	13.3
Fixed gate ($g = 0.5$)	39.0	10.2

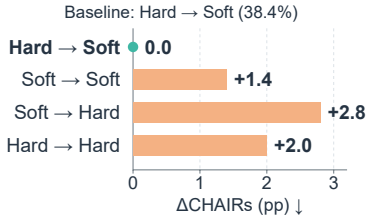


Figure 3: Effect of visual intervention assignment.

while the best-performing tested fixed gate ($g = 0.5$) yields 39.0%, compared with 38.4% for adaptive gating. Under the same visual interventions, assessing candidate responses across layers yields lower hallucination rates than applying final-layer logit contrast directly. Multi-layer assessment and gradient-based alignment better distinguish hallucination-prone tokens from benign prediction shifts, while adaptive gating tailors correction strength to each step.

Candidate selection and visual-token selection. The Q75 threshold on alignment scores reaches 38.4% CHAIRs, compared with 39.8% when the threshold is zero; the relative threshold allows each layer to contribute candidates according to its own response distribution, avoiding weakly affected tokens. For visual-token selection, retaining the least-attended keys at the initial step yields 38.4% CHAIRs versus 40.0% for the most-attended keys, as removing salient visual regions forces the model to reveal its uncertainty before committing to a prediction. At continuation steps, attenuating the most-attended keys yields 38.4% CHAIRs, compared with Bottom-100 and Random-100 attenuation at 44.0% and 46.8%, respectively; targeting highly attended regions more effectively exposes over-reliance on specific visual cues. The latter alternatives also increase CHAIRi from 10.1% to 12.7% and 12.6%, with mean generation lengths of 114.4, 114.5, and 116.5 tokens for Top-100, Bottom-100, and Random-100, respectively.

Table 4: Visual-token selection (%). Lower is better; * marks the default.

Selection	CHAIRs ↓	CHAIRi ↓
Hard: retained keys ($t = 0$)		
Bottom-100*	**38.4**	**10.1**
Top-100	40.0	**10.1**
Soft: attenuated keys ($t \geq 1$)		
Top-100*	**38.4**	**10.1**
Bottom-100	44.0	12.7
Random-100	46.8	12.6

Component and intervention assignment. Applying correction only at the initial step yields 44.2% CHAIRs, while applying it only during continuation yields 39.8%; combining both gives 38.4%. With correction rules fixed, Figure 3 shows the increase in CHAIRs relative to Hard-to-Soft (38.4%). Soft-to-Soft, Hard-to-Hard, and Soft-to-Hard increase CHAIRs by 1.4, 2.0, and 2.8 percentage points, respectively, supporting the proposed intervention schedule. The Hard-to-Soft pattern matches the generation process: hard restriction at the initial step prevents the model from anchoring on incomplete visual evidence, while soft attenuation during continuation allows adaptive correction without disrupting the established context. Complete results and the switching-position analysis are provided in Appendix B.

5 Conclusion

We presented CAVCD, a training-free decoding framework for mitigating object hallucinations in large vision-language models. CAVCD combines hard-to-soft visual intervention with candidate-aware target construction and gated token correction. During continuation, it assesses intervention-induced responses for individual candidates across decoder layers and selectively uses them to guide logit correction, while an adaptive gate regulates correction strength. Experiments show reduced object hallucination and improved object-existence accuracy, with ablation studies supporting the contributions of the proposed components. These findings highlight the importance of both assessing negative-branch responses and regulating their influence on token selection.

AI USE STATEMENT

In this work, we used generative AI tools to assist in implementing the research methods. We did not use generative AI tools for other tasks requiring disclosure under the ICLR 2027 AI Policy for Authors. Additionally, we used generative AI tools to improve the readability of the manuscript, assist in writing and editing code, and summarize and analyze existing literature. The authors reviewed and revised all AI-assisted text, checked literature summaries and interpretations against the original sources, and verified and tested AI-assisted code and method implementations. We take responsibility for the final content of this work, including text, claims, code, and other artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work evaluates an inference-time decoding method using established image and visual question answering benchmarks. We did not recruit human participants or collect new personal data. Although CAVCD reduces object hallucination on the evaluated benchmarks, it does not guarantee that generated descriptions or answers are accurate. Outputs should be checked before use in settings where incorrect visual claims could cause harm.

REPRODUCIBILITY STATEMENT

The methodology specifies the visual interventions, candidate selection, target construction, and decoding rules. Section 4 describes the evaluation protocols and model configurations, and the appendix reports additional ablations, parameter sensitivity, and efficiency measurements. An anonymized implementation of the core CAVCD method is provided in the supplementary materials.

REFERENCES

Nora Belrose, Igor Ostrovsky, Lev McKinney, Zach Furman, Logan Smith, Danny Halawi, Stella Biderman, and Jacob Steinhardt. Eliciting latent predictions from transformers with the tuned lens. *arXiv preprint arXiv:2303.08112*, 2023. URL https://arxiv.org/abs/2303.08112v6.

Zhaorun Chen, Zhuokai Zhao, Hongyin Luo, Huaxiu Yao, Bo Li, and Jiawei Zhou. HALC: Object hallucination reduction via adaptive focal-contrast decoding. In *International Conference on Machine Learning*, 2024.

Yung-Sung Chuang, Yujia Xie, Hongyin Luo, Yoon Kim, James Glass, and Pengcheng He. DoLa: Decoding by contrasting layers improves factuality in large language models. In *International Conference on Learning Representations*, 2024.

Wenliang Dai, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale Fung, and Steven Hoi. InstructBLIP: Towards general-purpose vision-language models with instruction tuning. In *Advances in Neural Information Processing Systems*, 2023.

Qidong Huang, Xiaoyi Dong, Pan Zhang, Bin Wang, Conghui He, Jiaqi Wang, Dahua Lin, Weiming Zhang, and Nenghai Yu. OPERA: Alleviating hallucination in multi-modal large language models via over-trust penalty and retrospection-allocation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

Fushuo Huo, Wenchao Xu, Zhong Zhang, Haozhao Wang, Zhicheng Chen, and Peilin Zhao. Self-introspective decoding: Alleviating hallucinations for large vision-language models. In *International Conference on Learning Representations*, 2025.

Sicong Leng, Hang Zhang, Guanzheng Chen, Xin Li, Shijian Lu, Chunyan Miao, and Lidong Bing. Mitigating object hallucinations in large vision-language models through visual contrastive decoding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

Xiang Lisa Li, Ari Holtzman, Daniel Fried, Percy Liang, Jason Eisner, Tatsunori B. Hashimoto, Luke Zettlemoyer, and Mike Lewis. Contrastive decoding: Open-ended text generation as optimization. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*, 2023a.

Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Wayne Xin Zhao, and Ji-Rong Wen. Evaluating object hallucination in large vision-language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, 2023b.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: Common objects in context. In *European Conference on Computer Vision*, 2014.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In *Advances in Neural Information Processing Systems*, 2023.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

Anna Rohrbach, Lisa Anne Hendricks, Kaylee Burns, Trevor Darrell, and Kate Saenko. Object hallucination in image captioning. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2018.

Zhiqing Sun, Sheng Shen, Shengcao Cao, Haotian Liu, Chunyuan Li, Yikang Shen, Chuang Gan, Liang-Yan Gui, Yu-Xiong Wang, Yiming Yang, Kurt Keutzer, and Trevor Darrell. Aligning large multimodal models with factually augmented RLHF. In *Findings of the Association for Computational Linguistics: ACL 2024*, 2024.

Xintong Wang, Jingheng Pan, Liang Ding, and Chris Biemann. Mitigating hallucinations in large vision-language models with instruction contrastive decoding. In *Findings of the Association for Computational Linguistics: ACL 2024*, 2024.

Sangmin Woo, Donguk Kim, Jaehyuk Jang, Yubin Choi, and Changick Kim. Don't miss the forest for the trees: Attentional vision calibration for large vision language models. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 1927–1951, 2025. doi: 10.18653/v1/2025.findings-acl.99.

Shukang Yin, Chaoyou Fu, Sirui Zhao, Tong Xu, Hao Wang, Dianbo Sui, Yunhang Shen, Ke Li, Xing Sun, and Enhong Chen. Woodpecker: Hallucination correction for multimodal large language models. *Science China Information Sciences*, 67(12):220105, 2024. doi: 10.1007/s11432-024-4251-x.

Qifan Yu, Juncheng Li, Longhui Wei, Liang Pang, Wentao Ye, Bosheng Qin, Siliang Tang, Qi Tian, and Yueting Zhuang. Hallucidoctor: Mitigating hallucinatory toxicity in visual instruction data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024a.

Tianyu Yu, Yuan Yao, Haoye Zhang, Taiwen He, Yifeng Han, Ganqu Cui, Jinyi Hu, Zhiyuan Liu, Hai-Tao Zheng, Maosong Sun, and Tat-Seng Chua. RLHF-V: Towards trustworthy MLLMs via behavior alignment from fine-grained correctional human feedback. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024b.

Zihao Yue, Liang Zhang, and Qin Jin. Less is more: Mitigating multimodal hallucination from an EOS decision perspective. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*, 2024.

Yiyang Zhou, Chenhang Cui, Jaehong Yoon, Linjun Zhang, Zhun Deng, Chelsea Finn, Mohit Bansal, and Huaxiu Yao. Analyzing and mitigating object hallucination in large vision-language models. In *International Conference on Learning Representations*, 2024.

Algorithm 1 CAVCD: Candidate-Aware Visual Counterfactual Decoding

Input: Image x, instruction u, model θ, parameters K, ρ, α, λ, β, I
Output: Generated sequence y

// Initial step ($t = 0$): Hard intervention
Select $\mathcal{R}_0 \leftarrow$ BottomK of visual tokens by attention at layer ℓ_s
Obtain z_0 (clean) and z_0^- (negative, retaining only $\mathcal{R}_0$)
Build candidate set $\mathcal{C}_0$ from z_0 (Eq. equation 2)
Select $y_0 \leftarrow \arg\max_i \left[z_{0,i} + \alpha(z_{0,i} - z_{0,i}^-)\right]_{i \in \mathcal{C}_0}$
Set $t \leftarrow 0$, $y \leftarrow (y_0)$; retain the clean prefill cache and check stopping conditions

// Continuation steps ($t \geq 1$): Soft intervention + candidate-aware correction
while not done **do**
    $t \leftarrow t + 1$
    Select $\mathcal{S}_t \leftarrow$ TopK of visual tokens by attention at layer ℓ_s
    Obtain z_t (clean) and hidden states $\{h_{t,\ell}, h_{t,\ell}^-\}_{\ell \in \mathcal{L}}$ (negative with attenuation ρ on $\mathcal{S}_t$)
    Both branches process y_{t-1} from the same incoming clean cache; retain only the updated clean cache
    Build candidate set $\mathcal{C}_t$ from z_t
    Set $p_t \leftarrow \mathrm{softmax}(z_t[\mathcal{C}_t])$, $q_t \leftarrow p_t$, $g_t \leftarrow 0$
    if $|\mathcal{C}_t| = 1$ **then**
        Set y_t to the sole candidate
    else
        Initialize $\mathcal{L}_{\mathrm{act},t} \leftarrow \varnothing$
        for each layer $\ell \in \mathcal{L}$ **do**
            Compute stabilized alignment scores $u_{t,\ell,i}$ (Eq. equation 4)
            Set $\tau_{t,\ell} \leftarrow Q_{75}(\{u_{t,\ell,i}\}_{i \in \mathcal{C}_t})$, $w_i \leftarrow [\max(u_{t,\ell,i} - \tau_{t,\ell}, 0)]^2$
            if $\sum_{i \in \mathcal{C}_t} w_i > 10^{-12}$ **then**
                Set $q_{t,\ell,i} \leftarrow w_i / \sum_{j \in \mathcal{C}_t} w_j$; add ℓ to $\mathcal{L}_{\mathrm{act},t}$
            end if
        end for
        if $\mathcal{L}_{\mathrm{act},t} \neq \varnothing$ **then**
            Average active-layer targets to obtain q_t (Eq. equation 6)
            Set $p_t^- \leftarrow \mathrm{softmax}(n_{t,\ell_f}[\mathcal{C}_t])$ and compute g_t (Eq. equation 7)
        end if
        Select $y_t \leftarrow \arg\max_{i \in \mathcal{C}_t}[z_{t,i} + \lambda g_t(q_{t,i} - p_{t,i})]$
    end if
    Append y_t to y and check stopping conditions
end while
return y

A Complete Algorithm

Algorithm 1 presents the complete CAVCD decoding procedure. The algorithm distinguishes between the initial step, which uses hard intervention and direct logit contrast, and continuation steps, which use soft intervention with candidate-aware response assessment and adaptive gating.

Numerical stability and fallback. We compute alignment scores in FP32 over the full vocabulary, with log-probability shifts obtained using `log_softmax`. Alignment scores are set to zero when the cosine denominator does not exceed 10^{-12}. Within each layer, the 75th-percentile threshold is computed over candidate alignment scores using linear interpolation. A layer is active only if the sum of squared positive excesses above this threshold exceeds 10^{-12}; otherwise, it is excluded from normalization and aggregation. We normalize the weights within each active layer and average the resulting distributions with equal weight. If no layer is active, we set $q_t = p_t$ and $g_t = 0$, yielding no correction and retaining the clean prediction. A singleton candidate set also produces zero excess above the threshold and therefore selects its sole candidate without correction.

Layer configurations. All layer indices are zero-based decoder-block indices, and ranges are inclusive. For LLaVA-1.5-7B, both hard and soft interventions select visual tokens using attention from block 1 and modify attention in blocks 2–31. For InstructBLIP-Vicuna-7B, the initial hard intervention uses block 4 for selection and operates in blocks 5–31, whereas the continuation soft

Table 5: Visual intervention assignment with fixed correction rules. All metrics are percentages.

Initial → continuation	CHAIRs ↓	CHAIRi ↓
Hard → Soft	**38.4**	**10.13**
Soft → Soft	39.8	10.17
Hard → Hard	40.4	10.97
Soft → Hard	41.2	11.18

Table 6: Adaptive versus fixed gating. All metrics are percentages.

Gate	CHAIRs ↓	CHAIRi ↓
Adaptive (default)	**38.4**	**10.13**
$g = 0.25$	40.6	10.62
$g = 0.50$	39.0	10.17
$g = 0.75$	39.4	10.14
$g = 1.00$	41.6	11.05

intervention uses block 1 for selection and operates in blocks 2–31. Both backbones extract readouts from the outputs of blocks 8, 16, 24, and 31, with block 31 providing the final-layer negative prediction used to compute the gate. LLaVA-1.5-7B has 576 CLIP patch tokens and uses a visual-token budget of $K = 100$; InstructBLIP-Vicuna-7B has 32 Q-Former query tokens and uses $K = 3$. Both use $\rho = 0.5$ for soft attenuation.

B Additional Ablations

All comparisons in this section use LLaVA-1.5-7B with CHAIR random500. These ablations examine the effects of modifying the predefined configuration; their results were not used to select the default settings.

Visual intervention assignment with fixed correction rules. Table 5 reports the complete results underlying Figure 3. Only the visual intervention at the initial and continuation steps is changed: Hard retains low-attention visual keys, whereas Soft attenuates high-attention visual keys. All variants retain direct logit contrast at the initial step and candidate-aware gated correction during continuation, with the remaining settings unchanged.

Hard-to-Soft achieves the lowest CHAIRs and CHAIRi among the tested assignments. Relative to Soft-to-Soft and Hard-to-Hard, it lowers CHAIRs by 1.4 and 2.0 percentage points, respectively. Because the correction rules are held fixed, these comparisons isolate the effect of visual intervention assignment.

Adaptive versus fixed gating. Table 6 compares the adaptive gate with four fixed values. All variants use the same target construction and visual interventions, with $\lambda = 3$; only g_t in $\lambda g_t(q_t - p_t)$ is replaced by a constant. The default Hard-to-Soft result and the previously evaluated $g = 1$ result are reused.

Adaptive gating has the lowest CHAIRs and CHAIRi among the tested settings. The closest fixed setting in CHAIRs is $g = 0.5$, which yields 39.0% versus 38.4% for adaptive gating.

Candidate-relative threshold. Table 7 reports alternative thresholds on alignment scores for constructing the layer-specific target distributions. The zero-threshold variant sets the cutoff to zero but retains positive-part squaring, per-layer target normalization, cross-layer aggregation, and the adaptive gate. Q50 and Q90 change only the quantile cutoff.

Among the tested thresholds, Q75 gives the lowest CHAIRs and CHAIRi. The zero-threshold variant still sets negative alignment scores to zero, squares the remaining scores, and normalizes them into layer-specific target distributions, yet its CHAIRs is 1.4 points higher than Q75. Moving the

Table 7: Effects of candidate-relative thresholds on alignment scores.

Alignment-score threshold	CHAIRs (%) ↓	CHAIRi (%) ↓
Zero threshold	39.8	10.6
Q50	41.8	10.9
Q75 (default)	**38.4**	**10.1**
Q90	39.8	10.5

Table 8: Contributions of logit correction at the initial step and continuation steps.

Initial step	Continuation steps	CHAIRs (%) ↓	CHAIRi (%) ↓
Clean	Clean	46.0	12.2
Hard	Clean	44.2	12.0
Clean	Soft	39.8	10.2
Hard	**Soft**	**38.4**	**10.1**

quantile to Q50 or Q90 also increases CHAIRs, indicating that the relative cutoff matters beyond the positive-part and normalization operations shared by all variants.

Initial-step and continuation-step corrections. Table 8 replaces the logit correction at either stage with clean decoding. Hard and Soft denote the full decoding configurations: Hard uses hard intervention with direct final-layer logit contrast, and Soft uses soft intervention with candidate-aware assessment and adaptive gating. In the main text, the hard-intervention group in Table 4 separately compares the visual-token subsets retained at the initial step.

Both stages of logit correction lower CHAIRs relative to clean decoding when applied individually. Applying the Soft configuration at continuation steps accounts for the larger change in this comparison (46.0% to 39.8%), while adding the Hard configuration at the initial step further lowers CHAIRs to 38.4%. The combined configuration also has the lowest CHAIRi of the four, supporting logit correction at both stages.

Joint configuration assignment and switching position. Unlike Table 5, this comparison changes both the visual intervention and the correction rule. The Hard configuration combines hard intervention with direct final-layer logit contrast, while the Soft configuration combines soft intervention with candidate-aware gated correction. Table 9 reports assignments of these complete configurations and switching positions between them; it does not isolate the effect of visual intervention alone.

Table 9(a) shows that applying Soft at continuation steps yields lower CHAIRs than applying Hard, regardless of the initial-step configuration. With Hard at the initial step, the difference is 13.0 percentage points (38.4% versus 51.4%). In panel (b), switching from Hard to Soft after the initial step gives the lowest CHAIRs among the tested switching positions; extending the number of steps using Hard does not further reduce CHAIRs. The two-step schedule matches the one-step schedule in CHAIRi at the reported precision but has a higher CHAIRs. These results support the predefined one-step switching schedule.

C Parameter Sensitivity

Table 10 reports one-parameter-at-a-time comparisons on LLaVA-1.5-7B with CHAIR random500. The default configuration uses $K = 100$, $\rho = 0.5$, $\lambda = 3$, and a Q75 threshold on alignment scores. Each variant changes only the named parameter. We report generation length and object-mention counts alongside CHAIRs and CHAIRi to document the resulting outputs.

The default setting gives the lowest CHAIRs among the tested ρ and λ values. Increasing K to 150 matches its CHAIRs but does not improve CHAIRi, whereas reducing K to 50 increases CHAIRs. With $\rho = 0.75$, CHAIRi is slightly lower but CHAIRs is higher than the default; the stronger

Table 9: Joint intervention-and-correction configuration assignment and switching-position analysis.

(a) Initial-step and continuation-step configurations			
Initial step → continuation steps	CHAIRs (%)	CHAIRi (%)	Mean tokens
Soft → Soft	39.6	10.2	115.2
Hard → Soft	**38.4**	**10.1**	114.4
Soft → Hard	49.6	12.2	119.4
Hard → Hard	51.4	12.4	118.3
Clean (reference)	46.0	12.2	116.5
(b) Number of leading steps using Hard			
Schedule	CHAIRs (%)	CHAIRi (%)	Mean tokens
$k = 0$	39.6	10.2	115.2
$k = 1$	**38.4**	**10.1**	114.4
$k = 2$	39.4	**10.1**	114.3
$k = 4$	40.8	10.8	114.9
$k = 8$	40.8	11.5	114.1
Hard throughout generation	51.4	12.4	118.3

Table 10: Sensitivity to the visual-token budget and correction parameters.

Configuration	CHAIRs (%) ↓	CHAIRi (%) ↓	Mean tokens	Object mentions
Default	38.4	10.1	114.4	3751
$K = 50$	41.6	10.3	112.4	3724
$K = 150$	38.4	10.3	113.7	3693
$\rho = 0.25$	42.0	10.8	114.6	3766
$\rho = 0.75$	39.6	10.0	114.2	3720
$\lambda = 1$	42.2	11.3	112.9	3751
$\lambda = 5$	39.4	10.4	113.6	3772

attenuation at $\rho = 0.25$ raises both hallucination measures. Changing λ to 1 or 5 also raises CHAIRs. These comparisons characterize sensitivity around the predefined configuration.

Readout-layer sensitivity. Table 11 compares three readout-layer sets on LLaVA-1.5-7B using the same 500-image CHAIR subset. All indices are zero-based decoder-block indices. Each set contains four layers, including the final block (31). Only the readout set used for alignment assessment and target construction changes; the selector and intervention layers, correction parameters, and remaining generation settings are held fixed. The final-layer distributions used to compute the gate always come from block 31.

Table 11: Sensitivity to readout-layer placement. CHAIRs and CHAIRi are percentages.

Readout layers	CHAIRs ↓	CHAIRi ↓	Mean tokens
[8,16,24,31] (default)	**38.4**	**10.13**	114.43
[4,12,20,31]	41.4	10.96	114.56
[12,20,28,31]	40.4	10.44	118.90

The predefined set [8,16,24,31] achieves the lowest CHAIRs and CHAIRi among the tested configurations. Moving the intermediate readouts earlier or later increases CHAIRs from 38.4% to 41.4% and 40.4%, respectively. The earlier configuration produces nearly the same average generation length, suggesting that its higher hallucination rate is not simply explained by longer outputs. These results indicate that readout-layer placement affects correction quality.

Table 12: Inference efficiency on the fixed 100-image CHAIR subset. Latency is the mean $\pm$ standard deviation across three repeats; other values are measured averages, except memory, which reports the peak.

Method	Latency (s/image)	TTFT (ms)	Continuation (ms/token)	Peak alloc./res. (GiB)	Output tokens
Clean	4.283 ± 0.103	174.3	35.21	14.01 / 14.18	117.69
VCD	8.424 ± 0.020	348.8	69.41	14.81 / 15.04	117.35
SID	8.546 ± 0.232	348.5	71.70	15.72 / 15.91	115.32
CAVCD	9.359 ± 0.039	351.6	78.51	14.74 / 15.04	115.74

D Inference Efficiency

We measure inference efficiency on the same fixed 100-image CHAIR subset using LLaVA-1.5-7B, with three serial repeats on one RTX 3090. All methods use native stopping conditions; output-token counts include EOS. Timing excludes image file I/O and image/prompt preprocessing but includes visual encoding, prefill, all method-specific forward passes, and decoding control. Time to first token (TTFT) includes the initial step. Continuation latency divides the total time after the first token by the number of subsequent generated tokens. Peak CUDA allocated and reserved memory are the maximum per-sample peaks measured after warmup and include model weights.

Table 12 shows that CAVCD takes 2.19 times as long as clean decoding per image and 1.11 and 1.10 times as long as VCD and SID, respectively. The methods generate similar numbers of tokens, and CAVCD's TTFT is close to those of VCD and SID; its additional time is primarily observed during continuation. Its peak allocated memory is 0.73 GiB above Clean, 0.07 GiB below VCD, and 0.98 GiB below SID. SID uses FastV attention masking rather than physical token removal. Clean, VCD, and SID use the frozen SID-main author runtime, whereas CAVCD uses the current project runtime. The timings therefore compare the available implementations and do not isolate the cost of CAVCD's algorithm from runtime differences.