ACTIV: REDUCING SPATIOTEMPORAL HALLUCINATIONS IN VIDEO LANGUAGE MODELS THROUGH TOKEN INTERVENTIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

Video language models have gained adoption in applications such as robotics and autonomous driving, where reliability and low inference latency are essential. Yet, these models still misreport where objects are and the order in which events occur. Contrastive decoding can reduce these errors without retraining by comparing predictions from original and altered visual inputs. However, constructing negatives by re-encoding altered videos increases inference latency. We target a useful accuracy–latency tradeoff by constructing spatial and temporal negatives directly from already-computed visual tokens. We introduce Adaptive Contrast with Token Interventions for Video (ACTIV), a training-free framework that constructs spatial and temporal views with only a single video encoding pass. ACTIV averages visual features across locations or time to bottleneck the corresponding information. During answer generation, ACTIV contrasts predictions from these alternatives with those from the original features, weighting each comparison by its effect on the prediction to favor words supported more strongly by the original video. Across three models and three hallucination benchmarks, ACTIV outperforms the evaluated training-free baselines in average score for each model; on LLaVA-Video-7B, it improves VidHalluc and EventHallusion scores by 3.87 and 5.38 points over the unmodified model, respectively, while reducing estimated model-inference latency by approximately **22–26**% relative to the state-of-the-art.

1 INTRODUCTION

Video large language models (Video-LLMs) have progressed from short-clip question answering toward reasoning over longer videos (Zohar et al., 2025), and are increasingly relevant to robots and other agents that must keep track of a changing scene (Fan et al., 2025). Reliability in these settings requires more than recognizing the entities and actions in a clip: a model must also determine *where* they appear and *when* they occur. Current models can fail along either axis. For example, a model may identify the correct object but incorrectly describe its location, or recognize two actions but reverse their order. These are ***spatial*** and ***temporal*** hallucinations in Video-LLMs: the answer remains linguistically plausible but asserts a spatial configuration or event sequence that the video does not support (Li et al., 2025a; Bae et al., 2025). In an embodied system, either error can change which *object* an agent approaches or which *action* it executes next. These applications also require low inference latency, making the cost of correcting such errors important.

Efforts to improve video understanding span input selection, model design and training, and test-time computation. Frame selection supplies informative evidence (Hu et al., 2025), while architectural changes and training strengthen spatial and temporal representations (Bae et al., 2025). We pursue *training-free test-time correction*, explored in recent work (Zhang et al., 2024a; Wu et al., 2026), because it can complement both input and model improvements. Specifically, contrastive decoding compares predictions from the original input with those from a weakened alternative, called a *negative*, to favor words supported more strongly by the original evidence (Leng et al., 2024).

However, avoiding retraining does not eliminate the cost of correction. Creating negatives through additional vision-encoder passes repeats the work of processing image patches across frames. Vision encoding can already be a major latency bottleneck when processing high-resolution images or long

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.

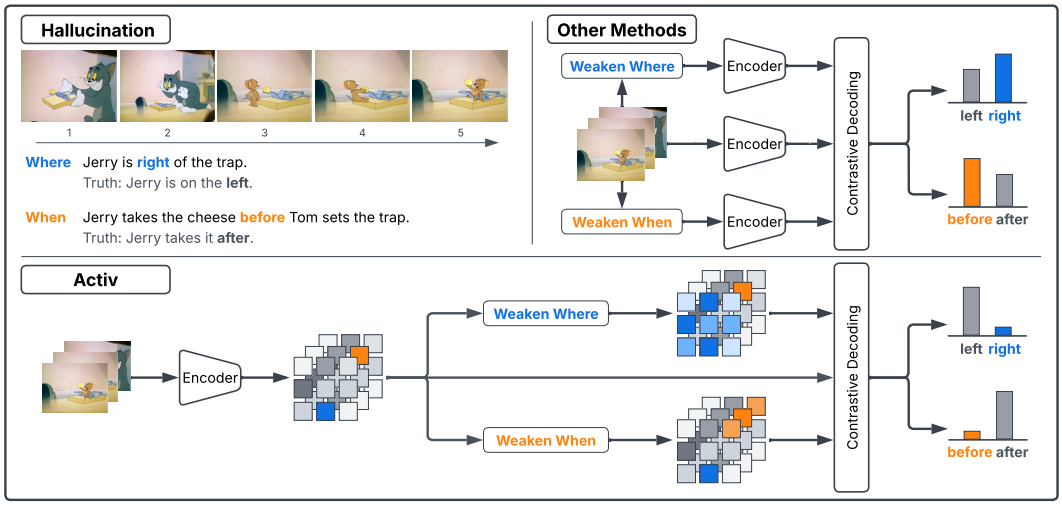

Figure 1: **ACTIV: correcting hallucinations with one video encoding.** Video language models can recognize objects and actions yet misreport their locations or order (top left). Methods that alter the video to weaken these cues require additional vision-encoder passes (top right). ACTIV instead averages the encoded tokens across locations or time to weaken spatial or temporal information directly (bottom). Comparing predictions before and after these changes helps the model favor answers supported by the video. The example frames are from Siddharth M (2025).

videos (Liu et al., 2025; Shi et al., 2026). This motivates our central intuition: *reuse the encoded video and change the evidence presented to the language model.* The language model receives visual tokens organized by location and time. Modifying these features while keeping the question and their positions fixed lets us examine how predictions respond to changed evidence without another vision-encoder pass.

Reusing a single video encoding directs our intervention to the visual tokens supplied to the language model. We ask: *does replacing these tokens weaken the model's support for its answer?* To measure this effect, we introduce the *Video Evidence Margin (VEM)*, which compares the same answer's likelihood under the original tokens and random replacements (Section 2). We find that replacing these tokens can leave an answer equally or more probable. This motivates correction that favors predictions supported more strongly by the original features, using targeted token changes to weaken spatial or temporal information.

Our intuition is that predictions based on information about location or event order should become less plausible when that information is weakened. We therefore seek to use the video's encoded features both as evidence for answering a question and as guidance for correcting the response. We develop this idea into Adaptive Contrast with Token Interventions for Video (ACTIV), a training-free framework that uses token interventions to improve spatial and temporal grounding while reusing a single video encoding (Section 3). Averaged over VidHalluc (Li et al., 2025a), VideoHallucer (Wang et al., 2024), and EventHallusion (Zhang et al., 2024a), ACTIV outperforms the evaluated competing training-free methods on Qwen2.5-VL-7B (Bai et al., 2025), LLaVA-OneVision-7B (Li et al., 2024), and LLaVA-Video-7B (Zhang et al., 2024b). It also reduces estimated model-inference latency by approximately 22–26% on LLaVA-Video-7B relative to our implementation of a state-of-the-art training-free method (Wu et al., 2026).

Our main contribution, ACTIV, uses spatial and temporal averaging of already-computed visual tokens to construct contrastive negatives. This enables both forms of correction with one video encoding and yields a useful accuracy–latency tradeoff (Section 3). We also introduce VEM, a new metric addressing a question accuracy alone cannot answer: *how strongly do the video's encoded features support the model's answer?* (Section 2). Our evaluation examines when correction helps, whether its benefits transfer, and what it costs at inference, with ablations testing the contribution of the token interventions and adaptive correction (Section 4).

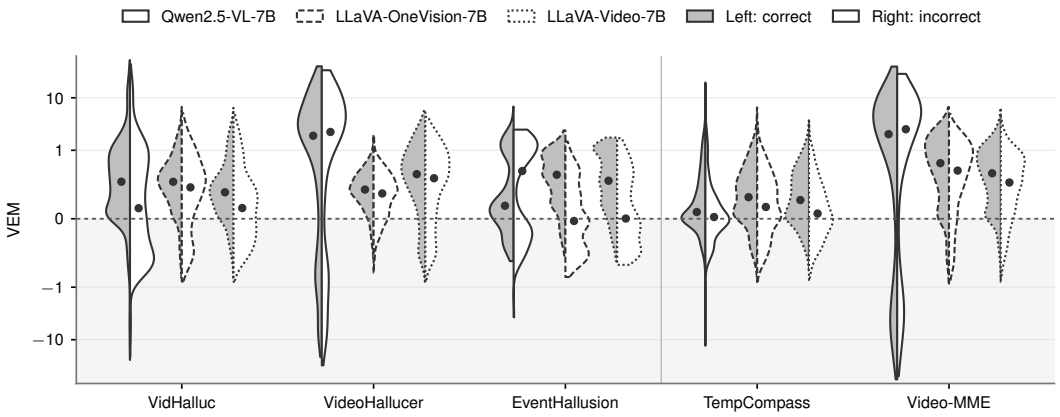

Figure 2: **Even correct answers can score higher with random visual features.** The model generates an answer from the real video. The Video Evidence Margin (VEM) compares that same answer's score with real features against its average score with random replacements. Above zero, real features score the answer higher; in the shaded region, random features score it equally or higher. This occurs across five benchmarks, motivating ACTIV's goal of favoring predictions supported by the video's spatial and temporal information. Markers denote medians.

2 MEASURING WHETHER THE VIDEO SUPPORTS THE ANSWER

Video-LLMs are usually judged by answer correctness, which does not tell us whether the video supported the answer. A correct answer may follow useful visual evidence or already be plausible from the question and learned language priors. Confidence does not separate these cases: it only measures how strongly the model favors the answer with the video present. We therefore ask: *does the real video make the generated answer more probable?*

2.1 VIDEO EVIDENCE MARGIN

We introduce the *Video Evidence Margin (VEM)* to answer this question for an individual prediction. The model first generates an answer y from the real video. We then score that same answer using the real visual tokens Z and $R = 3$ independent random replacements $\widetilde{Z}_r$. Each random replacement uses the original features' mean and standard deviation in each feature dimension, keeping the values on a comparable numerical scale. The index r identifies the replacement; averaging their scores reduces variation from individual random draws. The prompt, answer, token positions, and model parameters remain unchanged.

Scoring with only the question measures support without visual input. Our comparison keeps visual tokens present with the same layout and comparable numerical scale, testing whether real feature values support the answer more strongly than random replacements. Let $S(y; U)$ denote the model's length-normalized log-probability for answer y under visual representation U. We define

$$\mathrm{VEM}(x, q, y) = S(y; Z) - \frac{1}{R}\sum_{r=1}^{R} S(y; \widetilde{Z}_r). \tag{1}$$

A positive VEM means that the real features give the answer a higher score. At zero, the two scores are equal; below zero, the random replacements give the higher score. For example, if the model assigns the fixed answer probability 0.4 with the real features and 0.5 with each replacement, VEM is negative: the model scores the answer higher with random features. VEM therefore audits the visual support behind a fixed answer rather than generating a second answer or requiring a correctness label.

Attention maps describe how weights are distributed across tokens; VEM measures how changing visual features affects the answer's score.

2.2 Some Correct Answers Remain Probable After the Video Is Replaced

We measure VEM for the unmodified predictions of three 7B Video-LLMs across five benchmarks including both hallucinations and video understanding. Figure 2 compares correct and incorrect answers on the same scale. Correct answers let us ask whether getting the answer right also means the real video increased its score; incorrect answers show how that pattern differs among mistakes. Each percentage uses only the answers in its group as the denominator. Markers indicate medians; the vertical scale is nonlinear.

Across these model–benchmark comparisons, between 8.6% and 33.3% of correct answers remain equally or more probable after replacement. The effect is not limited to hallucination benchmarks. It reaches 33.3% for Qwen2.5-VL on TempCompass (Liu et al., 2024) and 24.3% on Video-MME (Fu et al., 2025), while reaching 29.9% on EventHallusion. For LLaVA-Video, it reaches 24.4% on TempCompass.

Nonpositive VEM is more common among incorrect answers in all five benchmark groups. On EventHallusion, its share rises from 19% of correct answers to 48% of incorrect answers, averaging equally across the three models.

These findings motivate a practical question: *can we favor predictions that receive stronger support from the video's spatial and temporal information?* VEM examines a completed answer using random replacements. The next section introduces ACTIV, which compares predictions at each generation step using more targeted alternatives. These reduce feature differences across space or time to weaken information about *where* objects appear or *when* events occur. Contrasting their predictions with those from the original features favors tokens with stronger support from the original representation. Both alternatives reuse a single video encoding.

3 ACTIV: Contrastive Decoding from a Single Video Encoding

To answer a question about a video reliably, a model must connect its words to where things are and how events unfold. ACTIV guides this connection by comparing predictions before and after these clues are weakened. From a single encoding, we construct two versions of the video's numerical representation that reduce differences between locations or moments (Figure 3). During generation, contrastive decoding (Leng et al., 2024; Wu et al., 2026) favors words supported more strongly by the original features. ACTIV gives more weight to the comparison that changes predictions more, adapting its correction without retraining the model.

3.1 Spatial and Temporal Token Negatives

For Video-LLMs with accessible visual tokens indexed by time and space, let $Z = E(x) \in \mathbb{R}^{T\times S\times d}$ denote the visual tokens after the model's native encoding, projection, and pooling or merging. Here T counts encoded video time steps and S spatial positions per step. A time step t can represent one frame or adjacent frames encoded together; $o_{t,s} \in \mathbb{R}^d$ is the original feature at time t and position s. We construct both negatives directly from Z, leaving the question, token layout, and positional indices unchanged.

We use o for an original feature, n for a negative feature, and a for an average feature. Let a_t^{S} be the average of all spatial tokens at time t. The spatial negative moves each token toward this average:

$$n_{t,s}^{\mathrm{S}} = (1-\beta_{\mathrm{S}})o_{t,s} + \beta_{\mathrm{S}} a_t^{\mathrm{S}}. \tag{2}$$

This makes locations more alike without mixing different times. The temporal negative applies the same idea across moments instead. Let a_s^{T} be the average feature at spatial position s over all T time steps. Moving each token toward this average gives

$$n_{t,s}^{\mathrm{T}} = (1-\beta_{\mathrm{T}})o_{t,s} + \beta_{\mathrm{T}} a_s^{\mathrm{T}}. \tag{3}$$

The two operations therefore reduce differences across space or time, respectively. Their strengths $\beta_{\mathrm{S}}, \beta_{\mathrm{T}} \in [0,1]$ range from unchanged tokens at zero to repeated means at one. Each negative is built independently from Z; the figure illustrates both at strength 0.8.

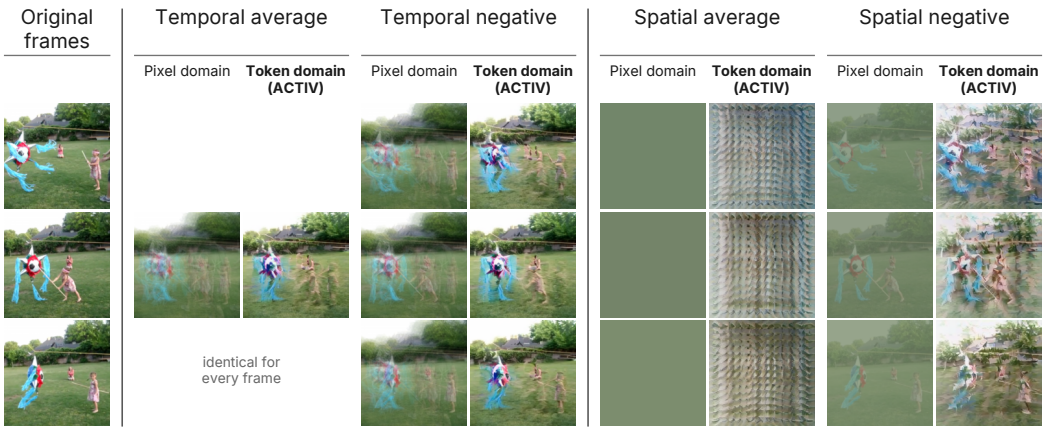

Figure 3: **Why construct negatives in token space?** Negatives blend the original representation with an average across frames or spatial locations. In the pixel-space negatives shown here, the people and hanging object remain recognizable, and changes in posture can still be followed. Temporal token averaging mixes features from different moments, while the spatial token-space reconstructions visibly distort body shapes and scene layout. ACTIV compares predictions from these weakened features with those from the original video to favor words supported by spatial and temporal cues, without re-encoding the video. Token-space panels are decoded illustrations using a separate SigLIP2 representation autoencoder (Tschannen et al., 2025; Zheng et al., 2026).

Averaging retains the mean feature and scales deviations from it. Omitting the spatial and temporal indices, either construction becomes

$$n = a + (1 - \beta)(o - a), \qquad \|n - a\|_2^2 = (1 - \beta)^2 \|o - a\|_2^2. \tag{4}$$

The mean is unchanged because deviations sum to zero; $\beta = 0.8$ leaves 20% of each deviation. This is an exact reduction in feature variation, intended to weaken spatial or temporal distinctions without assuming that every trace of a relation disappears. Both negatives cost $O(TSd)$ to construct and require no further vision-encoder pass.

3.2 Letting Prediction Changes Guide the Contrast

At generation step k, three decoder branches share the question q and the answer tokens already generated, $y_{<k}$, but receive original, spatial-negative, or temporal-negative features. Their next-token scores (logits) are ℓ_k^0, ℓ_k^{S}, and ℓ_k^{T}; softmax gives the corresponding probabilities p_k^0, p_k^{S}, and p_k^{T}.

The two negatives need not affect every prediction equally. We therefore give more weight to the one that changes the next-token distribution more, using Jensen–Shannon divergence (JSD) (Lin, 1991) to measure that difference. For branch $b \in \{\mathrm{S}, \mathrm{T}\}$, the divergence and normalized weight are

$$D_k^b = \mathrm{JSD}(p_k^0 \| p_k^b), \qquad w_k^b = \frac{D_k^b}{D_k^{\mathrm{S}} + D_k^{\mathrm{T}}}. \tag{5}$$

If both divergences are zero, both weights are $1/2$. The weights use existing branch predictions, requiring no attention maps, and combine the spatial and temporal contrasts to adjust the original scores:

$$\widetilde{\ell}_k = \ell_k^0 + \alpha \Big[w_k^{\mathrm{S}} (\ell_k^0 - \ell_k^{\mathrm{S}}) + w_k^{\mathrm{T}} (\ell_k^0 - \ell_k^{\mathrm{T}}) \Big], \tag{6}$$

Here $\alpha \geq 0$ controls correction strength. This rule favors candidates supported more strongly by the original features relative to the negatives. For example, if temporal averaging reduces the preference for "before" over "after," the temporal contrast reinforces that preference.

To avoid promoting originally implausible tokens, we retain only candidates v satisfying $p_k^0(v) \geq \lambda \max_u p_k^0(u)$, with $\lambda = 0.1$, following Leng et al. (2024). This requires at least one tenth of the highest original next-token probability. After applying the checkpoint's repetition penalty, we select the highest-scoring remaining token and pass it to all three branches. The branches run together with

separate decoder caches and reuse the same video encoding. Standard ACTIV therefore adds token transformations and decoder computation, but requires neither baseline-answer generation nor VEM scoring.

An optional variant, ACTIV+VEM, uses the answer-level evidence signal from Section 2 to control correction. It scores the unmodified model's answer y^0, replaces the per-step weights with equal spatial and temporal weights, and sets the strength to

$$\alpha_{\text{eff}} = \alpha \ \text{clip}\left(1 - \frac{\text{VEM}(x, q, y^0)}{v_0}, 0, 1\right), \tag{7}$$

Clipping restricts the multiplier to $[0, 1]$; $v_0 = 1$ in natural-log units per answer token. Nonpositive VEM applies full strength; positive VEM reduces it to zero at v_0. We then generate the corrected answer using Eq. 6 with α_{eff}. This variant also reuses the encoding, but adds baseline generation and VEM scoring.

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

Models, benchmarks, and comparisons. We evaluate Qwen2.5-VL-7B, LLaVA-OneVision-7B, and LLaVA-Video-7B against their unmodified counterparts, TCD (Zhang et al., 2024a), DINO-HEAL (Li et al., 2025a), SEASON, and released ArrowRL (Xue et al., 2025), TPO (Li et al., 2025b), and RRPO (Sarkar & Etemad, 2025) checkpoints. TCD follows its paper; we reimplement SEASON from Wu et al. (2026) because code is unreleased. DINO-HEAL uses the published equations rather than the differing released code, with reweighting after Qwen2.5-VL's patch merger and before LLaVA's projector. Appendix A.1 details these implementations and compares DINO-HEAL's equation-based and released-code rules. Hallucination evaluations follow benchmark prompts and scoring rules, with a shared eight-frame cap and sampling and decoding settings held fixed across methods. Scores cover all tasks: VidHalluc averages four task scores, and VideoHallucer (Wang et al., 2024) five subsets requiring both answers in each pair to be correct. EventHallusion reports accuracy over all questions. Overall averages the three benchmark scores.

Configurations. For the models above, ACTIV uses $(\beta_{\text{T}}, \beta_{\text{S}}, \alpha)$ values $(0.5, 0.5, 0.25)$, $(1, 0.25, 0.5)$, and $(0.75, 0.5, 1)$, respectively. ACTIV+VEM retains these negative strengths with base $\alpha = 0.25, 1, 2$; VEM uses three reproducible random replacements per item and a cutoff $v_0 = 1$ fixed before evaluation. For each model, we select $(\beta_{\text{T}}, \beta_{\text{S}}, \alpha)$ and baseline strengths by grid search on stratified 5% validation splits of each hallucination benchmark, maximizing the mean benchmark score. All methods are evaluated on the remaining 95%; selected configurations transfer unchanged to video-understanding benchmarks. We assess significance using 95% paired bootstrap confidence intervals from 10,000 resamples over evaluation videos or question pairs; intervals above zero indicate significant gains.

4.2 HALLUCINATION MITIGATION

ACTIV leads the competing training-free methods on average across backbones, but gains over the unmodified model depend on the backbone (Table 1). Both LLaVA models improve over the baseline and competing corrections on every benchmark; LLaVA-Video also surpasses TPO and RRPO without training. On Qwen2.5-VL, ACTIV's average score drops by 0.12 points from baseline, compared with 0.95–2.85 points for TCD, DINO-HEAL, and SEASON. VEM guidance recovers this loss without a significant gain over baseline and leads the training-free averages on Qwen and LLaVA-OneVision. ACTIV's overall gains over baseline are significant on LLaVA-OneVision (4.43 points, 95% CI: $[3.26, 5.60]$) and LLaVA-Video (3.11 points, 95% CI: $[1.86, 4.46]$).

We also test whether the gains reflect a shift in answer bias. On EventHallusion, matching the "yes" rate alone does not reproduce them. ACTIV raises LLaVA-Video's "yes" share from 42.3% to 50.6%, but matching this share by flipping the baseline's least-confident "no" answers yields 66.75, below the baseline's 67.24 and 5.87 points below ACTIV. The same control on LLaVA-OneVision scores 62.59 against ACTIV's 70.90.

Table 1: **Evaluations on three hallucination benchmarks** across three video language model backbones. Darker and lighter purple mark the best and second-best averages in each baseline and training-free block. † marks significant Overall gains over the unmodified model.

			VidHalluc					VideoHallucer						EventHallusion				Overall
Model	Method	Training-free	BQA	MCQ	TSH	STH	Avg.	ORH	Temp.	Sem.	Fact	Non-fact	Avg.	Entire	Mislead.	Mix	Avg.	Avg.
Qwen2.5-VL-7B	Baseline	–	84.80	89.87	84.04	76.59	83.82	62.63	48.86	71.05	26.32	56.84	53.14	62.28	88.24	58.55	66.99	67.98
	TCD	✓	84.55	88.14	82.98	72.71	82.10	62.63	47.16	72.11	27.37	50.53	51.96	61.40	84.31	60.10	66.50	66.85
	DINO-HEAL	✓	83.94	89.83	85.26	78.18	84.30	62.11	43.75	71.58	25.79	57.89	52.22	65.79	90.20	50.26	64.55	67.03
	SEASON	✓	85.54	85.90	69.65	54.73	73.95	61.58	40.91	67.89	28.42	56.32	51.02	57.02	83.33	71.50	70.42	65.13
	ACTIV	✓	84.80	89.31	84.74	73.58	83.11	62.11	50.57	72.63	25.26	56.84	53.48	62.28	86.27	59.59	66.99	67.86
	ACTIV+VEM	✓	84.83	89.59	85.09	75.17	83.67	62.63	49.43	71.58	26.32	57.37	53.46	62.28	87.25	59.59	67.24	68.12
	ArrowRL	✗	85.38	86.32	80.70	78.99	82.85	62.63	52.27	68.95	30.00	61.05	54.98	64.04	92.16	66.84	72.37	70.07
LLaVA-OneVision-7B	Baseline	–	88.29	89.85	88.95	59.53	81.65	50.00	43.75	44.74	15.79	50.00	40.86	49.12	90.20	53.89	61.61	61.37
	TCD	✓	87.67	88.87	87.19	59.70	80.86	48.42	46.59	45.26	15.26	48.42	40.79	49.12	89.22	56.99	62.84	61.50
	DINO-HEAL	✓	78.55	83.68	52.28	15.00	57.38	60.53	28.98	56.32	11.05	46.84	40.74	63.16	79.41	25.91	49.63	49.25
	SEASON	✓	86.47	88.33	82.46	65.24	80.62	46.84	46.02	42.63	15.26	44.74	39.10	48.25	86.27	72.54	69.19	62.97
	ACTIV	✓	87.42	89.24	88.25	75.94	85.21	49.47	44.32	45.26	15.26	52.11	41.29	49.12	89.22	74.09	70.90	65.80 †
	ACTIV+VEM	✓	87.51	89.45	89.65	76.33	85.73	50.53	44.89	46.32	15.79	53.16	42.14	49.12	89.22	73.06	70.42	66.10 †
LLaVA-Video-7B	Baseline	–	87.29	89.50	84.74	51.58	78.28	65.26	56.82	63.16	12.11	52.63	50.00	57.02	93.14	59.59	67.24	65.17
	TCD	✓	86.77	89.15	84.91	58.09	79.73	64.74	56.82	62.63	11.05	52.11	49.47	58.77	92.16	64.25	69.68	66.29
	DINO-HEAL	✓	78.53	83.14	71.75	15.00	62.11	63.68	59.09	61.58	13.68	37.37	47.08	50.00	83.33	35.23	51.34	53.51
	SEASON	✓	86.95	88.47	82.81	58.61	79.21	62.11	53.41	57.37	15.26	46.84	47.00	53.51	87.25	73.06	71.15	65.79
	ACTIV	✓	87.03	89.10	86.32	66.14	82.15	67.37	55.11	60.53	13.68	53.68	50.07	55.26	87.25	75.13	72.62	68.28 †
	ACTIV+VEM	✓	86.91	89.43	87.19	67.54	82.77	66.32	54.55	58.95	12.63	50.53	48.59	53.51	87.25	76.68	72.86	68.07 †
	TPO	✗	87.32	89.55	85.26	53.42	78.89	63.16	57.39	64.21	13.68	53.16	50.32	55.26	93.14	60.62	67.24	65.48
	RRPO	✗	82.80	89.94	86.67	48.69	77.02	63.68	59.09	68.42	15.79	51.05	51.61	70.18	96.08	56.99	70.42	66.35

4.3 INFERENCE EFFICIENCY

ACTIV uses one video encoding, compared with two for TCD and three for SEASON; DINO-HEAL additionally runs a DINOv2 encoder (Oquab et al., 2023). ACTIV achieves higher accuracy at lower estimated latency than our SEASON implementation on LLaVA-Video-7B (Table 2). Although TCD and DINO-HEAL are faster than ACTIV, they achieve lower average scores. Latency estimates use the median of per-sample sums of recorded visual processing, prefill, and decoding times, excluding video loading and input preprocessing. Appendix A.3 separates visual-processing and decoder costs at 16 and 32 frames.

Table 2: **Accuracy and estimated inference latency on LLaVA-Video-7B** using one RTX A6000. Settings follow Sec. 4.1. Relative latency uses paired baselines; Avg. denotes the benchmark score.

			VidHalluc			VideoHallucer			EventHallusion		
Method	Encoder calls	Decoder branches	Latency (s)	Relative	Avg.	Latency (s)	Relative	Avg.	Latency (s)	Relative	Avg.
Baseline	1	1	0.41	1.00×	78.28	0.47	1.00×	50.00	0.50	1.00×	67.24
TCD	2	2	0.66	1.63×	79.73	0.74	1.59×	49.47	0.78	1.60×	69.68
DINO-HEAL	1 + DINOv2	1	0.50	1.22×	62.11	0.57	1.22×	47.08	0.58	1.17×	51.34
SEASON	3	3	1.11	2.69×	79.21	1.35	2.87×	47.00	1.35	2.73×	71.15
ACTIV	1	3	0.82	2.02×	82.15	1.04	2.20×	50.07	1.06	2.13×	72.62

4.4 TRANSFER TO VIDEO UNDERSTANDING

Without retuning, ACTIV improves TempCompass by 0.43, 0.97, and 1.74 points on Qwen2.5-VL, LLaVA-OneVision, and LLaVA-Video, respectively. Across the five benchmarks in Table 3, ACTIV or its VEM variant achieves the highest training-free average on all three models. TempCompass uses three rule-scored formats, excluding caption generation because it requires an LLM judge. All methods use eight frames for TempCompass and thirty-two for Video-MME without subtitles.

4.5 ABLATION STUDIES

Spatial and temporal negative construction. Table 4 compares isolated negatives on LLaVA-OneVision-7B. ACTIV's temporal negative leads VideoHallucer and EventHallusion by 0.27 and 1.72 points over the strongest alternatives, respectively; its spatial negative narrowly leads VideoHallucer. Each negative is contrasted with the original video using the same decoder and $\lambda = 0.1$. ACTIV retains the main negative strengths, with $\alpha = 0.25$ for spatial and $\alpha = 1$ for temporal contrasts. Flip, shuffle, and reversal operate on tokens; SEASON uses re-encoded noisy frames for its spatial negative and homogenization ($\beta = 0.33$) for its temporal negative.

Table 3: **Comparison across hallucination and video-understanding benchmarks on three backbones**, with Overall averaging the five displayed benchmarks. Darker and lighter purple mark the best and second-best averages in each baseline and training-free block.

			Hallucination				Video understanding			Overall
Model	Method	Training-free	VidHalluc	VideoHallucer	EventHallusion	Avg.	TempCompass	Video-MME	Avg.	Avg.
Qwen2.5-VL-7B	Baseline	–	83.82	53.14	66.99	67.98	72.82	61.74	67.28	67.70
	TCD	✓	82.10	51.96	66.50	66.85	72.33	60.89	66.61	66.76
	DINO-HEAL	✓	84.30	52.22	64.55	67.03	72.65	61.15	66.90	66.97
	SEASON	✓	73.95	51.02	70.42	65.13	71.68	60.91	66.30	65.60
	ACTIV	✓	83.11	53.48	66.99	67.86	73.25	61.26	67.26	67.62
	ACTIV+VEM	✓	83.67	53.46	67.24	68.12	73.29	61.37	67.33	67.81
	ArrowRL	✗	82.85	54.98	72.37	70.07	72.23	61.33	66.78	68.75
LLaVA-OneVision-7B	Baseline	–	81.65	40.86	61.61	61.37	68.99	59.22	64.11	62.47
	TCD	✓	80.86	40.79	62.84	61.50	69.02	59.07	64.05	62.52
	DINO-HEAL	✓	57.38	40.74	49.63	49.25	60.98	50.89	55.94	51.92
	SEASON	✓	80.62	39.10	69.19	62.97	69.78	58.50	64.14	63.44
	ACTIV	✓	85.21	41.29	70.90	65.80	69.96	58.93	64.45	65.26
	ACTIV+VEM	✓	85.73	42.14	70.42	66.10	69.85	59.15	64.50	65.46
LLaVA-Video-7B	Baseline	–	78.28	50.00	67.24	65.17	70.47	63.00	66.74	65.80
	TCD	✓	79.73	49.47	69.68	66.29	70.74	62.33	66.54	66.39
	DINO-HEAL	✓	62.11	47.08	51.34	53.51	65.99	58.96	62.48	57.10
	SEASON	✓	79.21	47.00	71.15	65.79	72.20	62.78	67.49	66.47
	ACTIV	✓	82.15	50.07	72.62	68.28	72.21	61.81	67.01	67.77
	ACTIV+VEM	✓	82.77	48.59	72.86	68.07	72.29	61.81	67.05	67.66
	TPO	✗	78.89	50.32	67.24	65.48	70.15	63.11	66.63	65.94
	RRPO	✗	77.02	51.61	70.42	66.35	70.98	63.04	67.01	66.61

Table 4: **Comparison of spatial and temporal negative construction methods** on LLaVA-OneVision-7B. Bold marks the best score in each column.

(a) Spatial negatives

Negative	VideoHallucer Avg.	EventHallusion Avg.
Original Video	40.86	61.61
Spatial flip	40.39	62.10
Spatial shuffle	40.92	61.86
SEASON	41.18	**64.55**
ACTIV	**41.20**	61.86

(b) Temporal negatives

Negative	VideoHallucer Avg.	EventHallusion Avg.
Original Video	40.86	61.61
Gaussian noise	40.29	61.12
Temporal reversal	42.68	66.50
SEASON	39.33	71.39
ACTIV	**42.95**	**73.11**

Combining the two negatives. Table 5 tests how to combine ACTIV's spatial and temporal negatives on LLaVA-Video-7B, keeping the negatives and main settings fixed. We compare equal weights, SEASON's attention-based weights, and ACTIV's weights based on changes in predictions. ACTIV achieves the highest three-benchmark average (68.28), exceeding equal and attention weighting by 0.50 and 0.41 points, respectively. A gated variant skips correction when the negatives have little effect on predictions (summed divergence below 0.05); its average drops to 66.21.

Table 5: Weighting rules for combining spatial and temporal negatives on LLaVA-Video-7B.

Weighting rule	VidHalluc	VideoHallucer	EventHallusion	Overall
Equal weights	81.62	50.08	71.64	67.78
Attention weights (SEASON)	82.10	49.64	71.88	67.87
ACTIV+gating	78.94	**50.49**	69.19	66.21
ACTIV	**82.15**	50.07	**72.62**	**68.28**

Contribution of each negative. Table 6 shows that combining spatial and temporal negatives outperforms either alone on LLaVA-Video-7B, achieving 67.37% ORH and 86.32% TSH accuracy under the main settings.

Transfer to a larger model. Using its 7B counterpart's configuration without retraining or retuning, ACTIV improves LLaVA-OneVision-72B's event-order accuracy on VidHalluc TSH from 89.67% to 91.00%.

Table 6: Comparison of contrastive decoding with spatial negatives, temporal negatives, or both on LLaVA-Video-7B.

Configuration	VideoHallucer ORH	VidHalluc TSH
Original Video	65.26	84.74
Spatial only	64.21	84.21
Temporal only	64.74	85.26
Both (ACTIV)	**67.37**	**86.32**

Visual support after ACTIV. We score baseline and ACTIV answers with the unmodified model (Section 2). Averaged equally across three models and five benchmarks, the share of answers with VEM ≤ 0 falls from 21.9% to 18.1%, a 17.6% relative reduction. Reductions are 5.6%, 22.7%, and 29.0% for Qwen2.5-VL, LLaVA-OneVision, and LLaVA-Video, respectively.

5 RELATED WORK

Video understanding can improve through better input selection, representation learning, or inference-time correction. Frame selection prioritizes informative observations (Hu et al., 2025), while MASH-VLM learns representations that separate spatial scene context from temporal actions (Bae et al., 2025). We pursue correction with an existing model, avoiding additional training while remaining complementary to these improvements. Training-free inference still has a cost: FastVLM and AutoGaze highlight the expense of vision encoding, particularly at high resolution or over long videos (Vasu et al., 2025; Shi et al., 2026). This motivates correcting answers using visual features already computed.

Contrastive decoding corrects generation by comparing predictions under different visual conditions. VCD uses noisy images (Leng et al., 2024), M3ID removes image conditioning (Favero et al., 2024), and TCD uses fewer video frames (Zhang et al., 2024a). DINO-HEAL reweights features using saliency from an auxiliary encoder (Li et al., 2025a). SEASON (Wu et al., 2026) combines spatial and temporal contrasts with attention-based weighting. VADER (Xing et al., 2026) constructs negatives through decoder-attention masking. STEAR (Fan et al., 2026) also reuses a single video encoding but relies on attention-guided patch selection and middle-layer reinjection; ACTIV weights separate spatial and temporal contrasts over all visual tokens by prediction divergence, without attention maps, layer selection, or uncertainty thresholds.

These interventions also raise the question of whether an answer is supported by visual evidence. MM-SHAP measures modality contributions independently of accuracy (Parcalabescu & Frank, 2023); VIGIL uses likelihood differences under altered visual access for post-training (Xiao et al., 2026), and DICA uses visual-dependence signals for inference-time correction (Yang et al., 2026). VEM shows that replacing the video's encoded features can leave even a correct answer equally or more probable. This finding motivates ACTIV to favor predictions that lose support when spatial or temporal information is weakened, using features already computed by the model.

6 CONCLUSION

Video language models must reliably describe locations and event order while keeping low inference latency. ACTIV reduces spatial and temporal hallucinations without training by contrasting token interventions with the original features from a single video encoding. Averaging across locations or time weakens the corresponding information; weighting each contrast by its effect on predictions favors answers supported by the original features. Across three models and three hallucination benchmarks, ACTIV outperforms the evaluated competing training-free methods in average score for each model. On LLaVA-Video-7B, it improves VidHalluc and EventHallusion by 3.87 and 5.38 points over baseline, with 22–26% lower estimated inference latency than our state-of-the-art method implementation. These findings demonstrate a useful accuracy–latency tradeoff from constructing spatial and temporal contrasts in already-computed visual-token space.

7 AI USE STATEMENT

We used generative AI tools to refine the manuscript's wording and structure, edit and review code, and modify figures. All AI-assisted work was reviewed and verified by an author. We take responsibility for the final content, including the manuscript, code, figures, and reported results.

8 REPRODUCIBILITY STATEMENT

We provide the VEM definition in Section 2 and the token interventions and decoding procedure, including the VEM-guided variant, in Section 3. Section 4.1 describes the models, benchmarks, baseline implementations, parameter selection, and statistical testing. Section 4 reports the settings for transfer and ablation experiments; Table 2 and its accompanying text specify the hardware and latency calculation. Appendix A.1 details the baseline implementations and the settings of the 16- and 32-frame profiling runs.

REFERENCES

Kyungho Bae, Jinhyung Kim, Sihaeng Lee, Soonyoung Lee, Gunhee Lee, and Jinwoo Choi. Mash-vlm: Mitigating action-scene hallucination in video-llms through disentangled spatial-temporal representations. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13744–13753. IEEE, 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, 2025. URL https://arxiv.org/abs/2502.13923.

Linfeng Fan, Yuan Tian, Ziwei Li, and Zhiwu Lu. Stear: Layer-aware spatiotemporal evidence intervention for hallucination mitigation in video large language models. *arXiv preprint arXiv:2604.03045*, 2026.

Yue Fan, Xiaojian Ma, Rongpeng Su, Jun Guo, Rujie Wu, Xi Chen, and Qing Li. Embodied videoagent: Persistent memory from egocentric videos and embodied sensors enables dynamic scene understanding. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 6342–6352. IEEE, 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 *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14303–14312. IEEE, 2024.

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.

Kai Hu, Feng Gao, Xiaohan Nie, Peng Zhou, Son Tran, Tal Neiman, Lingyun Wang, Mubarak Shah, Raffay Hamid, Bing Yin, and Trishul Chilimbi. M-LLM based video frame selection for efficient video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13702–13712, 2025.

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

Chaoyu Li, Eun Woo Im, and Pooyan Fazli. Vidhalluc: Evaluating temporal hallucinations in multimodal large language models for video understanding. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13723–13733. IEEE, 2025a.

Rui Li, Xiaohan Wang, Yuhui Zhang, Orr Zohar, Zeyu Wang, and Serena Yeung-Levy. Temporal preference optimization for long-form video understanding. *arXiv preprint arXiv:2501.13919*, 2025b.

Jianhua Lin. Divergence measures based on the shannon entropy. *IEEE Transactions on Information theory*, 37(1):145–151, 1991.

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, 2024.

Zhijian Liu, Ligeng Zhu, Baifeng Shi, Zhuoyang Zhang, Yuming Lou, Shang Yang, Haocheng Xi, Shiyi Cao, Yuxian Gu, Dacheng Li, Xiuyu Li, Haotian Tang, Yunhao Fang, Yukang Chen, Cheng-Yu Hsieh, De-An Huang, An-Chieh Cheng, Jinyi Hu, Sifei Liu, Ranjay Krishna, Pavlo Molchanov, Jan Kautz, Hongxu Yin, Song Han, and Yao Lu. NVILA: Efficient frontier visual language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4122–4134, 2025.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

Letitia Parcalabescu and Anette Frank. Mm-shap: A performance-agnostic metric for measuring multimodal contributions in vision and language models & tasks. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 4032–4059, 2023.

Pritam Sarkar and Ali Etemad. Self-alignment of large video language models with refined regularized preference optimization. In *Advances in Neural Information Processing Systems*, volume 38, pp. 47603–47635, 2025.

Baifeng Shi, Stephanie Fu, Long Lian, Hanrong Ye, David Eigen, Aaron Reite, Jan Kautz, Boyi Li, David M Chan, Trevor Darrell, et al. Attend before attention: Efficient and scalable video understanding via autoregressive gazing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 17022–17034, 2026.

Siddharth M. Tom and Jerry Cartoon Complete Frames Collection. Kaggle dataset, 2025. URL https://www.kaggle.com/datasets/siddharthm1698/tom-and-jerry-cartoon-complete-frames-collection.

Michael Tschannen, Alexey Gritsenko, Xiao Wang, Muhammad Ferjad Naeem, Ibrahim Alabdulmohsin, Nikhil Parthasarathy, Talfan Evans, Lucas Beyer, Ye Xia, Basil Mustafa, et al. Siglip 2: Multilingual vision-language encoders with improved semantic understanding, localization, and dense features. *arXiv preprint arXiv:2502.14786*, 2025.

Pavan Kumar Anasosalu Vasu, Fartash Faghri, Chun-Liang Li, Cem Koc, Nate True, Albert Antony, Gokula Santhanam, James Gabriel, Peter Grasch, Oncel Tuzel, et al. Fastvlm: Efficient vision encoding for vision language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 19769–19780, 2025.

Yuxuan Wang, Yueqian Wang, Dongyan Zhao, Cihang Xie, and Zilong Zheng. Videohallucer: Evaluating intrinsic and extrinsic hallucinations in large video-language models. *arXiv preprint arXiv:2406.16338*, 2024.

Chang-Hsun Wu, Kai-Po Chang, Yu-Yang Sheng, Hung-Kai Chung, Kuei-Chun Wang, and Yu-Chiang Frank Wang. SEASON: Mitigating temporal hallucination in video large language models via self-diagnostic contrastive decoding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11096–11105, 2026.

Xi Xiao, Chen Liu, Chih-Ting Liao, Yunbei Zhang, Qizhen Lan, Yuxiang Wei, Lin Zhao, Janet Wang, Jianyang Gu, Muchao Ye, Tianyang Wang, and Hao Xu. Staying VIGILant: Mitigating visual laziness via counterfactual visual alignment in MLLMs. In *European Conference on Computer Vision*, 2026.

Dong Xing, Jiaxin Chen, Hang Yang, Peixun Liu, Qiushi Yang, and Yuqing Wang. Vader: Adaptive debiasing for hallucination mitigation in video large language models. *arXiv preprint arXiv:2608.08622*, 2026.

Zihui Sherry Xue, Romy Luo, and Kristen Grauman. Seeing the arrow of time in large multimodal models. In *Advances in Neural Information Processing Systems*, volume 38, pp. 100976–101006, 2025.

Hao Yang, Jin Wang, and Xuejie Zhang. Dica: Dual-indicator guided contrastive alignment in multimodal large language models. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 38797–38818, 2026.

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

Yuanhan Zhang, Jinming Wu, Wei Li, Bo Li, Zejun Ma, Ziwei Liu, and Chunyuan Li. Llava-video: Video instruction tuning with synthetic data. *arXiv preprint arXiv:2410.02713*, 2024b.

Boyang Zheng, Nanye Ma, Shengbang Tong, and Saining Xie. Diffusion transformers with representation autoencoders. In *International Conference on Learning Representations*, volume 2026, pp. 35791–35820, 2026.

Orr Zohar, Xiaohan Wang, Yann Dubois, Nikhil Mehta, Tong Xiao, Philippe Hansen-Estruch, Licheng Yu, Xiaofang Wang, Felix Juefei-Xu, Ning Zhang, et al. Apollo: An exploration of video understanding in large multimodal models. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18891–18901. IEEE, 2025.

A Appendix

A.1 Baseline Implementations

TCD and SEASON. We implement TCD (Zhang et al., 2024a) and SEASON (Wu et al., 2026) from their method descriptions under the shared evaluation protocol in Section 4.1. TCD separately encodes a sparse frame sample for its negative; SEASON uses encoder-level temporal homogenization and re-encoded noisy frames with attention-based weighting. Their additional encoder passes are included in the reported inference cost. In the 16- and 32-frame profiling runs, TCD uses 4 and 8 negative frames, respectively; SEASON uses contrast strength 1 and homogenization strength 0.33, and ACTIV retains its main LLaVA-Video configuration $(0.75, 0.5, 1)$.

DINO-HEAL. We consistently use the published Eqs. 9–12 of Li et al. (2025a): interpolate the DINOv2 saliency map, apply a sigmoid, and multiply the visual features by the resulting weights. Reweighting is applied before LLaVA's projector and after Qwen2.5-VL's patch merger. To assess the alternative released-code rule, which differs in saliency normalization and feature reweighting, we evaluate both rules on the same 600 VidHalluc TSH questions with an eight-frame cap (Table 7). The reported equation-based rule scores higher than the released-code rule on all three backbones in this controlled comparison; the LLaVA degradation also occurs with the released rule, and adding the released code's per-frame min-max normalization to the equation rule changes accuracy by at most 0.67 points.

Because the sigmoid is applied to saliency in $[0, 1]$ (softmax attention probabilities, with or without min-max normalization), every weight lies between 0.5 and 0.73, so the equation rule roughly halves the visual features. Qwen2.5-VL tolerates this, while both LLaVA models do not: rescaling each frame's weights to average one, a control of ours rather than part of DINO-HEAL, returns both LLaVA models to within one point of baseline (-0.83 and $+0.33$). The degraded outputs take three forms. On TSH, LLaVA-OneVision's replies remain well formed but fall to near the 50% chance

level of this two-way task. On STH, every LLaVA reply under the equation rule reports no scene change; a constant answer has zero Matthews correlation and no correct descriptions, which the STH metric scores as exactly 15.00. The released rule's 0.50% on LLaVA-OneVision reflects 591 of 600 replies that state no complete action order, 131 of them empty, while on LLaVA-Video it answers "Action A before Action B" to 543 of 600 questions. STEAR (Fan et al., 2026) and VADER (Xing et al., 2026) report neutral or positive DINO-HEAL results on LLaVA models under their own implementations and evaluation protocols. Together with the spread across rules in Table 7, this indicates that DINO-HEAL's effect is highly implementation-sensitive; our DINO-HEAL results are those of the published equations under our protocol.

Table 7: Auxiliary DINO-HEAL implementation comparison on all 600 VidHalluc TSH questions. Entries are accuracy (%). The equation-based rule is the implementation used throughout our reported comparisons. The min-max variant adds the released code's saliency normalization to the equation rule; the mean-one control rescales each frame's weights to average one and is not part of DINO-HEAL.

Model	Baseline	Equation	Min-max	Released code	Mean-one
Qwen2.5-VL-7B	84.00	85.33	85.00	78.17	83.83
LLaVA-OneVision-7B	89.00	52.33	53.00	0.50	88.17
LLaVA-Video-7B	84.67	71.67	72.00	51.67	85.00

A.2 Effect of Increasing the Number of Frames

We compare 16 and 32 input frames on 100 EventHallusion questions, whose category and answer shares match the full benchmark, using LLaVA-Video-7B, pooling results from nine jobs with one RTX A6000 per job (Tables 8 and 9). Latency is the median per-item sum of vision encoding, prefill, and decoding time, excluding video loading; relative latency is normalized to the baseline at the same frame count. Memory is PyTorch's peak allocated GPU memory, including approximately 16 GB of model weights. Accuracy is reported as the number of correct answers out of 100.

Table 8: Efficiency and accuracy with 16 input frames on 100 EventHallusion questions using LLaVA-Video-7B and one RTX A6000 per job.

Method	Latency (s)	Relative	Memory (GB)	Accuracy
Baseline	0.87	1.00×	18.5	68/100
DINO-HEAL	1.01	1.16×	19.3	59/100
TCD	1.16	1.34×	18.6	68/100
ACTIV	1.86	2.13×	18.8	76/100
SEASON	2.45	2.82×	18.8	72/100

Table 9: Efficiency and accuracy with 32 input frames on the same 100 EventHallusion questions using LLaVA-Video-7B and one RTX A6000 per job.

Method	Latency (s)	Relative	Memory (GB)	Accuracy
Baseline	1.65	1.00×	20.5	70/100
DINO-HEAL	1.91	1.16×	21.4	55/100
TCD	2.13	1.30×	20.6	74/100
ACTIV	3.72	2.26×	21.1	79/100
SEASON	4.84	2.94×	21.1	76/100

Findings. ACTIV retains 23–24% lower estimated latency than SEASON at both frame counts, with equal peak allocated memory at the reported precision. Doubling the frames doubles ACTIV's latency and increases its memory from 18.8 to 21.1 GB. Moving to 32 frames changes accuracy by at most six questions per method, raising it for all methods except DINO-HEAL. ACTIV is the most accurate method at both frame counts. In paired comparisons, it alone is correct on 9 and 11 questions versus 1 and 2 for the baseline (exact McNemar $p = 0.02$ in both cases), whereas its 5–1 and 4–1 splits against SEASON are not significant at this sample size.

A.3 LATENCY BREAKDOWN

Table 10 separates the recorded costs for the same 100 questions and nine profiling jobs used in Appendix A.2. Visual processing includes the encoder passes required by each method and, for ACTIV, construction of the token negatives. Both ACTIV and SEASON use three decoder branches, while requiring one and three video encoding passes, respectively. Each component is summarized by its median; the total is the median of the per-item sums, as in the frame-count tables. Consequently, the component medians need not sum exactly to the total.

Table 10: Latency breakdown in seconds on LLaVA-Video-7B over the same 100 EventHallusion questions, with one RTX A6000 per profiling job.

Frames	Method	Visual	Prefill	Decoding	Total
16	Baseline	0.292	0.516	0.061	0.869
	ACTIV	0.296	1.477	0.082	1.856
	SEASON	0.893	1.471	0.087	2.451
32	Baseline	0.549	1.026	0.071	1.646
	ACTIV	0.556	3.049	0.111	3.717
	SEASON	1.682	3.040	0.122	4.845

The saving is concentrated in visual processing: ACTIV's cost is approximately one third of SEASON's at both frame counts, while their prefill times differ by less than 1%. Constructing ACTIV's token negatives takes a median of 1.7 ms at 16 frames and 3.8 ms at 32 frames, included in the visual-processing column. Thus, reusing the encoded tokens removes repeated visual computation while retaining both contrastive branches; the remaining overhead is primarily decoder prefill.