EFFICIENT MULTIMODAL CONTRASTIVE DECODING VIA BUDGETED ACTIVATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Multimodal contrastive decoding (MCD) has been widely adopted to mitigate hallucination in multimodal large language models (MLLMs). It is performed by contrasting predictions from an original input with those from an auxiliary branch. The inference latency thus increases to at least $2\times$ that of vanilla decoding. However, our preliminary experiments show that MCD sparsely alters the output token (e.g., $\sim 7\%$ on POPE). This mismatch between computational cost and rate of altered tokens motivates a token-level analysis of MCD's activation. Accordingly, we introduce FLIPTRACE, a theoretical framework that characterizes MCD's effect as *flips*. We then prove a flip-immunity condition that bounds how many tokens MCD can affect and show that these effects concentrate where the bound predicts. Together, these findings lead to *flip susceptibility*, a statistic computed from the original logits alone that predicts token flips. Using this measure, we develop a training-free budgeted activation method. Within this budget, our method prioritizes MCD activation on decoding steps with high flip susceptibility. We evaluate our method across diverse combinations of six MCD methods, four MLLM backbones, and five benchmarks, spanning visual and audio-visual tasks. Across these settings, budgeted activation integrates with existing MCD methods to achieve a maximum reduction of 87% in inference latency, while preserving their performance. Our code is anonymously released here.

1 INTRODUCTION

Multimodal large language models (MLLMs) are prone to hallucination where the generated answers are not supported by their multimodal inputs (Li et al., 2023b; Rohrbach et al., 2018). Multimodal contrastive decoding (MCD) methods mitigate hallucination at inference time by contrasting predictions under the original input with those from an auxiliary branch. This increases the relative preference for tokens supported by the original input while reducing that for tokens favored by the auxiliary branch. MCD methods differ in how they construct the auxiliary branch: Some approaches augment the inputs by suppressing multimodal inputs (Leng et al., 2024; Im et al., 2026; Chen et al., 2024; Kan et al., 2024; Favero et al., 2024; Park et al., 2025a; Suo et al., 2025), or modifying textual instructions (Wang et al., 2024). Other methods obtain auxiliary predictions by modifying or leveraging model-internal signals (Jung et al., 2025; Huo et al., 2025; Chuang et al., 2024).

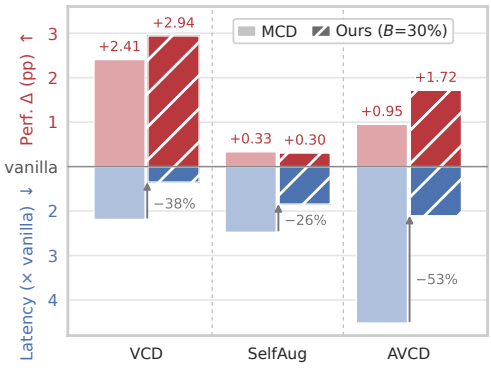


Figure 1: Comparison of performance *gain* over vanilla (red bars, top) and latency normalized to vanilla (blue bars, bottom). Our method significantly reduces MCD latency, while preserving and even surpassing the performance of MCD methods.

In general, *existing MCD methods share a common decoding paradigm: At each step, both a vanilla forward pass and an auxiliary forward pass are evaluated*. This incurs substantial computational overhead, with measured inference latency at least $2\times$ that of vanilla decoding, as shown in Fig. 1. However, our preliminary analysis shows that the auxiliary branch changes the selected token at

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.

only a small fraction of decoding steps. We refer to token decisions that differ from those of vanilla decoding as "flips". For example, on the POPE benchmark, VCD (Leng et al., 2024), a representative MCD method, flips only $\sim 7\%$ of vanilla token decisions. A few studies attempt to reduce this redundancy by activating the branch only when an uncertainty score exceeds a fixed threshold (Jung et al., 2025; Huang et al., 2026; Liu, 2026). However, they are limited by two deficiencies: (i) They control auxiliary computation indirectly with *a less flexible threshold*, rather than an explicit computation budget. As a result, the realized activation rate may vary across MLLM and dataset combinations. (ii) *On a more fundamental level, existing methods fail to establish a theoretical understanding of MCD's token-level flips*. Bridging this gap is essential for understanding the mechanism underlying MCD, and for establishing a foundation for eliminating redundant auxiliary computation.

To this end, we introduce FLIPTRACE, a theoretical framework that characterizes MCD's token-level effects. We first derive a *flip-immunity condition*. Specifically, we show that MCD cannot change the selected token when the vanilla top-2 margin is sufficiently large. Formally, this occurs when the margin exceeds the maximum relative gain attainable by the auxiliary branch. Thereafter, we convert this margin into a freely available *flip susceptibility* statistic that ranks decoding steps by how readily MCD can change the selected token. Unlike prior method-specific heuristics, our flip susceptibility statistic is theoretically grounded. Based on flip susceptibility, we allocate auxiliary computation under a user-specified activation budget B, which specifies the target fraction of generation steps at which the auxiliary branch is evaluated. Our *budgeted activation* performs online calibration to track this budget. It prioritizes decoding steps with high flip susceptibility, where MCD is more likely to alter the vanilla token decision. Importantly, our method integrates directly with any underlying MCD method by preserving its original computation on activated steps.

To assess the performance-efficiency trade-off, we evaluate our budgeted activation method across six MCD methods. In addition, our experiments span four MLLM backbones and five benchmarks, covering a diverse range of visual discriminative, visual generative, visual perception, visual reasoning, audio-visual discriminative, and audio-visual generative tasks. These experiments demonstrate broad applicability and generalized effectiveness of our method: **(i) Speedup.** As shown in Fig. 1, at a 30% budget, our budgeted activation reduces latency by 38%, 26%, and 53% on VCD, SelfAug, and AVCD, respectively. This corresponds to speedups of $1.62\times$, $1.34\times$, and $2.15\times$ over the MCD counterparts. **(ii) Performance.** In general, across multiple evaluated settings, budgeted MCD at $B{=}30\%$ (Fig. 1) or 50% matches or exceeds the performance of its MCD counterpart. Budgeted AVCD at $B{=}30\%$ also improves audio-visual comprehension from F1 76.5% to 77.3%. Additionally, on MVCD, a method for in-context learning (ICL), our budgeted activation at $B{=}50\%$ matches MVCD results across one to five ICL examples.

In summary, this work makes three contributions:

1. We introduce FLIPTRACE, a framework that formalizes MCD's token-level flips. We derive a flip-immunity condition on when flips can occur and show that these flips are sparse.
2. We propose a flip susceptibility statistic that is freely available from the vanilla forward pass alone. Based on these, we introduce a training-free budgeted activation method that can be applied seamlessly to existing MCD methods.
3. Our results demonstrate substantial latency reductions while largely preserving performance across diverse combinations of MCD methods, MLLM backbones, and benchmarks. These empirical results support our theoretical characterization that MCD's token-level effects are sparse and concentrated at decoding steps with high flip susceptibility.

2 RELATED WORK

Hallucination broadly refers to model-generated content that appears plausible but is ungrounded. In MLLMs, this commonly manifests as content that is inconsistent with factual knowledge or unsupported by the multimodal inputs. Although hallucination has been argued to be an inherent limitation of generative models (Xu et al., 2024), substantial effort has been devoted to mitigating it (Bai et al., 2024). Training-based approaches improve multimodal grounding through stronger supervision, including preference feedback (Sun et al., 2024; Yu et al., 2024; 2025), hallucination-aware contrastive learning (Jiang et al., 2024), and explicit supervision of generation termination (Yue et al., 2024).

On the flip side, MCD has become the *de facto* inference-only strategy for hallucination mitigation in MLLMs (Leng et al., 2024; Pang et al., 2025; Lee et al., 2026; Yin et al., 2025). Existing MCD methods differ primarily in the construction of auxiliary branches: Some methods directly alter the inputs, either by suppressing or removing multimodal evidence (Leng et al., 2024; Chen et al., 2024; Kan et al., 2024; Park et al., 2025a; Suo et al., 2025; Im et al., 2026; An et al., 2025; Woo et al., 2024; Lee & Song, 2025; Park et al., 2025b; Zhang et al., 2025a; Lyu et al., 2024; Favero et al., 2024; Chen et al., 2025), or modifying the prompts or textual contexts to reduce reliance on language priors (Wang et al., 2024; Manevich & Tsarfaty, 2024; Kim et al., 2024; Zhang et al., 2025b). Other methods leverage model-internal information (Huo et al., 2025; Woo et al., 2025; Gong et al., 2024; Deng & Yang, 2025; Zhou et al., 2025). They construct the auxiliary branch by modifying or selecting intermediate model representations, layers, or decoding signals without explicitly perturbing the input. Although earlier methods focus on vision-language hallucination, the same contrastive principle has also been further extended to audio and audio-visual models (Jung et al., 2025; Lin et al., 2026; Li et al., 2026; Hsu et al., 2025). Together, they establish a broad range of auxiliary constructions, but share the common two-pass design, resulting in an inference latency at least $2\times$ that of the vanilla MLLM. A few prior methods have attempted to resolve this high inference latency. They activate the auxiliary branch only when a *method-specific heuristic exceeds a fixed threshold* (Jung et al., 2025; Huang et al., 2026; Liu, 2026). These thresholds need to be tuned manually, with the resulting activation rate varying drastically across MLLM-benchmark combinations. Instead, our budgeted activation method uses theoretically grounded flip susceptibility to regulate the activation rate, and is independent of model and task settings.

3 FlipTrace: A Theoretical Framework

3.1 Preliminary

Each MCD method considered here combines a vanilla pass with a method-specific auxiliary pass. Specifically, an MLLM with parameters θ generates output $y_{1:T}$ from multimodal evidence v and text prompt x. At time step t, the vanilla pass produces logits $\boldsymbol{z}_t{=}f_\theta(v, x, y_{<t})$ over a vocabulary $\mathcal{V}$. The auxiliary pass then produces logits $\tilde{\boldsymbol{z}}_t$ from a method-specific auxiliary condition. For example, `VCD` replaces the image with a copy $\tilde{v}$ perturbed by Gaussian noise, i.e., $\tilde{\boldsymbol{z}}_t{=}f_\theta(\tilde{v}, x, y_{<t})$. MCD then combines the two sets of logits as follows:

$$\boldsymbol{z}_t^{\mathrm{mcd}} = (1+\alpha)\boldsymbol{z}_t - \alpha\tilde{\boldsymbol{z}}_t, \tag{1}$$

where $\alpha > 0$ controls the contrast strength. The resulting contrast promotes tokens preferred under the vanilla branch and suppresses tokens preferred under the auxiliary branch.

Although MCD evaluates the auxiliary branch at every decoding step, the contrast does not always alter the predicted token, resulting in unnecessary inference overhead. To characterize when the contrast affects the decoding decision, we define a *token flip* as a change in the predicted token:

Definition 1 (Token flip)**.** *Let $i_t^\star{=}\arg\max_i z_{t,i}$ be the vanilla winning token and $\hat{\imath}_t{=}\arg\max_i z_{t,i}^{\mathrm{mcd}}$ the MCD winning token at step t. Step t is a token flip (or flip for short) iff $\hat{\imath}_t \neq i_t^\star$.*

3.2 Budget bound on flips

We then derive a sufficient condition under which no flip can occur. Let $\mathcal{V}_t \subseteq \mathcal{V}$ denote the candidate set over which MCD selects its output at step t, with $i_t^\star \in \mathcal{V}_t$. We define the vanilla top-2 margin as:

$$m_t = z_{t,(1)} - z_{t,(2)}, \tag{2}$$

where $z_{t,(1)}$ and $z_{t,(2)}$ are the largest and second-largest vanilla logits. Every challenger $j \in \mathcal{V}_t\backslash\{i_t^\star\}$ trails the vanilla winner by at least m_t. A flip can therefore occur only if the MCD perturbation gives a challenger enough relative contrastive gain to close this gap. To quantify this gain, we write the perturbed logit as $z_{t,j}^{\mathrm{mcd}}{=}z_{t,j} + \delta_{t,j}$, where $\delta_{t,j}{=}\alpha(z_{t,j} - \tilde{z}_{t,j})$.

Definition 2 (Relative contrastive gain)**.** *The maximum relative contrastive gain over all challengers is:*

$$\Delta_t \;=\; \max_{j\in\mathcal{V}_t\backslash\{i_t^\star\}} \big(\delta_{t,j} - \delta_{t,i_t^\star}\big) \;=\; \alpha \max_{j\in\mathcal{V}_t\backslash\{i_t^\star\}} \Big[\big(z_{t,j} - z_{t,i_t^\star}\big) - \big(\tilde{z}_{t,j} - \tilde{z}_{t,i_t^\star}\big)\Big]. \tag{3}$$

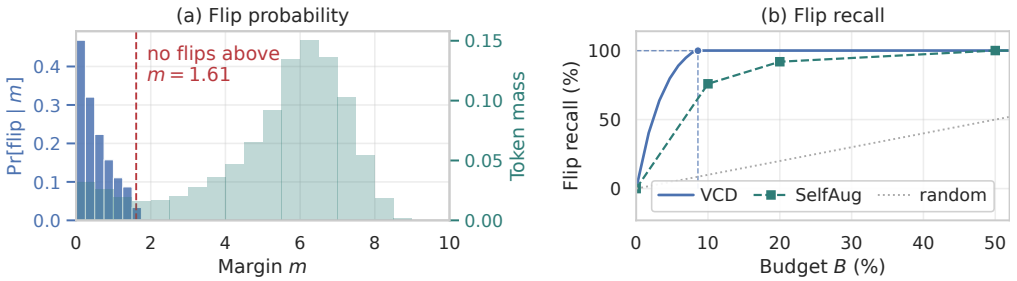


Figure 2: Analysis of MCD flips. (a) Blue bars show conditional flip probability for VCD. Green bars show the fraction of tokens in each bin. Together, these indicate that flips are sparse and concentrated in regions with low margin. (b) Teacher-forced flip recall with AUCs 95.0% and 91.0% for VCD and SelfAug. This establishes flip susceptibility as a natural statistic in the ranking of flips.

Δ_t measures the largest signed relative contrastive gain that the contrast can assign to a challenger relative to the vanilla winner. The second expression compares the winner's lead over each challenger in the auxiliary and vanilla logits. As MCD subtracts the auxiliary logits, a positive relative contrastive gain can erode the winner's lead, whereas a negative relative contrastive gain reinforces it. Comparing this maximum relative contrastive gain with the vanilla margin yields a sufficient condition for flip immunity.

Proposition 1 (Flip immunity). *If $m_t \geq \Delta_t$, then step t does not flip and $\hat{\imath}_t{=}i_t^\star$. Equivalently, every flip satisfies $m_t < \Delta_t$.*

Prop. 1 follows because every challenger must close a gap of at least m_t, while its relative contrastive gain is at most Δ_t. The condition is sufficient but not necessary because the smallest gap and the largest relative contrastive gain may correspond to different challengers. Thus, $m_t < \Delta_t$ permits a flip but does not imply that one occurs (see App. C.1.1 for proof). Applying this condition across a decoding run of T steps gives an oracle budget bound.

Corollary 1 (Budget bound). *Fix any $\bar{\Delta} \geq \max_t \Delta_t$ over a decoding run of T steps, and let $\mathcal{S}_{\bar{\Delta}}{=}\{t : m_t < \bar{\Delta}\}$. Then $N_{\text{flip}} \leq |\mathcal{S}_{\bar{\Delta}}|$, and activating MCD on $\mathcal{S}_{\bar{\Delta}}$ attains 100% flip recall at activation fraction $|\mathcal{S}_{\bar{\Delta}}|/T$.*

The budget bound identifies an oracle set $\mathcal{S}_{\bar{\Delta}}$ on which auxiliary computation suffices to retain every flip along the decoding run. Within this set, the vanilla margin provides a useful ordering because a smaller m_t imposes a weaker perturbation requirement for a flip. Ranking steps by increasing m_t therefore prioritizes decisions with the lowest barrier to an MCD-induced change.

3.3 Prediction of Flips

As $\mathcal{S}_{\bar{\Delta}}$ depends on the auxiliary logits, the oracle budget bound is not directly deployable: an activation rule must decide whether to invoke the auxiliary branch before computing these logits, where neither Δ_t nor $\bar{\Delta}$ can be used directly. We therefore seek a signal computable from the vanilla pass alone. The vanilla margin m_t, defined in Eq. 2, satisfies this requirement and, through its role in the immunity condition of Prop. 1, naturally characterizes how susceptible the vanilla decision is to auxiliary perturbation. We thus convert this margin into a bounded statistic that ranks steps by how readily MCD can change the winning token:

Definition 3 (Flip susceptibility). *The flip susceptibility at step t is $r_t{=}\exp(-\beta_r m_t)$ with $\beta_r > 0$.*

We use $\exp(\cdot)$ to convert the non-negative margin into a bounded statistic ($r_t \in (0, 1]$). This transformation preserves the margin-based ordering, while providing a bounded scale for online calibration in Sec. 4.1. We set $\beta_r{=}1$, so r_t equals the probability ratio between the runner-up and the vanilla winner, $p_{t,(2)}/p_{t,(1)}$, where $\boldsymbol{p}_t{=}\text{softmax}(\boldsymbol{z}_t)$. A value near one indicates that the two leading vanilla candidates have similar probabilities, making this ratio a natural statistic for flip susceptibility.

Empirical verification. We verify the relationship between margins and MCD token flips by examining a VCD trace from LLaVA-1.5-7B. Across 819,403 tokens, the conditional flip probability is concentrated in the lowest-margin bins, and no flip occurs above the observed cutoff $m{=}1.61$

(Fig. 2(a)). This low-margin region contains every observed flip but comprises only 8.2% of the tokens. Ranking steps by flip susceptibility retains many recorded flips even under a small activation budget. We test this ranking by replaying MCD trajectories and activating MCD on the steps with the highest flip susceptibility. The resulting budget-recall curves are shown in Fig. 2(b), with AUC 95.0% for `VCD` and 91.0% for `SelfAug`. These results motivate using flip susceptibility to allocate auxiliary computation under a user-specified budget.

4 BUDGETED ACTIVATION

Our analysis in FLIPTRACE provides a freely available measure of flip susceptibility, which identifies steps at which MCD is most likely to alter the generated token. Next, we translate this signal to allocate auxiliary evaluations under a user-specified budget, prioritizing steps with higher flip susceptibility. Specifically, we express this budget as a target activation rate $B \in [0\%, 100\%]$, where endpoints $B{=}0\%$ and 100% denote vanilla decoding and MCD, respectively. At step t, an activation variable $a_t \in \{0, 1\}$ records whether our method evaluates the auxiliary branch. Over a stream of T generated tokens, the realized activation rate should approximate this target:

$$\widehat{B}_T = \frac{1}{T}\sum_{t=1}^{T} a_t \approx B. \tag{4}$$

Under budget, our method allocates auxiliary evaluations to decoding steps at which MCD is most likely to alter the token decision. At each step, it computes the vanilla logits $\boldsymbol{z}_t$ and flip susceptibility r_t, with a binary variable a_t determining whether the auxiliary branch is invoked. Activated steps decode the next token from the MCD logits $\boldsymbol{z}_t^{\mathrm{mcd}}$, whereas inactive steps decode directly from the vanilla logits $\boldsymbol{z}_t$. Our method only determines when to activate the auxiliary branch. It makes no assumptions about the auxiliary construction and therefore preserves the underlying MCD computation exactly on activated steps.

4.1 ONLINE ACTIVATION-RATE CALIBRATION

A fixed flip susceptibility threshold can yield different activation rates across settings. Rather than selecting a threshold for each setting, our method takes a target activation budget as input and calibrates the threshold online through a warmup phase followed by online calibration.

Phase 1: Warmup. The warmup phase provides a data-dependent threshold initialization. For the first N_w tokens, our method activates MCD at every step and collects flip susceptibility values $S{=}\{r_1, \ldots, r_{N_w}\}$. It then initializes the activation threshold τ_t at the empirical $(1{-}B)$-quantile, $\tau_{N_w+1}{=}Q_{1-B}(S)$. This quantile selects a B-fraction of the observed warmup values and aligns the initial threshold with the target rate.

Phase 2: Online continuous calibration. After warmup, our method activates MCD iff $r_t{>}\tau_t$ and the controller updates the threshold after each decision:

$$\tau_{t+1} \;=\; \mathrm{clip}_{[0,1]}\big(\tau_t + \gamma_t\,(a_t - B)\big), \quad \text{with } \gamma_t = \frac{\gamma_0}{1 + t/T_0}. \tag{5}$$

Here, γ_t is a decaying step size with initial value γ_0 and decay constant T_0 (see App. D.2.2). Each activation raises τ_t and thereby discourages subsequent activations. Conversely, each skip lowers τ_t and encourages subsequent activations. For flip susceptibility values drawn from a fixed distribution F_r, the expected update has a stationary fixed point at $\mathbb{E}[a_t]{=}1 - F_r(\tau){=}B$. This fixed point is the $(1{-}B)$-quantile, as proven in App. D.1.1. The update supplies feedback that steers the activation frequency toward the specified budget. This is summarized in Alg. 1. τ_t persists throughout each benchmark, allowing B to constrain the average activation rate over the evaluation stream rather than per sample (App. D.1.1).

Together, flip susceptibility activation and online calibration convert B into a selective sequence of MCD activations. The first component prioritizes activations where MCD is most likely to flip a token, while the second controls how often they occur. As activated steps retain the underlying MCD method's computation exactly, *budgeted activation is agnostic to the auxiliary construction and can be seamlessly applied to any existing MCD variant*.

Algorithm 1: One generation step under budgeted activation, post warmup

Input: token step t, prefix $y_{<t}$, budget B, calibrated threshold τ_t

$\boldsymbol{z}_t \leftarrow f_\theta(v, x, y_{<t})$; // vanilla forward

$r_t \leftarrow \exp(-\beta_r m_t)$;

$a_t \leftarrow \mathbf{1}[r_t > \tau_t]$; // activation rule

if $a_t{=}1$ **then**

  CatchUpAuxiliary$(y_{<t})$; // App. D.1.2

  $\tilde{\boldsymbol{z}}_t \leftarrow$ MCD-Auxiliary$(y_{<t})$; // activate auxiliary branch

  emit from $\boldsymbol{z}_t^{\mathrm{mcd}}{=}(1{+}\alpha)\boldsymbol{z}_t - \alpha\tilde{\boldsymbol{z}}_t$ on $\mathcal{V}_t$;

else

  emit from $\boldsymbol{z}_t$;

$\tau_{t+1} \leftarrow \mathrm{clip}\big(\tau_t + \gamma_t(a_t - B)\big)$; // online calibration, Eq. 5

5 Experiments and Results

5.1 Setup

We evaluate budgeted activation across six underlying MCD methods: `VCD` (Leng et al., 2024), `SelfAug` (Im et al., 2026), `AVCD` (Jung et al., 2025), `HALC` (Chen et al., 2024), `MVCD` (Pang et al., 2025), and `ICD` (Wang et al., 2024), applied to four MLLM backbones (Liu et al., 2024; Bai et al., 2025; Cheng et al., 2024; Yang et al., 2024). Our evaluation spans a diverse range of five multimodal benchmarks, covering (i) visual discriminative (Li et al., 2023b), (ii) visual generative (Rohrbach et al., 2018), (iii) visual perception (Fu et al., 2025), (iv) visual reasoning (Goyal et al., 2017), (v) audio-visual discriminative, and (vi) audio-visual generative (Kim et al., 2025) tasks.

We integrate budgeted activation directly with each underlying MCD method by controlling auxiliary-branch activation, measuring performance at $B{\in}\{10\%, 30\%, 50\%\}$. On activated steps, we retain the method's original contrast design and hyperparameters, and preserve its released evaluation setting for faithful comparisons, detailed in App. D.2. For each underlying MCD method, we reference the vanilla MLLM ($B{=}0\%$) and the method's full-activation configuration ($B{=}100\%$).

We set the warmup length N_w based on the token count of each benchmark rather than through hyperparameter tuning. Specifically, we use $N_w{=}200$ warmup tokens for `POPE`, `CHAIR`, `MME`, and `VQAv2`, and $N_w{=}20$ for the shorter `AVHBench` stream. The warmup constitutes only a small fraction of the generated tokens, e.g., approximately 3% for each `POPE` split and 4% for `AVHBench`. Warmup questions are sampled randomly. Details are provided in App. D.2.4.

5.2 Performance under budgeted activation

(i) Visual discriminative tasks. We first evaluate budgeted MCD on `POPE`, a discriminative object-hallucination benchmark based on binary object-presence questions. We report the F1 over nine splits formed by `COCO`, `GQA`, and `A-OKVQA` under random, popular, and adversarial sampling.

As shown in Fig. 3, budgeted `VCD` and budgeted `SelfAug` match and even exceed their standard counterparts. On `LLaVA-1.5`, budgeted `VCD` at $B{=}50\%$ reaches an F1 score of 82.8% ($+3.4$ pp over vanilla), compared with 81.9% ($+2.4$ pp) by `VCD`. Budgeted `SelfAug` follows the same pattern, with its peak at $B{=}50\%$. On the stronger `Qwen2.5-VL` backbone, MCD improves less over vanilla decoding. Nevertheless, both budgeted methods are within 0.05 pp in F1 of their standard counterparts at $B{=}30\%$. Tables A6, A7, and A8 provide the full results. For `ICD` (Table A12), a budget of 50% results in an F1 of 80.92% ($+0.77$ pp), versus 81.41% ($+1.26$ pp) for `ICD`.

Analysis. The maximum results achieved by budgeted activation also coincide with saturated flip recall. In free generation, budgeted `VCD` already reaches 99.2% flip recall at $B{=}30\%$. At saturation, additional activations have few remaining MCD token flips to recover, but may instead introduce negative flips, which change correct predictions into incorrect ones.

(ii) Visual generative, (iii) perception, and (iv) reasoning tasks. Beyond binary object-presence questions, we evaluate on vision-based generative, perception, and reasoning tasks. **(ii)** Generative:

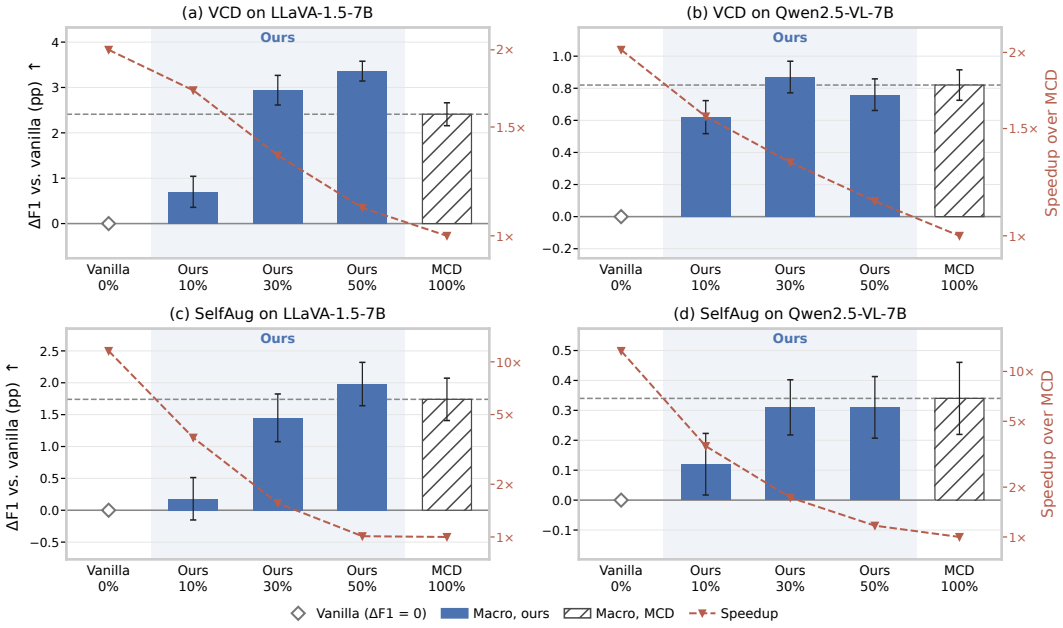


Figure 3: POPE F1 at various budgets on the COCO, GQA, and A-OKVQA splits. Each curve varies the activation budget of the same underlying method. Red curves average over all splits, while gray dashed lines show the speedup over MCD. Our budgeted activation method matches, and sometimes exceeds, the performance of MCD at up to 1.73× speedup.

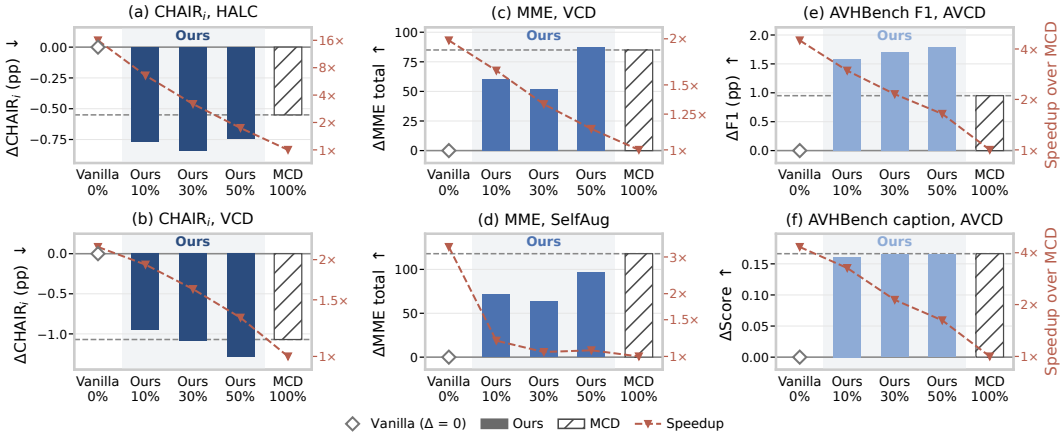


Figure 4: Performance-efficiency trade-offs across visual (a)-(b) generative, (c)-(d) perception, and (e)-(f) audio-visual tasks. Ours matches or exceeds MCD performance at up to 6.57× speedup.

Fig. 4(a)-(b) present the results on captioning performance (lower is better). Budgeted HALC attains its best score of 7.41 (−0.84 pp) at B=30%, compared with 7.70 (−0.55 pp) for HALC. Similarly, budgeted VCD attains 8.59 (−1.29 pp) at B=50%, compared with 8.81 (−1.07 pp) for VCD. **(iii)** Perception: On MME (Fig. 4(c)-(d)), budgeted VCD achieves a score of 1651.3, surpassing VCD while requiring only half as many auxiliary branch activations (B=50%). Complete results are reported in Table A9. **(iv)** Reasoning: We further evaluate MVCD, which leverages in-context learning (ICL) examples for *visual reasoning*. As shown in Fig. 5, budgeted activation substantially reduces inference cost while preserving or improving MVCD performance. At a budget of only 30%, our method matches the accuracy of MVCD across all ICL configurations. Full results are reported in Table A10.

(v) Audio-visual discriminative and (vi) generative tasks. Our method also extends naturally to *audio-visual* modalities. **(v)** On AVHBench's discriminative tasks (Fig. 4(e)), budgeted AVCD with the VideoLLaMA2.1-7B-AV backbone achieves a discriminative F1 score of 77.16% (+1.58 pp) at a budget of only 10%. Performance further improves to a peak F1 of 77.37% (+1.79 pp) at

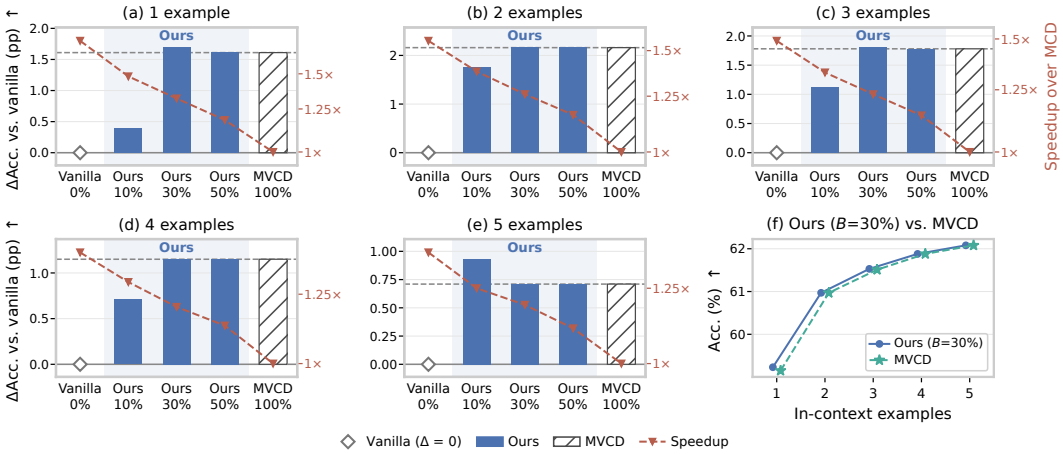


Figure 5: (a)-(e): Budget-performance comparison of MVCD at each ICL configuration. (f): Comparison between budgeted MVCD at B=30% and full-activation MVCD across all five configurations. Our 30% budget matches MVCD while delivering speedups up to 1.32× at 1 ICL example.

Table 1: End-to-end latency of VCD on LLaVA-1.5-7B for 40 short POPE answers and 12 captions of 256 generated tokens. Speedup denotes the ratio of the wall-clock time to VCD .

Budget	Realized rate	POPE		Captioning	
		Wall-clock (s)	speedup	Wall-clock (s)	speedup
0%	0.0%	3.29 ±0.00	2.00×	23.56 ±0.01	2.19×
10%	10.3%	3.83 ±0.00	1.72×	26.73 ±0.01	1.93×
30%	29.9%	4.88 ±0.00	1.35×	31.76 ±0.04	1.62×
50%	50.0%	5.93 ±0.01	1.11×	39.00 ±0.02	1.32×
100%	100.0%	6.57 ±0.01	1.00×	51.54 ±0.01	1.00×

B=50%, surpassing the 76.53% (+0.95 pp) achieved by AVCD. **(vi)** A similar trend can be observed on AVHBench's captioning task (Fig. 4(f)), where performance plateaus at a 10% activation rate, matching that of AVCD. Full results are reported in Table A11.

Summary of results. Together, budgeted activation transfers across (i) visual discriminative, (ii) visual generative, (iii) visual perception, (iv) visual reasoning, (v) audio-visual discriminative, and (vi) audio-visual generative tasks naturally, without the need for any additional tuning. Our budgeted activation method reduces inference latency, while matching or exceeding the performance of full-activation MCD. Although the budget B provides direct and setting-agnostic control over the activation rate, its performance effect remains configuration-dependent, reflecting the interaction between the MCD method, MLLM, and benchmark. Accordingly, *B serves primarily as a mechanism to control the activation rate of the auxiliary branch, and therefore inference latency of MCD.*

5.3 Efficiency

As presented in Fig. 1 and Table 1, budgeted activation reduces mean latency relative to MCD. At B=10%, the reduction corresponds to a 1.72× speedup on short POPE answers and a 1.93× speedup on 256-token captioning. For captioning, VCD requires 2.19× the latency of vanilla decoding, whereas B=10% requires 1.13×. Captioning yields the larger relative speedup as its longer outputs amortize the shared image-prefill cost. We provide a full latency analysis in App. D.4.

5.4 Controlled comparisons of activation policies

Efficiency baselines. Prior methods (Jung et al., 2025; Huang et al., 2026; Liu, 2026) design signals to selectively activate the auxiliary branch. They **(i)** leverage heuristic scores that are method-specific, and **(ii)** manually fix the activation threshold for each experiment setting. We assess these two factors through controlled comparisons. **(i)** To isolate the *first*, we replace our flip susceptibility

Table 2: Comparison of Random and Entropy (AVCD) to our flip susceptibility for budgeted VCD at matched target budgets.

Budget	Random	Entropy	Ours
Vanilla		79.45 ±0.21	
10%	79.73 ±0.23	79.52 ±0.41	**80.15** ±0.27
30%	80.19 ±0.24	81.05 ±0.15	**82.39** ±0.25
50%	80.75 ±0.11	81.87 ±0.05	**82.81** ±0.06
VCD		81.86 ±0.14	

Table 3: Comparison of Confidence (CHASD) to our flip susceptibility on budgeted CHASD at matched target budgets.

Budget	Conf.	Ours
Vanilla	1755.17 ±0.00	
10%	1744.82 ±1.98	**1779.85** ±4.06
30%	1760.02 ±4.86	**1761.02** ±4.62
50%	1759.73 ±4.62	1759.73 ±4.62
CHASD	1763.30 ±1.13	

with existing heuristics within our budgeted activation framework. For VCD on POPE (Table 2), we compare with random gating and the entropy heuristic of AVCD (Jung et al., 2025). For CHASD on MME (Table 3), we compare with its own confidence signal. Flip susceptibility outperforms the best baseline at every budget on POPE, and the confidence signal at $B{=}10\%$ and 30% on MME. It even exceeds the performance of VCD and CHASD. **(ii)** On the *second*, our method prevents cross-setting rate drift, a drawback of fixed thresholds. In Table A17, transferring a fixed threshold across nine POPE splits yields activation rates from 24.45% to 34.69%, with a mean absolute deviation of 3.14 pp from the 30% target. Our calibrated budget reduces this to 0.30 pp, ensuring a consistent activation rate across *any* MCD-MLLM-task combination. We compare against the fixed threshold of CHASD in App. D.3.2.

Flip agreement and correctness. In Table 4, we replace flip susceptibility with alternative heuristics within budgeted activation. We compare the resulting token flips with those of VCD by their Jaccard similarity and $|\Delta\,\mathrm{P/N}|$, which we define as the absolute difference in the ratio of flips that correct a vanilla token to those that corrupt one (Eq. 13). At $B{=}30\%$, flip susceptibility closely reproduces the token flips of VCD (Jaccard 0.916) and its P/N ($|\Delta\,\mathrm{P/N}|{=}0.007$, compared with 0.028 for random activation). Although confidence matches flip susceptibility here, it underperforms on MME (Table 3). As flip susceptibility estimates whether MCD will flip a token rather than whether it is positive, budgeted activation also retains harmful flips produced by VCD. This reflects a limitation of the underlying MCD method rather than of budgeted activation (App. D.5.2).

Table 4: Comparison of token flips to VCD, and distance of the positive-to-negative (P/N) ratio from VCD.

Config	Jaccard ↑	$\lvert\Delta\,\mathrm{P/N}\rvert$ ↓
	$B{=}30\%$	
Random	0.304	0.028
Entropy	0.857	0.013
Conf.	0.916	0.007
Ours	0.916	0.007

6 Conclusion

This work aims to address the high inference latency of MCD methods. Our preliminary studies find that MCD changes the selected token at only a small fraction of decoding steps. We therefore introduce FlipTrace, which formalizes these sparse changes as token flips, derives a sufficient condition for flip immunity, and uses the vanilla top-2 margin to define an auxiliary-free measure of flip susceptibility. We then develop a budgeted activation method that controls auxiliary-branch activation to track a user-specified budget. Our method integrates seamlessly with existing MCD methods, irrespective of their auxiliary construction or method-specific hyperparameters. It thus can be applied across combinations of six MCD methods, four MLLM backbones, and five benchmarks. Importantly, it delivers a maximum of 87% reduction in inference latency while preserving the performance of the original MCD.

Limitations. We acknowledge two limitations of our work. (i) While FlipTrace characterizes where MCD is likely to induce a token flip, it does not determine whether that flip improves the generated output. (ii) The flip-immunity condition is per-token and does not provide a sequence-level guarantee under free-running generation. Differences introduced by individual activation decisions may therefore propagate through subsequent decoding steps.

AI USE STATEMENT

In this work, we used AI tools for searching and summarizing relevant literature, programming experimental code, editing the manuscript and assisting in the development and verification of mathematical proofs. We did not use generative AI tools to generate core ideas, experimental results or independently substantiate the scientific claims of this work. Literature identified or summarized with the assistance of generative AI was further checked using an online citation verification tool. AI-assisted code, proofs, and manuscript content were manually reviewed and verified by the authors. The authors take full responsibility for the final content of this work, including all text, claims, results, and artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work studies the inference-time efficiency of public models on standard public benchmarks. We collect no new data and involve no human subjects. Reducing hallucination lowers the risk of models asserting unsupported content. Reducing inference cost may also lower the energy footprint of deploying such mitigations. We do not foresee ethical concerns specific to this method beyond those that already exist within the underlying models.

REPRODUCIBILITY STATEMENT

All experiments use publicly released models and benchmarks. Sec. 5.1 and App. D.2 list the hyperparameters used. Budgeted activation uses four interpretable hyperparameters: warmup length N_w, step size γ_0, decay T_0, and temperature parameter β_r for flip susceptibility. We use a common configuration for the experiments, and the budget B is the user-facing control rather than a tuned value.

REFERENCES

Wenbin An, Feng Tian, Sicong Leng, Jiahao Nie, Haonan Lin, Qianying Wang, Ping Chen, Xiaoqin Zhang, and Shijian Lu. Mitigating object hallucinations in large vision-language models with assembly of global and local attention. In *CVPR*, 2025.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-VL technical report. *arXiv preprint arXiv:2502.13923*, 2025.

Zechen Bai, Pichao Wang, Tianjun Xiao, Tong He, Zongbo Han, Zheng Zhang, and Mike Zheng Shou. Hallucination of multimodal large language models: A survey. *arXiv preprint arXiv:2404.18930*, 2024.

Wei Chen, Xin Yan, Bin Wen, Fan Yang, Tingting Gao, Di Zhang, and Long Chen. Decoupling contrastive decoding: Robust hallucination mitigation in multimodal large language models. In *NeurIPS*, 2025.

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

Zesen Cheng, Sicong Leng, Hang Zhang, Yifei Xin, Xin Li, Guanzheng Chen, Yongxin Zhu, Wenqi Zhang, Ziyang Luo, Deli Zhao, and Lidong Bing. VideoLLaMA 2: Advancing spatial-temporal modeling and audio understanding in video-LLMs. *arXiv preprint arXiv:2406.07476*, 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 *ICLR*, 2024.

Jingyuan Deng and Yujiu Yang. MaskCD: Mitigating LVLM hallucinations by image head masked contrastive decoding. In *EMNLP Findings*, 2025.

Alessandro Favero, Luca Zancato, Matthew Trager, Siddharth Choudhary, Pramuditha Perera, Alessandro Achille, Ashwin Swaminathan, and Stefano Soatto. Multi-modal hallucination control by visual information grounding. In *CVPR*, 2024.

Laura Fieback, Nishilkumar Balar, Jakob Spiegelberg, and Hanno Gottschalk. ECD: Efficient contrastive decoding with probabilistic hallucination detection. In *ECML PKDD*, 2025.

Chaoyou Fu, Peixian Chen, Yunhang Shen, Yulei Qin, Mengdan Zhang, Xu Lin, Jinrui Yang, Xiawu Zheng, Ke Li, Xing Sun, Yunsheng Wu, Rongrong Ji, Caifeng Shan, and Ran He. MME: A comprehensive evaluation benchmark for multimodal large language models. In *NeurIPS*, 2025.

Xuan Gong, Tianshi Ming, Xinpeng Wang, and Zhihua Wei. DAMRO: Dive into the attention mechanism of LVLM to reduce object hallucination. In *EMNLP*, 2024.

Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the V in VQA matter: Elevating the role of image understanding in visual question answering. In *CVPR*, 2017.

Tzu-wen Hsu, Ke-Han Lu, Cheng-Han Chiang, and Hung-yi Lee. Reducing object hallucination in large audio-language models via audio-aware decoding. In *IEEE ASRU*, 2025.

Xiaoyi Huang, Kejia Zhang, and Zhiming Luo. CHASD: Language increment-calibrated contrastive decoding against hallucination in LVLMs. *arXiv preprint arXiv:2605.23344*, 2026.

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

Eun Woo Im, Muhammad Kashif Ali, and Vivek Gupta. Self-Aug: Query and entropy adaptive decoding for large vision-language models. In *ICLR*, 2026.

Chaoya Jiang, Haiyang Xu, Mengfan Dong, Jiaxing Chen, Wei Ye, Ming Yan, Qinghao Ye, Ji Zhang, Fei Huang, and Shikun Zhang. Hallucination augmented contrastive learning for multimodal large language model. In *CVPR*, 2024.

Chaeyoung Jung, Youngjoon Jang, and Joon Son Chung. AVCD: Mitigating hallucinations in audio-visual large language models through contrastive decoding. In *NeurIPS*, 2025.

Zhehan Kan, Ce Zhang, Zihan Liao, Yapeng Tian, Wenming Yang, Junyuan Xiao, Xu Li, Dongmei Jiang, Yaowei Wang, and Qingmin Liao. CATCH: Complementary adaptive token-level contrastive decoding to mitigate hallucinations in LVLMs. *arXiv preprint arXiv:2411.12713*, 2024.

Junho Kim, Hyun Jun Kim, Yeon Ju Kim, and Yong Man Ro. CODE: Contrasting self-generated description to combat hallucination in large multi-modal models. In *NeurIPS*, 2024.

Sung-Bin Kim, Hyun-Bin Oh, Jung-Mok Lee, Arda Senocak, Joon Son Chung, and Tae-Hyun Oh. AVHBench: A cross-modal hallucination benchmark for audio-visual large language models. In *ICLR*, 2025.

Jihoon Lee and Min Song. Retrieval visual contrastive decoding to mitigate object hallucinations in large vision-language models. In *ACL Findings*, 2025.

Yi-Lun Lee, Yi-Hsuan Tsai, and Wei-Chen Chiu. Delve into visual contrastive decoding for hallucination mitigation of large vision-language models. In *ICPR*, 2026.

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 *CVPR*, 2024.

Xiang Lisa Li, Ari Holtzman, Daniel Fried, Percy Liang, Jason Eisner, Tatsunori Hashimoto, Luke Zettlemoyer, and Mike Lewis. Contrastive decoding: Open-ended text generation as optimization. In *ACL*, 2023a.

Yanda Li, Yuhan Liu, Zirui Song, Yunchao Wei, Martin Takáč, and Salem Lahlou. Temporal contrastive decoding: A training-free method for large audio-language models. In *ACL Findings*, 2026.

Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Wayne Xin Zhao, and Ji-Rong Wen. Evaluating object hallucination in large vision-language models. In *EMNLP*, 2023b.

Tzu-Quan Lin, Wei-Ping Huang, Yi-Cheng Lin, and Hung-yi Lee. How contrastive decoding enhances large audio language models. *arXiv preprint arXiv:2603.09232*, 2026.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. In *CVPR*, 2024.

Xinyun Liu. HTDC: Hesitation-triggered differential calibration for mitigating hallucination in large vision-language models. *arXiv preprint arXiv:2604.12115*, 2026.

Xinyu Lyu, Beitao Chen, Lianli Gao, Jingkuan Song, and Heng Tao Shen. Alleviating hallucinations in large vision-language models through hallucination-induced optimization. In *NeurIPS*, 2024.

Avshalom Manevich and Reut Tsarfaty. Mitigating hallucinations in large vision-language models (LVLMs) via language-contrastive decoding (LCD). In *ACL Findings*, 2024.

Yuqi Pang, Bowen Yang, Haoqin Tu, Yun Cao, and Zeyu Zhang. Language models can see better: Visual contrastive decoding for LLM multimodal reasoning. In *ICASSP*, 2025.

Woohyeon Park, Woojin Kim, Jaeik Kim, and Jaeyoung Do. SECOND: Mitigating perceptual hallucination in vision-language models via selective and contrastive decoding. In *ICML*, 2025a.

Yeji Park, Deokyeong Lee, Junsuk Choe, and Buru Chang. ConVis: Contrastive decoding with hallucination visualization for mitigating hallucinations in multimodal large language models. In *AAAI*, 2025b.

Anna Rohrbach, Lisa Anne Hendricks, Kaylee Burns, Trevor Darrell, and Kate Saenko. Object hallucination in image captioning. In *EMNLP*, 2018.

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

Wei Suo, Lijun Zhang, Mengyang Sun, Lin Yuanbo Wu, Peng Wang, and Yanning Zhang. Octopus: Alleviating hallucination via dynamic contrastive decoding. In *CVPR*, 2025.

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

Sangmin Woo, Jaehyuk Jang, Donguk Kim, Yubin Choi, and Changick Kim. RITUAL: Random image transformations as a universal anti-hallucination lever in large vision language models. *arXiv preprint arXiv:2405.17821*, 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 *ACL Findings*, 2025.

Ziwei Xu, Sanjay Jain, and Mohan Kankanhalli. Hallucination is inevitable: An innate limitation of large language models. *arXiv preprint arXiv:2401.11817*, 2024.

An Yang et al. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*, 2024.

Hao Yin, Guangzong Si, and Zilei Wang. The mirage of performance gains: Why contrastive decoding fails to mitigate object hallucinations in MLLMs? In *NeurIPS*, 2025.

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 *CVPR*, 2024.

Tianyu Yu, Haoye Zhang, Qiming Li, Qixin Xu, Yuan Yao, Da Chen, Xiaoman Lu, Ganqu Cui, Yunkai Dang, Taiwen He, Xiaocheng Feng, Jun Song, Bo Zheng, Zhiyuan Liu, Tat-Seng Chua, and Maosong Sun. RLAIF-V: Open-source AI feedback leads to super GPT-4V trustworthiness. In *CVPR*, 2025.

Zihao Yue, Liang Zhang, and Qin Jin. Less is more: Mitigating multimodal hallucination from an EOS decision perspective. In *ACL*, 2024.

Ce Zhang, Zifu Wan, Zhehan Kan, Martin Q. Ma, Simon Stepputtis, Deva Ramanan, Russ Salakhutdinov, Louis-Philippe Morency, Katia Sycara, and Yaqi Xie. Self-correcting decoding with generative feedback for mitigating hallucinations in large vision-language models. In *ICLR*, 2025a.

Yifan Zhang, Yang Shi, Weichen Yu, Qingsong Wen, Xue Wang, Wenjing Yang, Zhang Zhang, Liang Wang, and Rong Jin. Debiasing multimodal large language models via penalization of language priors. In *ACM MM*, 2025b.

Guanyu Zhou, Yibo Yan, Xin Zou, Kun Wang, Aiwei Liu, and Xuming Hu. Mitigating modality prior-induced hallucinations in multimodal large language models via deciphering attention causality. In *ICLR*, 2025.

SUPPLEMENTARY MATERIALS

CONTENTS

A Notation . . . 14
B Scope and Relationship to Prior Work . . . 14
C Analysis of MCD Token Flips . . . 14
C.1 Theoretical analysis . . . 14
C.2 Empirical analysis . . . 17
D Experiments . . . 17
D.1 Budgeted activation . . . 17
D.2 Experimental setup . . . 19
D.3 Performance under budgeted activation . . . 20
D.4 Latency analysis . . . 23
D.5 Activation-policy comparisons and diagnostics . . . 26

A NOTATION

Table A1 lists the symbols used and where each is introduced.

B SCOPE AND RELATIONSHIP TO PRIOR WORK

Table A2 states each MCD method's original task and the construction of its auxiliary branch. Table A3 further compares methods that decide per token whether to run the auxiliary branch. Their triggers are fixed thresholds on an uncertainty-based heuristic, or a trained router, so the realized activation rate follows from the configuration and the data rather than from a user setting. Each heuristic is also designed specifically for its auxiliary construction. To the best of our knowledge, no prior literature has provided a theoretical analysis of the token-level effects of MCD.

C ANALYSIS OF MCD TOKEN FLIPS

C.1 THEORETICAL ANALYSIS

C.1.1 PROOF OF PROP. 1 AND COR. 1

The condition in Sec. 3 accommodates MCD implementations that restrict selection to the adaptive plausibility set

$$\mathcal{V}_t = \{\, j \in \mathcal{V} : z_{t,j} \geq \log \beta + \max_i z_{t,i} \,\}, \tag{6}$$

where $\beta \in (0, 1]$ is the plausibility cutoff (Li et al., 2023a; Leng et al., 2024). Logits outside $\mathcal{V}_t$ are set to $-\infty$ before selection, and $i_t^\star \in \mathcal{V}_t$ always. For $\mathcal{V}_t{=}\{i_t^\star\}$, we set $\Delta_t{=}-\infty$, making immunity trivial.

Proof. Suppose $m_t \geq \Delta_t$ and let $j \in \mathcal{V}_t \backslash \{i_t^\star\}$. By definition, $z_{t,i_t^\star} - z_{t,j} \geq m_t$ and $\delta_{t,j} - \delta_{t,i_t^\star} \leq \Delta_t$. Thus, $z^{\text{mcd}}_{t,i_t^\star} - z^{\text{mcd}}_{t,j} {=} (z_{t,i_t^\star} - z_{t,j}) - (\delta_{t,j} - \delta_{t,i_t^\star}) \geq m_t - \Delta_t \geq 0$. Therefore, $i_t^\star$ remains a maximizer over $\mathcal{V}_t$, and $\hat{\imath}_t{=}i_t^\star$ when ties favor the incumbent argmax, as in the implementation. The contrapositive establishes the second claim. □

For Cor. 1, every flip satisfies $m_t < \Delta_t \leq \bar{\Delta}$. Activating on $\{t : m_t < \bar{\Delta}\}$ therefore covers every flip using $|\{t : m_t < \bar{\Delta}\}|$ auxiliary evaluations among T tokens. Since $r_t{=}e^{-\beta_r m_t}$ is strictly decreasing, the equivalent rule is $r_t > e^{-\beta_r \bar{\Delta}}$.

Table A1: Symbols used in the paper, grouped by the part that introduces them. Each row points to the definition, corollary, or equation that fixes the symbol, and repeats no definition.

Symbol	Meaning
	Model and decoding
θ	Frozen parameters of the multimodal LLM
v, x	Multimodal evidence and text prompt
$\tilde{v}$	Perturbed evidence of the auxiliary branch, such as a noised image
$y_{1:T}$, $y_{<t}$	Generated token sequence and the prefix before step t
T	Number of decoding steps in a run
$\mathcal{V}$, $\mathcal{V}_t$	Vocabulary and the adaptive plausibility set at step t
$\boldsymbol{z}_t$, $\tilde{\boldsymbol{z}}_t$	Vanilla and auxiliary-branch logits, $f_\theta(v, x, y_{<t})$ and $f_\theta(\tilde{v}, x, y_{<t})$
$\boldsymbol{z}_t^{\mathrm{mcd}}$	Contrastive logits of Eq. 1
$z_{t,j}$	Vanilla logit of token j
$z_{t,(1)}$, $z_{t,(2)}$	Largest and second-largest entries of $\boldsymbol{z}_t$
α	Contrast strength of Eq. 1
β	Plausibility cutoff that defines $\mathcal{V}_t$
	FLIPTRACE *framework*
$i_t^\star$, $\hat{\imath}_t$	Vanilla and MCD greedy tokens, $\arg\max_i z_{t,i}$ and $\arg\max_i z_{t,i}^{\mathrm{mcd}}$
ϕ_t	Flip indicator, $\mathbf{1}[\hat{\imath}_t \neq i_t^\star]$ (Def. 1)
f, N_{flip}	Flip rate $\frac{1}{T}\sum_t \phi_t$ and flip count of a run
m_t	Top-2 logit margin, $z_{t,(1)} - z_{t,(2)}$ (Eq. 2)
$\delta_{t,j}$	Logit perturbation of the contrast, $\alpha(z_{t,j} - \tilde{z}_{t,j})$
Δ_t	Maximum relative contrastive gain over all challengers (Def. 2, Eq. 3)
$\bar{\Delta}$	Run-level upper bound with $\bar{\Delta} \geq \max_t \Delta_t$ (Cor. 1)
$\mathcal{S}_{\bar{\Delta}}$	Low-margin step set $\{t : m_t < \bar{\Delta}\}$ (Cor. 1)
	Budgeted activation
r_t	Flip susceptibility, $\exp(-\beta_r m_t) \in (0, 1]$ (Def. 3)
β_r	Flip susceptibility temperature
B	Activation budget, the target fraction of steps that evaluate the contrast
τ_t	Activation threshold at step t
a_t	Activation decision, $\mathbf{1}[r_t > \tau_t]$
N_w	Warmup length
S	Warmup flip susceptibility set $\{r_1, \ldots, r_{N_w}\}$
Q_{1-B}	Empirical $(1-B)$-quantile
γ_t, γ_0, T_0	Step size $\gamma_0/(1 + t/T_0)$, its initial value, and its decay constant
	Measurement and experiment
$\mathrm{R}(B)$, $\mathrm{P}(B)$	Flip recall of Eq. 8 and flip precision at budget B
$\Omega_{\mathrm{mcd}}(B)$	Teacher-forced agreement with MCD (Eq. 7)
$\Omega_{\mathrm{van}}(B)$	Teacher-forced agreement with vanilla decoding (Eq. 7)
TV, K	Total variation between two step distributions, on a top-K trace
u_t	Top-1 uncertainty, the diagnostic baseline of Eq. 11

C.1.2 EMPIRICAL TIGHTNESS OF THE FLIP-IMMUNITY BOUND

On the top-K=100 trace (`POPE COCO`-adversarial, 1,500 questions, α=1), the observed flip set exactly matches $\{m_t < \Delta_t\}$. For this analysis, Δ_t is maximized over the recorded challengers in $\mathcal{V}_t \backslash \{i_t^\star\}$. Applying the plausibility restriction yields the same set. The maximum relative contrastive gain has median -0.27 and upper-tail quantiles of 1.67, 2.56, and 3.14 at the 90th, 99th, and 99.9th percentiles, respectively, with a maximum of 3.38. This upper tail is consistent with the light-tail assumption used in the distributional application of Cor. 1.

C.1.3 FROM MARGIN TO FLIP SUSCEPTIBILITY

For $\theta \in (0, 1]$, strict monotonicity gives $\{t : r_t > \theta\} = \{t : m_t < -\log\theta/\beta_r\}$. Thresholding flip susceptibility therefore generates the same activation sets as thresholding the margin, with β_r

Table A2: Methods evaluated with budgeted activation. *Field* gives the problem that each cited work was designed for. *Contrastive branch* gives the conditioning used by the second pass. The six evaluated methods span pixel-space, query-conditioned, field-of-view, in-model cross-modal, prompt-level, and in-context-example contrasts. `MVCD`, the last of these, is the cross-domain stress test. Sec. 5.1 gives the backbone, benchmark, and decoding regime used for each method. `ECD` (Fieback et al., 2025) is omitted because it is a single-pass comparator with no second pass for budgeted activation to control.

MCD method	Field	Contrastive branch
`VCD` (Leng et al., 2024)	Object hallucination in images	Diffusion-noised copy of the image
`SelfAug` (Im et al., 2026)	Object hallucination in images	Query-adaptive image augmentation
`HALC` (Chen et al., 2024)	Object grounding in open-ended captioning	Field-of-view focal contrast at object words
`AVCD` (Jung et al., 2025)	Cross-modal hallucination in audio-visual inputs	Attentive masking of a whole modality
`ICD` (Wang et al., 2024)	Object hallucination in images	Instruction perturbation, the one purely textual contrast
`MVCD` (Pang et al., 2025)	Multimodal reasoning	Prompt with in-context learning examples versus without

Table A3: Methods that decide per token whether to run a contrastive branch. *Trigger* gives the activation measure and decision rule. *Token-flip condition* indicates whether the work states a condition under which running the branch could not change the emitted token. *Rate budget* indicates whether the user sets an activation rate that is tracked online rather than obtained from a fixed threshold. *Training-free* means that the method uses neither a trained component nor a per-model calibration set.

Method	Trigger	Token-flip condition	Rate budget	Training-free
`AVCD` (Jung et al., 2025)	Fixed predictive entropy	✗	✗	✓
`HTDC` (Liu, 2026)	Layer-wise hesitation	✗	✗	✓
`CHASD` (Huang et al., 2026)	Fixed top-1 confidence	✗	✗	✓
`Octopus` (Suo et al., 2025)	Learned token router	✗	✗	✗
Ours	Flip susceptibility + online calibration	✓	✓	✓

absorbed into the threshold. The exponential maps $m \in [0, \infty)$ to $r \in (0, 1]$, allowing our method in Sec. 4.1 to clip τ to the range of $[0, 1]$.

C.1.4 Tail-based motivation for flip susceptibility

The heuristic argument in Sec. 3.3 follows from the light-tail assumption in Cor. 1. Let $g_{t,j}{=}z_{t,i_t^\star} - z_{t,j}$ be the clean gap to challenger j. A flip requires $\delta_{t,j} - \delta_{t,i_t^\star} > g_{t,j}$ for some plausible challenger. A union bound gives $\Pr[\text{flip}] \lesssim \sum_j \bar{G}(g_{t,j})$, where $\bar{G}$ denotes the upper tail of the relative contrastive gain. With $\bar{G}(g) \propto e^{-\lambda g}$, this expression factorizes into $e^{-\lambda m_t}$ and a correction for additional challengers. The leading term equals r_t at $\beta_r{=}\lambda$ and dominates when the remaining gaps are sufficiently larger than m_t.

C.2 Empirical analysis

C.2.1 Token agreement under teacher-forced replay

For the replay of Sec. 3.3, let $\phi_t{=}\mathbf{1}[\hat{\imath}_t \neq i_t^\star]$ denote a flip and $a_t(B){=}\mathbf{1}\big[r_t > Q_{1-B}(\{r_s\}_{s=1}^T)\big]$ select the highest-susceptibility fraction B. The empirical quantile conventions $Q_1{=}\max$ and $Q_0{=}-\infty$ recover vanilla and MCD decoding at the two budget endpoints, although ties can slightly alter the activated fraction.

The replay emits $\hat{\imath}_t$ when activated and $i_t^\star$ otherwise, differing from MCD only at missed flips and from vanilla only at activated flips. Correspondingly, we define agreement rates as:

$$\Omega_{\mathrm{mcd}}(B) = 1 - \frac{1}{T}\sum_{t=1}^{T} \phi_t\big(1 - a_t(B)\big), \qquad \Omega_{\mathrm{van}}(B) = 1 - \frac{1}{T}\sum_{t=1}^{T} \phi_t a_t(B). \tag{7}$$

Furthermore, flip recall measures the fraction of MCD flips recovered by the replay:

$$\mathrm{R}(B) = \frac{\sum_{t=1}^{T} \phi_t a_t(B)}{\sum_{t=1}^{T} \phi_t}. \tag{8}$$

Complementarily, flip precision measures the fraction of activated auxiliary calls that actually induce a flip:

$$\mathrm{P}(B) = \frac{\sum_t \phi_t a_t(B)}{\sum_t a_t(B)}. \tag{9}$$

As the budget increases, recall rises as more flips are recovered, whereas precision can decline once additional activations increasingly fall on non-flipping decisions.

With flip rate $f{=}\frac{1}{T}\sum_t \phi_t$, we can substitute Eq. 8 into Eq. 7 to get:

$$\Omega_{\mathrm{mcd}}(B) = 1 - f\big(1 - \mathrm{R}(B)\big), \qquad \Omega_{\mathrm{van}}(B) = 1 - f\big(\mathrm{R}(B)\big). \tag{10}$$

Both rates lie in $[1{-}f, 1]$ and sum to $2{-}f$. They therefore restate flip recall on a per-position basis rather than provide independent evidence: when flips are sparse, agreement remains high even if few flips are recovered, while activating a flip necessarily lowers vanilla agreement.

For this teacher-forced replay, we additionally compare our flip susceptibility against a top-1 uncertainty baseline:

$$u_t = 1 - \max_i \mathrm{softmax}(\boldsymbol{z}_t)_i. \tag{11}$$

This baseline ranks decoding steps according to uncertainty in the vanilla top-1 prediction rather than susceptibility to an MCD-induced decision change.

Table A4 reports replay agreement and precision using the trajectory traces of Sec. 3. Across budgets, r_t concentrates auxiliary evaluations more effectively on flipping decisions, while precision decreases only after flip recall approaches saturation. In contrast, the top-1 uncertainty baseline yields lower flip recall at matched budgets, particularly for `SelfAug` in the low-budget regime. These results indicate that a generic heuristic is less informative than our theory-backed r_t in identifying potential MCD flips. This complements our controlled comparison with `AVCD`'s entropy-based heuristic in Table 2.

D Experiments

D.1 Budgeted activation

D.1.1 Convergence of online activation-rate calibration

For independent flip susceptibility values drawn from a fixed distribution F_r, Eq. 5 is a stochastic-approximation recursion with fixed point $\mathbb{E}[a_t]{=}1 - F_r(\tau){=}B$. The schedule $\gamma_t{=}\gamma_0/(1 + t/T_0)$ satisfies $\sum_t \gamma_t{=}\infty$ and $\sum_t \gamma_t^2 < \infty$. If F_r is continuous and strictly increasing at the target quantile, τ_t converges to the $(1{-}B)$-quantile. Deployment values are neither independent nor stationary, so this analysis motivates the tracker; rate calibration is assessed empirically in Sec. 5.1. The token counter includes warmup, so tracking starts at $t{=}N_w$ with step size $\gamma_0/(1 + N_w/T_0)$.

Table A4: Teacher-forced replay on `LLaVA-1.5-7B`, comparing our theory-backed flip susceptibility r_t and the top-1 uncertainty heuristic u_t.

Measure	Budget	Ω_{mcd}	Ω_{van}	Flip recall	Flip precision
VCD (819,403 tokens, flip rate 2.03%)					
-	0%	97.97	100.00	0.00	-
r_t	10%	100.00	97.97	100.00	20.27
	30%	100.00	97.97	100.00	6.78
	50%	100.00	97.97	100.00	4.06
u_t	10%	99.99	97.98	99.61	20.19
	30%	100.00	97.97	100.00	6.76
	50%	100.00	97.97	100.00	4.16
-	100%	100.00	97.97	100.00	2.03
SelfAug (228,148 tokens, flip rate 1.68%)					
-	0%	98.32	100.00	0.00	-
r_t	10%	99.60	98.73	75.91	13.21
	30%	99.97	98.35	97.99	5.51
	50%	100.00	98.32	100.00	3.37
u_t	10%	99.36	98.96	61.87	10.42
	30%	99.96	98.36	97.89	5.48
	50%	100.00	98.32	100.00	3.36
-	100%	100.00	98.32	100.00	1.68

Scope of the budget. The threshold τ_t is shared across all examples within a benchmark and is reset only between benchmarks. Each `POPE` split of 3,000 questions is therefore treated as a separate benchmark. Accordingly, the budget B constrains the token-averaged activation rate over the benchmark, while the activation rate of an individual example may be higher or lower than B. When the distribution of r_t shifts within a benchmark, Eq. 5 adapts τ_t toward the new $(1-B)$-quantile, after which the realized activation rate converges back toward B. This adaptation becomes slower for shifts occurring later in the benchmark as γ_t decreases with t.

D.1.2 Exact auxiliary-branch evaluation via incremental prefill

To implement the budgeted activation in Sec. 4 efficiently, we maintain a separate KV cache for the auxiliary branch over the contrastive inputs, aligned with existing two-pass MCD methods. After k skipped steps, we use *incremental prefill* to process the skipped tokens in a single forward pass, extending the auxiliary KV cache before computing the current auxiliary logits. Under causal masking, this recovers the cache entries and exact activated-step logits of an uninterrupted auxiliary branch on the same prefix. Table A5 compares incremental prefill with sequential replay at B=30% under identical activation decisions. As both methods result in the same arithmetic output, the ablation study isolates the latency benefit of incremental prefill.

Scope of the efficiency claim. The total number of tokens processed by the auxiliary branch depends on its final activation in each sample. If the auxiliary branch is activated at the final decoding step, incremental prefill processes all preceding skipped tokens. Otherwise, tokens after the final activation remain unprocessed by the auxiliary branch. Our method therefore does not necessarily reduce total FLOPs of MCD methods. However, FLOPs do not translate directly into latency, which also depends on other factors, such as memory access and hardware utilization. Rather, our efficiency claim concerns inference latency. Even on the same k tokens, incremental prefill replaces k decoding passes with a *single* forward pass, thus yielding substantial speedups through more efficient execution.

Table A5: Incremental prefill on `LLaVA-1.5-7B` captioning at B=30%.

Catch-up method	Realized rate (%)	Wall-clock (s)	Throughput (tokens/s)	Relative speed
Batched (ours)	29.9	31.56 $\pm$0.03	46.3	**1.42**×
Per-token replay	29.9	44.81 $\pm$0.04	32.5	1.00×

D.2 Experimental setup

D.2.1 Multimodal contrastive decoding methods

We follow the published hyperparameters for MCD methods in all experiments:

- `VCD`: α=1.0, β=0.1, and diffusion noise step 500.
- `SelfAug`: α=1.0, `cd_tau`=0.5.
- `AVCD`: attentive modality masking, α=2.5 for binary `AVHBench` questions and α=0.5 for captioning, β=0.2.
- `HALC`: contrast weight 0.05, four focal windows, expansion ratio 0.6, two candidates, beam size one, `GroundingDINO` box threshold 0.45, and `BLIP` scoring.
- `MVCD`: contrast between prompts with and without in-context examples, one to five examples selected from a pool of 20, relative probability cutoff 0.1, masked score -1000.

D.2.2 Budgeted activation hyperparameters

The default activation measure is flip susceptibility $r = \exp(-\beta_r m)$ with β_r=1. Following Sec. 5.1, calibration uses:

- Warmup: MCD at every token for N_w=200 tokens on `POPE`, `MME`, `CHAIR`, and `VQAv2`; N_w=20 on `AVHBench`.
- Initial threshold: empirical $(1-B)$-quantile of warmup measures.
- Stochastic approximation: γ_0=0.01, decay T_0=1000, and threshold clipping to $[0, 1]$.

D.2.3 Benchmarks

Benchmarks retain each method's native decoding regime.

- `POPE`: 3,000 questions per split across {`COCO` / `GQA` / `A-OKVQA`} × {random / popular / adversarial}.
- `MME`: full perception and cognition suite (maximum score 2,800).
- `CHAIR`: 500 `COCO` images for captioning.
- `AVHBench`: 205 binary audio-visual hallucination questions and 1,106 captioning questions.

D.2.4 Repeats and variations in results

Following their published methodologies, we repeat each performance evaluation over three runs with seeds 55, 56, and 57. We report the mean and standard deviation. The reported variation in performance follows each method's native stochastic protocol (if any), and does not come from budgeted activation. This is with the exception of `HALC`, which uses deterministic decoding. Latency experiments use seed 55.

Stream order and warmup length. We fix N_w for each benchmark without hyperparameter tuning. Warmup questions are randomly sampled according to the evaluation seed. On `MME`, we further compare N_w=200 and N_w=1,267, corresponding to 1.5% and 10% of the stream, respectively. Across all budgets, the resulting difference in total score is at most 25 points, which remains within the standard deviation across seeds.

Table A6: Budgeted VCD and SelfAug on POPE with LLaVA-1.5-7B and Qwen2.5-VL-7B.

Budget	POPE F1 (%)			Macro (%)
	COCO	GQA	A-OKVQA	
VCD on LLaVA-1.5-7B				
0%	79.60 ±0.40	79.10 ±0.29	79.64 ±0.14	79.45 ±0.21
10%	80.49 ±0.59	79.92 ±0.49	80.05 ±0.23	80.15 ±0.27
30%	**82.75** ±0.35	**82.12** ±0.19	**82.29** ±0.21	**82.39** ±0.25
50%	**83.15** ±0.07	**82.52** ±0.23	**82.76** ±0.05	**82.81** ±0.06
100%	82.40 ±0.15	81.29 ±0.39	81.88 ±0.16	81.86 ±0.14
VCD on Qwen2.5-VL-7B				
0%	85.53 ±0.10	86.14 ±0.28	86.37 ±0.09	86.01 ±0.09
10%	86.24 ±0.05	86.59 ±0.04	87.06 ±0.12	86.63 ±0.05
30%	**86.57** ±0.11	86.85 ±0.08	**87.21** ±0.05	**86.88** ±0.04
50%	**86.44** ±0.10	86.79 ±0.07	87.09 ±0.20	86.77 ±0.04
100%	86.33 ±0.11	87.03 ±0.11	87.13 ±0.14	86.83 ±0.03
SelfAug on LLaVA-1.5-7B				
0%	80.30 ±0.46	79.96 ±0.38	80.42 ±0.18	80.23 ±0.31
10%	81.14 ±0.08	79.91 ±0.32	80.19 ±0.20	80.41 ±0.12
30%	82.48 ±0.53	80.89 ±0.13	81.68 ±0.27	81.68 ±0.21
50%	**82.90** ±0.21	**81.68** ±0.16	**82.05** ±0.30	**82.21** ±0.14
100%	82.65 ±0.45	81.43 ±0.17	81.82 ±0.13	81.97 ±0.12
SelfAug on Qwen2.5-VL-7B				
0%	85.53 ±0.10	86.14 ±0.28	86.37 ±0.09	86.01 ±0.09
10%	85.81 ±0.07	85.89 ±0.18	86.69 ±0.10	86.13 ±0.05
30%	86.07 ±0.10	85.95 ±0.10	**86.95** ±0.11	86.32 ±0.02
50%	86.00 ±0.13	**86.11** ±0.06	86.84 ±0.02	86.32 ±0.05
100%	86.11 ±0.05	86.05 ±0.20	86.88 ±0.09	86.35 ±0.08

D.2.5 HARDWARE

Wall-clock measurements use one NVIDIA A100-80GB GPU, batch size 1, across three repeats. These exclude model and data loading.

D.3 PERFORMANCE UNDER BUDGETED ACTIVATION

D.3.1 FULL PERFORMANCE RESULTS

Table A6 reports POPE macro F1 for budgeted VCD and SelfAug across activation budgets, while Tables A7-A8 report their F1 for each of the nine dataset-difficulty splits on LLaVA-1.5-7B and Qwen2.5-VL-7B, respectively. Table A9 reports CHAIR_i for budgeted HALC, VCD, and SelfAug, and MME scores for budgeted VCD and SelfAug, all on LLaVA-1.5-7B. Table A10 reports VQAv2 accuracy, runtime, sequence agreement, and speedup for budgeted MVCD across activation budgets and in-context shot counts. Table A11 reports AVHBench accuracy, F1, and caption score for budgeted AVCD.

Table A7: Budgeted `VCD` and `SelfAug` on all nine `POPE` splits with `LLaVA-1.5-7B`.

Budget	COCO			GQA			A-OKVQA		
	rand.	pop.	adv.	rand.	pop.	adv.	rand.	pop.	adv.
VCD									
0%	81.25	80.05	77.52	82.94	78.04	76.32	83.30	79.91	75.70
10%	82.64	80.72	78.12	84.17	78.88	76.71	84.14	80.77	75.23
20%	84.09	82.89	80.15	85.49	80.16	76.57	85.65	81.94	76.29
30%	84.53	83.35	80.36	86.71	81.26	78.40	86.23	82.80	77.84
50%	84.72	83.80	80.93	86.70	81.57	79.29	86.79	83.17	78.32
100%	84.34	82.66	80.19	85.64	80.15	78.07	85.70	82.10	77.85
SelfAug									
0%	82.03	80.72	78.14	83.94	79.11	76.83	84.08	80.80	76.38
10%	83.14	81.48	78.79	84.43	78.53	76.77	84.14	80.46	75.96
20%	83.79	81.69	79.41	84.75	78.38	76.49	85.35	80.71	75.70
30%	84.70	82.68	80.06	85.27	79.91	77.48	85.68	81.91	77.44
50%	84.67	83.17	80.85	86.57	80.46	78.01	86.25	82.59	77.31
100%	84.85	82.71	80.39	85.89	80.10	78.30	86.20	82.12	77.15

Table A8: Budgeted `VCD` and `SelfAug` on all nine `POPE` splits with `Qwen2.5-VL-7B`.

Budget	COCO			GQA			A-OKVQA		
	rand.	pop.	adv.	rand.	pop.	adv.	rand.	pop.	adv.
VCD									
0%	86.69	85.75	84.16	89.42	85.52	83.50	89.37	87.41	82.33
10%	87.26	86.45	85.02	90.00	86.01	83.75	90.15	87.87	83.17
20%	87.57	86.47	85.06	90.14	86.04	83.98	90.28	88.33	83.21
30%	87.55	86.87	85.30	90.01	86.12	84.43	90.24	88.04	83.34
50%	87.29	86.70	85.32	90.37	85.97	84.05	90.21	88.04	83.03
100%	87.44	86.56	85.00	90.39	86.24	84.45	90.18	87.97	83.24
SelfAug									
0%	86.69	85.75	84.16	89.42	85.52	83.50	89.37	87.41	82.33
10%	86.91	85.85	84.68	89.71	85.08	82.86	90.03	87.71	82.33
20%	87.13	86.38	84.62	89.57	84.99	82.67	90.24	87.70	82.67
30%	86.99	86.18	85.02	89.41	85.13	83.31	90.19	87.80	82.86
50%	87.02	86.16	84.84	89.55	85.44	83.33	89.94	87.89	82.68
100%	87.08	86.25	85.01	89.70	85.13	83.33	90.20	87.69	82.76

Table A9: Budgeted `HALC`, `VCD`, and `SelfAug` on `CHAIR`, and `VCD` and `SelfAug` on `MME`, with `LLaVA-1.5-7B`.

Budget	$CHAIR_i$ (↓)			MME total (↑)	
	HALC	VCD	SelfAug	VCD	SelfAug
0%	8.25 -	9.88 ±0.08	9.26 ±0.25	1563.9 ±10.4	1550.9 ±51.6
10%	<u>7.48</u> -	8.93 ±0.21	<u>8.60</u> ±0.95	1624.7 ±7.4	1622.1 ±21.9
30%	**7.41** -	<u>8.79</u> ±0.45	8.74 ±0.07	1616.2 ±14.1	1614.3 ±8.2
50%	7.51 -	**8.59** ±0.19	9.25 ±0.38	**1651.3** ±20.2	<u>1647.0</u> ±12.4
100%	7.70 -	8.81 ±0.20	**8.22** ±0.27	1648.9 ±37.5	**1668.8** ±27.9

Table A10: Budgeted `MVCD` on `VQAv2` with `Qwen2-7B-Instruct` at various counts of ICL examples.

Budget	VQA acc. (↑)	Wall-clock (s) (↓)	Speedup (↑)	Seq. agree. (%)
1 in-context learning example				
0%	57.54 ±1.13	39.80 ±0.24	1.78×	72.47
10%	57.93 ±1.00	47.83 ±1.14	1.48×	87.53
30%	**59.23** ±1.01	53.69 ±0.69	1.32×	99.53
50%	**59.15** ±1.02	60.30 ±1.13	1.18×	100.00
100%	59.15 ±1.02	70.93 ±1.73	1.00×	100.00
2 in-context learning examples				
0%	58.81 ±0.94	52.17 ±1.24	1.56×	72.93
10%	60.57 ±1.04	58.77 ±0.75	1.38×	88.53
30%	**60.97** ±0.88	64.41 ±0.36	1.26×	99.87
50%	**60.97** ±0.88	70.30 ±0.50	1.16×	100.00
100%	60.97 ±0.88	81.39 ±0.85	1.00×	100.00
3 in-context learning examples				
0%	59.73 ±0.90	61.27 ±1.44	1.49×	71.07
10%	60.86 ±1.29	68.68 ±0.83	1.33×	87.53
30%	**61.53** ±1.04	74.01 ±1.67	1.23×	99.73
50%	**61.51** ±1.00	80.03 ±1.55	1.14×	99.93
100%	61.51 ±1.00	91.20 ±1.94	1.00×	100.00
4 in-context learning examples				
0%	60.73 ±0.80	69.92 ±0.51	1.43×	72.00
10%	61.44 ±0.53	76.58 ±0.36	1.30×	88.47
30%	**61.88** ±0.60	83.09 ±0.44	1.20×	100.00
50%	**61.88** ±0.60	88.48 ±0.71	1.13×	100.00
100%	61.88 ±0.60	99.65 ±0.19	1.00×	100.00
5 in-context learning examples				
0%	61.37 ±0.52	78.10 ±0.77	1.39×	71.00
10%	**62.30** ±0.79	86.81 ±1.17	1.25×	87.80
30%	**62.08** ±0.64	91.50 ±1.18	1.19×	99.73
50%	**62.08** ±0.64	97.31 ±1.30	1.11×	99.80
100%	62.08 ±0.64	108.44 ±1.24	1.00×	100.00

Table A11: Budgeted `AVCD` on `AVHBench` with `VideoLLaMA2.1-7B-AV`.

Budget	Discriminative		Caption score
	Acc. (%)	F1 (%)	
0%	74.31 ±0.6	75.58 ±0.9	2.84 -
10%	75.93 ±0.2	77.16 ±0.2	3.001 -
30%	**76.59** ±0.0	77.29 ±0.0	3.005 -
50%	**76.59** ±0.4	**77.37** ±0.3	3.005 -
100%	75.77 ±0.6	76.53 ±0.6	**3.006** -

Table A12: POPE results of ICD with the LLaVA-1.5-7B backbone.

Budget	COCO			GQA			A-OKVQA			Macro
	rand.	pop.	adv.	rand.	pop.	adv.	rand.	pop.	adv.	
0%	85.21	82.39	79.13	83.83	77.93	74.93	83.66	79.56	74.71	80.15 ±0.05
10%	84.84	82.35	79.25	83.75	77.81	75.01	83.78	79.56	74.71	80.12 ±0.04
30%	84.69	82.88	79.33	84.32	77.96	75.18	84.07	79.74	74.59	80.31 ±0.07
50%	85.72	83.51	79.78	85.01	78.35	75.77	84.56	80.50	75.12	80.92 ±0.13
100%	86.27	84.11	80.08	85.62	78.72	75.97	85.43	81.09	75.40	81.41 ±0.04

Table A13: Performance comparison of CHASD's confidence threshold (0.5) to our flip susceptibility at a matched budget (8.3%) on MME.

Method	Fixed threshold	MME total
Conf. (CHASD)	0.5	1753.91 ±0.00
Flip susceptibility	0.779	**1786.00** ±8.52

D.3.2 Additional MCD methods

Complementing the MCD methods evaluated in the main paper, Table A12 reports results for ICD, whose contrast is defined against the LLaVA-1.5 system prompt. Performance generally improves with the activation budget, offering a quality-latency trade-off.

Efficient MCD baselines. We further evaluate CHASD (Huang et al., 2026), an MCD baseline with a built-in efficiency mechanism. *As the authors' code is unavailable, we reimplement the contrastive construction and activation gate according to details provided in the paper to the best of our abilities.* First, we replace our flip susceptibility with the confidence signal proposed by CHASD while retaining the same budgeted activation framework. Table 3 reports the resulting MME performance across different activation budgets. Flip susceptibility achieves higher performance at B=10% and B=30%. At B=50%, both signals activate the same steps and give identical results. Second, we compare against the original fixed-threshold policy of CHASD, which yields an activation rate of 8.3%. Table A13 compares its MME performance with our method under the same B=8.3% activation budget.

D.3.3 Per-run results and paired tests

Table A14 reports per-experiment POPE macro F1, pairing each budgeted experiment with MCD using the same random seed. The performance results show consistent gains over full MCD at B=50%. Furthermore, budgeted activation adds no randomness: its variability arises from sampling or contrastive perturbations.

D.4 Latency analysis

End-to-end inference speedup on benchmarks. Fig. A1 reports measured inference speedups at various budgets relative to full-activation MCD on real benchmarks. Speedup generally grows as the budget falls, where the relative gain is determined by the MCD method's auxiliary branch: HALC, whose focal-contrast detector fires per detectable word, reaches 3.19× at B=30%. SelfAug performs a compute-heavy augmentation-selection process before decoding, thus heavily reducing the maximum latency improvements at any non-zero budget. On MME, whose answers are a handful of tokens, this unbudgeted setup dominates total latency and the curve flattens near 1.05×, against 1.31× on CHAIR's much longer captions, where gated decoding is the larger share of the total.

Latency decomposition. We conduct two controlled timing experiments to study the effects of activation budget and output length on inference latency. We evaluate VCD and SelfAug with LLaVA-1.5-7B and Qwen2.5-VL-7B, and AVCD with VideoLLaMA2.1-7B-AV, using one fixed COCO image for the image models and one fixed AVHBench clip for the audio-visual model,

Table A14: Per-run POPE macro F1 for budgeted VCD and SelfAug on LLaVA-1.5-7B and Qwen2.5-VL-7B, with runs paired across budgets.

	VCD				SelfAug			
Budget	Run 1	Run 2	Run 3	Mean	Run 1	Run 2	Run 3	Mean
				LLaVA-1.5-7B				
0%	79.18	79.48	79.69	79.45 ±0.21	79.80	80.51	80.38	80.23 ±0.31
10%	80.02	80.53	79.91	80.15 ±0.27	80.44	80.25	80.54	80.41 ±0.12
20%	81.58	81.63	81.19	81.47 ±0.20	80.60	80.85	80.64	80.70 ±0.11
30%	82.23	82.19	82.73	82.39 ±0.25	81.88	81.78	81.39	81.68 ±0.21
50%	82.87	82.72	82.84	82.81 ±0.06	82.01	82.30	82.31	82.21 ±0.14
100%	81.69	81.84	82.03	81.86 ±0.14	82.06	82.04	81.80	81.97 ±0.12
				Qwen2.5-VL-7B				
0%	86.14	85.96	85.94	86.01 ±0.09	86.14	85.96	85.94	86.01 ±0.09
10%	86.68	86.56	86.65	86.63 ±0.05	86.14	86.06	86.18	86.13 ±0.05
20%	86.76	86.85	86.75	86.79 ±0.05	86.24	86.19	86.24	86.22 ±0.03
30%	86.91	86.90	86.82	86.88 ±0.04	86.29	86.34	86.34	86.32 ±0.02
50%	86.75	86.84	86.74	86.77 ±0.04	86.38	86.32	86.25	86.32 ±0.05
100%	86.87	86.80	86.81	86.83 ±0.03	86.35	86.25	86.45	86.35 ±0.08

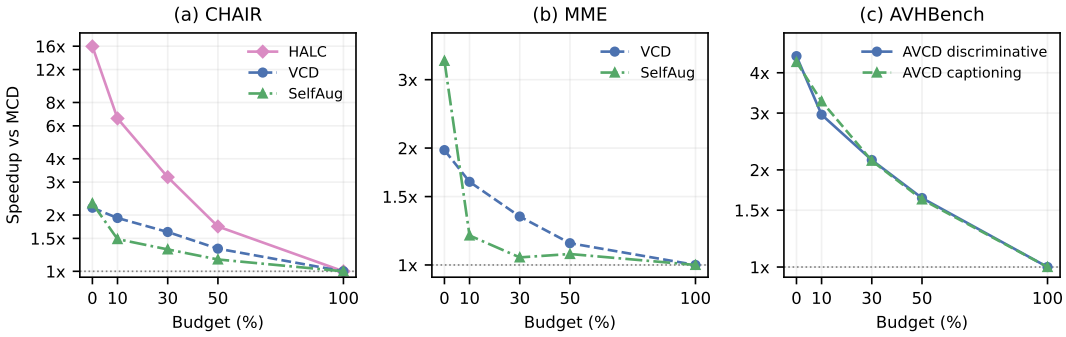


Figure A1: Measured end-to-end wall-clock speedup versus MCD (B=100%) at various budgets.

with greedy decoding for fixed output lengths $N \in \{32, 128, 256, 512, 1024\}$. Tables A15 and A16 decompose latency into primary setup, auxiliary setup, and decoding at short and long output lengths:

- Primary setup includes modality encoding and prompt prefill.
- Auxiliary setup includes auxiliary construction and its initial encoding and prefill.
- Decoding covers subsequent forward passes, cache maintenance, and generation overhead.

The reported percentages denote each component's fraction of total latency, while per-token latency is computed from decoding time only. Notably, for SelfAug with Qwen2.5-VL-7B at B=30%, auxiliary setup accounts for 64.9% of total latency at 32 tokens but only 5.6% at 1024 tokens, showing that its fixed setup cost is rapidly amortized as generation length increases.

Latency versus output length. Fig. A2 shows the relationship between total latency and output length for each activation budget, averaged over three runs. We include the endpoint of 0 tokens as a prefill-only measurement. For AVCD, the timing harness caches the primary prefix and retains current-token modality statistics, recomputing the three auxiliary branches at each activation since their attention masks depend on the current token; after the initial auxiliary setup, these full-prefix auxiliary refreshes belong to the decoding component.

Table A15: Latency decomposition at 32 output tokens. Total time shows the mean and sample standard deviation of three repeats. Component entries show mean seconds and percentage of mean total time.

Budget	Total (s)	Primary		Auxiliary		Decoding		ms/token
		s	%	s	%	s	%	
VCD / LLaVA-1.5-7B								
0%	0.574±0.001	0.074	13.0%	0.000	0.0%	0.499	87.0%	15.60
10%	0.687±0.001	0.074	10.8%	0.076	11.1%	0.537	78.2%	16.78
30%	0.778±0.036	0.075	9.7%	0.075	9.7%	0.628	80.7%	19.62
50%	0.863±0.046	0.075	8.7%	0.077	8.9%	0.711	82.4%	22.21
100%	1.146±0.005	0.075	6.5%	0.077	6.8%	0.994	86.7%	31.07
SelfAug / LLaVA-1.5-7B								
0%	0.573±0.001	0.075	13.0%	0.000	0.0%	0.499	87.0%	15.59
10%	1.716±0.001	0.074	4.3%	1.107	64.5%	0.536	31.2%	16.76
30%	1.808±0.038	0.075	4.1%	1.106	61.2%	0.627	34.7%	19.60
50%	1.891±0.047	0.075	3.9%	1.107	58.5%	0.710	37.5%	22.19
100%	2.177±0.006	0.075	3.4%	1.109	50.9%	0.993	45.6%	31.04
VCD / Qwen2.5-VL-7B								
0%	0.578±0.000	0.087	15.0%	0.000	0.0%	0.491	85.0%	15.34
10%	0.725±0.004	0.087	12.0%	0.104	14.4%	0.534	73.7%	16.68
30%	0.826±0.040	0.087	10.6%	0.103	12.5%	0.635	76.9%	19.86
50%	0.916±0.047	0.087	9.5%	0.104	11.3%	0.725	79.1%	22.65
100%	1.172±0.006	0.087	7.4%	0.104	8.9%	0.981	83.7%	30.65
SelfAug / Qwen2.5-VL-7B								
0%	0.578±0.001	0.087	15.1%	0.000	0.0%	0.491	84.9%	15.33
10%	1.962±0.004	0.086	4.4%	1.342	68.4%	0.534	27.2%	16.68
30%	2.061±0.043	0.087	4.2%	1.338	64.9%	0.635	30.8%	19.85
50%	2.153±0.048	0.087	4.0%	1.341	62.3%	0.725	33.7%	22.67
100%	2.409±0.008	0.087	3.6%	1.341	55.7%	0.981	40.7%	30.64
AVCD / VideoLLaMA2.1-7B-AV								
0%	1.239±0.003	0.536	43.2%	0.000	0.0%	0.703	56.8%	21.98
10%	6.906±0.041	0.538	7.8%	1.878	27.2%	4.490	65.0%	140.31
30%	17.060±4.413	0.537	3.1%	1.884	11.0%	14.639	85.8%	457.46
50%	26.483±5.728	0.537	2.0%	1.875	7.1%	24.071	90.9%	752.23
100%	61.695±0.226	0.532	0.9%	1.869	3.0%	59.294	96.1%	1852.93

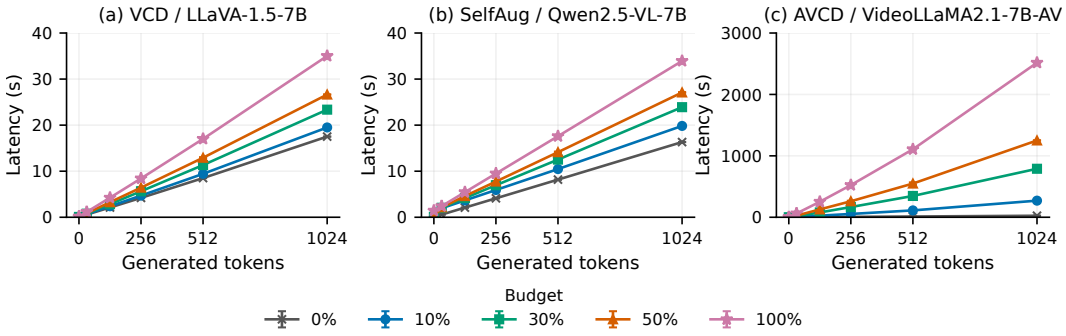


Figure A2: Latency versus output lengths at various activation budgets. 0 generated tokens indicate prefill time.

Table A16: Latency decomposition at 1024 output tokens, using the same notation and timing boundary as Table A15.

Budget	Total (s)	Primary		Auxiliary		Decoding		ms/token
		s	%	s	%	s	%	
VCD / LLaVA-1.5-7B								
0%	17.512±0.003	0.076	0.4%	0.000	0.0%	17.435	99.6%	17.03
10%	19.495±0.198	0.074	0.4%	0.075	0.4%	19.345	99.2%	18.89
30%	23.362±0.086	0.074	0.3%	0.076	0.3%	23.213	99.4%	22.67
50%	26.637±0.048	0.074	0.3%	0.077	0.3%	26.487	99.4%	25.87
100%	35.018±0.013	0.075	0.2%	0.078	0.2%	34.866	99.6%	34.05
SelfAug / LLaVA-1.5-7B								
0%	17.497±0.009	0.076	0.4%	0.000	0.0%	17.421	99.6%	17.01
10%	20.526±0.199	0.074	0.4%	1.106	5.4%	19.345	94.2%	18.89
30%	24.386±0.087	0.074	0.3%	1.106	4.5%	23.206	95.2%	22.66
50%	27.662±0.051	0.074	0.3%	1.108	4.0%	26.481	95.7%	25.86
100%	36.041±0.014	0.075	0.2%	1.109	3.1%	34.857	96.7%	34.04
VCD / Qwen2.5-VL-7B								
0%	16.325±0.004	0.088	0.5%	0.000	0.0%	16.237	99.5%	15.86
10%	18.551±0.211	0.087	0.5%	0.101	0.5%	18.363	99.0%	17.93
30%	22.660±0.093	0.087	0.4%	0.104	0.5%	22.469	99.2%	21.94
50%	25.849±0.092	0.086	0.3%	0.101	0.4%	25.662	99.3%	25.06
100%	32.691±0.012	0.087	0.3%	0.102	0.3%	32.502	99.4%	31.74
SelfAug / Qwen2.5-VL-7B								
0%	16.324±0.019	0.088	0.5%	0.000	0.0%	16.236	99.5%	15.86
10%	19.830±0.289	0.087	0.4%	1.338	6.7%	18.406	92.8%	17.97
30%	23.901±0.107	0.086	0.4%	1.337	5.6%	22.478	94.0%	21.95
50%	27.094±0.085	0.086	0.3%	1.339	4.9%	25.669	94.7%	25.07
100%	33.927±0.006	0.087	0.3%	1.339	3.9%	32.501	95.8%	31.74
AVCD / VideoLLaMA2.1-7B-AV								
0%	24.776±0.010	0.536	2.2%	0.000	0.0%	24.240	97.8%	23.67
10%	270.464±27.689	0.534	0.2%	1.884	0.7%	268.046	99.1%	261.76
30%	789.728±12.904	0.534	0.1%	1.876	0.2%	787.318	99.7%	768.87
50%	1250.516±3.140	0.535	0.0%	1.882	0.2%	1248.099	99.8%	1218.85
100%	2515.936±11.460	0.537	0.0%	1.876	0.1%	2513.524	99.9%	2454.61

D.5 Activation-policy comparisons and diagnostics

D.5.1 Fixed thresholds versus online calibration

To assess the difficulty in transferability of fixed thresholds used by existing methods, we calibrate a threshold on `LLaVA-1.5-7B` using the `COCO`-adversarial split to achieve a target activation rate of 30%. We then apply this threshold unchanged to all nine `POPE` splits. As shown in Table A17, the resulting activation rate varies substantially across splits, whereas our budgeted activation policy consistently tracks the target rate more closely.

The short `AVHBench` stream also motivates the warmup in Sec. 4.1. With approximately 470 generated tokens, N_w=200 imposes an activation floor near 42%, precluding low-rate operation. At B=10%, reducing warmup to N_w=20 cuts the realized rate by approximately two thirds at equal accuracy. Sec. 5.1 therefore sets warmup according to benchmark stream length.

D.5.2 Analysis of flip valence

Under an activation policy π with decisions a_t, step t is a flip of π iff $a_t\phi_t$=1 (Def. 1). Let i_t^{gt} denote the correct token at step t. On a benchmark d, the positive flips of π, which correct the vanilla token,

Table A17: Realized activation rates across nine POPE splits on LLaVA-1.5-7B using a fixed threshold $\tau^\star$=0.0342 calibrated for a 30% target.

Dataset	Split	Realized rate (%)
COCO	random	24.45
COCO	popular	25.77
A-OKVQA	random	28.49
GQA	random	28.57
COCO	adversarial	29.30
A-OKVQA	popular	31.72
GQA	popular	33.89
A-OKVQA	adversarial	34.59
GQA	adversarial	34.69
Mean absolute deviation (pp, ↓)		
Fixed threshold		3.14
Budgeted activation (Ours)		0.30

Table A18: Answer-flip counts and valence AUROC for VCD on LLaVA-1.5-7B across POPE split types.

POPE Split	Positive flips	Negative flips	Net effect	Negative share	Valence AUROC (%)
Random	175.7 ±3.3	278.0 ±2.2	−102.3 ±5.2	61.3%	58.8
Popular	211.0 ±3.7	389.7 ±6.3	−178.7 ±5.8	64.9%	60.8
Adversarial	234.7 ±9.4	411.0 ±5.7	−176.3 ±5.3	63.7%	58.6
All	621.3 ±12.7	1078.7 ±12.3	−457.3 ±4.0	63.5%	59.3

and its negative flips, which corrupt a correct vanilla token, are

$$\mathcal{F}_d^+(\pi) = \{t : a_t\phi_t{=}1,\ \hat{\imath}_t = i_t^{\mathrm{gt}}\}, \qquad \mathcal{F}_d^-(\pi) = \{t : a_t\phi_t{=}1,\ i_t^\star = i_t^{\mathrm{gt}}\}, \tag{12}$$

where t ranges over the decoding steps on d, and their ratio is $\mathrm{P/N}_d(\pi) = |\mathcal{F}_d^+(\pi)| \,/\, |\mathcal{F}_d^-(\pi)|$. Over a set $\mathcal{D}$ of benchmarks, the distance of the macro-averaged ratio of π from that of full MCD (a_t=1 for all t) is

$$|\Delta\,\mathrm{P/N}|(\pi) = \Big|\frac{1}{|\mathcal{D}|}\sum_{d\in\mathcal{D}}\mathrm{P/N}_d(\pi) - \frac{1}{|\mathcal{D}|}\sum_{d\in\mathcal{D}}\mathrm{P/N}_d(\mathrm{MCD})\Big|. \tag{13}$$

Under greedy VCD on LLaVA-1.5-7B, negative flips outnumber positive flips on every POPE split type and make up 64% of all flips (Table A18). Flip susceptibility separates positive from negative flips only slightly above chance (AUROC 59%; Fig. A3), since it measures whether MCD will change the vanilla decision rather than whether the change is correct. Consequently, budgeted activation preserves the P/N of VCD (Table 4). The prevalence of negative flips is therefore inherited from MCD rather than introduced by budgeted activation.

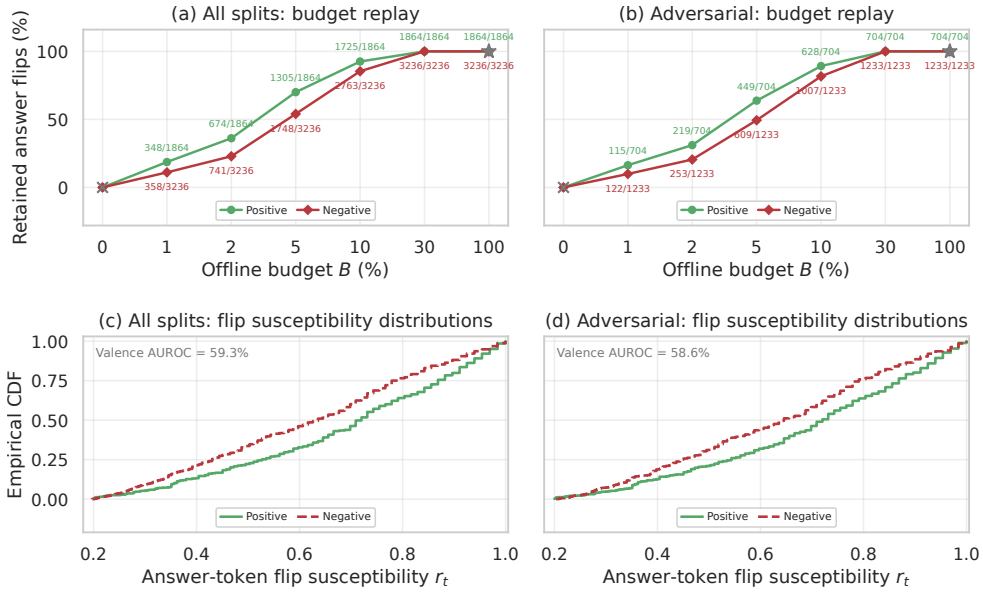


Figure A3: Answer-flip recall and flip susceptibility distributions by valence for VCD on POPE. (a), (c): all nine splits; (b), (d): adversarial splits only.