MITIGATING SCENE TRANSITION MISUNDERSTANDING IN VIDEO LLMS VIA SEMANTIC-SHIFT-AWARE CONTRASTIVE DECODING

Anonymous authors
Paper under double-blind review

ABSTRACT

Recognizing scene transitions is crucial for long-video understanding, yet Video Large Language Models (Video LLMs) often hallucinate nonexistent transitions or omit real ones. We analyze these failures through semantic-shift regions and find that over-attending to non-boundary shifts promotes hallucination, while under-attending to boundary-relevant shifts promotes omission. Controlled interventions provide evidence that such mis-calibrated attention causally contributes to scene-transition misunderstanding. Based on this finding, we propose SSACD (Semantic Shift-Aware Contrastive Decoding), a training-free method that preserves the original prediction in a clean branch while amplifying the identified attention tendency in an auxiliary branch and contrastively suppressing the resulting failure-prone predictions. SSACD realizes this principle through attention-level sharpening and input-level perturbation. Across multiple Video LLMs and benchmarks, SSACD outperforms existing training-free decoding methods on scene-transition understanding and improves multi-scene long-video performance.

1 INTRODUCTION

Understanding *scene transitions* is a fundamental capability for video understanding, especially in long-form videos where multiple scenes unfold over extended time spans (Wang et al., 2025; Fu et al., 2025; Fang et al., 2024). A scene transition marks a change in location, environment, or visual context, and provides a structural boundary for organizing a video into coherent events (Krishna et al., 2017; Yang et al., 2023). When such boundaries are misunderstood, a model may maintain an incorrect temporal context, leading to errors in event segmentation, temporal grounding, and causal reasoning (Li et al., 2024b; Liu et al., 2024b).

Despite recent advances in Video Large Language Models (Video LLMs) (Lin et al., 2024; Li et al., 2024a; Bai et al., 2025), current models still struggle to recognize scene transitions reliably (Li et al., 2025; Choong et al., 2024; Gao et al., 2025). In scene-transition evaluation, these failures can be categorized into two complementary types: *hallucination* and *omission*. Hallucination refers to predicting a scene transition when the scene remains continuous, while omission refers to missing an actual transition. These errors distort the temporal structure inferred from the video: hallucination may split a continuous event into unrelated scenes, whereas omission may cause a model to interpret a new scene using visual or semantic evidence from a previous one.

To better understand these failures, we examine how Video LLMs attend to *semantic-shift* regions, where adjacent frames or segments exhibit abrupt changes in visual-semantic content (Radford et al., 2021; Ye et al., 2023). Importantly, semantic shifts do not necessarily indicate scene transitions: while true boundaries often involve substantial semantic changes, similar shifts can also arise within a continuous scene due to object motion, camera movement, or other visual variations (Zhang et al., 2016; Ma et al., 2002). We therefore use semantic-shift regions as diagnostic locations where transition-relevant evidence may emerge without necessarily implying an actual scene boundary.

As illustrated in Figure 1, hallucination and omission can exhibit contrasting attention patterns around semantic-shift regions. In the hallucination example, the incorrect prediction places relatively stronger attention on a semantic change that does not correspond to a scene transition than the correct prediction. Conversely, in the omission example, the incorrect prediction places comparatively weaker

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.

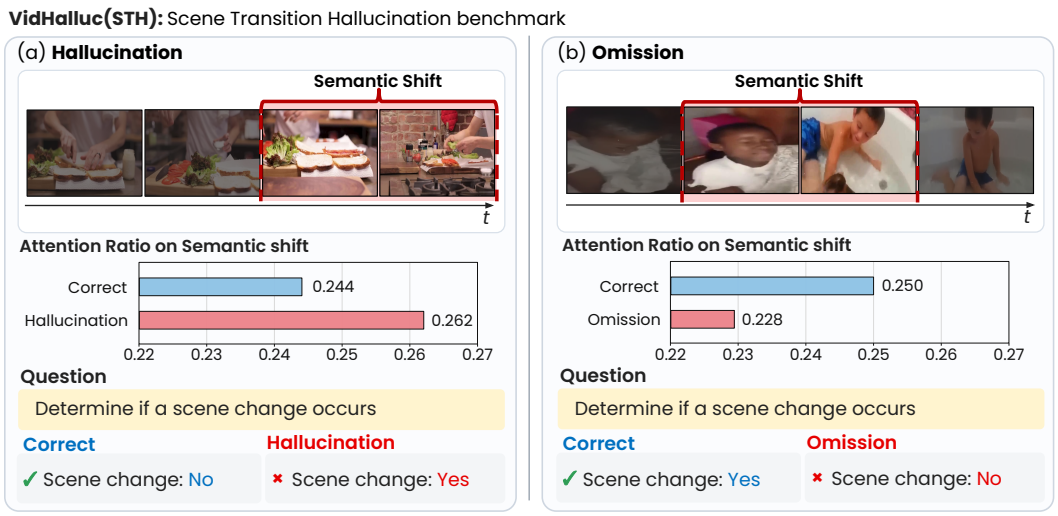

Figure 1: Examples and attention analysis of scene-transition hallucination and omission. (a) Hallucination: the model predicts a transition in a continuous scene. (b) Omission: the model misses a true transition. Each panel presents an illustrative video and its corresponding question and responses, while the bar chart compares the average semantic-shift attention ratios between correct and erroneous predictions on VidHalluc (STH).

attention around a true scene boundary. These examples suggest that erroneous scene-transition judgments may be accompanied by over- or under-allocation of attention to semantic-shift regions, motivating us to examine whether this pattern holds systematically.

Motivated by this observation, we hypothesize that scene-transition errors are associated with *mis-calibrated attention* to semantic-shift regions, i.e., over- or under-allocation of attention relative to their relevance to a scene boundary. To examine this possibility beyond qualitative examples, we conduct controlled interventions on attention to these regions. Specifically, we increase attention to semantic shifts in non-transition cases and examine whether hallucination becomes more frequent, while suppressing attention to semantic shifts at true boundaries and examining whether omission increases. Across benchmarks and Video LLM backbones, these interventions consistently strengthen the corresponding failure modes. These results provide evidence that mis-calibrated attention to semantic-shift regions causally contributes to scene-transition misunderstanding.

Guided by this finding, we propose SSACD (Semantic Shift-Aware Contrastive Decoding), which mitigates scene-transition misunderstanding by deliberately amplifying the observed over- and under-attention tendencies around semantic-shift regions and using the resulting failure-prone predictions as contrastive signals. Directly recalibrating attention may disrupt the model's use of semantic-shift evidence that remains useful for general video understanding. Instead, SSACD preserves the original video evidence through an untouched clean branch and constructs a failure-amplified branch that exaggerates the model's existing attention tendency. The discrepancy between the two branches reveals predictions that are disproportionately sensitive to these error-inducing tendencies, allowing them to be suppressed through contrastive decoding.

We realize this failure-amplification principle through two distinct intervention strategies. The first directly sharpens the model's attention distribution around semantic-shift regions, while the second perturbs the corresponding input frames to induce a similar amplification without directly modifying internal attention. Although they operate at different levels, both strategies amplify failure-prone predictions associated with semantic-shift attention and contrast them against the clean prediction. Both variants are training-free and require no parameter updates.

We evaluate SSACD on scene-transition understanding and general video understanding benchmarks, including long-form video tasks (Li et al., 2025; 2024b; Wang et al., 2025; Zhou et al., 2025). Across models and benchmarks, SSACD consistently reduces both hallucinated and omitted scene transitions and outperforms existing training-free decoding methods (Leng et al., 2024; Park et al., 2025; Jung et al., 2026). SSACD also preserves or improves performance on broader video understanding tasks.

Further analyses show that both attention-level sharpening and input-level perturbation amplify the error-inducing tendencies identified in our motivating analysis and yield effective contrastive signals.

Our contributions: (1) We identify attention mis-calibration around semantic-shift regions as a key factor in scene-transition misunderstanding, showing that over-attending to non-boundary shifts promotes hallucination, while under-attending to boundary-relevant shifts promotes omission. (2) We propose SSACD, a training-free method that amplifies these error-inducing attention tendencies through attention-level sharpening and input-level perturbation, using the resulting predictions as contrastive signals against a clean branch. (3) We demonstrate that SSACD reduces both hallucination and omission, outperforms existing training-free decoding methods, and preserves or improves general video understanding across benchmarks.

2 Related Work

Scene transition understanding concerns how a model detects and interprets changes between temporally adjacent video states. Early work studied physical shot boundary detection, including DeepSBD (Hassanien et al., 2017) and ClipShots (Tang et al., 2018), but these methods mainly address edit-level discontinuities rather than semantic changes in story or event structure. Later studies moved toward semantic scene segmentation in long-form videos, with MovieNet emphasizing scene-level annotation (Huang et al., 2020), LGSS modeling local-global structure (Rao et al., 2020), BaSSL introducing boundary-aware self-supervision (Mun et al., 2022), TranS4mer targeting long-range temporal modeling (Islam et al., 2023), and Scene-VLM introducing multimodal video scene segmentation with vision-language models (Berman et al., 2026). Recent Video LLM benchmarks extend scene transition understanding from localization to reasoning: MVBench and TVBench include a scene-transition task (Li et al., 2024b; Cores et al., 2024), EventHallusion studies event-level hallucination and proposes TCD (Zhang et al., 2024a), and VidHalluc evaluates action, temporal-sequence, and scene-transition hallucinations while introducing DINO-HEAL (Li et al., 2025). However, these works primarily focus on evaluating or mitigating scene-transition failures, with less attention to understanding the mechanisms underlying hallucination and omission. Our work instead investigates how models use local semantic changes around scene boundaries, taking semantic-shift regions as a diagnostic lens for analyzing these failure modes.

3 Motivating Analysis

Objective. Figure 1 shows that incorrect scene-transition judgments can accompany contrasting attention patterns around semantic-shift regions: relatively strong attention to a non-boundary shift in hallucination and relatively weak attention to a true boundary in omission. While suggestive, these qualitative examples do not establish whether such attention patterns merely correlate with the errors or actively contribute to them. We therefore conduct controlled attention interventions to test whether deliberately strengthening these tendencies increases the corresponding failure modes. Specifically, we examine whether emphasizing non-boundary semantic shifts promotes hallucination and whether suppressing boundary-relevant shifts promotes omission.

Setting. We uniformly sample $K = 8$ frames from each video and define a *semantic shift* as an abrupt change in visual-semantic content between temporally adjacent sampled frames. Given the sampled video $v = [I_1, \ldots, I_K]$ and a visual encoder $E(\cdot)$, we compute the semantic distance between adjacent frames as

$$d_i = 1 - \cos\big(E(I_i), E(I_{i+1})\big), \qquad i = 1, \ldots, K-1. \tag{1}$$

We then identify the set of semantic shifts as

$$\mathcal{S} = \{(i, i+1) \mid d_i > \tau; i = 1, \ldots, K-1\}, \tag{2}$$

where τ is the detection threshold. Each semantic shift is treated only as a candidate region containing potentially transition-relevant evidence, rather than as a predicted scene-transition boundary.

We then apply post-hoc attention steering (Zhang et al., 2024b) to the final transformer layer while leaving the remaining model parameters unchanged. For each semantic shift $(i, i+1) \in \mathcal{S}$, we define the *target attention ratio* as the fraction of visual attention assigned to its two constituent frames. For

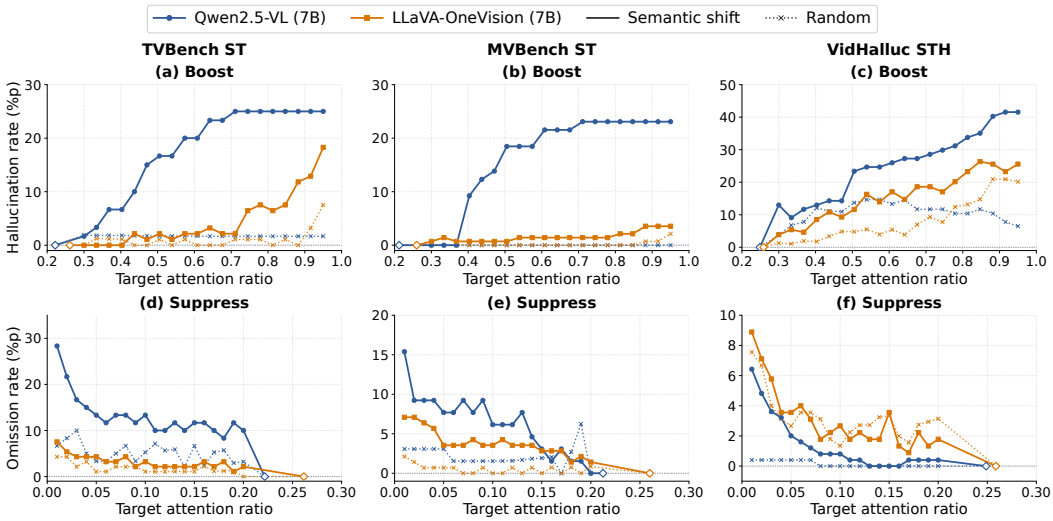

Figure 2: **Attention intervention on semantic-shift regions.** Hallucination and omission rates under interventions on semantic-shift regions and size-matched random regions across VidHalluc (STH), TVBench (ST), and MVBench (ST). *Boost* increases the target attention ratio in non-transition samples and measures hallucination, whereas *Suppress* decreases the target attention ratio in true-transition samples and measures omission. Solid and dotted lines denote interventions on semantic-shift and random regions, respectively.

non-transition cases, *Boost* increases this ratio and tests whether stronger emphasis on non-boundary semantic shifts increases hallucination. For true-transition cases, *Suppress* decreases the ratio and tests whether weakening boundary-relevant semantic shifts increases omission.

To isolate semantic-shift-specific effects, we compare each intervention with a size-matched random-region control under the same target attention ratio. Ground-truth transition labels are used only to separate non-transition and true-transition samples for this controlled analysis.

We conduct the analysis using Qwen2.5-VL-7B and LLaVA-OneVision-7B on the scene-transition subsets of VidHalluc (Li et al., 2025), TVBench (Cores et al., 2024), and MVBench (Li et al., 2024b). For semantic-shift detection, we use the CLIP visual encoder (ViT-B/32) (Radford et al., 2021).

Results. Figure 2 summarizes the results across the three benchmarks. Under *Boost*, hallucination increases as more attention is allocated to non-boundary semantic shifts, with semantic-shift interventions generally producing stronger effects than size-matched random-region controls. Conversely, under *Suppress*, omission increases as attention to semantic shifts at true boundaries is reduced, again with stronger effects for semantic-shift regions than for random controls.

Although the magnitude of the effect varies across models, benchmarks, and intervention strengths, the observed direction consistently follows our hypothesis. These controlled interventions provide evidence that mis-calibrated attention to semantic-shift regions *causally contributes* to scene-transition misunderstanding: over-emphasizing non-boundary shifts promotes hallucination, whereas under-emphasizing boundary-relevant shifts promotes omission. The results therefore indicate that scene-transition failures depend not only on the presence of visual-semantic changes, but also on how strongly the model relies on such changes when making transition judgments.

4 Method

4.1 Semantic Shift-Aware Contrastive Decoding

Our motivating analysis in Section 3 shows that over-emphasizing non-boundary semantic shifts increases hallucination, whereas under-emphasizing boundary-relevant shifts increases omission. While these findings reveal how the corresponding failure tendencies can be strengthened, they do not specify an optimal attention distribution for directly correcting them. Direct recalibration, as explored

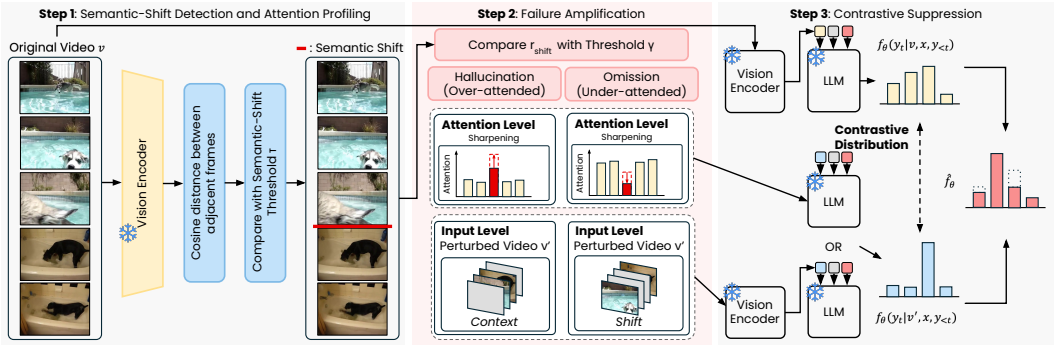

Figure 3: Overview of SSACD. SSACD first detects semantic-shift regions and profiles the model's attention to them, then amplifies the identified attention tendencies through either attention-level sharpening or input-level perturbation. Finally, failure-amplified predictions are suppressed through contrastive decoding.

in prior work (Liu et al., 2024a; Jiang et al., 2025), would therefore require imposing an uncertain attention target and may interfere with visual evidence that remains useful for video understanding.

Instead, SSACD preserves the model's original behavior in a *clean branch* and deliberately amplifies the identified failure tendency in an auxiliary *failure-amplified branch*. Contrastive decoding provides a natural mechanism for exploiting this amplified signal: predictions that become disproportionately favored as the failure tendency is strengthened can be suppressed relative to the clean prediction.

Given a video $v = [I_1, \ldots, I_K]$ and instruction x, let $f_{\text{clean}} = f_\theta(\cdot \mid v, x, y_{<t})$ denote the logits produced by the original Video LLM at decoding step t, and let $f_{\text{amplified}}$ denote the logits from the failure-amplified branch. Following prior contrastive decoding methods (Leng et al., 2024; Park et al., 2025), SSACD computes

$$\hat{f}_\theta = (1+\alpha) f_{\text{clean}} - \alpha f_{\text{amplified}}, \tag{3}$$

where $\alpha \geq 0$ controls the contrast strength. Predictions disproportionately strengthened in the failure-amplified branch are thereby down-weighted relative to the clean branch, while the clean branch retains the original video evidence and model behavior.

As illustrated in Figure 3, SSACD consists of three stages: (1) semantic shift detection and attention profiling, (2) failure amplification, and (3) contrastive suppression. We realize failure amplification either through *attention-level sharpening* or *input-level perturbation*. The latter is used as the default SSACD configuration in our main experiments.

4.2 Semantic Shift Detection and Attention Profiling

SSACD first detects semantic shifts from the input video using the criterion defined in Section 3, yielding $\mathcal{S} = \{s_n\}_{n=1}^{N}$, where $s_n = (i_n, i_n + 1)$ and $N = |\mathcal{S}|$. It then performs a preliminary clean decoding pass to characterize the model's attention to each shift. Let $\mathcal{T}_j$ denote the visual tokens corresponding to frame I_j. At decoding step t, the final-layer attention mass assigned to frame I_j is computed as

$$A_j^{(t)} = \frac{1}{H} \sum_{h=1}^{H} \sum_{q \in \mathcal{T}_j} a_{t,h,q}^{(L)}, \tag{4}$$

where H is the number of attention heads and $a_{t,h,q}^{(L)}$ denotes the attention weight assigned to visual token q by head h in the final transformer layer.

For each semantic shift s_n, we measure the fraction of accumulated visual attention assigned to its two constituent frames:

$$r_n = \frac{\sum_{t=1}^{T} \left(A_{i_n}^{(t)} + A_{i_n+1}^{(t)} \right)}{\sum_{t=1}^{T} \sum_{j=1}^{K} A_j^{(t)}}, \tag{5}$$

where T is the number of generated tokens. The ratio r_n represents how strongly the model attends to the semantic-shift region relative to the overall visual input.

Guided by the intervention findings in Section 3, SSACD uses r_n to determine the amplification mode for each shift. When $r_n < \gamma$, the shift is treated as *under-attended*; otherwise, it is treated as *over-attended*, where γ is the attention threshold. We denote the resulting mode by $z_n \in \{\text{UNDER}, \text{OVER}\}$. The former is subsequently amplified toward weaker attention, while the latter is amplified toward stronger attention.

4.3 Semantic Shift-Aware Failure Amplification

Given the amplification mode z_n determined from the preliminary clean pass, SSACD constructs a failure-amplified branch that deliberately strengthens the corresponding attention tendency around each semantic shift. We realize this principle through two strategies: *attention-level sharpening* and *input-level perturbation*.

4.3.1 Attention-Level Sharpening

The attention-level variant directly exaggerates the model's existing attention tendency around semantic shifts. Following post-hoc attention steering (Zhang et al., 2024b), we implement this intervention by multiplicatively rescaling selected attention weights and renormalizing the resulting attention distribution. We modify only the attention weights in the final transformer layer, leaving all preceding layers and model parameters unchanged.

Let $\mathcal{T}_{s_n}$ denote the visual tokens corresponding to the two frames of semantic shift s_n. For an under-attended shift, sharpening further decreases attention to $\mathcal{T}_{s_n}$; for an over-attended shift, it further increases their attention:

$$\tilde{a}^{(L)}_{t,h,q} = \frac{\eta_q a^{(L)}_{t,h,q}}{\sum_k \eta_k a^{(L)}_{t,h,k}}, \qquad \eta_q = \begin{cases} \lambda_{\text{under}}, & q \in \mathcal{T}_{s_n},\ z_n = \text{UNDER}, \\ \lambda_{\text{over}}, & q \in \mathcal{T}_{s_n},\ z_n = \text{OVER}, \\ 1, & \text{otherwise}, \end{cases} \tag{6}$$

where $0 < \lambda_{\text{under}} < 1$ and $\lambda_{\text{over}} > 1$.

For multiple semantic shifts, the same scaling is applied shift-wise according to each z_n, sharpening each region in its corresponding direction. Thus, under-attended shifts become further under-attended, whereas over-attended shifts become further over-attended. Decoding under the sharpened attention distribution yields the failure-amplified logits $f_{\text{amplified}}$.

4.3.2 Input-Level Perturbation

Rather than explicitly manipulating internal attention, the input-level variant amplifies the identified tendency by perturbing the underlying visual evidence. This avoids prescribing where and how strongly attention should be modified, and instead allows the corresponding attention response to arise through the model's own computation.

For each semantic shift, the perturbation mode is selected according to z_n.

Context Perturbation. For an over-attended shift (*i.e.*, $z_n = \text{OVER}$), we preserve its two shift-related frames and instead perturb the surrounding context. Specifically, we add Gaussian noise $\epsilon_j \sim \mathcal{N}(0, \sigma^2 I)$ to the selected context frames. Weakening contextual evidence makes the semantic shift relatively more salient, thereby reinforcing the existing over-attention tendency.

Shift Perturbation. For an under-attended shift (*i.e.*, $z_n = \text{UNDER}$), we apply the same Gaussian perturbation to its two shift-related frames. Weakening the shift-specific evidence further reinforces the existing under-attention tendency.

Multiple semantic shifts. When multiple shifts are detected, we process each shift independently: under-attended shifts select their shift frames, whereas over-attended shifts select their associated context. The selected regions are combined into a single perturbation mask, with each frame perturbed at most once, and the resulting video v' is used as the failure-amplified input. Full construction details are provided in Appendix A.

Applying the perturbation mask to the original video yields a perturbed video v', which is then passed through the same Video LLM to construct the failure-amplified branch and obtain $f_{\text{amplified}}$.

4.4 CONTRASTIVE SUPPRESSION

At each decoding step, the clean and failure-amplified branches are evaluated using the same decoding history. The two branches are then combined using Equation 3, suppressing predictions that become disproportionately strengthened in the failure-amplified branch.

When no semantic shift is detected, i.e., $\mathcal{S} = \emptyset$, no failure-amplified branch is constructed and SSACD reduces to standard decoding with the clean branch.

Following prior contrastive decoding methods (Leng et al., 2024; Park et al., 2025), we further restrict contrastive adjustment to tokens sufficiently supported by the clean branch:

$$\mathcal{V}_{\text{plausible}} = \left\{ y \in \mathcal{V} \;\middle|\; f_{\text{clean}}(y) \geq \log \beta + \max_{z \in \mathcal{V}} f_{\text{clean}}(z) \right\}, \tag{7}$$

where $\mathcal{V}$ denotes the output vocabulary and β controls the plausibility threshold. Tokens outside $\mathcal{V}_{\text{plausible}}$ are masked during contrastive decoding.

5 EXPERIMENTS

5.1 EXPERIMENTAL SETUP

We evaluate SSACD on both scene-transition-specific and broader video understanding benchmarks. Our experiments address four questions: (1) Does SSACD improve scene-transition understanding across different Video LLMs and benchmarks? (2) Does SSACD preserve broader video understanding while remaining effective in multi-scene long-video settings? (3) Does the proposed failure-amplification mechanism effectively strengthen the targeted semantic-shift attention tendency, and does this amplified tendency provide a more effective contrastive signal than smoothing it? (4) Which components contribute to the effectiveness of SSACD?

Benchmarks. We evaluate scene-transition understanding on VidHalluc (STH) (Li et al., 2025), TVBench (ST) (Cores et al., 2024), and MVBench (ST) (Li et al., 2024b), covering scene-transition classification, description, and multiple-choice reasoning. For broader video understanding, we use LVBench (Wang et al., 2025) and MLVU (Zhou et al., 2025), which include long-video and diverse temporal reasoning tasks.

Video LLMs and baselines. We evaluate Qwen2.5-VL-7B (Bai et al., 2025) and LLaVA-OneVision-7B (Li et al., 2024a), comparing SSACD with the base models and training-free baselines DINO-HEAL (Li et al., 2025), TCD (Zhang et al., 2024a), and VCD (Leng et al., 2024).

Implementation details. Unless otherwise specified, SSACD denotes the input-level variant used in our main comparisons; attention-level sharpening and smoothing are evaluated separately in the design analysis. Following prior contrastive decoding work (Zhang et al., 2024a), we uniformly sample eight frames and detect adjacent semantic shifts using CLIP ViT-B/32 (Radford et al., 2021) with $\tau = 0.2$. Results are averaged over three seeds, with further implementation details, hyperparameter analyses, and the effect of perturbation noise discussed in Appendices D and B.

5.2 MAIN RESULTS

Scene-transition understanding. Table 1 reports results on VidHalluc (STH), TVBench (ST), and MVBench (ST). Across all three benchmarks and both Video LLM backbones, SSACD consistently outperforms the corresponding base models and existing training-free decoding baselines, improving both classification and description on VidHalluc (STH) and yielding gains on TVBench (ST) and MVBench (ST).

Unlike generic hallucination mitigation or contrastive decoding methods, SSACD constructs its contrastive signal around semantic-shift regions, providing a transition-specific signal for mitigating hallucination and omission. Additional experiments with 72B-scale Video LLMs show that these gains persist at larger model scales; detailed results are provided in Appendix E.

Long-video understanding. Table 1 also reports results on LVBench and MLVU, which contain long, multi-scene videos with frequent semantic changes. SSACD consistently improves performance

Model	Method	Scene Transition Benchmark					General Benchmark	
		VidHalluc (STH)			MVBench (ST)	TVBench (ST)	LVBench	MLVU
		Class.↑	Desc.↑	Overall↑	Accuracy↑	Accuracy↑	Overall↑	M-Avg↑
Qwen2.5-VL	Base	63.3	48.6	57.4	77.8	69.2	23.3	31.8
	DINO-HEAL	65.8	49.3	59.2	79.5	70.3	24.6	32.8
	TCD	65.5	53.5	60.7	81.5	72.8	25.0	**37.4**
	VCD	69.2	52.5	62.4	80.3	71.1	24.6	34.7
	SSACD	**73.4**	**54.1**	**65.7**	**82.3**	**73.6**	**32.8**	**37.4**
LLaVA-OneVision	Base	32.9	38.6	35.2	47.8	72.8	25.4	35.1
	DINO-HEAL	31.7	40.8	35.3	49.0	73.0	26.6	34.8
	TCD	44.7	49.2	46.5	52.8	76.1	25.9	32.5
	VCD	45.0	48.9	46.6	49.7	75.1	25.0	36.2
	SSACD	**50.5**	**51.1**	**50.7**	**53.3**	**78.2**	**32.0**	**39.1**

Table 1: Performance comparison across models, methods, and video understanding benchmarks. MVBench (ST) and TVBench (ST) report accuracy on the Scene Transition subset. LVBench reports the overall accuracy, while MLVU reports M-Avg, the average accuracy across its seven multiple-choice tasks. Values are reported in %.

Model	Type	Auxiliary Design	VidHalluc (STH)		
			Class.↑	Desc.↑	Overall↑
Qwen2.5-VL	Base	–	63.3	48.6	57.4
	Smoothing	Direct Intervention	66.3	46.5	58.4
	$\text{SSACD}_{\text{Attn}}$	Attention Sharpening	72.2	43.0	60.5
	$\text{SSACD}_{\text{Input}}$	Input Perturbation	**73.4**	**54.1**	**65.7**
LLaVA-OV	Base	–	32.9	38.6	35.2
	Smoothing	Direct Intervention	36.8	46.5	40.7
	$\text{SSACD}_{\text{Attn}}$	Attention Sharpening	39.4	48.4	43.0
	$\text{SSACD}_{\text{Input}}$	Input Perturbation	**50.5**	**51.1**	**50.7**

Table 2: Comparison of auxiliary designs on VidHalluc (STH). Smoothing is a comparison, while $\text{SSACD}_{\text{Attn}}$ and $\text{SSACD}_{\text{Input}}$ denote the attention- and input-level SSACD variants, respectively.

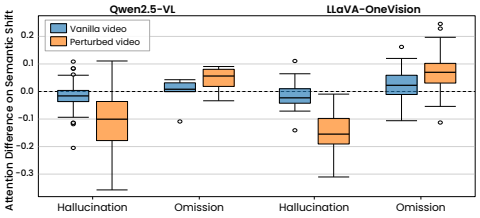

Figure 4: Attention behavior around semantic shifts. Blue boxes show clean-branch attention patterns, while orange boxes show how Context and Shift Perturbations amplify over- and under-attention, respectively.

across both Video LLM backbones, suggesting that its benefit extends beyond transition-specific questions to long-video settings where distinguishing scene transitions from incidental semantic changes is important for long-range temporal understanding.

5.3 Design Choice Analysis

Failure amplification versus direct intervention. Table 2 compares Smoothing with the two failure-amplification strategies of SSACD. While Smoothing improves over the base model, both SSACD variants achieve stronger performance across the two Video LLM backbones. One possible explanation is that directly modifying attention requires deciding how the attention distribution should be adjusted without knowing its optimal allocation, which may also disturb semantic-shift evidence that remains useful for prediction. SSACD instead preserves the original prediction in an untouched clean branch and confines the intervention to an auxiliary branch, using the amplified tendency only as a contrastive signal.

Among the two amplification strategies, input-level perturbation ($\text{SSACD}_{\text{Input}}$) further outperforms attention-level sharpening ($\text{SSACD}_{\text{Attn}}$). Since both variants preserve the same clean branch and contrastive decoding scheme, this difference highlights the importance of how the failure-amplified branch is constructed. Attention-level sharpening requires explicitly specifying where and how strongly internal attention should be amplified, whereas input-level perturbation modifies the visual evidence and allows the corresponding attention response to arise through the model's internal computation. This suggests that inducing failure amplification through the input provides a more effective contrastive signal than directly manipulating the attention distribution.

Does input perturbation induce the intended amplification? Unlike attention-level sharpening, which directly modifies internal attention, input-level perturbation operates only on the visual input. We therefore examine whether it induces the intended change in semantic-shift attention.

Method	VidHalluc (STH)		
	Hallu.↓	Omiss.↓	Overall↑
SSACD	**22.2**	**35.8**	**50.7**
w/o semantic shift detection	25.8	40.4	47.3
w/o shift-aware perturbation	31.0	37.6	46.6
w/o adaptive perturbation selection (shift only)	31.0	38.1	46.4
w/o adaptive perturbation selection (context only)	24.7	42.1	47.0

Table 3: Ablation study of input-level SSACD on VidHalluc (STH). Values are reported in %.

Model	Method	Latency (s)	Tokens/s
LLaVA-OV (7B)	Base	1.20	33.23
	DINO-HEAL	1.27	28.74
	TCD	1.94	17.28
	VCD	2.52	16.88
	$\text{SSACD}_{\text{Attn}}$	1.92	13.14
	$\text{SSACD}_{\text{Input}}$	2.89	17.40

Table 4: Inference efficiency on LLaVA-OneVision-7B using an NVIDIA RTX PRO 6000. Latency is measured per sample.

As shown in Figure 4, *Context Perturbation* increases attention to over-attended shifts by weakening their surrounding context, whereas *Shift Perturbation* further reduces attention to under-attended semantic shifts. These results show that input-level perturbation amplifies the attention tendency identified during the clean profiling pass without directly intervening in the attention distribution.

Together, these results support two key design choices of SSACD: preserving the clean prediction while exploiting an amplified auxiliary signal through contrastive decoding, and inducing this amplification through input-level perturbation rather than explicit attention manipulation.

5.4 Additional Analysis

Ablation study. Table 3 evaluates the key design components of input-level SSACD. Replacing detected semantic shifts with random regions degrades performance, confirming the importance of localizing semantic changes when constructing the failure-amplified branch. Likewise, applying noise uniformly to all frames performs worse than semantic-shift-aware perturbation, supporting targeted intervention around semantic-shift regions.

The two perturbation modes also play distinct roles. Removing Context Perturbation, which targets over-attended shifts, increases hallucination, whereas removing Shift Perturbation, which targets under-attended shifts, increases omission. These results support adaptive perturbation selection based on the shift-specific attention profile.

Inference efficiency. Table 4 reports SSACD efficiency on LLaVA-OV (7B). SSACD incurs additional cost over vanilla decoding due to profiling and failure-amplified branch evaluation, trading efficiency for improved scene-transition understanding. $\text{SSACD}_{\text{Attn}}$ is faster than $\text{SSACD}_{\text{Input}}$, whose throughput remains comparable to TCD and VCD. When no semantic shift is detected, SSACD falls back to standard decoding, avoiding branch overhead.

6 Limitations

SSACD relies on fixed-threshold CLIP-based semantic-shift detection, which may be less robust to gradual or domain-dependent semantic changes. In addition, input-level SSACD uses frame-level Gaussian perturbations, providing relatively coarse control over the visual evidence used for failure amplification. SSACD also incurs additional inference cost from attention profiling and the evaluation of both clean and failure-amplified branches. Future work could explore adaptive shift detection, finer-grained perturbations, and more efficient branch evaluation.

7 Conclusion

We investigated scene-transition misunderstanding in Video LLMs and found that mis-calibrated attention to semantic-shift regions causally contributes to hallucination and omission. Based on this finding, we introduced SSACD, a training-free method that preserves a clean branch while amplifying failure-prone tendencies in an auxiliary branch for contrastive suppression. SSACD consistently improves scene-transition understanding across models and benchmarks while remaining effective on multi-scene long-video tasks, highlighting semantic-shift-aware failure amplification as an effective strategy for Video LLMs.

AI USE STATEMENT

We used generative AI tools, including ChatGPT and Codex, to assist with brainstorming and refining research directions, supporting implementation and experimental workflows, sourcing and summarizing relevant background literature, and drafting and polishing portions of the manuscript. These tools served only as assistive aids. All research decisions, experimental designs, implementations, analyses, claims, and manuscript content were reviewed and verified by the authors. We did not use generative AI to generate synthetic datasets or to formulate or prove mathematical claims. The authors take full responsibility for the final content of this work.

ETHICS STATEMENT

This work studies hallucination and omission in Video LLMs using existing models and research benchmarks and does not introduce new human-subject data. SSACD is intended to improve the reliability of scene-transition understanding, but improved performance on the evaluated benchmarks does not eliminate other forms of video hallucination or guarantee reliability in safety-critical applications.

REPRODUCIBILITY STATEMENT

We provide the methodological details required to reproduce SSACD in Section 4, including semantic-shift detection, attention profiling, failure amplification, and contrastive decoding. The experimental setup specifies the evaluated Video LLMs, benchmarks, baselines, frame-sampling procedure, and semantic-shift detection settings. Additional details are provided in the Appendix, including multiple-shift handling (Appendix A), input perturbation strategies (Appendix B), the adaptive plausibility constraint (Appendix C), hyperparameter sensitivity (Appendix D), scaling results (Appendix E), and efficiency and implementation details (Appendix F).

REFERENCES

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, et al. Qwen2. 5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025.

Nimrod Berman, Adam Botach, Emanuel Ben-Baruch, Shunit Haviv Hakimi, Asaf Gendler, Ilan Naiman, Erez Yosef, and Igor Kviatkovsky. Scene-vlm: Multimodal video scene segmentation via vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 39848–39857, 2026.

Wey Yeh Choong, Yangyang Guo, and Mohan Kankanhalli. Vidhal: Benchmarking temporal hallucinations in vision llms. *arXiv preprint arXiv:2411.16771*, 2024.

Daniel Cores, Michael Dorkenwald, Manuel Mucientes, Cees GM Snoek, and Yuki M Asano. Tvbench: Redesigning video-language evaluation. 2024.

Xinyu Fang, Kangrui Mao, Haodong Duan, Xiangyu Zhao, Yining Li, Dahua Lin, and Kai Chen. Mmbench-video: A long-form multi-shot benchmark for holistic video understanding, 2024. URL https://arxiv.org/abs/2406.14515.

Chaoyou Fu, Yuhan Dai, Yongdong Luo, Lei Li, Shuhuai Ren, Renrui Zhang, Zihan Wang, Chenyu Zhou, Yunhang Shen, Mengdan Zhang, et al. Video-mme: The first-ever comprehensive evaluation benchmark of multi-modal llms in video analysis. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 24108–24118. IEEE, 2025.

Hongcheng Gao, Jiashu Qu, Jingyi Tang, Baolong Bi, Yue Liu, Hongyu Chen, Li Liang, Li Su, and Qingming Huang. Exploring hallucination of large multimodal models in video understanding: Benchmark, analysis and mitigation. *arXiv preprint arXiv:2503.19622*, 2025.

Ahmed Hassanien, Mohamed Elgharib, Ahmed Selim, Sung-Ho Bae, Mohamed Hefeeda, and Wojciech Matusik. Large-scale, fast and accurate shot boundary detection through spatio-temporal convolutional neural networks. *arXiv preprint arXiv:1705.03281*, 2017.

Qingqiu Huang, Yu Xiong, Anyi Rao, Jiaze Wang, and Dahua Lin. Movienet: A holistic dataset for movie understanding. In *European conference on computer vision*, pp. 709–727. Springer, 2020.

Md Mohaiminul Islam, Mahmudul Hasan, Kishan Shamsundar Athrey, Tony Braskich, and Gedas Bertasius. Efficient movie scene detection using state-space transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 18749–18758, 2023.

Zhangqi Jiang, Junkai Chen, Beier Zhu, Tingjin Luo, Yankun Shen, and Xu Yang. Devils in middle layers of large vision-language models: Interpreting, detecting and mitigating object hallucinations via attention lens. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 25004–25014. IEEE, 2025.

Chaeyoung Jung, Youngjoon Jang, and Joon Son Chung. Avcd: Mitigating hallucinations in audio-visual large language models through contrastive decoding. *Advances in Neural Information Processing Systems*, 38:63143–63174, 2026.

Ranjay Krishna, Kenji Hata, Frederic Ren, Li Fei-Fei, and Juan Carlos Niebles. Dense-captioning events in videos. In *Proceedings of the IEEE international conference on computer vision*, pp. 706–715, 2017.

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*, pp. 13872–13882, 2024.

Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Peiyuan Zhang, Yanwei Li, Ziwei Liu, et al. Llava-onevision: Easy visual task transfer. *arXiv preprint arXiv:2408.03326*, 2024a.

Chaoyu Li, Eun Woo Im, and Pooyan Fazli. Vidhalluc: Evaluating temporal hallucinations in multimodal large language models for video understanding. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 13723–13733, 2025.

Kunchang Li, Yali Wang, Yinan He, Yizhuo Li, Yi Wang, Yi Liu, Zun Wang, Jilan Xu, Guo Chen, Ping Luo, et al. Mvbench: A comprehensive multi-modal video understanding benchmark. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 22195–22206, 2024b.

Bin Lin, Yang Ye, Bin Zhu, Jiaxi Cui, Munan Ning, Peng Jin, and Li Yuan. Video-llava: Learning united visual representation by alignment before projection. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 5971–5984, 2024.

Shi Liu, Kecheng Zheng, and Wei Chen. Paying more attention to image: A training-free method for alleviating hallucination in lvlms. In *European Conference on Computer Vision*, pp. 125–140. Springer, 2024a.

Yuanxin Liu, Shicheng Li, Yi Liu, Yuxiang Wang, Shuhuai Ren, Lei Li, Sishuo Chen, Xu Sun, and Lu Hou. Tempcompass: Do video llms really understand videos? In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 8731–8772, 2024b.

Yu-Fei Ma, Lie Lu, Hong-Jiang Zhang, and Mingjing Li. A user attention model for video summarization. In *Proceedings of the tenth ACM international conference on Multimedia*, pp. 533–542, 2002.

Jonghwan Mun, Minchul Shin, Gunsoo Han, Sangho Lee, Seongsu Ha, Joonseok Lee, and Eun-Sol Kim. Bassl: Boundary-aware self-supervised learning for video scene segmentation. In *Proceedings of the Asian Conference on Computer Vision*, pp. 4027–4043, 2022.

Yeji Park, Deokyeong Lee, Junsuk Choe, and Buru Chang. Convis: Contrastive decoding with hallucination visualization for mitigating hallucinations in multimodal large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 6434–6442, 2025.

Daiqing Qi, Dongliang Guo, Hanzhang Yuan, Handong Zhao, Mengxuan Hu, Lehan Yang, and Sheng Li. Improve temporal reasoning in multimodal large language models via video contrastive decoding. *Advances in Neural Information Processing Systems*, 38:159835–159857, 2026.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PmLR, 2021.

Anyi Rao, Linning Xu, Yu Xiong, Guodong Xu, Qingqiu Huang, Bolei Zhou, and Dahua Lin. A local-to-global approach to multi-modal movie scene segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10146–10155, 2020.

Shitao Tang, Litong Feng, Zhanghui Kuang, Yimin Chen, and Wei Zhang. Fast video shot transition localization with deep structured models. In *Asian Conference on Computer Vision*, pp. 577–592. Springer, 2018.

Weihan Wang, Zehai He, Wenyi Hong, Yean Cheng, Xiaohan Zhang, Ji Qi, Ming Ding, Xiaotao Gu, Shiyu Huang, Bin Xu, et al. Lvbench: An extreme long video understanding benchmark. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 22958–22967, 2025.

Antoine Yang, Arsha Nagrani, Paul Hongsuck Seo, Antoine Miech, Jordi Pont-Tuset, Ivan Laptev, Josef Sivic, and Cordelia Schmid. Vid2seq: Large-scale pretraining of a visual language model for dense video captioning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10714–10726, 2023.

Qinghao Ye, Guohai Xu, Ming Yan, Haiyang Xu, Qi Qian, Ji Zhang, and Fei Huang. Hitea: Hierarchical temporal-aware video-language pre-training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 15405–15416, 2023.

Jiacheng Zhang, Yang Jiao, Shaoxiang Chen, Na Zhao, Zhiyu Tan, Hao Li, and Jingjing Chen. Eventhallusion: Diagnosing event hallucinations in video llms. *arXiv preprint arXiv:2409.16597*, 2024a.

Ke Zhang, Wei-Lun Chao, Fei Sha, and Kristen Grauman. Video summarization with long short-term memory. In *European conference on computer vision*, pp. 766–782. Springer, 2016.

Qingru Zhang, Chandan Singh, Liyuan Liu, Xiaodong Liu, Bin Yu, Jianfeng Gao, and Tuo Zhao. Tell your model where to attend: Post-hoc attention steering for llms. In *International Conference on Learning Representations*, volume 2024, pp. 42411–42430, 2024b.

Junjie Zhou, Yan Shu, Bo Zhao, Boya Wu, Zhengyang Liang, Shitao Xiao, Minghao Qin, Xi Yang, Yongping Xiong, Bo Zhang, et al. Mlvu: Benchmarking multi-task long video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13691–13701, 2025.

Model	Perturbation	VidHalluc (STH)		
		Hallu.↓	Omiss.↓	Overall↑
Qwen2.5-VL	Blank	**23.2**	6.3	61.0
	Remove	25.5	5.8	57.2
	Noise	31.2	**3.1**	**64.9**
LLaVA-OneVision	Blank	25.5	36.0	48.5
	Remove	29.8	46.1	44.6
	Noise	**22.2**	**35.8**	**50.7**

Table 5: Comparison of perturbation strategies on VidHalluc (STH). Noise perturbation achieves the best overall performance for both Video LLMs. Values are reported in %.

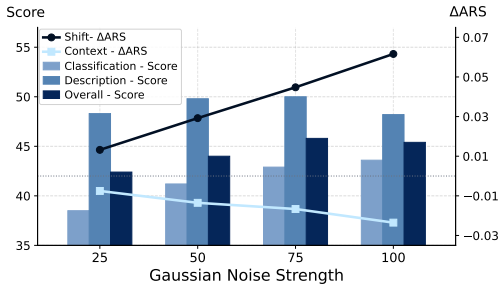

Figure 5: Influence of Gaussian perturbation strength on semantic-shift attention and scene-transition performance.

A Handling Multiple Semantic Shifts

A video may contain multiple detected semantic shifts with different amplification modes. For the input-level variant, we process each shift independently and combine the selected perturbation regions into a single mask.

Let $\mathcal{S} = \{s_n\}_{n=1}^{N}$ with $s_n = (i_n, i_n + 1)$. Frames belonging to detected shift pairs are first collected into

$$\mathcal{B} = \bigcup_{n=1}^{N} \{i_n, i_n + 1\}.$$

Each remaining frame is assigned to its nearest semantic shift, defining a shift-specific context region $\mathcal{C}_n$. When a frame is equidistant from two shifts, it is assigned to the temporally later one.

For an under-attended shift ($z_n = \text{UNDER}$), its two shift frames are added to the perturbation mask. For an over-attended shift ($z_n = \text{OVER}$), its associated context region is added instead. The final mask is formed by set union, so each frame is perturbed at most once.

Using the resulting mask $\mathcal{M}$, we construct $v' = [\tilde{I}_1, \ldots, \tilde{I}_K]$ by adding $\epsilon_j \sim \mathcal{N}(0, \sigma^2 I)$ to every frame $j \in \mathcal{M}$ while leaving all remaining frames unchanged. The resulting v' serves as the failure-amplified input.

Algorithm 1 Multiple-Shift Mask Construction

Require: $\mathcal{S} = \{s_n\}_{n=1}^{N}, \{z_n\}_{n=1}^{N}, K$
Ensure: Perturbation mask $\mathcal{M}$
$\mathcal{B} \leftarrow \bigcup_{n=1}^{N} \{i_n, i_n + 1\}$
for $n = 1, \ldots, N$ **do**
  $\mathcal{C}_n \leftarrow \emptyset$
end for
for $j = 1, \ldots, K$ **do**
  if $j \notin \mathcal{B}$ **then**
    $n^\star \leftarrow \arg\min_n \min(|j - i_n|, |j - (i_n + 1)|)$
    $\mathcal{C}_{n^\star} \leftarrow \mathcal{C}_{n^\star} \cup \{j\}$
  end if
end for
$\mathcal{M} \leftarrow \emptyset$
for $n = 1, \ldots, N$ **do**
  if $z_n = \text{UNDER}$ **then**
    $\mathcal{M} \leftarrow \mathcal{M} \cup \{i_n, i_n + 1\}$
  else
    $\mathcal{M} \leftarrow \mathcal{M} \cup \mathcal{C}_n$
  end if
end for
return $\mathcal{M}$

B Comparison of Input Perturbation Strategies

Why Gaussian noise? We further compare Gaussian noise with two alternative ways of degrading visual evidence: *blanking* the selected frames and *removing* them from the input sequence.

Blanking strongly suppresses the visual evidence contained in the selected frames and can therefore be viewed as a substantially more aggressive perturbation than adding noise. As observed in Figure 5, excessively strong perturbation can degrade descriptive generation even when some scene-transition decisions improve. This suggests that completely eliminating the visual content of selected frames may remove useful evidence in addition to amplifying the intended failure tendency.

Frame removal introduces a different issue. Removing selected frames changes the sampled temporal sequence itself, including its frame count and relative temporal arrangement. Consequently, the

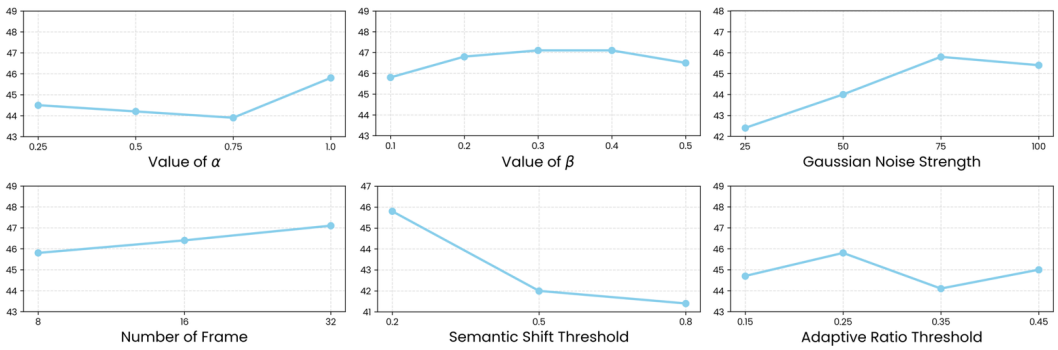

Figure 6: Sensitivity to Hyperparameter Settings on VidHalluc(STH).

resulting prediction may reflect not only the intended degradation of semantic-shift evidence but also changes in temporal alignment.

In contrast, Gaussian noise preserves both the number and temporal positions of the sampled frames while allowing continuous control over perturbation strength through σ. As shown in Table 5, Gaussian noise achieves the highest overall performance for both Video LLMs. We therefore use Gaussian noise as the default perturbation mechanism in input-level SSACD.

How input perturbation modulates attention. Figure 5 examines how perturbation strength affects semantic-shift attention. As Gaussian noise increases, the shift-region attention ratio deviates further from the clean branch, showing that input perturbation can indirectly modulate attention to semantic-shift evidence. This links the direct attention intervention in our motivating analysis to the input-level amplification used in SSACD.

As the amplified signal becomes stronger, both classification and description performance initially improve. However, description accuracy declines under excessive perturbation, indicating that overly strong noise can degrade useful visual evidence. These results highlight a trade-off: effective failure amplification should sufficiently expose failure-prone behavior while retaining enough visual information to form an informative auxiliary signal.

C Adaptive Plausibility Constraint

SSACD combines predictions from the clean and failure-amplified branches through contrastive decoding. However, the failure-amplified branch is not assumed to be uniformly incorrect. Even after semantic-shift-aware failure amplification, it may preserve visual evidence that remains useful for generation. Applying the contrastive adjustment indiscriminately to the entire vocabulary can therefore over-penalize plausible predictions.

To reduce this effect, SSACD follows prior contrastive decoding methods (Leng et al., 2024; Park et al., 2025) and restricts contrastive adjustment to tokens sufficiently supported by the clean branch. As defined in Section 4.4, a token is considered plausible when its clean-branch logit lies within the threshold controlled by β relative to the maximum clean logit. Tokens outside this plausibility set are masked during contrastive decoding.

This constraint ensures that the failure-amplified branch is used primarily to distinguish among predictions that remain plausible under the original video evidence. It therefore limits excessive contrastive suppression while preserving the failure-amplification signal used by SSACD.

D Hyperparameter Sensitivity

Figure 6 examines the sensitivity of the default input-level SSACD to the contrastive strength α, plausibility threshold β, perturbation strength σ, number of sampled frames, semantic-shift detection threshold τ, and failure-selection threshold γ.

Model	Method	VidHalluc (STH)		
		Class.↑	Desc.↑	Overall↑
Qwen2.5-VL (72B)	Base	52.2	50.5	51.5
	DINO-HEAL	57.0	49.8	54.1
	VCD	54.3	51.7	53.3
	SSACD	**57.5**	**52.3**	**55.4**
LLaVA-OneVision (72B)	Base	48.6	41.3	45.7
	DINO-HEAL	50.2	42.1	47.0
	VCD	54.6	48.4	52.1
	SSACD	**58.4**	**49.3**	**54.8**

Table 6: Performance comparison on VidHalluc (STH) for 72B-scale Video LLMs. Values are reported in %.

Scaling to larger Video LLMs. We further evaluate SSACD on the 72B-scale variants of Qwen2.5-VL and LLaVA-OneVision to examine whether its effectiveness persists as model capacity increases.
As shown in Table 6, SSACD consistently improves VidHalluc (STH) performance over the corresponding base models across both model families. It also achieves stronger overall performance than the evaluated training-free decoding baselines.
These results indicate that the benefit of semantic-shift-aware failure amplification is not limited to 7B-scale Video LLMs and persists at a substantially larger model scale.

SSACD remains effective across a range of settings, while excessively weak or strong failure amplification reduces performance. When perturbation is too weak, the failure-amplified branch remains close to the clean branch and provides limited contrastive guidance. Conversely, overly strong perturbation can remove useful visual evidence and make the auxiliary predictions less informative. These results indicate that effective failure amplification requires a balance between strengthening the relevant failure tendency and preserving sufficient semantic information for meaningful contrastive decoding.

E Scaling to Larger Video LLMs

We further evaluate SSACD on 72B-scale variants of Qwen2.5-VL and LLaVA-OneVision to examine whether its effectiveness persists as model capacity increases.

F Efficiency and Implementation Details

SSACD is a training-free decoding-time method based on contrastive decoding. For the default input-level variant, SSACD first performs a preliminary clean decoding pass to estimate shift-specific attention statistics and determine the failure-prone tendency associated with each detected semantic shift. During the subsequent contrastive decoding stage, each generation step evaluates a clean branch conditioned on the original video and a failure-amplified branch conditioned on the semantic-shift-aware perturbed video.

As a result, SSACD introduces additional inference cost compared with vanilla decoding, as is common in contrastive decoding methods. This overhead reflects a trade-off between decoding efficiency and improved scene-transition understanding. Importantly, SSACD requires no additional training or parameter updates and uses the same underlying Video LLM for both branches.

F.1 Parallel Branch Evaluation

The clean and failure-amplified branches are independent before their logits are combined. As also discussed in prior video contrastive decoding work (Qi et al., 2026), this structure allows the two branch evaluations to be scheduled concurrently when sufficient GPU resources are available.

A sequential implementation can be written as:

```
outputs_clean = self.forward(...)  # clean branch
outputs_fail  = self.forward(...)  # failure-amplified branch

# combine branch logits
```

Alternatively, the two forward passes can be submitted to separate CUDA streams:

```
stream_clean = torch.cuda.Stream()
stream_fail  = torch.cuda.Stream()
```

```
clean_holder = {}
fail_holder = {}

with torch.cuda.stream(stream_clean):
    clean_holder["out"] = self.forward(...)

with torch.cuda.stream(stream_fail):
    fail_holder["out"] = self.forward(...)

torch.cuda.synchronize()

outputs_clean = clean_holder["out"]
outputs_fail  = fail_holder["out"]

# combine branch logits
```

After synchronization, the branch logits are combined using the same contrastive objective as in the sequential implementation. Thus, parallel scheduling does not change the SSACD decoding objective and only changes how the two independent branch evaluations are executed. When the hardware supports effective concurrent execution, this can reduce wall-clock latency, although it may increase peak GPU memory usage because both branches can be resident simultaneously.

G Additional Qualitative Results

We provide additional qualitative examples on VidHalluc (STH) to illustrate the behavior of SSACD under different scene-transition cases. Figure 7 presents additional scene-description examples, while Figures 8 and 9 show hallucination and omission cases, respectively. These examples complement the quantitative results in the main paper by illustrating how SSACD suppresses predictions associated with failure-prone semantic-shift behavior.

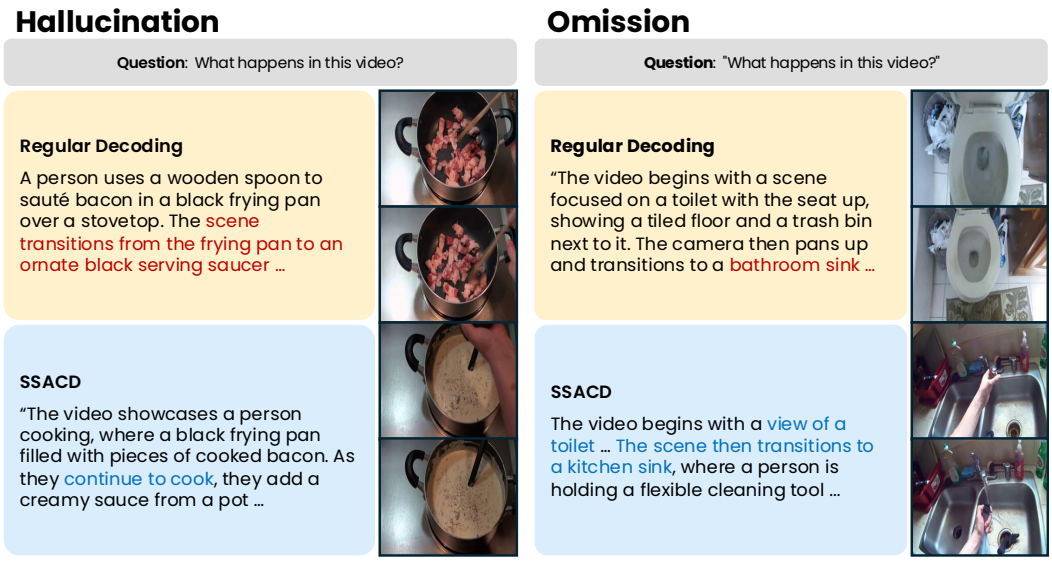

Figure 7: Qualitative captioning examples on the VidHalluc (STH) dataset. The example compares regular decoding and SSACD under the same video input, illustrating how SSACD produces a more accurate scene-transition-aware description.

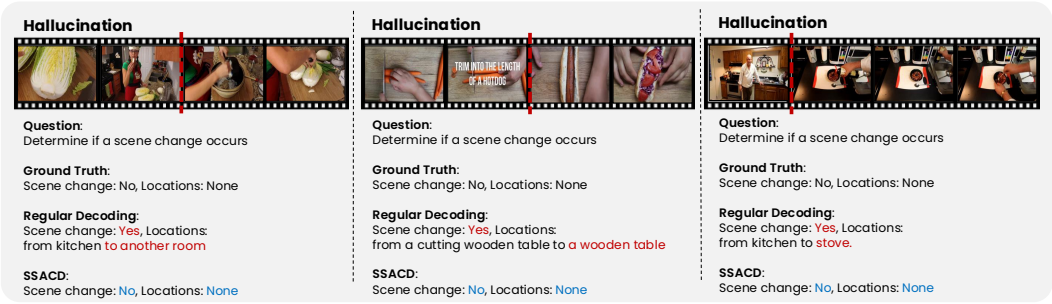

Figure 8: Qualitative hallucination examples on the VidHalluc (STH) dataset. Regular decoding predicts a scene transition even though the scene remains continuous, while SSACD suppresses the hallucinated transition by reducing reliance on misleading semantic shift evidence.

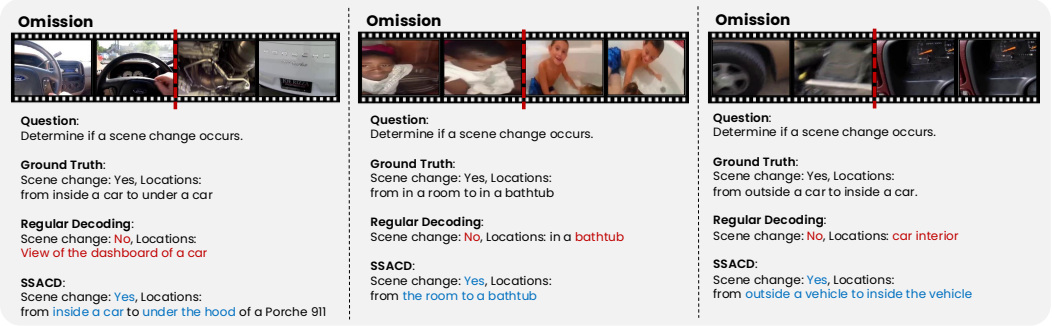

Figure 9: Qualitative omission examples on the VidHalluc (STH) dataset. Regular decoding misses an actual scene transition and maintains the previous scene context, whereas SSACD recovers the transition by contrasting the clean branch with a semantic-shift-aware perturbed branch.