SPANPROBE: AN ADAPTIVE HALLUCINATION DETECTION FRAMEWORK BASED ON TOKEN EVIDENCE PROFILES

Anonymous authors
Paper under double-blind review

ABSTRACT

Multimodal large language models (MLLMs) can generate fluent visual descriptions but remain prone to hallucinations, limiting their reliable deployment. Existing hallucination detection methods based on model-internal hallucination signals typically rely on predefined readout strategies. Because tokens within a span carry different information, fixed readout strategies struggle to adequately capture the relative importance of individual tokens, making it difficult to reliably assess the hallucination risk of a complete semantic span. We find that hallucination signals tend to concentrate on a small number of tokens within each span and that the token carrying the strongest signal varies across samples. Therefore, accurately characterizing the relative importance of individual tokens for hallucination discrimination under such a non-uniform distribution is essential for reliable span-level hallucination detection. To address this challenge, we propose SpanProbe, an adaptive hallucination detection framework based on Token Evidence Profiles (TEPs). SpanProbe extracts visual-attention mass, full-attention entropy, and attention-head output norms from every attention head in every decoder layer to characterize each token's visual dependence, attention dispersion, and activation strength. It then dynamically generates sample-dependent aggregation weights from the token TEPs and uses these weights to construct a span-level representation for hallucination detection. Furthermore, SpanProbe employs a risk-gated local rollback-and-regeneration strategy. Using the risk scores produced by the pooling module, it identifies high-risk semantic spans and selectively revises high-risk content while preserving previously accepted low-risk content, thereby mitigating hallucinations during generation. Extensive experiments across multiple benchmarks demonstrate that SpanProbe generally outperforms existing baselines. On M-HalDetect, which contains multi-token semantic spans, SpanProbe achieves an average AUPRC of 77.43%, outperforming the strongest baseline by 8.05 percentage points.

1 INTRODUCTION

Despite their powerful vision-language generation capabilities, multimodal large language models (MLLMs) still produce hallucinations, limiting their reliability (Guan et al., 2024; Wang et al., 2024a; Bai et al., 2025b). Accurately detecting hallucinations (Jing et al., 2024; Chen et al., 2024a; Petryk et al., 2024; Qiu et al., 2024) and effectively mitigating them (Yin et al., 2024; Zhou et al., 2024) are key to improving the trustworthiness of MLLMs and ensuring their safe and reliable deployment in real-world settings.

Existing research on hallucinations in MLLMs generally follows two approaches. The first directly mitigates hallucinations during generation by adjusting decoding strategies or strengthening the use of visual information (Woo et al., 2024; Zhu et al., 2025), as exemplified by VCD (Leng et al., 2024), OPERA (Huang et al., 2024), and PAI (Liu et al., 2025). The second detects hallucinations and then uses the resulting signals to guide mitigation (Sahu et al., 2024; Lee et al., 2024). For example, HaloProbe (Zohrabi et al., 2026) guides decoding using token-level hallucination probabilities, while VIB-Probe (Zhang et al., 2026) employs a variational information bottleneck to extract

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.

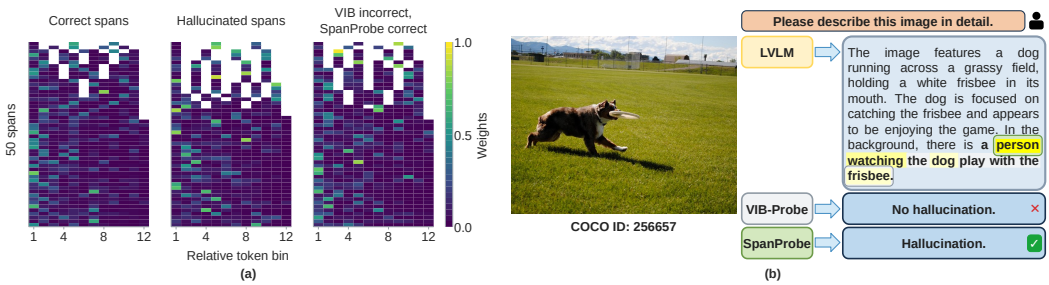

Figure 1: Motivation for token-adaptive span-level hallucination detection. (a) Token-weight heatmaps for hallucination-free spans, spans containing hallucinations, and spans misclassified by VIB-Probe but correctly classified by SpanProbe. Weights are non-uniform within spans, and high-weight positions vary across samples; white cells indicate empty positions. (b) A disagreement case in which VIB-Probe fails to detect a hallucination while SpanProbe correctly identifies it. Bold text marks the semantic span under evaluation. The gray outline marks the span-final token used by VIB-Probe. Yellow shading indicates relative token weights, and the green outline marks the token assigned the highest weight by SpanProbe. Check marks and crosses indicate correct and incorrect predictions, respectively.

discriminative information from attention-head outputs across layers and uses probe gradients to identify and suppress hallucination-sensitive attention heads.

Although hallucination detection methods based on internal signals have shown promise, most still rely on predefined rules to read detection signals from semantic spans (Fieback et al., 2025; Zhang et al., 2025; 2026). Because different tokens within a span carry different information, fixed readout strategies cannot adequately capture their relative importance, making it difficult to reliably assess the hallucination risk of the complete semantic span.

To investigate how hallucination signals vary across tokens within a semantic span, we examine within-span token weights. Figure 1(a) summarizes multiple samples, showing that the weights are distributed non-uniformly and that the positions of high-weight tokens vary across samples. Figure 1(b) provides a concrete example: VIB-Probe reads detection signals only at the span-final token, missing informative cues elsewhere in the span and failing to detect the hallucination. In contrast, SpanProbe adaptively aggregates information across the span and makes the correct prediction. Together, these observations suggest that hallucination-discriminative information is not confined to a fixed position. Accurately characterizing the relative importance of tokens is therefore crucial for reliable span-level hallucination detection.

To address this challenge, we propose SpanProbe, an adaptive hallucination detection framework based on Token Evidence Profiles (TEPs). SpanProbe extracts three types of features from every attention head in every decoder layer of the model to construct a compact TEP along three complementary dimensions: evidence source, distribution, and strength. Specifically, visual-attention mass, full-attention entropy, and attention-head output norms characterize each token's reliance on visual information, attention dispersion, and attention-head output strength, respectively. SpanProbe then employs token-adaptive attention pooling to dynamically generate sample-dependent aggregation weights from the TEPs of tokens in the current span. Using these weights, it aggregates token representations into a span-level representation and estimates the hallucination risk of the complete semantic span.

Furthermore, SpanProbe employs a risk-gated local rollback-and-regeneration strategy to mitigate hallucinations. Using the risk scores derived from the pooled span representations, it identifies high-risk semantic spans and selectively revises their content while preserving previously accepted low-risk content, thereby effectively mitigating hallucinations during generation.

Extensive experiments on four representative MLLMs and three hallucination benchmarks demonstrate that SpanProbe generally outperforms existing hallucination detection methods and can effectively guide downstream hallucination mitigation. On M-HalDetect, which contains multi-token semantic spans, SpanProbe achieves an average AUPRC of 77.43% across four backbones, outperforming the strongest baseline by 8.05 percentage points. In addition, on LLaVA-v1.6, risk-

gated rollback achieves relative reductions of more than 60% in both CHAIR_i and CHAIR_s on MSCOCO.

Our main contributions are summarized as follows:

1. We reveal that hallucination signals are distributed non-uniformly within complete semantic spans and that the positions of the tokens carrying the strongest signals vary across samples, exposing a key limitation of fixed-position readout strategies.
2. We propose SpanProbe, which combines Token Evidence Profiles with token-adaptive attention pooling. The profiles guide the weighting and aggregation of tokens within each span to construct a span-level representation, enabling SpanProbe to model the hallucination risk of complete semantic spans and perform effective span-level hallucination detection. SpanProbe further uses the estimated risk to drive local rollback and regeneration, thereby mitigating hallucinations in generated content.
3. We validate the detection capability of SpanProbe and its effectiveness for inference-time hallucination mitigation across four representative MLLMs and three hallucination benchmarks.

2 Related Work

2.1 Hallucination Detection Based on Internal Signals

Existing MLLM hallucination detectors typically verify generated content with external tools (Yin et al., 2024; Park & Li, 2025) or directly exploit output statistics, hidden states, and attention representations produced during inference (Kogilathota et al., 2026). Internal-signal methods further adopt different token readouts: MetaToken (Fieback et al., 2025) aggregates token log-probabilities within a target mention and combines them with predictive uncertainty and visual attention at the mention-start token; ContextualLens (Phukan et al., 2025) and NTP (Azachi et al., 2025) respectively use mean pooling and predefined statistics to summarize information within a span; DHCP (Zhang et al., 2025) and VIB-Probe (Zhang et al., 2026) detect hallucinations using cross-modal attention patterns and cross-layer attention-head outputs, respectively.

These studies show that diverse internal signals and multi-token information are useful for hallucination detection, but their readouts still rely on selected tokens or predefined aggregation rules. SpanProbe instead treats the complete semantic span as the detection unit, learns sample-adaptive relative token weights, and aggregates their internal evidence into a unified span-level representation.

2.2 Hallucination Mitigation in MLLMs

Existing MLLM hallucination mitigation methods fall into two categories. The first directly adjusts decoding without explicit hallucination detection (Wang et al., 2024b; Chen et al., 2024b; Wei & Zhang, 2024; Huo et al., 2025; Zhuang et al., 2025; An et al., 2025; Woo et al., 2025): VCD (Leng et al., 2024) contrasts output distributions under different visual conditions, while OPERA (Huang et al., 2024), PAI (Liu et al., 2025), and VISTA (Li et al., 2025) reduce hallucinations through decoding constraints, enhanced visual attention, and visual-information guidance, respectively. The second triggers interventions using detection signals (Wu et al., 2024). EAZY (Che et al., 2025) suppresses image tokens that induce hallucinations, HaloProbe (Zohrabi et al., 2026) uses token-level hallucination probabilities to guide decoding, and VIB-Probe (Zhang et al., 2026) uses probe gradients to identify and suppress hallucination-sensitive attention heads before regenerating the current token. In contrast, SpanProbe uses complete-span risk as a gating signal and performs rollback and regeneration only on localized high-risk content.

3 Method

Figure 2 presents the overall framework of SpanProbe. Given a candidate semantic span, SpanProbe extracts internal hallucination signals from every attention head in every decoder layer of a frozen MLLM to construct a Token Evidence Profile (TEP) for each token in the span. Based on these TEPs,

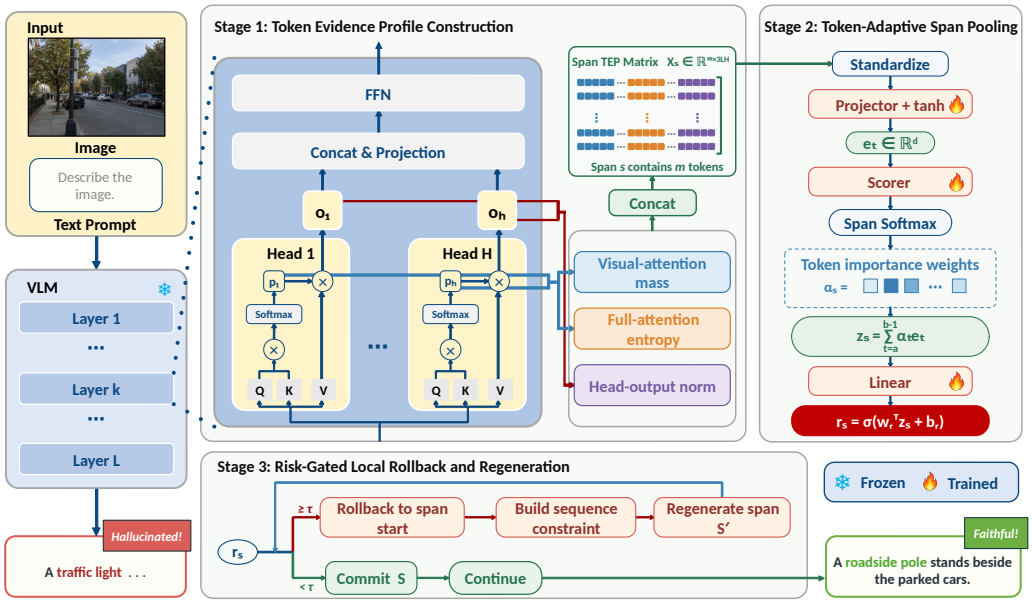

Figure 2: Overview of the complete SpanProbe framework, comprising span-level hallucination detection and risk-gated local rollback and regeneration. SpanProbe constructs a TEP for each subtoken using internal signals extracted from all decoder layers and attention heads of a frozen MLLM. A shared projection and token-adaptive attention pooling then produce a span-level hallucination risk. Low-risk candidates are committed directly, whereas for high-risk candidates, generation rolls back to the span start and proceeds under sequence constraints.

SpanProbe generates adaptive aggregation weights for the current span and uses them to combine token representations into a span-level representation and estimate hallucination risk. This risk further guides local rollback and constrained regeneration, so that only high-risk semantic content is revised.

3.1 Overview and Problem Formulation

Given an image I, a textual instruction q, and a frozen multimodal large language model F, the model autoregressively generates a token sequence

$$Y = (y_1, y_2, \ldots, y_T). \tag{1}$$

We represent a complete semantic unit in the response as a contiguous span

$$s = [a, b), \qquad Y_s = (y_a, y_{a+1}, \ldots, y_{b-1}), \tag{2}$$

where $1 \le a < b \le T + 1$.

Let $c_s \in \{0, 1\}$ denote the hallucination label of span s, where $c_s = 1$ indicates that the span expresses hallucinated content lacking visual grounding, whereas $c_s = 0$ indicates visually grounded content. Let θ denote the trainable parameters of SpanProbe. SpanProbe estimates

$$r_s = P_\theta(c_s = 1 \mid I, q, Y_{<b}; F), \tag{3}$$

where $r_s \in [0, 1]$ denotes the hallucination risk of the complete semantic span. SpanProbe aggregates token evidence only within the candidate interval $t \in [a, b)$, thereby assessing the risk associated with the current semantic unit.

The base model F remains frozen throughout, and only the shared token projection, importance scorer, and span-level risk head of SpanProbe are optimized.

3.2 TOKEN EVIDENCE PROFILE

To characterize the diagnostic information carried by different tokens within the same semantic span, SpanProbe constructs a Token Evidence Profile for each token using three types of statistics.

Suppose that the underlying MLLM contains L decoder layers, each with H self-attention heads, and that each attention head has dimensionality d_h. For response position t, decoder layer ℓ, and attention head h, let $\mathcal{K}_t$ denote all key positions accessible from the current position under the causal mask, and let $\mathcal{V}_t \subseteq \mathcal{K}_t$ denote the positions corresponding to visual tokens. The normalized attention probability is defined as

$$p_{t,\ell,h,j} = \frac{\exp\left(q_{t,\ell,h}^{\top} k_{j,\ell,h}/\sqrt{d_h}\right)}{\sum_{j' \in \mathcal{K}_t} \exp\left(q_{t,\ell,h}^{\top} k_{j',\ell,h}/\sqrt{d_h}\right)}, \qquad j \in \mathcal{K}_t. \tag{4}$$

After value-vector aggregation and before the output projection `o_proj`, the output of an individual attention head is

$$o_{t,\ell,h} = \sum_{j \in \mathcal{K}_t} p_{t,\ell,h,j} v_{j,\ell,h}. \tag{5}$$

3.2.1 VISUAL ATTENTION MASS

$$A_{t,\ell,h}^{\text{vis}} = \sum_{j \in \mathcal{V}_t} p_{t,\ell,h,j}. \tag{6}$$

This statistic is the probability mass that the current attention head assigns to all visual tokens, characterizing the extent to which the current token relies on visual input.

3.2.2 FULL-ATTENTION ENTROPY

$$H_{t,\ell,h} = -\sum_{j \in \mathcal{K}_t} p_{t,\ell,h,j} \log(\max\{p_{t,\ell,h,j}, \epsilon\}), \qquad \epsilon = 10^{-12}, \tag{7}$$

where ϵ is a constant introduced for numerical stability. This statistic characterizes how concentrated or dispersed the attention of the current token is over all causally accessible key positions.

3.2.3 ATTENTION-HEAD OUTPUT NORM

$$N_{t,\ell,h} = \|o_{t,\ell,h}\|_2. \tag{8}$$

This statistic measures the magnitude of an individual attention-head output after value-vector aggregation, characterizing the strength of the activation propagated to subsequent computations.

Finally, the three groups of statistics are concatenated in a fixed order:

$$x_t = \text{Concat}\left[\text{vec}\left(A_t^{\text{vis}}\right); \text{vec}(H_t); \text{vec}(N_t)\right] \in \mathbb{R}^{3LH}. \tag{9}$$

Thus, x_t provides a compact Token Evidence Profile for each token while preserving information across decoder layers and attention heads.

3.3 TOKEN-ADAPTIVE SPAN POOLING

For a semantic span $s = [a, b)$, we standardize each TEP and apply a shared projection to obtain a low-dimensional token representation:

$$\widetilde{x}_t = \frac{x_t - \mu}{\sigma}, \qquad e_t = \tanh(W_p \widetilde{x}_t + b_p), \qquad t \in [a, b), \tag{10}$$

where μ and σ are the feature-wise mean and standard deviation estimated from the training set. Because the most diagnostic token varies across individual spans, a shared importance scorer computes an importance logit and a relative weight for each token:

$$a_t = w_a^{\top} e_t + b_a, \qquad \alpha_t = \frac{\exp(a_t)}{\sum_{k=a}^{b-1} \exp(a_k)}. \tag{11}$$

SpanProbe aggregates the token representations within the span using the learned weights and predicts its hallucination risk:

$$z_s = \sum_{t=a}^{b-1} \alpha_t e_t, \qquad u_s = w_r^\top z_s + b_r, \qquad r_s = \mathrm{sigmoid}(u_s). \tag{12}$$

Given a training set of spans $\mathcal{D}$, the detector is optimized using class-weighted binary cross-entropy:

$$\mathcal{L}_{\mathrm{det}} = -\frac{1}{|\mathcal{D}|} \sum_{(s,c_s)\in\mathcal{D}} \left[w_+ c_s \log r_s + (1-c_s)\log(1-r_s) \right], \qquad w_+ = \frac{N_-}{N_+}. \tag{13}$$

Here, N_- and N_+ are the numbers of grounded and hallucinated spans in the corresponding training partition. In our implementation, we apply `BCEWithLogitsLoss` to u_s for numerical stability. During training, only the shared projection, importance scorer, and span-level risk head are updated; the underlying MLLM and TEP feature extraction remain frozen. When a span contains a single token, the within-span softmax reduces to $\alpha_a = 1$, allowing SpanProbe to handle single-token and multi-token semantic spans within a unified framework.

3.4 Risk-Gated Local Regeneration

In addition to standalone hallucination detection, the continuous risk score produced by SpanProbe can control autoregressive generation. A span constructor $\mathcal{C}$ monitors the current generation prefix $y_{1:t}$ and determines whether a candidate semantic span has been completed. When it returns boundaries $s = [a, b)$, the candidate content, token evidence, lookahead tokens required to confirm the boundary, and associated decoding states are temporarily stored in a private buffer and are not committed to the final output until risk assessment is complete. This procedure uses only content generated up to the current position.

SpanProbe computes r_s using all TEPs within the candidate span. Given a threshold τ fixed before evaluation, the risk-gating rule is

$$\mathrm{Gate}(s) = \begin{cases} \text{commit}, & r_s < \tau, \\ \text{rollback and regenerate}, & r_s \geq \tau. \end{cases} \tag{14}$$

When $r_s < \tau$, the candidate is committed and generation continues. When $r_s \geq \tau$, the model removes the candidate and the lookahead used to confirm its boundary, restores the decoding state aligned with the preceding prefix $Y_{<a}$, and resumes generation from the same position. All content preceding the candidate span therefore remains unchanged.

To prevent the rejected local content from being generated again, we retain the rejected span's original token sequence $\mathbf{y}_s^{\mathrm{rej}} = (y_a, \ldots, y_{b-1})$ and use it as a constraint during the current regeneration. If a next-token candidate w would complete this rejected sequence, its logit is masked:

$$\widetilde{\ell}_t(w) = \begin{cases} -\infty, & \text{if } w \text{ would complete } \mathbf{y}_s^{\mathrm{rej}}, \\ \ell_t(w), & \text{otherwise.} \end{cases} \tag{15}$$

Once a regenerated candidate span s' has been completed, it is handled according to the task-specific rollback policy. When iterative re-evaluation is enabled, SpanProbe re-evaluates s' and triggers another rollback if its risk still reaches the threshold and the rollback budget has not been exhausted. Both the underlying MLLM and SpanProbe remain frozen throughout generation. Details of candidate construction, state restoration, and task-specific constraint and rollback policies are provided in Appendix A.

4 Experiments

We evaluate SpanProbe from two perspectives: hallucination detection and inference-time mitigation. First, we compare SpanProbe with existing internal-signal-based detectors across three complementary benchmarks and four representative MLLMs. We then examine whether span-level risk can guide selective generation control on LLaVA-v1.6 and Qwen2.5-VL. Finally, we analyze the effects of different span readout and aggregation strategies, as well as the three feature groups in the Token Evidence Profile. Experiments on feature selection across decoder layers are provided in the appendix.

Table 1: Hallucination detection results across four MLLMs. AUROC and AUPRC are percentages (%), with hallucination as the positive class and higher values preferred. Average is the arithmetic mean across backbones.

Dataset	Method	InstructBLIP		LLaVA-v1.5		LLaVA-v1.6		Qwen2.5-VL		Average	
		A-ROC	A-PR	A-ROC	A-PR	A-ROC	A-PR	A-ROC	A-PR	A-ROC	A-PR
POPE	MetaToken (2025)	94.12	94.42	93.99	94.14	95.48	95.67	91.43	91.61	93.76	93.96
	DHCP (2025)	93.83	94.15	91.78	91.67	92.14	92.67	94.49	94.76	93.06	93.31
	VIB-Probe (2026)	94.88	95.10	96.59	**96.65**	95.99	95.51	96.26	96.44	95.93	95.93
	Ours	**96.24**	**96.41**	**96.64**	96.61	**96.20**	**96.21**	**96.41**	**96.57**	**96.37**	**96.45**
M-Hal Detect	SVAR (2025)	51.15	23.43	56.53	29.03	55.29	25.93	44.51	21.93	51.87	25.08
	IC (2025)	56.44	35.35	62.10	38.36	59.98	37.05	61.21	35.25	59.93	36.50
	MetaToken (2025)	81.08	59.38	79.55	57.81	80.27	58.18	79.90	56.54	80.20	57.98
	DHCP (2025)	85.35	69.15	70.93	43.73	72.72	47.68	78.72	59.03	76.93	54.90
	VIB-Probe (2026)	81.34	66.49	83.86	69.98	84.27	70.67	84.54	70.38	83.50	69.38
	Ours	**88.09**	**75.71**	**89.71**	**77.48**	**90.14**	**79.16**	**89.11**	**77.38**	**89.26**	**77.43**
MSCOCO	SVAR (2025)	55.02	18.31	53.12	17.99	55.21	18.47	54.85	18.68	54.55	18.36
	IC (2025)	69.81	25.73	61.92	21.37	69.61	25.61	66.88	24.17	67.06	24.22
	MetaToken (2025)	92.58	73.23	92.65	73.68	92.97	76.35	92.31	70.68	92.63	73.49
	DHCP (2025)	89.21	67.51	76.05	45.18	75.82	40.15	83.66	57.55	81.19	52.60
	VIB-Probe (2026)	89.51	70.31	89.53	71.28	90.80	74.40	89.48	70.67	89.83	71.67
	Ours	**94.69**	**78.03**	**94.47**	**79.59**	**96.33**	**85.69**	**95.38**	**80.45**	**95.22**	**80.94**

4.1 Experimental Setup

Datasets. We conduct hallucination detection experiments on POPE (Li et al., 2023), M-HalDetect (Gunjal et al., 2024), and MSCOCO (Lin et al., 2014; Rohrbach et al., 2018), and hallucination mitigation experiments on POPE and MSCOCO.

Backbone models. For hallucination detection, we use InstructBLIP (Dai et al., 2023), LLaVA-v1.5-7B (Liu et al., 2024a), LLaVA-v1.6-Mistral-7B (Liu et al., 2024b), and Qwen2.5-VL-7B-Instruct (Bai et al., 2025a). For hallucination mitigation, we use LLaVA-v1.6-Mistral-7B and Qwen2.5-VL-7B-Instruct.

Baselines. For hallucination detection, we compare SpanProbe with SVAR (Jiang et al., 2025b), IC (Jiang et al., 2025a), MetaToken (Fieback et al., 2025), DHCP (Zhang et al., 2025), and VIB-Probe (Zhang et al., 2026). As the detection baseline most closely related to our method, VIB-Probe compresses attention-head outputs collected across decoder layers using a variational information bottleneck for token-level hallucination detection. For hallucination mitigation, we compare against Vanilla decoding, Beam Search, PAI (Liu et al., 2025), VCD (Leng et al., 2024), and EAZY (Che et al., 2025). Complete details of dataset construction, evaluation protocols, and training configurations are provided in Appendix A.

4.2 Hallucination Detection

As shown in Table 1, SpanProbe achieves the best overall performance across three benchmarks and four backbone models. On POPE, whose semantic units consist of single tokens, SpanProbe performs comparably to or better than the strongest baseline, demonstrating that its span-level design remains effective in simple single-token detection settings. Its advantage becomes more pronounced on M-HalDetect and MSCOCO, which contain multi-token semantic units. These results indicate that different tokens within the same span provide complementary evidence and that adaptive evidence aggregation characterizes the hallucination risk of complete semantic content more effectively than fixed-position readout.

Cross-Dataset and Cross-Granularity Zero-Shot Transfer. On LLaVA-v1.5-7B, we train detectors on M-HalDetect annotated spans and evaluate them on MSCOCO sentences, and reverse

the direction from MSCOCO sentences to M-HalDetect spans. Each detector and its feature-standardization statistics are learned only from the source training set and frozen. We use no target training, model selection, or threshold calibration, and remove overlapping images from the target test sets. Appendix C details the protocol and sample counts.

Table 2: Bidirectional zero-shot transfer across datasets and detection granularities.

Transfer Direction	Method	AUROC (%)↑	AUPRC (%)↑
M-HalDetect → MSCOCO	VIB-Probe	81.59	50.71
	MeanPool	89.06	66.15
	StatPool	78.80	43.56
	SpanProbe	**91.32**	**68.49**
MSCOCO → M-HalDetect	VIB-Probe	63.92	40.43
	MeanPool	66.45	43.61
	StatPool	57.07	41.77
	SpanProbe	**73.21**	**56.13**

As shown in Table 2, SpanProbe outperforms fixed-endpoint and predefined aggregation methods in both directions. Its advantage persists between annotated spans and full sentences, indicating that adaptive token aggregation transfers risk ranking across datasets and detection granularities. Boundary robustness is examined in Appendix D.

4.3 Hallucination Mitigation

As shown in Table 3, SpanProbe achieves the best POPE accuracy and F1 scores and the lowest MSCOCO CHAIR scores among the compared methods on both LLaVA-v1.6 and Qwen2.5-VL. In particular, on LLaVA-v1.6, CHAIR_i and CHAIR_s decrease by more than 60% relative to Vanilla. Meanwhile, SpanProbe retains most of the object recall, with SPICE scores and output lengths comparable to Vanilla decoding (Appendix Table 7). These results indicate that selectively rolling back and regenerating high-risk spans based on span-level risk can reduce hallucinations while largely preserving grounded object mentions and descriptive content.

Table 3: Hallucination mitigation results on LLaVA-v1.6-Mistral-7B and Qwen2.5-VL-7B-Instruct. MSCOCO metrics are percentages except SPICE (Anderson et al., 2016), which is scaled by 100.

Method	POPE Acc.↑	POPE F1↑	CHAIR_i ↓	CHAIR_s ↓	Obj. Rec.↑	SPICE↑
LLaVA-v1.6-Mistral-7B						
Vanilla	89.17	89.12	8.12	29.40	65.19	11.58
Beam Search	89.17	88.81	7.02	26.40	64.68	11.56
PAI (2025)	89.17	88.47	9.73	22.40	64.81	**18.11**
VCD (2024)	88.94	89.58	9.75	36.60	**67.60**	11.19
EAZY (2025)	89.39	89.36	6.56	25.60	63.18	11.42
SpanProbe (Ours)	**91.00**	**90.97**	**3.15**	**11.60**	61.10	11.47
Qwen2.5-VL-7B-Instruct						
Vanilla	81.67	77.91	7.42	18.80	**55.84**	12.96
Beam Search	80.61	76.28	4.83	14.80	47.01	12.36
PAI (2025)	85.42	83.36	6.03	12.00	39.94	**13.38**
VCD (2024)	83.64	80.86	7.93	21.60	54.94	11.42
EAZY (2025)	73.89	64.93	5.75	12.40	50.26	12.79
SpanProbe (Ours)	**87.20**	**85.73**	**4.29**	**6.40**	50.65	12.78

4.4 Ablation Studies

Component-Level Ablation. Starting from the SpanProbe model, we separately remove the Token Evidence Profile and token-adaptive pooling to evaluate the contribution of each component; detailed definitions are provided in Appendix A. As shown in Table 4, removing the TEP causes

a larger performance drop, indicating that it provides important discriminative information, while token-adaptive pooling further integrates complementary evidence across tokens within the span.

Table 4: Component ablations on M-HalDetect (AUROC/AUPRC, %).

Base Model	Setting	AUROC	AUPRC
LLaVA-v1.5	SpanProbe	**89.71**	**77.48**
	w/o Adaptive Pooling	87.50	72.75
	w/o Token Evidence Profile	86.60	71.59
LLaVA-v1.6	SpanProbe	**90.14**	**79.16**
	w/o Adaptive Pooling	88.70	76.89
	w/o Token Evidence Profile	88.40	74.56

Span Readout and Aggregation. Having established the overall contribution of adaptive pooling, we compare five span readout and aggregation strategies on M-HalDetect using LLaVA-v1.5. As shown in Table 5, complete-span aggregation generally outperforms fixed-endpoint readout. SpanProbe further outperforms predefined aggregation strategies, indicating that learning token-relative importance is more effective than fixed aggregation rules. Definitions of all five strategies, five-seed results, and paired statistical analyses are provided in Appendix B.

Table 5: Comparison of span readout and aggregation strategies on M-HalDetect.

Method	AUROC (%)↑	AUPRC (%)↑	Recall (%)↑	F1 (%)↑	Brier×100 ↓	R@P80 (%)↑
StartProbe	84.52	64.60	70.29	61.99	16.71	22.43
EndProbe	81.98	65.73	67.28	59.31	17.56	33.01
MeanPool	87.50	72.75	72.14	65.96	14.96	43.50
StatPool	88.18	74.79	74.56	66.84	14.47	51.07
SpanProbe	**89.71**	**77.48**	**83.30**	**67.22**	**14.19**	**56.99**

Token Evidence Profile. On two backbones, the complete TEP outperforms single-feature and feature-removal variants (Table 6), supporting the complementary contributions of visual-attention mass, full-attention entropy, and head-output norms.

Table 6: TEP feature ablation on M-HalDetect (AUPRC, %).

Configuration	LLaVA-v1.6	Qwen2.5-VL
Visual only	77.14	74.95
Entropy only	76.72	75.90
Norm only	76.71	76.40
w/o Visual mass	78.35	76.94
w/o Attention entropy	76.98	76.41
w/o Head-output norm	77.62	76.39
Full TEP	**79.16**	**77.38**

5 Conclusion

We propose SpanProbe, which assesses the hallucination risk of complete semantic spans by adaptively aggregating token-level evidence and uses the resulting risk to guide local rollback and regeneration. Extensive experiments across multiple benchmarks demonstrate its effectiveness in detecting hallucinations in complete semantic spans and performing selective intervention. More broadly, this study reveals that hallucination-discriminative evidence is distributed non-uniformly across tokens, offering a new research direction for improving hallucination detection and mitigation through adaptive aggregation of internal evidence.

AI USE STATEMENT

During manuscript preparation, the authors used generative AI solely to assist with English language editing. The assistance was limited to suggestions concerning grammar, phrasing, and clarity of expression. The authors reviewed the resulting text, revised suggestions where necessary to preserve the intended scientific meaning, and made all final decisions about the wording. The authors take full responsibility for the accuracy and content of the final manuscript.

REFERENCES

Wenbin An, Feng Tian, Sicong Leng, Jiahao Nie, Haonan Lin, Qianying Wang, Ping Chen, Xiaoqin Zhang, and Shijian Lu. Mitigating object hallucinations in large vision-language models with assembly of global and local attention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 29915–29926, June 2025.

Peter Anderson, Basura Fernando, Mark Johnson, and Stephen Gould. SPICE: Semantic propositional image caption evaluation. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling (eds.), *Computer Vision – ECCV 2016*, pp. 382–398, 2016.

Ofir Azachi, Kfir Eliyahu, Eyal El Ani, Rom Himelstein, Roi Reichart, Yuval Pinter, and Nitay Calderon. Leveraging NTPs for efficient hallucination detection in VLMs. In Aman Sinha, Raúl Vázquez, Timothee Mickus, Rohit Agarwal, Ioana Buhnila, Patrícia Schmidtová, Federica Gamba, Dilip K. Prasad, and Jörg Tiedemann (eds.), *Proceedings of the 1st Workshop on Confabulation, Hallucinations and Overgeneration in Multilingual and Practical Settings (CHOMPS 2025)*, December 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, 2025a. URL https://arxiv.org/abs/2502.13923.

Zechen Bai, Pichao Wang, Tianjun Xiao, Tong He, Zongbo Han, Zheng Zhang, and Mike Zheng Shou. Hallucination of multimodal large language models: A survey, 2025b. URL https://arxiv.org/abs/2404.18930.

Liwei Che, Tony Qingze Liu, Jing Jia, Weiyi Qin, Ruixiang Tang, and Vladimir Pavlovic. Hallucinatory image tokens: A training-free EAZY approach to detecting and mitigating object hallucinations in LVLMs. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 21635–21644, October 2025.

Xiang Chen, Chenxi Wang, Yida Xue, Ningyu Zhang, Xiaoyan Yang, Qiang Li, Yue Shen, Lei Liang, Jinjie Gu, and Huajun Chen. Unified hallucination detection for multimodal large language models. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3235–3252, August 2024a.

Zhaorun Chen, Zhuokai Zhao, Hongyin Luo, Huaxiu Yao, Bo Li, and Jiawei Zhou. HALC: Object hallucination reduction via adaptive focal-contrast decoding. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, pp. 7824–7846, 21–27 Jul 2024b.

Wenliang Dai, Junnan Li, DONGXU LI, Anthony Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale N Fung, and Steven Hoi. InstructBLIP: Towards general-purpose vision-language models with instruction tuning. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, pp. 49250–49267, 2023.

Laura Fieback, Jakob Spiegelberg, and Hanno Gottschalk. MetaToken: Detecting hallucination in image descriptions by meta classification. In *Proceedings of the 20th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications - Volume 2: VISAPP*, pp. 126–137, 2025.

Tianrui Guan, Fuxiao Liu, Xiyang Wu, Ruiqi Xian, Zongxia Li, Xiaoyu Liu, Xijun Wang, Lichang Chen, Furong Huang, Yaser Yacoob, Dinesh Manocha, and Tianyi Zhou. HallusionBench: An advanced diagnostic suite for entangled language hallucination and visual illusion in large vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14375–14385, June 2024.

Anisha Gunjal, Jihan Yin, and Erhan Bas. Detecting and preventing hallucinations in large vision language models. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(16):18135–18143, 2024. doi: 10.1609/aaai.v38i16.29771.

Qidong Huang, Xiaoyi Dong, Pan Zhang, Bin Wang, Conghui He, Jiaqi Wang, Dahua Lin, Weiming Zhang, and Nenghai Yu. OPERA: Alleviating hallucination in multi-modal large language models via over-trust penalty and retrospection-allocation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13418–13427, June 2024.

Fushuo Huo, Wenchao Xu, Zhong Zhang, Haozhao Wang, Zhicheng Chen, and Peilin Zhao. Self-introspective decoding: Alleviating hallucinations for large vision-language models. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), *International Conference on Learning Representations*, volume 2025, pp. 24272–24295, 2025.

Nick Jiang, Anish Kachinthaya, Suzanne Petryk, and Yossi Gandelsman. Interpreting and editing vision-language representations to mitigate hallucinations. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), *International Conference on Learning Representations*, volume 2025, pp. 63582–63605, 2025a.

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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 25004–25014, June 2025b.

Liqiang Jing, Ruosen Li, Yunmo Chen, and Xinya Du. FaithScore: Fine-grained evaluations of hallucinations in large vision-language models. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 5042–5063, November 2024.

Sai Akhil Kogilathota, Sripadha Vallabha E G, Luzhe Sun, and Jiawei Zhou. HALP: Detecting hallucinations in vision-language models without generating a single token. In Vera Demberg, Kentaro Inui, and Lluís Marquez (eds.), *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 6067–6085, March 2026.

Seongyun Lee, Sue Hyun Park, Yongrae Jo, and Minjoon Seo. Volcano: Mitigating multimodal hallucination through self-feedback guided revision. In Kevin Duh, Helena Gomez, and Steven Bethard (eds.), *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 391–404, June 2024.

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 (CVPR)*, pp. 13872–13882, June 2024.

Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Xin Zhao, and Ji-Rong Wen. Evaluating object hallucination in large vision-language models. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 292–305, December 2023.

Zhuowei Li, Haizhou Shi, Yunhe Gao, Di Liu, Zhenting Wang, Yuxiao Chen, Ting Liu, Long Zhao, Hao Wang, and Dimitris N. Metaxas. The hidden life of tokens: Reducing hallucination of large vision-language models via visual information steering. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 35799–35819, 13–19 Jul 2025.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: Common objects in context. In David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars (eds.), *Computer Vision – ECCV 2014*, pp. 740–755. Springer International Publishing, 2014.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 26296–26306, June 2024a.

Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee. LLaVA-NeXT: Improved reasoning, OCR, and world knowledge, January 2024b. URL https://llava-vl.github.io/blog/2024-01-30-llava-next/.

Shi Liu, Kecheng Zheng, and Wei Chen. Paying more attention to image: A training-free method for alleviating hallucination in LVLMs. In *Computer Vision – ECCV 2024*, pp. 125–140, 2025.

Seongheon Park and Sharon Li. GLSim: Detecting object hallucinations in LVLMs via global-local similarity. In *Advances in Neural Information Processing Systems*, pp. 37747–37776, 2025.

Suzanne Petryk, David M. Chan, Anish Kachinthaya, Haodi Zou, John Canny, Joseph E. Gonzalez, and Trevor Darrell. ALOHa: A new measure for hallucination in captioning models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers)*, pp. 342–357, June 2024.

Anirudh Phukan, Divyansh, Harshit Kumar Morj, Vaishnavi, Apoorv Saxena, and Koustava Goswami. Beyond logit lens: Contextual embeddings for robust hallucination detection & grounding in VLMs. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 9661–9675, April 2025.

Haoyi Qiu, Wenbo Hu, Zi-Yi Dou, and Nanyun Peng. VALOR-EVAL: Holistic coverage and faithfulness evaluation of large vision-language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 1783–1805, August 2024.

Anna Rohrbach, Lisa Anne Hendricks, Kaylee Burns, Trevor Darrell, and Kate Saenko. Object hallucination in image captioning. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 4035–4045, October-November 2018.

Pritish Sahu, Karan Sikka, and Ajay Divakaran. Pelican: Correcting hallucination in vision-LLMs via claim decomposition and program of thought verification. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 8228–8248, November 2024.

Junyang Wang, Yuhang Wang, Guohai Xu, Jing Zhang, Yukai Gu, Haitao Jia, Jiaqi Wang, Haiyang Xu, Ming Yan, Ji Zhang, and Jitao Sang. AMBER: An LLM-free multi-dimensional benchmark for MLLMs hallucination evaluation, 2024a. URL https://arxiv.org/abs/2311.07397.

Xintong Wang, Jingheng Pan, Liang Ding, and Chris Biemann. Mitigating hallucinations in large vision-language models with instruction contrastive decoding. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 15840–15853, August 2024b.

Jinfeng Wei and Xiaofeng Zhang. DOPRA: Decoding over-accumulation penalization and reallocation in specific weighting layer. In *Proceedings of the 32nd ACM International Conference on Multimedia*, 2024. doi: 10.1145/3664647.3681076.

Sangmin Woo, Jaehyuk Jang, Donguk Kim, Yubin Choi, and Changick Kim. RITUAL: Random image transformations as a universal anti-hallucination lever in large vision language models, 2024. URL https://arxiv.org/abs/2405.17821.

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

Junfei Wu, Qiang Liu, Ding Wang, Jinghao Zhang, Shu Wu, Liang Wang, and Tieniu Tan. Logical closed loop: Uncovering object hallucinations in large vision-language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 6944–6962, August 2024.

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

Feiran Zhang, Yixin Wu, Zhenghua Wang, Xiaohua Wang, Changze Lv, Xuanjing Huang, and Xiaoqing Zheng. VIB-Probe: Detecting and mitigating hallucinations in vision-language models via variational information bottleneck. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 23509–23521, July 2026.

Yudong Zhang, Ruobing Xie, Xingwu Sun, Yiqing Huang, Jiansheng Chen, Zhanhui Kang, Di Wang, and Yu Wang. DHCP: Detecting hallucinations by cross-modal attention pattern in large vision-language models. In *Proceedings of the 33rd ACM International Conference on Multimedia*, pp. 3555–3564, 2025. doi: 10.1145/3746027.3755118.

Yiyang Zhou, Chenhang Cui, Jaehong Yoon, Linjun Zhang, Zhun Deng, Chelsea Finn, Mohit Bansal, and Huaxiu Yao. Analyzing and mitigating object hallucination in large vision-language models. In *International Conference on Learning Representations*, volume 2024, pp. 56969–56998, 2024.

Lanyun Zhu, Deyi Ji, Tianrun Chen, Peng Xu, Jieping Ye, and Jun Liu. IBD: Alleviating hallucinations in large vision-language models via image-biased decoding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pp. 1624–1633, June 2025.

Xianwei Zhuang, Zhihong Zhu, Yuxin Xie, Liming Liang, and Yuexian Zou. VASparse: Towards efficient visual hallucination mitigation via visual-aware token sparsification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4189–4199, June 2025.

Reihaneh Zohrabi, Hosein Hasani, Akshita Gupta, Mahdieh Soleymani Baghshah, Anna Rohrbach, and Marcus Rohrbach. HaloProbe: Bayesian detection and mitigation of object hallucinations in vision-language models, 2026. URL `https://arxiv.org/abs/2604.06165`.

A Experimental and Implementation Details

A.1 Detection Benchmarks and Candidate Construction

POPE. We split POPE's 500 images by image identifier into 400 training images and 100 held-out test images, keeping all questions associated with the same image in one partition. For MetaToken, DHCP, VIB-Probe, and SpanProbe, we evaluate the fixed candidate responses "Yes" and "No" separately for each question; a candidate is labeled hallucinated if it disagrees with the ground-truth answer and grounded otherwise. Under the tokenization used by the evaluated backbones, each candidate answer contains a single subtoken, so SpanProbe's token-adaptive pooling reduces to the identity mapping. For DHCP, each candidate feature averages the visual-attention tensors from two decoding steps: the prompt-end state used to predict the candidate token and the candidate-conditioned state used to predict EOS. The latter therefore differs between the "Yes" and "No" candidates. Results are computed separately on the Random, Popular, and Adversarial subsets and macro-averaged across them.

M-HalDetect. We use all 3,164 records in the released M-HalDetect validation set. The 786 unique images are partitioned by image identifier into 628 training images and 158 held-out images using random seed 20260805, with no image overlap. The resulting partitions contain 17,726 and 4,366 annotated spans, respectively; the held-out set contains 1,030 hallucinated and 3,336 grounded spans. We map INACCURATE and UNSURE to the hallucinated class and ACCURATE and ANALYSIS to the grounded class. Candidate boundaries are taken from the released character intervals and mapped to all tokenizer tokens that overlap each interval. We use the released model responses as fixed inputs to extract backbone-specific internal signals. The released annotations mark variable-length segments within detailed image descriptions.

MSCOCO. For open-ended detection, we use the MSCOCO 2014 validation images together with the instance and caption annotations required by CHAIR. We exclude the fixed 500-image set reserved for mitigation and sample 2,000 images from the remaining validation images using random seed 20260806. The image-disjoint split contains 1,600 training images and 400 held-out images, corresponding to 8,164 and 2,067 sentence units, respectively. Captions are generated with the prompt "Please describe this image in detail." and then used as fixed inputs for feature extraction. Each sentence is treated as one evaluation unit and is labeled hallucinated if it contains at least one hallucinated object according to the COCO annotations, and grounded otherwise. Table 1 reports sentence-level results only.

A.2 SpanProbe Implementation and Training

For each response token, we construct the Token Evidence Profile (TEP) from three features for every decoder-layer–attention-head pair: visual-attention mass, full-attention entropy computed with the natural logarithm, and the ℓ_2 norm of the individual head output before $\texttt{o_proj}$. The features are concatenated in group-major order: all visual-attention features, followed by all entropy features and all head-output-norm features; within each group, entries are ordered first by layer and then by head. For a backbone with L decoder layers and H query heads, the resulting TEP dimension is $3LH$. The 3,072-dimensional TEP used for the 32-layer, 32-head backbones is therefore a backbone-specific instance rather than a fixed architectural requirement of SpanProbe. For example, the 28-layer, 28-head Qwen2.5-VL backbone produces a 2,352-dimensional TEP.

Before entropy computation, attention weights are cast to FP32, clamped to nonnegative values, and renormalized over the causally accessible key positions. We use $\epsilon = 10^{-12}$.

Feature-wise means and standard deviations are estimated exclusively from token rows in the training partition and are fixed for held-out evaluation. The standardized TEP of every token is mapped by a shared linear projection to 128 dimensions followed by a $\tanh$ nonlinearity. A shared linear scorer produces one scalar per token, and a masked softmax over the current span converts these scores into relative token weights. The weighted sum of the 128-dimensional token representations is passed to a linear span-level risk head.

We train a separate detector for each backbone–dataset pair while keeping the MLLM frozen. Only the shared projection, token-importance scorer, and risk head are optimized. We use AdamW with

a learning rate of 10^{-3}, weight decay of 10^{-4}, and a batch size of 128. The weighted binary cross-entropy loss uses N_-/N_+ as the positive-class weight, where N_- and N_+ are the numbers of grounded and hallucinated training spans. Training runs for at most 30 epochs and stops after five consecutive epochs without improvement in inner-validation AUPRC. For the final fit on the complete training partition, the number of epochs is the median of the best epochs selected across five image-grouped training folds. The binary decision threshold is selected by maximizing F1 on pooled out-of-fold predictions from the training partition, with ties resolved in favor of the larger threshold. Held-out labels are never used for normalization, model selection, epoch selection, or threshold selection.

Core-component ablations. To isolate the contributions of the Token Evidence Profile and token-adaptive pooling, we construct two single-component ablations. In "− Adaptive Pooling," we retain the complete TEP, feature-wise standardization, the shared 128-dimensional projection, and the risk head, but remove the token-importance scorer and replace the span representation with uniform averaging over the projected token representations, $z_s = |s|^{-1} \sum_{t \in s} e_t$. In "− Token Evidence Profile," we replace each TEP with the aligned token hidden state $h_t^{(L)}$ from the final decoder layer of the frozen MLLM. The hidden state is standardized feature-wise using statistics estimated on the training partition and projected to 128 dimensions; the token-importance scorer, within-span softmax pooling, and risk head are otherwise unchanged. All three configurations use identical data partitions, training hyperparameters, model-selection criteria, and evaluation protocols.

A.3 Evaluation Metrics and Baseline Configuration

We report AUROC and AUPRC for hallucination detection and use AUPRC as the primary metric on class-imbalanced generative benchmarks. POPE results are macro-averaged across its three subsets. For the diagnostic span-readout experiment, we additionally report recall, F1, Brier score, and recall at 80% precision (R@P80).

All metrics treat hallucination as the positive class. We compute AUROC using `roc_auc_score` and average precision using `average_precision_score`; following common practice, we report the latter as AUPRC. Precision, recall, and F1 are computed after applying the threshold frozen from training-only out-of-fold predictions. The Brier score is computed on the original probability scale and multiplied by 100 only for presentation in Table 5. Expected calibration error uses ten equal-width bins. R@P80 is the maximum recall among operating points whose precision is at least 0.80.

MetaToken is evaluated using a matched reimplementation with its 42-dimensional highest-layer feature representation, feature standardization, and a gradient-boosting classifier. DHCP aggregates cross-modal attention representations over candidate-aligned decoding steps before applying its MLP classifier. VIB-Probe uses attention-head outputs collected across decoder layers and a 256-dimensional variational latent representation. For mitigation, the matched Beam Search baseline uses five beams, deterministic decoding, unit length penalty, and no top-k or top-p filtering. PAI, VCD, and EAZY follow their public implementations, with all decoding parameters fixed before evaluation.

Candidate-level adaptation for multi-token spans. Within each backbone–dataset pair, all detectors use the same frozen model responses, labels, data partitions, and final evaluation units, while internal evidence extraction and readout follow each method's original design or a predetermined adaptation rule. For methods without released code, we use matched reimplementations based on the method descriptions and retrain them on the same partitions. For a candidate span $s = [a, b)$, VIB-Probe reads the attention-head output representation at the final span token and produces a span risk with its probe. For multi-token candidates, DHCP averages its cross-modal attention representations over the decoding steps corresponding to all tokens in the span before applying its detector. The matched MetaToken uses attention and predictive-uncertainty statistics at the span start together with sequence statistics, including the summed token log-probability over the complete span. SpanProbe instead constructs a TEP for every token in the span and learns their relative weights. These rules are fixed before evaluation and applied consistently across backbones; token indices follow each backbone's tokenizer and are never selected using held-out results.

On M-HalDetect, the resulting score directly corresponds to an annotated span. On MSCOCO, all methods are evaluated on the same sentence units and sentence-level labels.

A.4 Risk-Gated Rollback and Regeneration

Mitigation uses a frozen SpanProbe detector and performs no parameter updates during generation. For MSCOCO mitigation, SpanProbe is trained and calibrated on object-mention spans constructed from the training partition. Each mention is labeled as hallucinated or grounded according to the COCO annotations, and the intervention threshold is selected from mention-level training-only out-of-fold predictions by maximizing recall subject to precision being at least 0.80. For POPE, the threshold is selected from training-only out-of-fold predictions by maximizing F1. Neither task accesses ground-truth hallucination labels while generating responses.

Independent evaluation of the mitigation detector. Using the same image split described above, the mention-level SpanProbe used for MSCOCO mitigation is trained on 9,603 object mentions constructed from the 1,600 training images, of which 768 (8.00%) are hallucinated. We independently evaluate the checkpoint used for mitigation on 2,597 object mentions from 400 held-out images that overlap with neither the training images nor the fixed 500-image mitigation evaluation set; 223 mentions (8.59%) are hallucinated. The detector achieves 90.44% AUROC and 62.09% AUPRC. Neither this held-out set nor the mitigation evaluation set is used for detector training, model selection, or threshold selection.

MSCOCO captioning. A task-specific span constructor identifies completed object mentions from the visible character stream produced by incremental subtoken decoding. It follows the CHAIR object-mention matching procedure but does not read the image's ground-truth object labels. Once an object mention is completed, SpanProbe evaluates the cached TEP rows of that candidate. If its risk exceeds the threshold, generation rolls back to the first token of the candidate. The generated token sequence, KV cache, cached TEP rows, span-constructor state, next-token logits, and logit history are restored to the state at the rollback point; the candidate span and any lookahead tokens used to confirm its boundary are discarded.

For a rejected span $s = [a, b)$, its original token sequence $\mathbf{y}_s^{\mathrm{rej}} = (y_a, \ldots, y_{b-1})$ serves as the constraint during regeneration; its constituent subtokens are not masked independently. Regeneration is greedy with a maximum of 512 new tokens. Whenever the highest-logit token would complete the rejected sequence, decoding selects the highest-ranked legal token instead. A regenerated mention is evaluated again by the same frozen detector. At most three rollbacks are permitted per caption, and the same rejected sequence is not repeatedly triggered within the current sentence. Sentence-local constraints are cleared at sentence-ending punctuation or generation termination; when the rollback budget is exhausted, generation retains the current continuation.

POPE question answering. The candidate is the first semantically valid "Yes" or "No" answer token in the completed response. A high-risk candidate triggers one rollback to the answer start, restoration of the prompt-side cache and preceding response tokens, and greedy regeneration while prohibiting the rejected answer's exact token sequence. The regenerated POPE answer is accepted without a second risk-gated rollback. Thus, POPE uses a single-token, single-revision procedure, whereas MSCOCO uses object-mention candidates, exact rejected-sequence constraints, and repeated detection within a budget.

For MSCOCO, CHAIR_i is the fraction of generated object mentions that are hallucinated, and CHAIR_s is the fraction of captions containing at least one hallucinated object mention. We additionally report object recall, micro-averaged over object categories across the complete test set:

$$\mathrm{Obj.Recall} = \frac{\sum_i |\widehat{G}_i \cap G_i|}{\sum_i |G_i|},$$

where G_i is the set of ground-truth object categories for image i and $\widehat{G}_i$ is the set mentioned in its generated caption under the standard CHAIR evaluation procedure. The resulting ratio is reported as a percentage. SPICE is computed against the reference captions on all 500 mitigation images and multiplied by 100 for presentation.

Table 7: Auxiliary MSCOCO mitigation statistics. Avg. Length is the mean decoded length under the evaluated backbone's tokenizer, and Avg. Objects is the mean number of object mentions identified by the span constructor per caption. Lengths are comparable within, but not across, backbone tokenizers.

Backbone	Method	Avg. Length	Avg. Objects
LLaVA-v1.6	Vanilla	220.67	5.81
	Beam Search	232.04	5.95
	PAI	112.01	5.57
	VCD	226.66	6.09
	EAZY	222.64	5.70
	SpanProbe (Ours)	222.00	5.28
Qwen2.5-VL	Vanilla	219.44	5.26
	Beam Search	308.46	5.22
	PAI	78.12	2.85
	VCD	210.70	4.79
	EAZY	219.14	4.66
	SpanProbe (Ours)	218.44	4.57

Inference efficiency. We evaluate the inference overhead of the optimized implementation of the complete SpanProbe framework on the 500-image MSCOCO mitigation test set with LLaVA-v1.6-Mistral-7B. The images are distributed across six NVIDIA RTX 4090 GPUs (24 GiB each), with each GPU processing its assigned images independently at a per-GPU batch size of 1. Experiments use PyTorch 2.10.0, Transformers 4.37.2, and CUDA 12.8. Vanilla and SpanProbe both use FP16 inference, eager attention, and KV caching; the model, detector, and generation configurations otherwise match the corresponding LLaVA-v1.6 experiment in Table 3.

SpanProbe computes the TEP statistics for all decoder layers and attention heads online in FP32 on the GPU. It does not persistently cache the full attention matrices; the only additional per-token state is the 3,072-dimensional TEP, which is truncated in sync with the generation state upon rollback. Candidate spans are constructed through incremental token-fragment matching, avoiding repeated decoding of the complete prefix.

Each image is run once per method, without explicit warm-up. We measure wall-clock time with `perf_counter` and synchronize the corresponding GPU immediately before and after each timed run. Timing covers image preprocessing, prefill, autoregressive decoding, and, for SpanProbe, detection, rollback, and regeneration; model loading is excluded. Latency is aggregated over the 500 per-image runtimes and reported as the mean. Effective output throughput is the total number of finally retained tokens divided by the sum of the per-image runtimes. Peak memory is the maximum allocated GPU memory recorded after resetting the CUDA peak-memory counter before each experimental condition.

Table 8: End-to-end inference efficiency on LLaVA-v1.6-Mistral-7B. Latency is averaged over 500 images.

Method	Effective Output Throughput ↑ (token/s)	Latency ↓ (s/image)	Peak Memory ↓ (GiB)	Avg. Rollbacks
Vanilla	34.15	6.42	17.19	0
SpanProbe	30.77	7.21	17.19	0.574

As shown in Table 8, under the same eager-attention configuration, SpanProbe increases mean end-to-end latency from 6.42 to 7.21 seconds, an increase of 12.3%, and reduces effective output throughput by 9.9%, while peak allocated memory is unchanged at the reported precision. The two configurations produce similar mean final output lengths (220.67 and 222.00 tokens, respectively; Table 7), indicating that the latency difference is not primarily caused by longer retained outputs. Across the 500 images, SpanProbe triggers 287 rollbacks, with 186 images (37.2%) triggering at least one; at most three rollbacks are allowed per image.

Together with the hallucination-mitigation results in Table 3, these measurements show that SpanProbe reduces object hallucinations with a modest inference-time overhead under the optimized implementation, reflecting a trade-off between generation reliability and inference efficiency.

B Additional Controlled Comparisons

Risk Detector Comparison under a Shared Rollback Policy. To distinguish the contribution of the risk detector from the gains shared across local rollback-and-regeneration configurations, we compare StartProbe, EndProbe, StatPool, MeanPool, MetaToken, VIB-Probe, and SpanProbe under a common rollback framework on LLaVA-v1.5-7B and Qwen2.5-VL-7B-Instruct. All experiments use the same fixed set of 500 MSCOCO images.

Within each backbone, all detector-guided configurations use identical candidate-span construction, rejected-sequence constraint rules, rollback budget, decoding strategy, and CHAIR evaluation procedure. Each detector only assigns a risk score to a completed candidate span; the subsequent rollback and regeneration process remains unchanged. All detector thresholds are determined using training data only and frozen before test evaluation. For online use, MetaToken relies only on information available when the candidate span has been completed, while VIB-Probe estimates risk from the internal representation at the final token of the span. Vanilla decoding performs neither risk detection nor rollback and serves as the non-intervention reference.

In addition to CHAIR_i and CHAIR_s, we report the mean number of rollbacks triggered per image and object recall. The former measures how frequently each detector intervenes in generation, while the latter assesses whether reducing hallucinations comes at the cost of omitting objects that are present in the image. Rollbacks/image is an observed test outcome under each detector's frozen threshold rather than an explicitly matched constraint; we therefore pay particular attention to comparisons with similar intervention frequencies.

Table 9: Risk-detector comparison under a shared rollback policy on LLaVA-v1.5-7B and Qwen2.5-VL-7B-Instruct. CHAIR_i, CHAIR_s, and Obj. Recall are percentages. Rollbacks/image is the mean number of rollbacks triggered per image.

Configuration	Rollbacks/image	$\mathrm{CHAIR}_i \downarrow$	$\mathrm{CHAIR}_s \downarrow$	Obj. Recall↑
LLaVA-v1.5-7B				
Vanilla, without rollback	0.000	15.31	54.00	**79.29**
StartProbe + rollback	0.460	11.88	44.80	78.31
EndProbe + rollback	0.640	11.58	43.00	77.60
StatPool + rollback	0.590	11.52	41.60	77.60
MeanPool + rollback	0.638	11.56	41.60	77.92
MetaToken + rollback	0.610	13.91	49.60	78.83
VIB-Probe + rollback	0.674	11.56	42.80	76.30
SpanProbe	0.612	**10.67**	**37.80**	78.44
Qwen2.5-VL-7B-Instruct				
Vanilla, without rollback	0.000	7.42	18.80	**55.84**
StartProbe + rollback	0.564	6.01	10.40	49.38
EndProbe + rollback	0.514	5.07	10.20	49.77
StatPool + rollback	0.558	4.86	9.80	48.86
MeanPool + rollback	0.484	6.14	11.00	50.96
MetaToken + rollback	0.582	5.39	14.40	51.32
VIB-Probe + rollback	1.644	7.48	10.20	41.23
SpanProbe	0.552	**4.29**	**6.40**	50.65

On LLaVA-v1.5-7B, every detector-guided configuration reduces object hallucinations relative to Vanilla, but SpanProbe achieves the lowest CHAIR_i and CHAIR_s. Relative to MeanPool, it reduces the two metrics by 0.89 and 3.80 percentage points, respectively, while increasing object recall by 0.52 points and using slightly fewer rollbacks. Relative to VIB-Probe, it reduces CHAIR_i and CHAIR_s by 0.89 and 5.00 points, increases object recall by 2.14 points, and also intervenes less frequently. These comparisons indicate that the gain from token-adaptive pooling does not arise from more frequent intervention.

On Qwen2.5-VL-7B-Instruct, SpanProbe again obtains the lowest CHAIR_i and CHAIR_s. Excluding VIB-Probe, the detector-guided configurations trigger 0.484–0.582 rollbacks per image. SpanProbe and StatPool have nearly identical intervention frequencies, yet SpanProbe lowers CHAIR_i and CHAIR_s by a further 0.57 and 3.40 points and increases object recall by 1.79 points. Compared with EndProbe, it likewise reduces the two CHAIR metrics by 0.78 and 3.80 points while retaining higher object recall. Thus, its advantage on Qwen2.5-VL cannot be explained by a higher intervention rate.

Overall, SpanProbe attains the lowest object-hallucination rates on both backbones at moderate intervention frequencies while maintaining competitive object recall among detector-guided methods. The shared-policy comparison therefore supports the effectiveness of token-adaptive span-level risk estimation in balancing hallucination reduction, content preservation, and intervention frequency.

Cross-study context for VIB-Probe mitigation. Table 10 summarizes the mitigation results reported by VIB-Probe (Zhang et al., 2026) together with our results on the corresponding backbones. Each method is paired with the Vanilla result reported by the same source. Because the two studies do not use aligned 500-image MSCOCO test sets or fully matched decoding configurations, these values are provided only as cross-study context and are not used to establish a direct performance ranking.

Table 10: Published VIB-Probe mitigation results and our corresponding results on MSCOCO. † denotes values reported in Table 3 of VIB-Probe. CHAIR_i and CHAIR_s are percentages; lower is better.

Backbone	Method	$\mathrm{CHAIR}_i \downarrow$	$\mathrm{CHAIR}_s \downarrow$
LLaVA-v1.5-7B	Vanilla†	18.2	59.3
	VIB-Probe†	14.1	44.9
	Vanilla	15.31	54.00
	SpanProbe	10.67	37.80
LLaVA-v1.6-Mistral-7B	Vanilla†	11.8	40.7
	VIB-Probe†	8.7	32.1
	Vanilla	8.12	29.40
	SpanProbe	3.15	11.60

Both studies show that detector-derived signals can support inference-time hallucination mitigation. The controlled within-study comparison of risk detectors under a shared rollback policy is reported in Table 9.

Statistical Protocol for Multi-Seed Comparisons. Both controlled comparisons below treat hallucination as the positive class. We repeat every configuration with five training seeds and report the mean and sample standard deviation of AUROC and AUPRC. For each baseline, ΔAUPRC is defined as SpanProbe minus the baseline in percentage points. We compute 95% confidence intervals from 2,000 paired image-level percentile bootstrap resamples of the 158-image M-HalDetect test set, retaining all spans associated with each sampled image and using identical resampled image indices for all compared methods. Within each bootstrap replicate, we compute the paired AUPRC difference for each seed and average the five differences. The seeds themselves are held fixed, so these intervals quantify test-image sampling uncertainty conditional on the five trained checkpoints. The test set contains 4,366 spans, including 1,030 hallucinated spans. Table 5 reports the run obtained with the prespecified reference seed to retain the complete set of threshold-dependent metrics; its values therefore differ slightly from the five-seed summaries below.

Representation and Aggregation Controls. To further distinguish the contributions of the representation and adaptive aggregation, we conduct a controlled comparison on M-HalDetect using LLaVA-v1.5. All configurations use the same data partitions, fixed model responses, candidate spans, and hallucination labels, while the underlying MLLM remains frozen. The original VIB-Probe reads the VIB representation at the final token of each span. VIB + Adaptive Pooling instead constructs VIB representations for all tokens in the span and aggregates them adaptively; its VIB

encoder and downstream pooling and classification modules are jointly optimized with span-level supervision. This adaptation is used only in the controlled comparison reported in Table 11; all other VIB-Probe results in this paper retain the original endpoint-readout architecture. SpanProbe w/o Adaptive Pooling retains the complete TEP, shared projection, and risk head but replaces token-adaptive pooling with uniform averaging. It is therefore identical to MeanPool in Table 12, and the two rows use the same experimental runs. The complete SpanProbe combines TEPs with token-adaptive pooling.

Table 11: Representation and aggregation controls on M-HalDetect using LLaVA-v1.5. AUROC and AUPRC are percentages reported as mean ± sample standard deviation over five training seeds. ΔAUPRC denotes the improvement of SpanProbe over each baseline in percentage points; brackets give 95% paired bootstrap confidence intervals.

Method	AUROC↑	AUPRC↑	ΔAUPRC [95% CI]
VIB-Probe	83.17 ± 0.45	69.76 ± 0.39	+7.36 $[4.61, 10.14]$
VIB + Adaptive Pooling	86.80 ± 1.04	72.34 ± 1.72	+4.78 $[2.75, 6.78]$
SpanProbe w/o Adaptive Pooling	87.50 ± 0.06	72.60 ± 0.13	+4.52 $[2.89, 6.19]$
SpanProbe	$\mathbf{89.37 \pm 0.22}$	$\mathbf{77.12 \pm 0.41}$	—

As shown in Table 11, adding adaptive pooling to VIB-Probe increases mean AUPRC from 69.76 to 72.34, showing that adaptive span aggregation also improves detection with VIB representations. The complete SpanProbe further exceeds this variant by 4.78 AUPRC points, with a 95% confidence interval of $[2.75, 6.78]$. With the TEP representation and all remaining downstream components fixed, adaptive pooling improves mean AUPRC over uniform averaging by 4.52 points, with a confidence interval of $[2.89, 6.19]$. These results support complementary contributions from the TEP representation and adaptive span aggregation.

Definitions of Span Readout and Aggregation Strategies. The five strategies in Table 5 use the same candidate spans and frozen LLaVA-v1.5 backbone on M-HalDetect. StartProbe and EndProbe read from the first and final token of each span, respectively. MeanPool uniformly averages token representations across the complete span. StatPool concatenates five summary vectors: the final-token representation (End), the elementwise mean (Mean), maximum (Max), and standard deviation (Std) across the span, and the difference between the final and first token representations (End−Start). SpanProbe instead learns token-importance weights from the TEPs and uses them to aggregate token representations, as defined in Section 3.

Multi-Seed Validation of Span Readout and Aggregation. To test whether the span readout and aggregation results in Table 5 are stable across training initializations, we repeat the five strategies with five training seeds under the same data split and evaluation protocol. Table 5 retains the prespecified reference run to provide the complete comparison, including threshold-dependent metrics. Table 12 reports the five-seed summaries of AUROC and AUPRC together with paired AUPRC differences between SpanProbe and each baseline.

Table 12: Five-seed comparison of span readout and aggregation strategies on M-HalDetect using LLaVA-v1.5. AUROC and AUPRC are percentages reported as mean ± sample standard deviation. ΔAUPRC denotes the improvement of SpanProbe over each baseline in percentage points; brackets give 95% paired bootstrap confidence intervals.

Method	AUROC↑	AUPRC↑	ΔAUPRC [95% CI]
StartProbe	84.60 ± 0.06	64.89 ± 0.25	+12.23 $[9.65, 14.60]$
EndProbe	82.07 ± 0.09	65.81 ± 0.07	+11.31 $[8.00, 14.82]$
MeanPool	87.50 ± 0.06	72.60 ± 0.13	+4.52 $[2.89, 6.19]$
StatPool	88.21 ± 0.06	74.85 ± 0.12	+2.27 $[1.09, 4.29]$
SpanProbe	$\mathbf{89.37 \pm 0.22}$	$\mathbf{77.12 \pm 0.41}$	—

As shown in Table 12, complete-span aggregation achieves higher mean AUROC and AUPRC than fixed-endpoint readout. SpanProbe obtains the strongest mean detection performance, improving AUPRC over MeanPool and StatPool by 4.52 and 2.27 percentage points, respectively; neither 95% confidence interval contains zero. The gain over StatPool is smaller, but the overall pattern agrees with the qualitative conclusion of Table 5 and further supports learning token-relative importance within a span.

C Bidirectional Cross-Dataset and Cross-Granularity Zero-Shot Transfer

C.1 Experimental Setup

To examine whether SpanProbe learns hallucination evidence that transfers across benchmark construction and task formulation, rather than depending solely on dataset-specific lexical, positional, or label-distribution cues, we conduct bidirectional cross-dataset and cross-task zero-shot transfer experiments.

M-HalDetect → MSCOCO. Detectors are trained on the human-annotated multi-token semantic spans in the M-HalDetect training split and then applied directly to the MSCOCO sentence-level hallucination detection test set. The source units are annotated semantic spans, whereas the target units are complete generated sentences; the benchmark, generated text, and detection granularity therefore change simultaneously. To prevent image-content leakage, we remove from the target test set three images whose COCO identifiers overlap with the source training split, together with their 16 grounded sentences. The deduplicated target test set contains 397 images and 2,051 sentence-level evaluation units, including 383 hallucinated positives and 1,668 grounded negatives.

MSCOCO → M-HalDetect. Detectors are trained on the sentence-level hallucination detection units in the MSCOCO training split and then applied directly to the human-annotated spans in M-HalDetect. Using the same COCO-identifier rule, we remove eight target images that overlap with the source training images. The deduplicated M-HalDetect test set contains 150 images, 604 complete responses, and 4,158 annotated spans, of which 985 are hallucinated positives. Annotated spans are the actual evaluation units in this direction; the 604 complete responses serve only as the fixed generated texts containing those spans and are not sentence-level evaluation units.

All experiments use a frozen LLaVA-v1.5-7B backbone. For each direction, the detectors are trained only on the training split of the corresponding source dataset. During transfer evaluation, no target-dataset labels are used for parameter updates, model selection, threshold calibration, or hyperparameter tuning. Detector parameters and the means and standard deviations used for feature standardization are estimated only from the source training split and remain frozen during target-dataset evaluation. Both directions therefore constitute strict zero-shot transfer settings.

We compare VIB-Probe, MeanPool, StatPool, and SpanProbe. VIB-Probe reads the internal representation at the final token of each candidate semantic unit. MeanPool uniformly averages the token representations over the complete unit, while StatPool aggregates them using a predefined set of summary statistics. SpanProbe instead learns the relative importance of individual tokens and adaptively aggregates their evidence into a span-level representation.

We report AUROC and AUPRC with hallucinated samples treated as the positive class. Because both target test sets are class-imbalanced, AUPRC is the primary metric. Both metrics are threshold-independent and directly evaluate how well risk scores learned only from the source dataset rank examples in the target dataset.

C.2 Results and Additional Analysis

Table 2 presents the primary comparison. In the M-HalDetect → MSCOCO direction, SpanProbe exceeds MeanPool by 2.26 AUROC and 2.34 AUPRC percentage points, and VIB-Probe by 9.73 and 17.78 points. In the reverse direction, it exceeds MeanPool by 6.76 AUROC and 12.52 AUPRC points. The target class proportions and evaluation units differ across directions, so absolute scores should be interpreted within each direction.

D CANDIDATE-BOUNDARY ROBUSTNESS AND AUTOMATIC-BOUNDARY TRANSFER

D.1 ROBUSTNESS UNDER SYNTHETIC BOUNDARY NOISE

We first examine whether SpanProbe depends on exact human-annotated intervals by perturbing or expanding the annotated test spans of M-HalDetect. Experiments use LLaVA-v1.5 and the fixed test set of 158 images, 636 complete responses, and 4,366 annotated spans. Token-level internal features are read from the cached complete responses, so the backbone outputs and response text remain fixed. EndProbe, MeanPool, StatPool, and SpanProbe use frozen checkpoints fitted on the complete training split.

All detectors are trained only on the original training spans. After the test boundaries are changed, we perform no retraining, parameter updates, feature-standardization re-estimation, threshold calibration, or hyperparameter tuning. We consider five boundary settings. **Gold** uses the original annotations. **Jitter** ±1/±2 randomly selects one side of each span and shifts that boundary by at most one or two tokens within the valid response range while retaining the original label. **Merge** combines adjacent annotated spans and labels the merged candidate hallucinated if any constituent span is hallucinated. **Add function word** expands each span to an adjacent function word when available while retaining its original label. All randomized operations use the fixed seed 20260912.

Jitter is a synthetic boundary-noise test under inherited labels rather than a newly annotated test set. Merge changes the number and granularity of candidates as well as the positive-class prevalence, so its AUPRC should not be compared directly with Gold. Within each boundary setting, however, every detector is evaluated on identical candidates and labels.

Table 13: AUPRC (%) under different candidate-boundary settings on M-HalDetect using LLaVA-v1.5. All detectors are trained on the dataset-provided training spans and applied to the modified test boundaries without retraining or calibration.

Boundary	Candidates	Pos. (%)	Avg. tokens	EndProbe	MeanPool	StatPool	SpanProbe
Gold	4,366	23.59	15.27	65.73	72.75	74.79	**77.48**
Jitter ±1	4,366	23.59	15.19	49.35	70.58	67.33	**74.41**
Jitter ±2	4,366	23.59	15.14	41.99	64.39	62.95	**70.17**
Merge	2,346	38.49	28.73	71.59	**78.72**	76.51	78.51
Add function word	4,366	23.59	16.06	57.84	71.23	69.37	**76.63**

Under local boundary shifts, SpanProbe retains the strongest absolute AUPRC. It exceeds the strongest baseline by 3.83 points under Jitter ±1 and by 5.78 points under Jitter ±2. EndProbe degrades sharply as the candidate endpoint moves, whereas aggregating evidence across the candidate provides a more reliable risk ranking under endpoint noise.

Adding an adjacent function word changes SpanProbe's AUPRC by only 0.85 points relative to Gold, and it remains 5.40 points above the strongest baseline in that setting. This result indicates that adaptive pooling can limit the influence of weakly informative boundary tokens. Under Merge, MeanPool and SpanProbe are nearly tied, reaching 78.72 and 78.51 AUPRC, respectively. Because merging changes candidate granularity and class prevalence, this row supports only within-setting comparisons.

Overall, SpanProbe achieves the strongest result under exact boundaries, both local perturbation levels, and function-word expansion, while remaining close to the best method under merged boundaries. Its advantage therefore does not require perfectly precise annotated intervals, although performance still depends on boundary quality.

D.2 TRANSFER TO AUTOMATICALLY SEGMENTED SENTENCES

We next evaluate adaptation to automatically generated candidate boundaries. We keep the images, model responses, and fixed M-HalDetect test split unchanged, but replace the dataset-provided span candidates with sentences automatically segmented through spaCy's `doc.sents` interface. This

setting changes only candidate boundaries and evaluation granularity while holding the image and generated text fixed.

All detectors use frozen checkpoints trained on the human-annotated spans in the M-HalDetect training split and are applied directly to the automatically constructed sentences. We perform no parameter updates, retraining, model selection, hyperparameter tuning, feature-standardization re-estimation, or target-boundary calibration. An automatic sentence is labeled hallucinated if its character interval overlaps any dataset interval labeled INACCURATE or UNSURE, and grounded otherwise; the gold annotations are used only to construct evaluation labels.

The fixed test set contains 158 images, 636 responses, and 4,366 annotated spans, including 1,030 hallucinated spans. Automatic segmentation produces 2,901 sentences, of which 2,893 yield complete token-aligned internal features and are included for every method. These evaluation sentences contain 954 hallucinated positives (32.98%). Their mean and median lengths are 23.04 and 22 input tokens, respectively, and all are multi-token candidates. Each evaluated sentence overlaps 1.54 annotated segments on average; 490 sentences (16.94%) overlap both grounded and hallucinated segments, making them coarser mixed-semantic units than the original annotations.

The automatic boundaries retain nearly all annotated hallucination content: all 1,030 hallucinated gold spans overlap at least one evaluated sentence, and 14,203 of 14,204 hallucinated token positions are covered (99.99%). Thus, the automatic construction does not produce an easier evaluation set by systematically omitting hallucinated content.

We compare EndProbe, MeanPool, StatPool, and SpanProbe on the same automatic sentences. EndProbe reads the TEP representation at the final token of each sentence; MeanPool averages token representations uniformly; StatPool applies the predefined statistical aggregation used in the main experiments; and SpanProbe learns token-relative weights over the complete sentence.

Table 14: Zero-shot transfer from human-annotated spans to spaCy sentence boundaries on M-HalDetect. All detectors are trained on the annotated spans and applied without retraining or calibration to the same 2,893 automatic sentences, including 954 hallucinated positives. AUROC and AUPRC are percentages; higher is better.

Method	AUROC↑	AUPRC↑
EndProbe	81.02	73.34
MeanPool	80.26	72.90
StatPool	80.89	71.98
SpanProbe	**84.02**	**76.97**

SpanProbe achieves the strongest performance under automatic sentence boundaries. Relative to EndProbe, the strongest baseline in this setting, it improves AUROC and AUPRC by 3.00 and 3.63 percentage points, respectively. Because every method is evaluated on identical automatic sentences and labels, these differences are not attributable to changes in candidate count or class prevalence across methods.

The result shows that the benefit of learning token-relative importance is not confined to precise human-annotated spans and transfers to coarser automatic candidate units without target-boundary training or calibration. However, `doc.sents` supplies sentence boundaries rather than discovering fine-grained semantic spans for objects, attributes, actions, or relations; this experiment therefore evaluates boundary transfer rather than general semantic-span proposal.

E ABLATION ON DECODER-LAYER SELECTION

Layer Selection. To determine whether SpanProbe requires evidence from all decoder layers, we compare four contiguous eight-layer windows, eight uniformly sampled layers, and all 32 layers on M-HalDetect. As shown in Table 15, All-32 achieves the highest AUROC and AUPRC on both backbone models. Among the fixed windows, Layers 9–16 perform best, suggesting that middle-layer signals are more discriminative in isolation than signals from shallow or deep layers. Nevertheless, compared with the best fixed window, All-32 still improves AUPRC by 2.23 and 2.57 percentage

points on the two backbone models, respectively. Compared with Uniform-8, the corresponding improvements are 1.58 and 3.85 percentage points. These consistent gains indicate that hallucination evidence is distributed across different decoder depths and that dense cross-layer coverage captures complementary information missed by restricted or sparsely sampled layer selections.

Table 15: Results with different decoder-layer selections on M-HalDetect. Results are reported as percentages. Uniform-8 denotes eight uniformly sampled layers, and Layers 25–32 is equivalent to Last-8.

Layer Selection	LLaVA-v1.5		LLaVA-v1.6	
	AUROC	AUPRC	AUROC	AUPRC
Layers 1–8	86.22	69.95	86.19	69.86
Layers 9–16	88.60	75.25	88.72	76.59
Layers 17–24	87.18	73.08	88.71	75.39
Layers 25–32	86.09	71.16	87.22	73.17
Uniform-8	88.33	75.90	88.48	75.31
All-32	**89.71**	**77.48**	**90.14**	**79.16**

F Position, Length, and Lexical Shortcut Analysis

To examine whether SpanProbe's performance is primarily driven by the generation position, length, or lexical information of candidate spans, we conduct additional shortcut-control experiments on M-HalDetect using LLaVA-v1.5. All experiments follow the same image-level training and test split as in the main experiments. The test set contains 4,366 spans, including 1,030 hallucinated spans and 3,336 grounded spans. All feature standardization and model training use only the training set; test labels are used only for final evaluation and construction of the matched subset described below, and do not participate in model selection or threshold selection.

We construct three baselines that do not use images, attention tensors, hidden states, or Token Evidence Profiles (TEPs). PositionProbe uses only structural features, including the start and end positions of the candidate span, span length, natural-language word count, generation-prefix length, sentence or mention index, and repetition status. LexicalProbe predicts hallucination risk using lowercase unigram/bigram TF–IDF representations of the complete surface text of each span. Position+Lexical Probe combines these structural and lexical features. All three baselines are fitted only on the training set and evaluated on the same test samples as SpanProbe.

In addition to the complete test set, we construct a position–length–lexical-group-matched subset within the fixed test set. For each hallucinated span, we first seek a grounded span with the same complete normalized span text; if none exists, we fall back to a grounded span with the same central content word. Matched samples are additionally required to fall into the same span-length bin and generation-position bin and to have the same repetition status. Each grounded span is used at most once, and matching does not participate in training or model selection. This procedure yields 191 pairs, comprising 382 spans: 6 pairs share the same complete normalized span text, while the remaining 185 pairs share the same central content word. The subset covers 18.5% of the hallucinated spans in the test set.

On the complete test set, PositionProbe, LexicalProbe, and Position+Lexical Probe achieve 36.20, 64.87, and 66.07 AUPRC, respectively, showing that lexical content is more predictive than the measured structural features. SpanProbe reaches 77.48 AUPRC, exceeding the strongest explicit shortcut baseline by 11.41 points (95% confidence interval: $[8.01, 14.70]$). It also exceeds its uniform-pooling variant, which obtains 72.75 AUPRC, by 4.73 points ($[2.98, 6.57]$).

On the matched subset, PositionProbe reaches 54.23 AUPRC, close to the 50% random baseline under class balance, while LexicalProbe remains at 67.11 because most pairs share only the central content word. SpanProbe reaches 75.51 AUPRC, exceeding Position+Lexical Probe by 8.58 points ($[2.24, 15.20]$) and its uniform-pooling variant by 6.33 points ($[1.97, 10.38]$). These comparisons show that the gains cannot be attributed solely to aggregating multiple span tokens and support learning token-relative importance after controlling for the measured shortcuts.

Table 16: Position, length, and lexical shortcut analysis on M-HalDetect using LLaVA-v1.5. AUPRC is reported as a percentage. The matched set contains 191 hallucinated–grounded span pairs matched by normalized text or central content word, span-length bin, generation-position bin, and repetition status. Values should not be compared across columns because the class proportions differ.

Method	Complete Test Set AUPRC	Position–Length–Lexical-Group Matched Set AUPRC
PositionProbe	36.20	54.23
LexicalProbe	64.87	67.11
Position+Lexical Probe	66.07	66.93
SpanProbe w/o Adaptive Pooling	72.75	69.18
SpanProbe	**77.48**	**75.51**

As an additional control, we replace the raw attention entropy with $H_{t,\ell,h}/\log(\max\{|K_t|, 2\})$, where $|K_t|$ is the number of accessible keys at step t, while keeping all other features and training unchanged. The variant obtains 76.09 and 74.31 AUPRC on the complete and matched sets, versus 77.48 and 75.51 originally, retaining most performance after controlling for entropy's mechanical dependence on context length.

Overall, position, length, and lexical information are each correlated with span-level hallucination risk to some extent, with lexical information exhibiting stronger predictive capability. SpanProbe achieves consistent further gains on both the complete test set and the matched subset, indicating that the model-internal signals captured by the TEP provide discriminative information complementary to these explicit structural and lexical features.

G Visual Sensitivity under Counterfactual Image Pairing

To further examine whether SpanProbe responds to visual content rather than relying only on fixed text, lexical identity, or position, we construct counterfactual image pairs on the LLaVA-v1.6 mention-level MSCOCO held-out set described in Appendix A.4. This set contains 2,597 object mentions from 400 images that are disjoint from both detector training and the mitigation evaluation set.

For each canonical object mention c, we construct a pair of image conditions, I^+ and I^-, where the COCO annotations mark c as present in I^+ and absent in I^-. Depending on the original mention label, the source image supplies one condition and the other image is selected from the same 400-image held-out pool. The two conditions use the identical prompt, complete response, candidate-span boundaries, and response-relative token positions; only the input image changes. We replay the same fixed response under both image conditions to re-extract the TEPs and use the frozen SpanProbe checkpoints to compute the hallucination risk for each candidate.

Because LLaVA-v1.6 uses AnyRes image processing, different images can produce different numbers of visual tokens. We therefore additionally construct a strictly matched subset in which I^+ and I^- have the same visual-token count. Of the 2,597 pairs, 2,470 satisfy this requirement; the remaining 127 pairs, for which no suitable image exists in the held-out pool, are included only in the complete paired set. In the strictly matched subset, response tokens have the same absolute sequence positions and the same number of causally accessible keys under both conditions.

We measure whether the detector assigns greater risk to the image condition without annotated support for the target object using paired accuracy:

$$\mathrm{PairAcc} = \frac{1}{N}\sum_{i=1}^{N} \mathbb{I}\big[r(I_i^-) > r(I_i^+)\big] .$$

We also report the risk difference $\Delta r = r(I^-) - r(I^+)$, together with AUROC and AUPRC computed by treating the I^- condition as hallucinated and the I^+ condition as grounded. Table 17 reports the mean and standard deviation across five existing frozen SpanProbe checkpoints. These

checkpoints are not selected, retrained, or ensembled for this analysis. Confidence intervals are computed from 2,000 bootstrap samples clustered by the original image identifier.

Table 17: Counterfactual image-pair results on LLaVA-v1.6. The COCO annotations mark the target object as present in I^+ and absent in I^-; both conditions retain identical text, span boundaries, and response-relative token positions. The AnyRes-matched set additionally requires the same number of visual tokens. Results are mean $\pm$ standard deviation across five frozen checkpoints. Because the paired construction contains equal numbers of I^- and I^+ conditions, its AUPRC is not directly comparable to AUPRC under the natural mention-level class distribution.

Evaluation Set	PairAcc (%)	Mean Δr	Median Δr	AUROC (%)	AUPRC (%)
All 2,597 pairs	94.64 ± 0.11	0.447 ± 0.037	0.395 ± 0.094	84.04 ± 2.01	80.65 ± 1.95
AnyRes-matched 2,470 pairs	94.86 ± 0.10	0.450 ± 0.037	0.403 ± 0.095	84.25 ± 2.03	81.26 ± 1.98

Across all pairs, SpanProbe attains a mean PairAcc of 94.64%, with an image-clustered bootstrap 95% confidence interval of $[93.47\%, 95.72\%]$. The mean risk difference is 0.447, with a 95% confidence interval of $[0.428, 0.466]$, showing that the image condition without annotated target-object support generally receives higher hallucination risk even when the text is fixed. After matching the AnyRes visual-token count, PairAcc remains 94.86% ($[93.64\%, 96.01\%]$), while AUROC and AUPRC reach 84.25% and 81.26%, respectively. The consistency across the complete and matched sets indicates that the observed effect is not explained by image-resolution-induced differences in visual-token count, response-token position, or the number of accessible keys. PairAcc measures within-text paired ordering, whereas AUROC and AUPRC measure global ordering across all positive and negative conditions; their absolute values therefore need not coincide.

These results show that, when the complete candidate text, token sequence, and positions are held fixed, SpanProbe's predicted risk changes systematically with whether the target object is supported by the image annotations, providing counterfactual evidence of sensitivity to visual content.

H Token-Level Evidence Heterogeneity Analysis

To further examine differences in the information carried by tokens within complete semantic spans, we first analyze how the benefit of complete-span modeling varies with span length. We then use two supplementary token-level diagnostics: counterfactual image pairing measures how individual tokens respond to changes in visual support, and a cross-fitted remove-and-retrain protocol tests whether high-weight tokens carry predictive information that is less readily replaced by the remaining tokens. The two token-level diagnostics use previously cached Token Evidence Profiles (TEPs), without rerunning the MLLM or re-extracting internal representations.

H.1 Span Length and the Benefit of Complete-Span Modeling

We group candidate spans in M-HalDetect by their number of natural-language words: 1, 2, 3–4, and 5–8. Within each group, we compare the AUPRC of SpanProbe with that of EndProbe, which reads from the final token of a span. Figure 3 reports the AUPRC difference, with positive values indicating an advantage for SpanProbe.

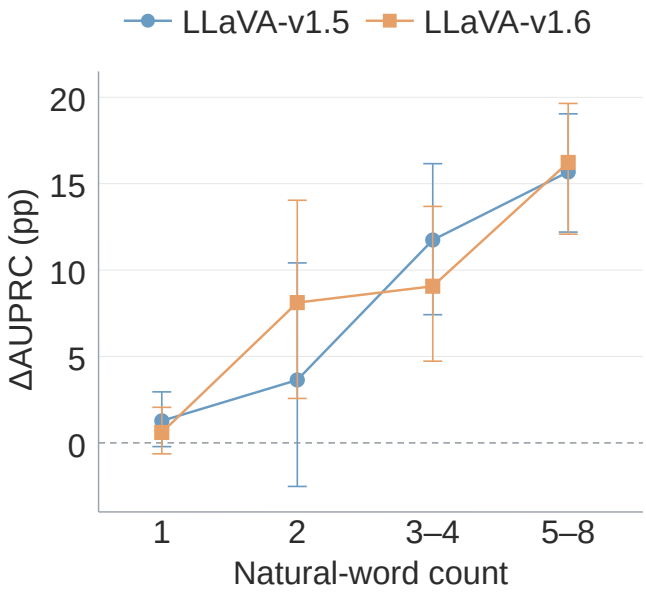

Figure 3: AUPRC advantage of SpanProbe over EndProbe across M-HalDetect spans grouped by natural-language word count. The horizontal axis denotes the number of natural-language words in a span; the vertical axis shows the AUPRC difference in percentage points. Results are shown for LLaVA-v1.5 and LLaVA-v1.6.

As shown in Figure 3, SpanProbe's advantage over EndProbe generally increases with span length on both backbones. The gap is smaller for short spans, whereas it reaches approximately 16 AUPRC percentage points for spans containing 5–8 natural-language words. This pattern suggests that reading from a fixed endpoint may miss information useful for span-level prediction when semantic content is expressed across more words, making complete-span modeling more beneficial.

H.2 Token-Level Visual Sensitivity under Counterfactual Image Pairing

Building on the counterfactual image pairs in Appendix G, we analyze token-level responses to visual support. For each candidate span, I^+ denotes an image condition in which the target object is supported by the image annotations, whereas I^- denotes a condition without such support. The two conditions use identical prompts, generated text, token sequences, span boundaries, and response-relative positions; only the input image changes.

To obtain a token-level signal independent of SpanProbe's aggregation weights, we use a separately trained token probe g. This probe takes the TEP x_t of an individual token as input, outputs a hallucination-risk score, and shares no parameters with SpanProbe's token scorer. For token t in a span, we define its counterfactual visual sensitivity as

$$\delta_t = g\big(x_t(I^-)\big) - g\big(x_t(I^+)\big)\,.$$

A larger positive value means that removing visual support for the target object causes the independent probe to assign a larger increase in hallucination risk to that token.

Because one- and two-token candidates provide too few positions for a stable characterization of within-span evidence distributions, we retain only candidates containing at least three tokens. The resulting set contains 266 spans and 860 tokens. For each span, we define

$$v_t = \max(\delta_t, 0), \qquad p_t = \frac{v_t}{\sum_{k\in s} v_k},$$

and quantify within-span concentration using the Gini coefficient, normalized entropy, and the top-20% evidence share. Normalized entropy is

$$H_{\text{norm}}(s) = -\frac{\sum_{t\in s} p_t \log p_t}{\log |s|}.$$

Lower normalized entropy and a higher Gini coefficient indicate a more uneven distribution of token-level sensitivity. The top-20% share is the fraction of sensitivity assigned to the $\lceil 0.2|s| \rceil$ tokens with the largest p_t. Because spans are short and the number of selected tokens is rounded upward, the corresponding uniform baseline is not exactly 20%.

Table 18: Token-level distribution of counterfactual visual sensitivity within candidate spans containing at least three tokens. The uniform baseline is the expected top-20% share when all tokens in a span of the same length have equal sensitivity.

Spans	Tokens	Mean δ_t	Median δ_t	Gini	Norm. Entropy	Top-20% Share	Uniform
266	860	0.335	0.111	0.341	0.717	59.20%	31.41%

The top 20% of tokens account for 59.20% of the positive counterfactual sensitivity on average, 27.79 percentage points above the length-matched uniform baseline. Together with a Gini coefficient of 0.341 and normalized entropy of 0.717, this result shows that changes in visual support are reflected unevenly across tokens within the analyzed spans, with clear but not extreme token-level heterogeneity.

H.3 Cross-Fitted Remove-and-Retrain Analysis

Directly deleting the highest-weight token from a trained attention-pooling detector can introduce two confounds. The deletion creates inputs not observed during training, and removing the highest-weight token mechanically causes a large change in the current weighted aggregation. A test-time deletion therefore shows that the existing detector depends on a token, but does not establish whether the token itself carries predictive information that cannot be replaced by the remaining tokens.

To reduce these confounds, we use a cross-fitted remove-and-retrain protocol. A teacher SpanProbe that has not observed the corresponding evaluation fold first ranks the tokens. We then apply the same removal rule to the training and evaluation data and retrain an otherwise identical mean-pooling detector on the modified training inputs, allowing the downstream detector to adapt to the post-removal distribution.

The experiment uses TEPs previously cached from LLaVA-v1.5 and LLaVA-v1.6 on M-HalDetect and retains only candidate spans containing at least three tokens. No MLLM inference or representation extraction is rerun. The two backbones use the same image-grouped cross-fitted protocol, including image-level partitions, teacher training and selection, token ranking, removal rules, downstream-detector training and model selection, and random-removal repetitions. Within each backbone, all conditions use the same filtered candidates and folds. Because tokenization and the resulting filtered candidate sets can differ across backbones, absolute AUPRC values should not be compared between the two model columns.

For each evaluation fold, the teacher SpanProbe is trained only on the remaining folds and therefore has not observed any image or span in the fold it ranks. For each backbone and fold, we use one teacher checkpoint chosen by the same prespecified model-selection rule, which does not access the remove-and-retrain evaluation results. These fold-specific teacher checkpoints are fixed before the removal experiments.

We compare four conditions: **No removal** retains every span token; **Remove random** deletes a randomly selected token from the same span; **Remove endpoint** deletes the final token; and **Remove top-weight** deletes the token assigned the largest aggregation weight by the teacher. In each condition, we retrain a mean-pooling detector with identical architecture, optimization, and model-selection procedures using the remaining token TEPs, allowing the detector to adapt to the post-removal input distribution. Random removal is repeated 20 times and the results are averaged. All conditions retain the feature-normalization statistics estimated from the original, unmodified training data, preventing re-estimated means and variances from introducing an additional source of variation.

Because hallucinated spans are the minority class and the purpose of this diagnostic is to compare removal-induced changes, we use AUPRC as the primary metric. We define $\Delta\mathrm{AUPRC} =$

$\mathrm{AUPRC}_{\text{no removal}} - \mathrm{AUPRC}_{\text{condition}}$. A positive value denotes a performance decrease after removal, whereas a negative value denotes a small improvement.

Table 19: Cross-fitted remove-and-retrain results on M-HalDetect spans containing at least three tokens. Every condition retrains the same mean-pooling detector on the corresponding post-removal training inputs. AUPRC and ΔAUPRC are percentages and percentage-point changes, respectively; bold marks the largest degradation within each backbone.

Condition	LLaVA-v1.5		LLaVA-v1.6	
	AUPRC↑	ΔAUPRC	AUPRC↑	ΔAUPRC
No removal	60.60	0.00	69.62	0.00
Remove random	60.76	-0.16	68.98	0.64
Remove endpoint	61.02	-0.42	69.45	0.17
Remove top-weight	58.74	**1.86**	67.11	**2.51**

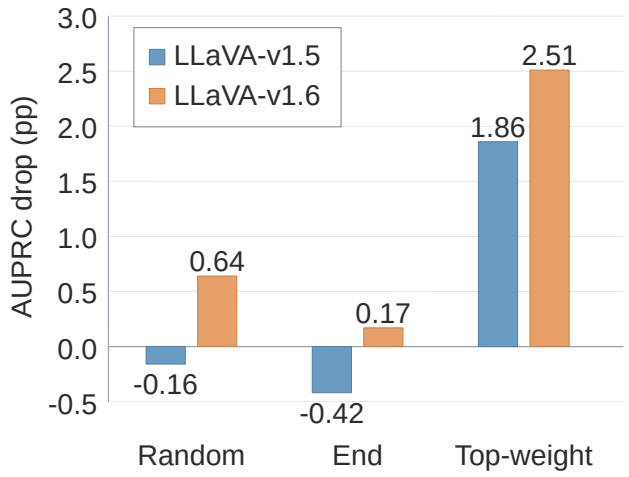

Figure 4: AUPRC changes under the cross-fitted remove-and-retrain protocol on M-HalDetect spans containing at least three tokens, visualizing Table 19. Each condition retrains the same mean-pooling detector after removing a random token, the span-final token, or the token assigned the highest weight by the cross-fitted teacher SpanProbe. Positive values indicate a decrease relative to no removal; negative values indicate a small improvement. Top-weight removal produces the largest decrease on both backbones.

Figure 4 visualizes these results. The two backbones exhibit the same relative pattern. On LLaVA-v1.5, removing the highest-weight token decreases AUPRC by 1.86 percentage points. By contrast, random and endpoint removal increase AUPRC slightly, by 0.16 and 0.42 points, respectively. Thus, relative to random and endpoint removal, top-weight removal produces additional losses of 2.02 and 2.28 points. On LLaVA-v1.6, top-weight removal decreases AUPRC by 2.51 points, whereas random and endpoint removal decrease it by only 0.64 and 0.17 points. The corresponding additional losses are 1.87 and 2.34 points.

The small increases after random or endpoint removal on LLaVA-v1.5 may reflect the removal of redundant or noisy tokens or ordinary training and sampling variation; they should not be interpreted as systematic improvements from these deletion strategies. More importantly, top-weight removal is the only condition that produces a clear decrease on both backbones, and its decrease is consistently larger than those of the random and endpoint controls.

Because each detector is retrained on its corresponding post-removal training inputs, this pattern cannot be explained solely by the original attention-pooling formula's mechanical dependence on the highest-weight token or by a mismatch between the training and evaluation input distributions.

The consistent descriptive trend across backbones indicates that the teacher's token ranking preferentially identifies tokens carrying span-label predictive information that is less readily replaced by the remaining tokens. For each backbone and fold, the analysis uses a single prespecified teacher checkpoint rather than averaging across multiple training initializations; it therefore tests whether the remove-and-retrain pattern reproduces across backbones, not whether token rankings are stable across random seeds. Moreover, learned aggregation weights reflect tokens' relative information value for span-level prediction and should not be interpreted as causal contributions to the generation of hallucinations.

H.4 Qualitative Comparison with VIB-Probe

Figure 5 presents four disagreement cases in which VIB-Probe is incorrect and SpanProbe is correct, illustrating cases where the token assigned the highest aggregation weight by SpanProbe differs from the fixed endpoint used by VIB-Probe.

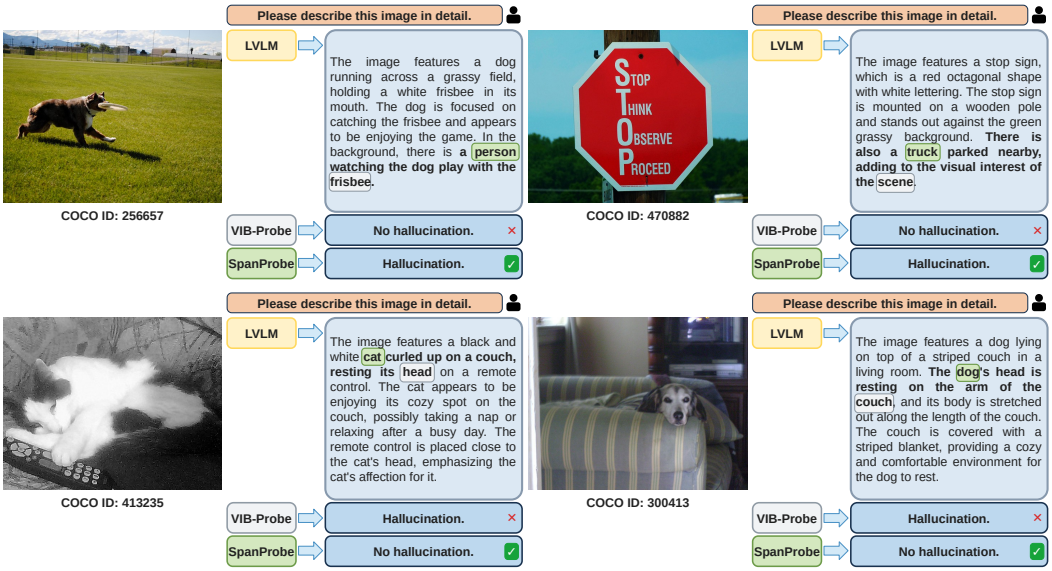

Figure 5: Qualitative comparison between VIB-Probe and SpanProbe on four disagreement cases. Each panel shows the input image, the LVLM response, and the predictions of the two detectors. Bold text denotes the candidate span evaluated for hallucination. The gray outline marks the span-final token whose internal representation is used by VIB-Probe, whereas the green outline marks the token assigned the highest aggregation weight by SpanProbe. VIB-Probe misses the hallucinated spans in the top row and falsely flags the grounded spans in the bottom row, while SpanProbe classifies all four spans correctly. Check marks and crosses indicate correct and incorrect predictions, respectively.