Cross-Layer Evolution Graph Learning for Fine-grained VLM Hallucination Detection

Shaoxuan Li **Shiliang Zhang***

State Key Laboratory of Multimedia Information Processing,
School of Computer Science, Peking University

shaoxuanli@stu.pku.edu.cn, slzhang.jdl@pku.edu.cn

Abstract

Although Vision-Language Models (VLMs) demonstrate impressive capabilities, their deployment in high-stakes domains is severely hindered by hallucinations. Existing detection methods mostly focus on static internal feature probing with coarse-grained annotations, failing to achieve fine-grained token-level localization without expensive dense labels. In this paper, we reconceptualize VLM hallucination detection as a dynamic graph learning task, identifying hallucinations as distinct *trajectory drifts* across network layers. To address these challenges, we propose TRACE (Token Representation Analysis with Cross-layer Evolution), a non-intrusive probing framework that models internal states as heterogeneous spatio-temporal graphs. By employing Graph Neural Networks (GNNs) to capture computational dynamics, TRACE effectively isolates hallucinatory signals. Furthermore, a token-first Multiple Instance Learning (MIL) readout elegantly bridges the granularity gap, distilling macroscopic sentence-level supervision into token-level risk attribution without token-level training labels. Extensive evaluations on M-HalDetect, ViGoR, and HalLoc demonstrate that TRACE achieves state-of-the-art detection performance while being competitive in localization accuracy, potentially offering an efficient and interpretable solution for trustworthy VLMs. Our code is available at `https://github.com/Shaoxuan-Li/TRACE`.

1 Introduction

Vision-Language Models (VLMs) demonstrate remarkable capabilities in multimodal comprehension and reasoning. However, their trustworthy deployment in high-stakes domains, such as medical diagnostics and autonomous driving, remains impeded by hallucinations. In the context of VLMs, hallucinations manifest as the generation of text that is factually ungrounded, logically inconsistent, or structurally misaligned with the visual context [31, 43]. Although extensive efforts have been dedicated to mitigating this phenomenon [45, 41, 16, 18, 23, 9], accurately detecting and localizing hallucinations post hoc remains an open and critical challenge.

Existing detection paradigms face a persistent granularity dilemma. Most current methodologies operate at the macroscopic response level (e.g., sentence level) [20, 52, 31, 45]. While inexpensive to obtain, these labels fail to localize the exact onset of hallucinations within the sequence. Conversely, token-level detection [38] offers fine-grained localization but requires expensive human annotations, rendering it difficult to scale for massive multimodal corpora.

Bridging this gap requires extracting well-designed, highly discriminative internal token representations. Prior mechanistic interpretability works typically probe static activations [9, 4] or sequential

*Corresponding author.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

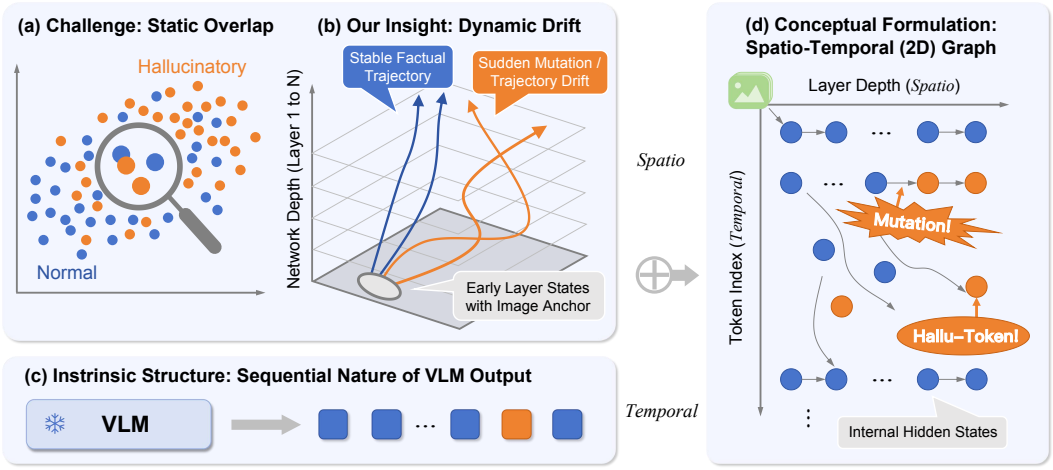


Figure 1: Conceptual Formulation of our proposed framework. (a) Static single-layer representations exhibit severe semantic entanglement. (b) Layerwise tracking unveils *trajectory drift* as a highly discriminative indicator. (c) The autoregressive decoding inherently forms a sequential structure over time. (d) Combining these insights, we conceptualize the internal states as a 2D spatio-temporal graph, capturing hallucination signals via graph learning.

attention maps [16, 2]. However, as illustrated in Figure 1(a), static hidden states suffer from severe semantic entanglement, blurring the decision boundary. In striking contrast, shifting to a dynamic layerwise perspective unveils a highly informative pattern (Figure 1(b)). While factual tokens maintain a stable trajectory anchored to the visual context, hallucinatory tokens undergo a sudden mutation in deeper layers—a phenomenon we term *trajectory drift* [18, 47].

Furthermore, the autoregressive text generation of VLMs inherently possesses a sequential structure (Figure 1(c)). Crucially, combining this sequential coherence with the aforementioned cross-layer evolution naturally manifests as a structured two-dimensional **spatio-temporal graph** (Figure 1(d)). In this formulation, the spatial dimension represents the network depth (layerwise transformations), while the temporal dimension corresponds to the autoregressive sequence steps (token-wise context). Within this topology, nodes represent internal hidden states, and edges encode the dense interdependencies among vertical evolution, horizontal context, and cross-modal grounding.

To effectively navigate this complex computational landscape, we leverage the structural inductive bias of Graph Neural Networks (GNNs) [44, 54]. Accordingly, we propose **TRACE** (Token Representation Analysis with Cross-layer Evolution), a non-intrusive probing framework that formulates hallucination detection as a graph learning task. By performing message passing along this heterogeneous graph, TRACE seamlessly captures the dynamic structural shifts that characterize hallucinations.

Leveraging these informative graph representations, TRACE elegantly resolves the granularity dilemma through a Multiple Instance Learning (MIL) readout [12, 25]. It acts as an information bottleneck, distilling macroscopic sentence-level supervision into token-level risk attribution, thereby enabling localization without token-level training labels.

In summary, our main contributions are as follows:

- **Spatio-temporal Graph Modeling**: We introduce TRACE, a non-intrusive probing framework that transcends static feature analysis by formulating VLM hallucination detection as a graph learning task over heterogeneous spatio-temporal evolution graphs.
- **Weakly-supervised Localization**: We design a token-first MIL readout mechanism that elegantly bridges the granularity gap, distilling token-risk attribution solely from coarse sentence-level supervision.
- **State-of-the-Art Performance**: Extensive experiments demonstrate that our framework achieves superior sentence-level detection across multiple benchmarks, delivering competitive fine-grained localization accuracy that rivals fully supervised baselines on the HalLoc benchmark.

2 Related Work

2.1 Hallucination Detection in VLMs

The evolution from Large Language Models (LLMs) to Vision-Language Models (VLMs) has been accompanied by the persistent challenge of hallucinations [31, 43, 33], where models generate seemingly coherent but factually inaccurate content. In VLMs, hallucinations primarily stem from distributional shifts, data biases, and a fundamental lack of visual grounding. Specifically, when internal linguistic priors overwhelm visual anchoring signals, models tend to fabricate objects or attributes disconnected from the input image.

Current research predominantly focuses on sentence-level or segment-level detection [45, 41, 16, 18, 23, 9]. For instance, benchmarks like M-HalDetect [20] and ViGoR [52] provide coarse-grained annotations for detailed image descriptions. These methodologies typically yield binary classification verdicts for the entire response sequence. While progress has been made in coarse detection, research into fine-grained token-level detection is also emerging, notably with the introduction of the HalLoc benchmark [38]. However, this task remains challenging due to the implicit nature of visual grounding and the high cost of manual token-level labels. TRACE aims to bridge this gap by distilling precise token-level localization capabilities solely from macroscopic sentence-level supervision.

2.2 External Verification of Visual Claims

Post-generation verification assesses whether a response is supported by visual evidence without accessing the generating model's internal states. FaithScore [27] decomposes free-form responses into atomic facts and checks their visual entailment. UniHD [14] extracts claims and selects auxiliary tools to gather evidence for hallucination judgments.

Natural language inference (NLI) [11, 48] compares paired texts for entailment or contradiction, providing a complementary consistency signal when reference descriptions are available. These methods assess generated claims through image- or text-derived evidence. For instance, MOCHa [10] uses contradiction between generated and reference captions as a fidelity signal for training captioners against open-vocabulary hallucinations. TRACE instead models the generating VLM's internal trajectories, learning response-level detection and token-risk attribution from sentence-level labels.

2.3 Internal State Probing

To transcend black-box assessment, researchers have turned to probing internal states for mechanistic interpretability [4, 13, 9, 21]. Attention-based methods assume that attention weights directly encode visual grounding information [16, 36, 2, 50]. For instance, VADE [41] models multi-layered and multi-headed raw attention maps as a sequence to identify hallucinatory patterns, while other approaches detect specific knowledge aggregation patterns [46, 23]. However, raw attention maps are inherently high-dimensional and noisy; naive aggregation often leads to substantial information loss.

Furthermore, under the linear representation hypothesis [39], many works attempt to detect and steer within a hallucination subspace [40, 53, 17]. Although intermediate hidden state probing can be effective in-domain, these static classifiers often overfit to task-specific linguistic priors, resulting in significant performance degradation without explicit vision-text fusion. TRACE addresses this limitation by shifting the focus from static snapshots to dynamic trajectory drifts. By integrating hidden states as node features and attention maps as edge priors within a spatio-temporal graph, our framework captures the evolutionary drift of semantics across layers, thereby achieving superior robustness.

2.4 Graph-based Modeling for VLM Analysis

Representing neural computation as a graph structure for mechanistic analysis is a pioneering direction [51, 37, 19]. In the domain of pure-text LLMs, the CHARM framework [19] represents computational traces as attributed graphs to cast hallucination detection as a graph learning task via GNNs [44, 54, 49]. While CHARM represents each token once and encodes layer and head information in features, TRACE extends this paradigm beyond the single-modality and horizontally-restricted structures of prior work. Our framework introduces a heterogeneous spatio-temporal graph

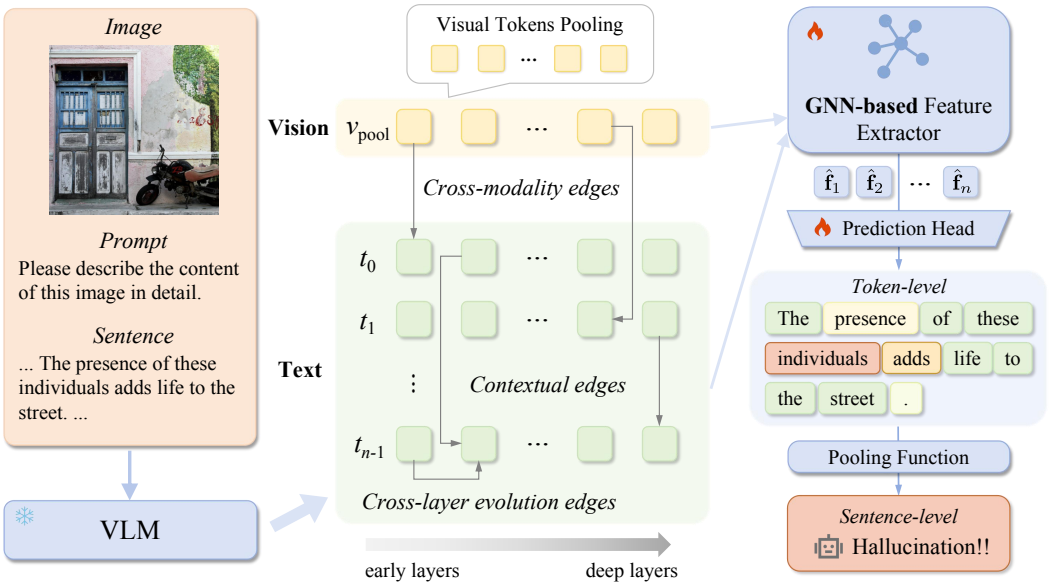


Figure 2: **Overview of the TRACE framework.** (1) **Graph Construction**: We extract hidden states from a frozen VLM to build a spatio-temporal graph, connecting nodes via *cross-layer evolution*, *contextual*, and *cross-modality* edges. (2) **Detection & Readout**: A heterogeneous GNN (f_{mp}) aggregates these computational dynamics into refined token representations $\hat{\mathbf{f}}_i$. Finally, an MIL-based readout ($f_{pool} \circ f_{pred}$) translates them to simultaneously output fine-grained token localization scores s_i and a macroscopic sentence-level hallucination probability $\hat{y}$.

that explicitly models cross-modal bridging and vertical evolution edges, providing a more holistic characterization of VLM dynamics.

Regarding VLM evolution analysis, recent research has explored how multimodal knowledge transforms across network depth [47], identifying critical mutation stages demarcated by critical layers. Analysis indicates that hallucinatory tokens often undergo significant mutations, or *layerwise trajectory drifts*, in deep layers. While these prior studies provide valuable descriptive insights, TRACE is the first to operationalize this evolutionary perspective into an actionable predictive framework. By coupling rich spatio-temporal representations with an MIL-based pooling module [12, 25], we transcend purely qualitative analysis to achieve emergent, fine-grained token-risk attribution from sentence-level weak supervision, successfully bridging the gap between macroscopic observation and microscopic detection.

3 Method

We present **TRACE** (Token Representation Analysis with Cross-layer Evolution), a lightweight and non-intrusive hallucination detector operating over a frozen vision-language model (VLM). Instead of isolating static layer outputs, TRACE conceptualizes the internal computations of a VLM as a dynamic spatio-temporal graph, capturing the semantic trajectory drifts that fundamentally characterize hallucinations.

3.1 Overview

Let $\mathcal{M}$ be a frozen VLM. Given an input triplet (I, P, R), consisting of an image, a text prompt, and an auto-regressively generated response $R = [t_0, t_1, \ldots, t_{n-1}]$, our objective is twofold. The primary macroscopic goal is to predict the sentence-level hallucination probability $\hat{y} \in [0, 1]$, supervised purely by a binary macro-label $y \in \{0, 1\}$ (where $y = 1$ indicates hallucination). Concurrently, our microscopic goal is to extract fine-grained, token-level localization scores $s_i \in [0, 1]$.

To achieve this dual-granularity detection, we formulate our framework as a composite mapping $\mathcal{F} : (I, P, R) \to (\hat{y}, \{s_i\}_{i=0}^{n-1})$, structured sequentially as:

$$\mathcal{F} = f_{pool} \circ f_{pred} \circ f_{mp} \circ f_{graph}. \tag{1}$$

Specifically, the framework operates through the following data flow: First, f_{graph} extracts intermediate hidden states from the frozen backbone to construct a heterogeneous spatio-temporal graph. Then f_{mp} propagates information via a GNN to capture a refined token representation $\hat{\mathbf{f}}_i \in \mathbb{R}^H$. Finally, a Multiple Instance Learning (MIL) readout bridges the two granularities: f_{pred} maps each high-dimensional representation $\hat{\mathbf{f}}_i$ into the scalar token-level score s_i, and f_{pool} aggregates these microscopic scores into the macroscopic graph-level prediction $\hat{y}$. The results with dual-granularity are visualized in Figure 3 and more in the Appendix A.8.

The entire framework is trained end-to-end using a Binary Cross-Entropy (BCE) loss on the sentence-level prediction $\hat{y}$. Crucially, optimizing this macroscopic objective through the MIL bottleneck concurrently forces the network to specialize the well-designed intermediate representations $\hat{\mathbf{f}}_i$.

3.2 Spatio-Temporal Graph Construction

To reduce memory usage and computational overhead, we register hooks at K uniformly-spaced layers $\mathcal{L} = \{l_k\}_{k=1}^{K}$, extracting hidden states H^{l_k} and self-attention matrices A^{l_k} averaged over all heads. We then construct a directed graph $G = (\mathcal{V}_{vis} \cup \mathcal{V}_{text}, \mathcal{E}_{evolve} \cup \mathcal{E}_{context} \cup \mathcal{E}_{cross})$.

Node types. The graph relies on two heterogeneous node types to represent the multimodal context, *vision nodes* and *text nodes*. For vision nodes, we average-pool all image-token hidden states at each layer l_k to form a visual anchor v^{l_k}. For text nodes, each response token t_i at layer l_k is independently represented by its corresponding hidden state $t_i^{l_k}$. The connectivity between these nodes is defined by three distinct computational dimensions, each parameterized by a scalar attribute.

Edge type 1: Cross-layer evolution edges $\mathcal{E}_{evolve}$. The cross-layer evolution edges are designed to connect identical tokens across adjacent sampled layers, from $t_i^{l_{k-1}}$ to $t_i^{l_k}$, capturing the layerwise evolution information which may indicate hallucination:

$$e_{evolve}\left(t_i^{l_{k-1}}, t_i^{l_k}\right) = \cos\left(\mathbf{x}_{t_i^{l_{k-1}}}, \mathbf{x}_{t_i^{l_k}}\right), \; 2 \le k \le K. \tag{2}$$

Here $\mathbf{x}_{t_i^{l_k}}$ denotes the hidden state of token t_i at layer l_k. As formulated, the edge attribute is defined by the cosine similarity of their hidden states to isolate directional semantic shifts from depth-induced norm accumulation. A mutation in this similarity serves as a strong prior of a trajectory drift.

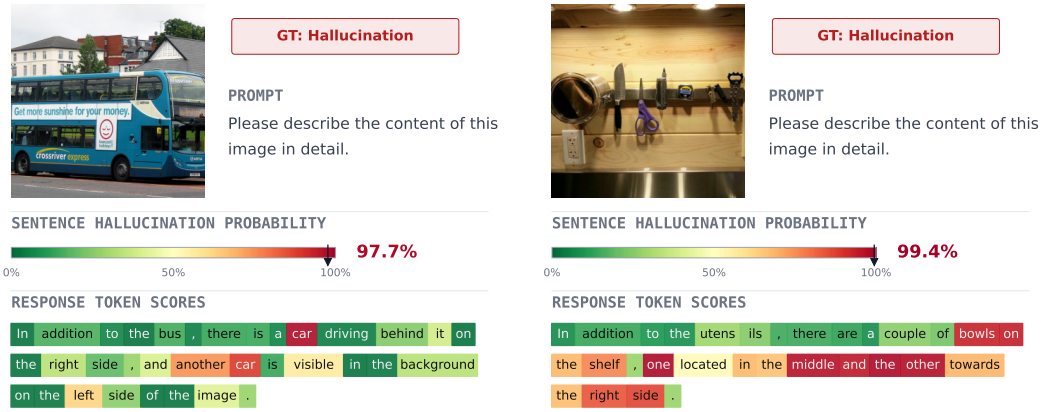
Edge type 2: Contextual edges $\mathcal{E}_{context}$. The contextual edges connect past tokens to current tokens within the same layer, modeling how erroneous information propagates forward through the auto-regressive generation. A directed edge from $t_j^{l_k}$ to $t_i^{l_k}$ $(j \le i)$ exists if the attention weight $A^{l_k}[i, j]$ exceeds a sparsification threshold τ, using the weight itself as the edge attribute:

$$e_{context}\left(t_j^{l_k}, t_i^{l_k}\right) = \begin{cases} A^{l_k}[i,j], & \text{if } A^{l_k}[i,j] > \tau \text{ and } j \le i, \\ 0, & \text{otherwise}, \end{cases} \tag{3}$$

Edge type 3: Cross-modality edges $\mathcal{E}_{cross}$. The cross-modality edges connect the visual anchor v^{l_k} to text tokens $t_i^{l_k}$ at the same layer. The edge attribute is quantified by the top-k average cross-modality attention weight. Specifically, at each sampled layer l_k, the vision anchor node v^{l_k} is connected to every response-token node at the same layer as follows:

$$e_{cross}\left(v^{l_k}, t_i^{l_k}\right) = \frac{1}{k} \sum_{j \in \mathcal{V}_{top\text{-}k}^{i,l_k}} A^{l_k}[i,\, j], \tag{4}$$

This specific design avoids the dilution of attention by irrelevant background patches. A persistent degradation in this cross-modality attribute across layers explicitly signals a collapse in visual grounding, making it a critical feature for hallucination detection.



(a) LLaVA-1.5 (Example 1)

(b) Qwen3-VL (Example 2)

Figure 3: **Visualization of zero-shot token-level hallucination localization.** TRACE accurately identifies hallucinatory spans (highlighted in red) under macroscopic sentence-level supervision, across different VLM backbones. Higher token scores indicate greater predicted hallucination risk.

3.3 Heterogeneous Graph Message Passing

To bridge the dimensionality gap between the massive VLM hidden states and the lightweight GNN working space, initial node features are independently projected and normalized via $\tilde{\mathbf{x}} = \mathrm{LN}(W\mathbf{x})$.

The core of the message passing phase executes M iterations of heterogeneous graph convolutions, where $M \geq K$ guarantees the completeness of cross-layer evolutionary message passing. For each relation type $r \in \{evolve, context, cross\}$, we deploy an independent Graph Attention Convolution with an edge-conditioned scalar gate (GAT_r^{edge}) to process the scalar edge attributes. The text node representations are updated via a post-norm, edge-conditioned residual architecture:

$$\mathbf{h}_{t_i^{l_k}}^{(m+1)} = \mathrm{LN}\left(\mathrm{GELU}\left(\sum_{r\in\mathcal{R}} \mathrm{GAT}_r^{edge}\left(\mathbf{h}^{(m)}, \{e_r\}\right)\right) + \mathbf{h}_{t_i^{l_k}}^{(m)}\right), \tag{5}$$

During this process, vision nodes act strictly as source-only anchors. Their representations remain un-updated during message passing to preserve the uncorrupted visual grounding prior, allowing a stable reference point for the evolving text representations.

After completing M rounds of propagation, we extract only the text nodes at the deepest sampled layer l_K to form the final spatio-temporal representations $\hat{\mathbf{f}}_i = \mathbf{h}_{t_i^{l_K}}^{(M)}$. These representations implicitly aggregate the evolutionary dynamics across layer depth, context width, and multimodal alignment, providing a holistic trajectory embedding ready for the subsequent readout phase.

3.4 Dual-Granularity Readout

To bridge the token-level trajectory features $\hat{\mathbf{f}}_i$ with the macroscopic sentence-level label y, we employ a token-first Multiple Instance Learning (MIL) readout mechanism [12, 25]. The prediction head f_{pred} projects each token representation into a scalar logit $z_i = \mathrm{MLP}_{pred}(\hat{\mathbf{f}}_i)$, which explicitly represents the fine-grained hallucination propensity of that specific token. Instead of pooling the high-dimensional features directly, the pooling module f_{pool} aggregates these scalar logits into a graph-level response logit z_{graph} using the Log-Sum-Exp (LSE) function:

$$z_{graph} = \frac{1}{r}\log\left(\frac{1}{n}\sum_{i=0}^{n-1}\exp(r\, z_i)\right), \tag{6}$$

where $r > 0$ is a temperature parameter controlling the sharpness of the maximum approximation. The final sentence-level hallucination probability is then obtained via a standard sigmoid activation, $\hat{y} = \sigma(z_{graph})$. Under ideal conditions, the sentence-level BCE loss will force the network to specialize the token-level logits z_i. At inference time, token-level localization scores are obtained as $s_i = \sigma(z_i)$.

4 Experiments

To rigorously evaluate the proposed TRACE framework, we design our experiments to answer three core research questions: **RQ1:** Can modeling spatio-temporal dynamics improve sentence-level hallucination detection and out-of-distribution (OOD) transferability? **RQ2:** Does the MIL-based readout mechanism successfully induce emergent token-level hallucination localization without fine-grained supervision? **RQ3:** How do the different computational edges and temporal depths, as well as other hyperparameters contribute to the overall graph dynamics?

4.1 Experimental Setup

Datasets. We conduct evaluations across two distinct granularity levels. For sentence/segment-level detection, we utilize **M-HalDetect** [20] as our source training domain and **ViGoR** [52] as the target domain for OOD transfer evaluation. Both datasets provide binary annotations indicating whether a generated response contains hallucinations. For token-level localization, we employ the recently introduced **HalLoc** [38] dataset, which provides precise token-level hallucination annotations across diverse subsets including VQA, Instruct, and Caption scenario.

Baselines and Evaluation Metrics. We benchmark TRACE against three families of white-box detection methods: (1) *Attention Scores* (VisAttnAgg, VisTexAttnAgg, AttnSink) [23, 50]; (2) *Probing Classifiers* trained on static hidden states from various layers [4, 9]; and (3) *Attention Map based Detectors* representing the prior state-of-the-art (VisAttnDet, VisTexAttnDet, VADE) [16, 41]. For sentence-level tasks, we report the Area Under the Precision-Recall Curve (AUPR) due to the inherent class imbalance in hallucination datasets. For token-level, we report the F1 score.

To provide context for TRACE's white-box results, we also evaluate output-level CLIP, BLIP–NLI, and VLM verifiers on ViGoR (see Appendix A.7).

Implementation Details. We extract intermediate hidden states and attention matrices from state-of-the-art VLMs with a parameter scale of 7B/8B, including the LLaVA series [35, 34] and Qwen-VL series [6, 8, 7]. We construct the spatio-temporal graph using $K = 8$ uniformly sampled layers. The GNN-based detector is trained using the AdamW optimizer with a learning rate of 2e-3. All experiments, including feature extraction, training, and evaluation, were conducted on four NVIDIA RTX 3090 GPUs. Full architectural and experimental details are provided in Appendix A.3.

Table 1: Comparison of hallucination detection methods on M-HalDetect and ViGoR(AUPR).

	M-HalDetect (Source)			**ViGoR** (Target)		
Method	LLaVA-1.5	Qwen1-VL	Qwen3-VL	LLaVA-1.5	Qwen1-VL	Qwen3-VL
Random	0.24	0.24	0.24	0.25	0.25	0.25
Attention Scores						
VisAttnAgg	0.33	0.31	0.31	0.33	0.34	0.35
VisTexAttnAgg	<u>0.35</u>	<u>0.38</u>	<u>0.40</u>	<u>0.37</u>	<u>0.40</u>	<u>0.42</u>
AttnSink	0.30	0.20	0.20	0.33	0.26	0.26
Probing Classifiers						
Last Layer	0.66	0.67	0.70	<u>0.51</u>	<u>0.49</u>	<u>0.58</u>
Last Four Layers	0.68	0.63	<u>0.71</u>	0.50	0.48	0.56
All Layers	<u>0.70</u>	<u>0.69</u>	0.70	0.46	0.47	<u>0.58</u>
Attention Map based Detectors						
VisAttnDet	0.75	0.70	0.78	0.62	0.60	0.66
VisTexAttnDet	0.78	0.71	0.80	0.64	0.64	0.67
VADE	**0.82**	<u>0.74</u>	**0.85**	<u>0.67</u>	<u>0.69</u>	<u>0.73</u>
Evolution Graph Analysis						
Ours	0.80	**0.81**	**0.85**	**0.71**	**0.72**	**0.77**

4.2 Main Results

Table 1 presents the sentence-level detection performance. TRACE achieves competitive performance on M-HalDetect, reaching an AUPR of 0.85 with Qwen3-VL, and outperforms all compared baselines across the three VLM backbones on ViGoR.

More importantly, TRACE exhibits remarkable **out-of-distribution robustness**. While static probing classifiers (e.g., *All Layers*) achieve decent in-domain performance (up to 0.70), their performance drastically collapses when transferred to the ViGoR dataset (dropping to $0.46 \sim 0.58$). In stark contrast, TRACE maintains strong transferability, retaining an AUPR of 0.77 on ViGoR. This corroborates our hypothesis: static features often overfit to dataset-specific linguistic priors, whereas the dynamic trajectory drifts captured by our heterogeneous graph represent the fundamental, invariant mechanism of hallucinations across different domains.

Table 2: Output-level verification baselines and TRACE on ViGoR with Qwen3-VL. The compared methods are described in Appendix A.7.

Method	AUPR	AUROC	F1
Frozen CLIP similarity	0.41	0.64	0.46
BLIP [30] evidence + DeBERTa-NLI [22]	0.33	0.56	0.42
Supervised CLIP interaction probe	0.56	0.75	0.56
Qwen3-VL-8B output-only verifier	0.65	0.80	0.59
TRACE (*white-box*)	**0.77**	**0.86**	**0.68**

Table 2 extends our comparative analysis beyond standard white-box detectors to black-box, output-level verifiers. Although TRACE operates as a white-box framework, it exhibits clear superiority over black-box baselines. Specifically, TRACE outperforms the strongest output-level verifier, Qwen3-VL, by 0.12 in AUPR while maintaining a consistent lead in both AUROC and F1 score. This comparison highlights the intrinsic benefit of leveraging trajectory-inspired representations over purely output-based verification.

Inference Efficiency. To evaluate real-world deployability, we benchmark the runtime overhead of TRACE on 73 Qwen3-VL responses (ranging from 6 to 109 tokens) with a batch size of one on a single NVIDIA RTX 3090 GPU. As summarized in Appendix A.6.7, TRACE introduces a negligible paired latency overhead of 15.2 ms at p50 and 21.0 ms at p95, alongside a modest GPU memory consumption of 53.8 MiB at p95 (28.6 MiB at p50). This computational footprint demonstrates that TRACE can be seamlessly integrated into real-time verification pipelines with little impact on overall system throughput.

Table 3: Hallucination localization F1 scores across subsets and methods on HalLoc with embeddings from LLaVA-1.5.

Subset	Method	Supervision Level	Object	Attribute	Relationship	Scene	Overall
VQA	CHAIR	–	0.19	–	–	–	–
	HalLocalizer	*Token-level*	0.69	0.92	0.68	0.87	–
	Ours	*Sentence-level*	0.62	0.86	0.65	0.85	0.79
		Token-level	**0.97**	**0.99**	**0.98**	**0.98**	**0.98**
Instruct	CHAIR	–	0.13	–	–	–	–
	HalLocalizer	*Token-level*	0.79	0.95	0.83	0.94	–
	Ours	*Sentence-level*	0.41	0.74	0.67	0.78	0.68
		Token-level	**0.81**	**0.97**	**0.98**	**0.98**	**0.94**
Caption	CHAIR	–	0.04	–	–	–	–
	HalLocalizer	*Token-level*	0.45	0.32	**0.47**	0.56	–
	Ours	*Sentence-level*	0.21	0.27	0.35	0.29	0.42
		Token-level	**0.49**	**0.46**	**0.47**	**0.58**	**0.60**

4.3 Emergent Token-Level Localization

The most compelling property of TRACE is its ability to perform fine-grained localization using only macroscopic labels. We empirically validate this on the token-level HalLoc benchmark in Table 3.

We evaluate two variants of our model: ***Sentence-level***, which is trained exclusively on sentence-level binary labels from M-HalDetect and HalLoc, and ***Token-level***, an upper-bound variant fine-tuned with oracle token-level labels. Astonishingly, despite never explicitly seeing token-level annotations during training, TRACE achieves competitive F1 scores, surpassing CHAIR and approaching heavily-engineered baselines like HalLocalizer.

When provided with full token-level supervision, our spatio-temporal graph framework achieves impressive localization. This result serves as powerful empirical evidence for our gradient routing property: the MIL bottleneck successfully forces the network to concentrate precisely on the tokens experiencing trajectory drift. It is important to point out that HalLocalizer employs a fragmented strategy, training *separate, specialized detector heads* tailored for each hallucination category. In contrast, TRACE operates under a unified training paradigm.

4.4 Ablation Studies

To dissect the contribution of our graph components, we perform extensive ablation studies. Specifically, we investigate the necessity of each edge topology, evaluate the impact of graph depth and readout mechanics, and analyze model sensitivity under key hyperparameter configurations.

Edge Topology Analysis. The topology of our spatio-temporal graph is the cornerstone of TRACE. Table 4 ablates the three computational edges. Removing the *Cross-modal* edges (S1) leads to a catastrophic performance drop (from 0.77 to 0.55), highlighting that hallucinations are fundamentally a failure of vision-text grounding rather than purely linguistic anomalies. Removing *Context* (S2) or *Layerwise evolution* (S3) also distinctly degrades performance, proving that horizontal dependencies and vertical dynamics are both indispensable for characterizing the hallucination space.

Table 4: Edge-type ablation with Qwen3-VL (AUPR).

Setting	Cross-modal	Context	Cross-layer evolution	ViGoR (Target)
S1	✗	✓	✓	0.55
S2	✓	✗	✓	0.71
S3	✓	✓	✗	0.70
S4	✓	✓	✓	**0.77**

Graph Depth and Readout. In Table 5, we investigate the temporal depth and the readout pattern of our graph. Constructing the graph using only the last layer yields a suboptimal AUPR of 0.70, whereas expanding the graph depth to 8 layers maximizes performance (0.77), confirming that trajectories are more informative than static snapshots. Besides, integrating the token-level readout (LSE pooling) not only enables fine-grained interpretability but also strictly improves or maintains the macroscopic sentence-level detection accuracy compared to a naive layer-level information interaction from the last response token.

Table 5: Graph depth and readout pattern ablation on M-HalDetect and ViGoR (AUPR).

Model	Level	**M-HalDetect** (Source)	**ViGoR** (Target)
Qwen3-VL	*1 layer (last)*	0.82	0.70
	4 layers	0.84	0.75
	8 layers	**0.85**	**0.77**
LLaVA-1.5	*layer*	**0.80**	0.70
	layer+token	**0.80**	**0.71**
Qwen3-VL	*layer*	0.83	0.76
	layer+token	**0.85**	**0.77**

Hyperparameter Sensitivity: Attention Threshold and MIL Sharpness. We further investigate the robustness of TRACE to two critical hyperparameters, with results visualized in Figure 4.

First, the contextual attention threshold τ controls the sparsity of contextual edges. As shown in Figure 4(a), a permissive threshold retains excessive linguistic noise and spurious correlations, while aggressive pruning ($\tau \geq 0.1$) leads to a sharp decline in performance. Setting $\tau = 0.05$ elegantly resolves this dilemma, achieving $86\% \sim 97\%$ sparsity while maintaining the detection capability.

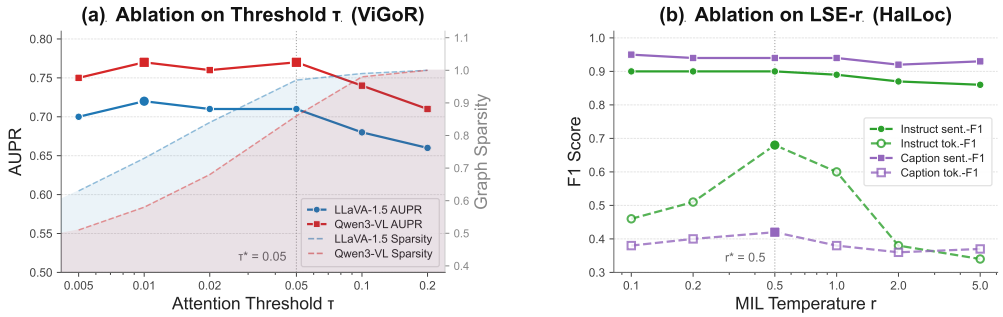


Figure 4: Ablation results of attention threshold τ and MIL temperature r.

Second, the MIL temperature parameter r modulates the sharpness of gradient routing during optimization. As illustrated in Figure 4(b), extreme values inherently degrade fine-grained localization. A small r approaches mean pooling, heavily diluting the hallucination signals. Conversely, a large r approximates hard max pooling, which severely concentrates gradients on a single peak token and thus fails to capture contiguous hallucination spans. Setting $r = 0.5$ yields an ideal effect.

In summary, our ablation experiments confirm that TRACE's superior performance stems from the holistic interplay of its spatio-temporal edge topologies, deep trajectory modeling, and fine-grained MIL readouts, while remaining robust under reasonable hyperparameter configurations ($\tau = 0.05, r = 0.5$). For more comprehensive ablation studies, we refer the reader to Appendix A.6.

5 Limitations and Future Work

Despite its strong empirical efficacy, TRACE has several limitations that suggest promising avenues for future research. First, processing extremely long text spans incurs non-negligible computational overhead, because storing internal representations across multiple layers requires memory that grows with sequence length. Future work will explore efficient long-context compression mechanisms to alleviate this memory overload. Second, TRACE requires white-box access to the target VLM, which precludes its use with closed-source models accessible only through APIs. Finally, while sentence-supervised token scores provide strong localization signals, they may occasionally miss subtle sub-spans within a hallucinated segment, and silent omissions remain challenging to pinpoint due to the absence of explicit target tokens. Building upon these insights, a key direction is integrating TRACE's diagnostic signals into real-time decoding pipelines to actively mitigate hallucinations in downstream applications.

6 Conclusion

In this paper, we present TRACE, a non-intrusive probing framework that re-conceptualizes VLM hallucination detection as a graph learning problem over internal *trajectory drifts*. By leveraging a Multiple Instance Learning (MIL) readout mechanism, TRACE effectively bridges the granularity gap, distilling fine-grained image-conditioned token-risk attributions from macroscopic sentence-level supervision to achieve state-of-the-art performance. Although deployment must guard against *automation bias*—where users might over-rely on automated verifiers and develop a false sense of security—our framework demonstrates the strong potential of cross-layer evolution graphs in enhancing the interpretability, reliability, and safety of Vision-Language Models.

Acknowledgments and Disclosure of Funding

This work was supported in part by New Generation Artificial Intelligence-National Science and Technology Major Project 2026ZD0128001, in part by the Natural Science Foundation of China under Grant 62625202, in part by National Science and Technology Major Project 2025ZD1601300, in part by Grant 2023-JCJQ-LA-001-088.

References

[1] Hervé Abdi and Lynne J Williams. Principal component analysis. *Wiley Interdisciplinary Reviews: Computational Statistics*, 2(4):433–459, 2010.

[2] 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 Computer Vision and Pattern Recognition Conference*, pages 29915–29926, 2025.

[3] Xiang An, Yin Xie, Kaicheng Yang, Wenkang Zhang, Xiuwei Zhao, Zheng Cheng, Yirui Wang, Songcen Xu, Changrui Chen, Didi Zhu, et al. Llava-onevision-1.5: Fully open framework for democratized multimodal training. *arXiv preprint arXiv:2509.23661*, 2025.

[4] Amos Azaria and Tom Mitchell. The internal state of an llm knows when it's lying. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 967–976, 2023.

[5] Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. Qwen technical report. *arXiv preprint arXiv:2309.16609*, 2023.

[6] Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. Qwen-vl: A versatile vision-language model for understanding, localization. *Text Reading, and Beyond*, 2(1):1, 2023.

[7] Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*, 2025.

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

[9] Yonatan Belinkov. Probing classifiers: Promises, shortcomings, and advances. *Computational Linguistics*, 48(1):207–219, 2022.

[10] Assaf Ben-Kish, Moran Yanuka, Morris Alper, Raja Giryes, and Hadar Averbuch-Elor. Mitigating open-vocabulary caption hallucinations. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 22680–22698, 2024.

[11] Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. A large annotated corpus for learning natural language inference. In Lluís Màrquez, Chris Callison-Burch, and Jian Su, editors, *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal, September 2015. Association for Computational Linguistics.

[12] Marc-André Carbonneau, Veronika Cheplygina, Eric Granger, and Ghyslain Gagnon. Multiple instance learning: A survey of problem characteristics and applications. *Pattern Recognition*, 77:329–353, 2018.

[13] Chao Chen, Kai Liu, Ze Chen, Yi Gu, Yue Wu, Mingyuan Tao, Zhihang Fu, and Jieping Ye. Inside: Llms' internal states retain the power of hallucination detection. *arXiv preprint arXiv:2402.03744*, 2024.

[14] 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 *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3235–3252, 2024.

[15] Wei-Lin Chiang, Zhuohan Li, Ziqing Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E Gonzalez, et al. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality. *See https://vicuna.lmsys.org (accessed 14 April 2023)*, 2(3):6, 2023.

[16] Yung-Sung Chuang, Linlu Qiu, Cheng-Yu Hsieh, Ranjay Krishna, Yoon Kim, and James Glass. Lookback lens: Detecting and mitigating contextual hallucinations in large language models using only attention maps. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 1419–1436, 2024.

[17] Jinhao Duan, Fei Kong, Hao Cheng, James Diffenderfer, Bhavya Kailkhura, Lichao Sun, Xiaofeng Zhu, Xiaoshuang Shi, and Kaidi Xu. Truthprint: Mitigating large vision-language models object hallucination via latent truthful-guided pre-intervention. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7372–7382, 2025.

[18] Alessandro Favero, Luca Zancato, Matthew Trager, Siddharth Choudhary, Pramuditha Perera, Alessandro Achille, Ashwin Swaminathan, and Stefano Soatto. Multi-modal hallucination control by visual information grounding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14303–14312, 2024.

[19] Fabrizio Frasca, Guy Bar-Shalom, Yftah Ziser, and Haggai Maron. Neural message-passing on attention graphs for hallucination detection. *arXiv preprint arXiv:2509.24770*, 2025.

[20] Anisha Gunjal, Jihan Yin, and Erhan Bas. Detecting and preventing hallucinations in large vision language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 18135–18143, 2024.

[21] Chi Han, Jialiang Xu, Manling Li, Yi Fung, Chenkai Sun, Nan Jiang, Tarek Abdelzaher, and Heng Ji. Word embeddings are steers for language models. *arXiv preprint arXiv:2305.12798*, 2023.

[22] Pengcheng He, Jianfeng Gao, and Weizhu Chen. Debertav3: Improving deberta using electra-style pre-training with gradient-disentangled embedding sharing. *arXiv preprint arXiv:2111.09543*, 2021.

[23] 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*, pages 13418–13427, 2024.

[24] Drew A Hudson and Christopher D Manning. Gqa: A new dataset for real-world visual reasoning and compositional question answering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6700–6709, 2019.

[25] Maximilian Ilse, Jakub Tomczak, and Max Welling. Attention-based deep multiple instance learning. In *International Conference on Machine Learning*, pages 2127–2136. PMLR, 2018.

[26] Andrew Jaegle, Felix Gimeno, Andy Brock, Oriol Vinyals, Andrew Zisserman, and Joao Carreira. Perceiver: General perception with iterative attention. In *International Conference on Machine Learning*, pages 4651–4664. PMLR, 2021.

[27] Liqiang Jing, Ruosen Li, Yunmo Chen, and Xinya Du. Faithscore: Fine-grained evaluations of hallucinations in large vision-language models. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 5042–5063, 2024.

[28] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International Journal of Computer Vision*, 123(1):32–73, 2017.

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

[30] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *International conference on machine learning*, pages 12888–12900. PMLR, 2022.

[31] Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Wayne Xin Zhao, and Ji-Rong Wen. Evaluating object hallucination in large vision-language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 292–305, 2023.

[32] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European Conference on Computer Vision*, pages 740–755. Springer, 2014.

[33] Hanchao Liu, Wenyuan Xue, Yifei Chen, Dapeng Chen, Xiutian Zhao, Ke Wang, Liping Hou, Rongjun Li, and Wei Peng. A survey on hallucination in large vision-language models. *arXiv preprint arXiv:2402.00253*, 2024.

[34] 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*, pages 26296–26306, 2024.

[35] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in Neural Information Processing Systems*, 36:34892–34916, 2023.

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

[37] Vedant Palit, Rohan Pandey, Aryaman Arora, and Paul Pu Liang. Towards vision-language mechanistic interpretability: A causal tracing tool for blip. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2856–2861, 2023.

[38] Eunkyu Park, Minyeong Kim, and Gunhee Kim. Halloc: Token-level localization of hallucinations for vision language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 29893–29903, 2025.

[39] Kiho Park, Yo Joong Choe, and Victor Veitch. The linear representation hypothesis and the geometry of large language models. *arXiv preprint arXiv:2311.03658*, 2023.

[40] Seongheon Park, Xuefeng Du, Min-Hsuan Yeh, Haobo Wang, and Yixuan Li. Steer llm latents for hallucination detection. *arXiv preprint arXiv:2503.01917*, 2025.

[41] Vishnu Prabhakaran, Purav Aggarwal, Vinay Kumar Verma, Gokul Swamy, and Anoop Saladi. Vade: Visual attention guided hallucination detection and elimination. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 14949–14965, 2025.

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

[43] 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*, pages 4035–4045, 2018.

[44] Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. The graph neural network model. *IEEE Transactions on Neural Networks*, 20(1):61–80, 2008.

[45] Zhiqing Sun, Sheng Shen, Shengcao Cao, Haotian Liu, Chunyuan Li, Yikang Shen, Chuang Gan, Liangyan Gui, Yu-Xiong Wang, Yiming Yang, et al. Aligning large multimodal models with factually augmented rlhf. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 13088–13110, 2024.

[46] Jesse Vig and Yonatan Belinkov. Analyzing the structure of attention in a transformer language model. In *Proceedings of the 2019 ACL workshop BlackboxNLP: analyzing and interpreting neural networks for NLP*, pages 63–76, 2019.

[47] Sudong Wang, Yunjian Zhang, Yao Zhu, Jianing Li, Zizhe Wang, Yanwei Liu, and Xiangyang Ji. Towards understanding how knowledge evolves in large vision-language models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 29858–29868, 2025.

[48] Adina Williams, Nikita Nangia, and Samuel R. Bowman. A broad-coverage challenge corpus for sentence understanding through inference. In Marilyn Walker, Heng Ji, and Amanda Stent, editors, *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122, New Orleans, Louisiana, June 2018. Association for Computational Linguistics.

[49] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and Philip S Yu. A comprehensive survey on graph neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 32(1):4–24, 2020.

[50] Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. Efficient streaming language models with attention sinks. *arXiv preprint arXiv:2309.17453*, 2023.

[51] Junjie Xu, Zongyu Wu, Minhua Lin, Xiang Zhang, and Suhang Wang. Llm and gnn are complementary: Distilling llm for multimodal graph learning. In *2025 IEEE International Conference on Big Data (BigData)*, pages 119–124. IEEE, 2025.

[52] Siming Yan, Min Bai, Weifeng Chen, Xiong Zhou, Qixing Huang, and Li Erran Li. Vigor: Improving visual grounding of large vision language models with fine-grained reward modeling. In *European Conference on Computer Vision*, pages 37–53. Springer, 2024.

[53] Le Yang, Ziwei Zheng, Boxu Chen, Zhengyu Zhao, Chenhao Lin, and Chao Shen. Nullu: Mitigating object hallucinations in large vision-language models via halluspace projection. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 14635–14645, 2025.

[54] Jie Zhou, Ganqu Cui, Shengding Hu, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun. Graph neural networks: A review of methods and applications. *AI Open*, 1:57–81, 2020.

A Technical appendices and supplementary material

A.1 Datasets

M-HalDetect M-HalDetect [20] is a fine-grained multimodal hallucination detection dataset built on images from the MS COCO 2014 dataset [32]. It contains 16,000 image-prompt-response samples (12,800 for training and 3,200 for validation). The responses consist of detailed image descriptions segmented into text chunks of varying lengths. Expert annotators classify each segment into four categories: *Accurate*, *Analysis*, *Inaccurate*, and *Unsure*. For the hallucination detection classification task in our experiments, we systematically re-categorize these labels by merging the *Analysis* class into *Accurate*, and the *Unsure* class into *Inaccurate*. This segment-level annotation makes it highly suitable for evaluating complex, detailed hallucinations in Vision-Language Models.

ViGoR The ViGoR [52] (Visual Grounding through Reward Modeling) dataset is designed to evaluate and enhance the visual grounding capabilities of VLMs. It comprises 15,400 samples based on 7,200 images from the MS COCO 2017 dataset [32]. Each image is paired with diverse instruction prompts and paragraph-length, detailed responses. Human annotators provide rigorous sentence-level evaluations, categorizing each sentence based on its correctness relative to the image and its creativity. We primarily utilize ViGoR to evaluate the out-of-distribution (OOD) transferability of hallucination detection methods.

HalLoc HalLoc [38] is a large-scale, token-level annotated dataset specifically designed for fine-grained hallucination localization in VLMs. It features approximately 155,000 samples spanning three core vision-language tasks: Visual Question Answering (VQA), instruction-following, and image captioning. Notably, the foundational data and scene graphs for constructing HalLoc are heavily derived from the Visual Genome dataset [28] and its derivative GQA [24], ensuring rich and dense semantic grounding. Notably, the HalLoc benchmark features fine-grained token-level annotations across four distinct hallucination categories: objects, attributes, relationships, and scene types. To align with our weakly-supervised setting, we derive coarse sentence-level labels from this dataset by flagging a sentence as hallucinatory if it contains at least one hallucinated token.

A.2 Models

LLaVA-1.5 LLaVA-1.5 [34] (specifically the `llava-1.5-7b` version) is a prominent fully-autoregressive VLM that leverages Vicuna-7B [15] as its large language model backbone and CLIP-ViT-L-336px [42] as its vision encoder. Structurally, its LLM backbone consists of 32 hidden layers, a hidden state dimension of 4096, and 32 attention heads. To bridge the modalities, it employs a simple two-layer MLP cross-modal connector, which projects the visual features directly into the word embedding space of the language model. The concatenated visual and textual tokens are then processed by the LLM, demonstrating strong zero-shot instruction-following capabilities and visual reasoning.

Qwen-VL Qwen-VL [6] (specifically the `Qwen-VL-Chat` 7B version) is a highly capable multi-modal model based on the Qwen-7B [5] language model and OpenCLIP's ViT-bigG [42] vision encoder. Similar to LLaVA, the language model backbone features 32 transformer layers, a hidden dimension of 4096, and 32 attention heads. To connect the modalities, it employs a Perceiver Resampler module [26]. This resampler effectively compresses the variable-length visual feature sequence into a fixed length of 256 visual tokens, which are subsequently integrated with the language model. It excels in diverse tasks including image captioning, fine-grained localization, and text reading.

Qwen3-VL Qwen3-VL [7] (specifically the `Qwen3-VL-8B-Instruct` version) represents the latest iteration in the Qwen vision-language model series, offering comprehensive architectural upgrades. Structurally, its language model backbone is deeper than its predecessors, consisting of 36 transformer layers, a hidden state dimension of 4096, and 32 attention heads. It introduces advanced components such as DeepStack, which fuses multi-level ViT features for fine-grained image-text alignment, and Interleaved-MRoPE for robust spatial and temporal positional embeddings [7]. Supporting both Dense and Mixture-of-Experts (MoE) architectures, Qwen3-VL natively supports ultra-long contexts and achieves state-of-the-art performance in multimodal reasoning and visual agent interactions.

A.3 Full Implementation Details

To facilitate full reproducibility of our TRACE framework, we detail the graph construction process, network architecture, and training hyperparameters below in Table 6.

Table 6: Summary of default hyperparameters used in our framework.

Hyperparameter	**Value**
Graph & Network Architecture	
Sampled Transformer Layers (K)	8
Top-k Attention Edges (k)	5
GNN Hidden Dimension (d_{hidden})	256
GNN Message Passing Layers (M)	8
GAT Attention Heads	4
Dropout Rate	0.1
Training & Optimization	
MIL LSE Temperature (r)	0.5
Batch Size	32
Training Epochs	$15 \sim 20$
Optimizer	AdamW
Initial Learning Rate	$2e{-}3$
Minimum Learning Rate	$2e{-}5$
Weight Decay	$1e{-}4$
Learning Rate Schedule	Cosine Annealing
Max Gradient Norm (Clipping)	1.0

Graph Construction and Architecture. The spatio-temporal graph is constructed by extracting hidden states from $K = 8$ uniformly sampled transformer layers. The input dimensions dynamically adapt to the backbone's native hidden size (e.g., 4096) and are linearly projected into a unified graph working space of $d_{hidden} = 256$. For cross-modality edges, we restrict the attention connectivity by selecting the top-k attention weights from each text token to visual tokens ($k = \min(5, N_{vis})$). To process the scalar edge attributes, the feature extractor employs a M-layer Graph Attention Network with edge-conditioned scalar gates, GAT^{edge}. The message-passing processing becomes as follows:

$$h'_i = \Sigma_j \alpha_{ij} \cdot g(e_{ij}) \cdot W h_j, \tag{7}$$

where α_{ij} is the attention score derived in a same way as GAT, and $g(\cdot) = \sigma\left(\mathrm{MLP}(\cdot)\right)$ denotes the edge-conditioned scalar gate.

Finally after M round of message passing, only text nodes corresponding to the deepest sampled layer are routed to the Prediction Head, which is instantiated as a two-layer MLP: Linear$(256, 128) \rightarrow$ GELU $\rightarrow$ Dropout$(0.1) \rightarrow$ Linear$(128, 1)$.

Optimization and Hyperparameters. The model is trained end-to-end using the AdamW optimizer. We employ a cosine annealing learning rate scheduler (`CosineAnnealingLR`) that smoothly decays the learning rate to 1% of its initial value.

A.4 Conditional Gradient Analysis of the MIL Readout

Inspired by Multiple Instance Learning (MIL) [12, 25], treating features of each response as a *bag* $X = \{\hat{\mathbf{f}}_0, \ldots, \hat{\mathbf{f}}_{n-1}\}$ with bag label $y \in \{0, 1\}$, we propose a token-first formulation that distills sentence-level supervision into token-level risk attribution.

Assumption 1 (*Feature Separability*)**.** *We assume the graph-augmented representations $\hat{\mathbf{f}}_i$ exhibit regional separability. Specifically, factual and hallucinatory representations naturally map to distinct manifolds in the cross-modality, contextual and cross-layer feature space, allowing a continuous mapping MLP_{pred} to satisfy $\mathbb{E}[z_i \mid \mathbf{f}_{hallu}] > \mathbb{E}[z_i \mid \mathbf{f}_{fact}]$, where $z_i = \mathrm{MLP}_{pred}(\hat{\mathbf{f}}_i)$.*

* *This separability is an empirical condition on learned representations.*

Token-first readout. Instead of pooling before predicting, MLP_{pred} is applied at the *token level*. The per-token logits z_i are aggregated via Log-Sum-Exp (LSE) to compute the sentence-level logit:

$$z_{graph} = \frac{1}{r} \log \left(\frac{1}{n} \sum_{i=0}^{n-1} \exp(r\, z_i) \right), \tag{8}$$

where $r > 0$ is a temperature parameter. As $r \to \infty$, LSE recovers hard max-pooling; at r near 1 it provides a smooth, differentiable approximation of the maximum.

Conditional gradient analysis. To demonstrate why this mechanism converges to token-level risk attribution under Assumption 1, we analyze the gradient of the sentence-level BCE loss $\mathcal{L}$ with respect to token logit z_i:

$$\frac{\partial \mathcal{L}}{\partial z_i} = (\hat{y} - y) \cdot \underbrace{\frac{\exp(r\, z_i)}{\sum_j \exp(r\, z_j)}}_{p_i^{(r)}}, \tag{9}$$

where $\hat{y} = \sigma(z_{graph})$ and $p_i^{(r)}$ is the softmax distribution over token logits at temperature r. This gradient acts as an adaptive routing mechanism:

- **Positive Bag ($y = 1$):** By Assumption 1, true hallucinatory tokens inherently produce higher initial z_i. The softmax weighting $p_i^{(r)}$ concentrates the positive gradient signal almost exclusively on these tokens, amplifying their confidence while leaving normal tokens un-updated.
- **Negative Bag ($y = 0$):** Since no hallucinations exist, the term $(\hat{y} - 0)$ penalizes any token with a spuriously high logit, driving the entire normal subspace toward zero.

Consequently, under ideal conditions, this bottleneck forces token-level specialization without requiring explicit dense annotations.

Inference and supervised extension. During inference, token-level hallucination scores are obtained directly as $s_i = \sigma(z_i)$, preserving identical computational paths to avoid distribution shifts. Furthermore, when dense annotations are available, this architecture seamlessly supports fully-supervised training by directly applying token-level BCE to each z_i, enabling fair comparisons between weak and full supervision.

A.5 Empirical Validation of Trajectory Drift

To substantiate the conceptual motivation presented in Figure 1 (in the main text), we conduct an empirical analysis using real internal hidden states extracted from LLaVA-1.5. Specifically, we sample a subset of factual and hallucinatory tokens and apply detrended Principal Component Analysis (PCA) [1] to project their high-dimensional representations and cross-layer evolution trajectories into a 2D space.

As illustrated in Figure 5(a)-(c), when viewed from a static, single-layer perspective (e.g., early layer $L = 4$, middle layer $L = 20$, and deep layer $L = 32$), the representations of normal and hallucinatory tokens are highly entangled. This static overlap severely blurs the decision boundary, validating our claim that relying solely on static features for hallucination detection is suboptimal.

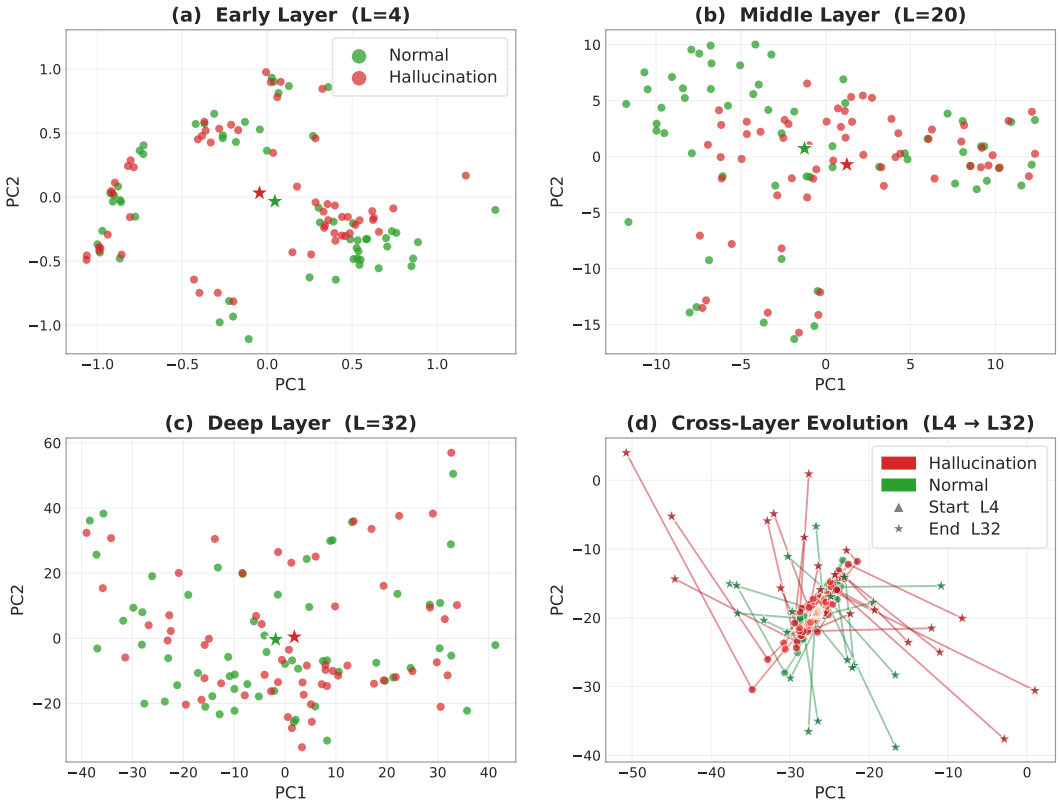


Figure 5: **Empirical visualization of static representations versus dynamic cross-layer trajectories.** We project the internal hidden states of factual (green) and hallucinatory (red/star) tokens into a 2D space using detrended Principal Component Analysis (PCA). **(a)-(c)** In static single-layer snapshots—such as the early layer ($L = 4$), middle layer ($L = 20$), and deep layer ($L = 32$)—the distributions of normal and hallucinatory tokens exhibit severe semantic overlap, making static classification challenging. **(d)** By tracking the cross-layer evolution of individual tokens from $L = 4$ to $L = 32$, every four layers, we observe that hallucinatory tokens undergo more distinct and dramatic trajectory mutations, empirically validating our motivation for dynamic spatio-temporal graph modeling.

In striking contrast, Figure 5(d) visualizes the dynamic cross-layer evolution of these tokens. Compared to factual tokens, which generally follow a relatively stable computational path, hallucinatory tokens exhibit more prominent *trajectory drifts*—characterized by sudden and erratic mutations in the cross-layer evolution of the hidden states ($L4 \rightarrow \cdots \rightarrow L32$). This real-world empirical evidence directly corroborates our core insight: hallucinations are fundamentally evolutionary anomalies, underscoring the necessity of the proposed TRACE framework that captures these dynamics via spatio-temporal graph learning.

A.6 Additional Results

A.6.1 Detailed Ablation Data

In this section, Tables 7 and 8 provide the exact numerical data corresponding to the ablation curves (Figure 4) discussed in the main text. Specifically, in Table 7, we additionally report the absolute number of retained contextual edges (*Num. Edges*). As observed, increasing the threshold τ substantially reduces the number of horizontal edges. The data confirm that $\tau = 0.05$ effectively filters out the vast majority of redundant connections (e.g., reducing edges from hundreds to around 44 for LLaVA-1.5) while maintaining optimal detection performance.

Table 7: Ablation on the contextual attention threshold (τ) evaluated on ViGoR. $\tau = 0.05$ provides the optimal trade-off between graph sparsity and detection performance (AUPR).

Threshold (τ)	**LLaVA-1.5**			**Qwen3-VL**		
	Num. Edges	Sparsity	AUPR	Num. Edges	Sparsity	AUPR
0.005	775.4	0.63	0.70	835.5	0.51	0.75
0.010	521.2	0.73	**0.72**	667.0	0.58	**0.77**
0.020	277.9	0.84	<u>0.71</u>	457.4	0.68	<u>0.76</u>
0.050	44.6	0.97	<u>0.71</u>	177.3	0.86	**0.77**
0.100	3.4	0.99	0.68	21.8	0.98	0.74
0.200	0.0	1.00	0.66	0.0	1.00	0.71

Table 8: Ablation on the MIL temperature parameter r across Instruct and Caption subsets. The optimal performance is achieved at $r = 0.5$, which achieves an ideal balance.

LSE-r	**Instruct**		**Caption**	
	sent.-F1	*tok.*-F1	*sent.*-F1	*tok.*-F1
0.1	0.90	0.46	**0.95**	0.38
0.2	0.90	0.51	0.94	0.40
0.5	**0.90**	**0.68**	0.94	**0.42**
1.0	0.89	0.60	0.94	0.38
2.0	0.87	0.38	0.92	0.36
5.0	0.86	0.34	0.93	0.37

A.6.2 Diverse VLM Architectures

To further validate our framework, we extend our evaluation to a broader range of VLM architectures, encompassing different models. Table 9 summarizes the sentence-level detection performance on both the in-domain M-HalDetect(Source) and out-of-distribution ViGoR(Target) benchmarks.

Table 9: Performance on M-HalDetect and ViGoR with different VLM models

Model	**M-HalDetect**(Source)	**ViGoR**(Target)
LLaVA-1.5 [34]	0.80	0.71
LLaVA-OV [29]	<u>0.85</u>	0.76
LLaVA-OV-1.5 [3]	0.83	0.75
Qwen-VL [6]	0.81	0.72
Qwen2.5-VL [8]	0.84	<u>0.76</u>
Qwen3-VL [7]	**0.85**	**0.77**

The results demonstrate that TRACE consistently provides strong hallucination detection capabilities regardless of the underlying VLM backbone. Notably, as the representation quality and internal alignment of the base models improve across generations, the probing performance of TRACE scales positively. As expected, newer architectures (e.g., Qwen3-VL) and models integrating richer visual contexts (e.g., LLaVA-OV) yield stronger intrinsic representations; nevertheless, our framework maintains robust and stable detection efficacy across this diverse spectrum of models.

A.6.3 Top-Token Alignment

To evaluate localization precision, we measure Hit@1 on all positive long-form HalLoc test examples, defined as whether the highest-scored token falls within an annotated hallucination span (Table 10). Chance performance corresponds to the empirical gold-token fraction. VQA is omitted due to its short response length, which renders Hit@1 uninformative.

Sentence-level supervision substantially exceeds chance across both subsets, demonstrating that global response labels provide strong signals for directing attention toward hallucinated spans. Notably, non-gold top predictions (often punctuation or functional tokens) decrease markedly when

Table 10: Top-token alignment (Hit@1) on positive HalLoc responses.

Task	Chance	Sentence-Supervised	Token-Supervised
Caption ($n = 3{,}273$)	0.054	0.457	0.634
Instruct ($n = 5{,}009$)	0.316	0.589	0.933

token-level supervision is introduced, confirming that direct token supervision effectively sharpens readout alignment.

A.6.4 Sparse Token Supervision.

We further assess data efficiency by varying token-label availability on a fixed 4,000-train / 1,000-validation Caption subset (Table 11). Without token labels (0%), the model achieves 0.359 F1 (compared to 0.42 under full benchmark settings in Table 3).

Table 11: Impact of token-label availability on Caption localization performance.

Token-Labeled Fraction	Token AUPR	Token F1
0% (0)	0.249	0.359
1% (40)	0.283	0.379
5% (200)	0.475	0.493
25% (1,000)	0.529	0.538
100% (4,000)	0.571	0.566

Crucially, incorporating even a tiny fraction of token labels yields significant gains—just 5% token supervision captures the majority of full-supervision performance (0.475 vs. 0.571 AUPR), demonstrating excellent annotation efficiency.

A.6.5 Error Operations

We evaluate TRACE across distinct error operations using dedicated test subsets.

Table 12: Response-level pair accuracy across error operations.

Error Operation	Pair Accuracy
Substitution	VQA: 0.862; MME count/color/position: 0.848
Polarity/negation	VQA: 0.840; Instruct: 0.859
Distributed multi-span	Instruct ($n = 2{,}026$ pairs): 0.898
Object existence	Commission: 0.645; False-negative: 0.940
Spatial relation swap	0.645

As shown in Table 12, pair accuracy is higher for explicit substitutions and polarity inversions than for object commissions and spatial relation swaps. The gap between commission (0.645) and false-negative claims (0.940) reflects an asymmetric sensitivity to object existence. Besides, *silent omissions* offer no target tokens to mark, fundamentally requiring a separate coverage mechanism beyond token grounding.

A.6.6 Category-Dependent Drift and Graph Scope

We analyze all 3,273 positive HalLoc-Caption test responses (2,483 image clusters; 11,998 annotated hallucination and 230,648 factual tokens). For token i, define total adjacent-layer cosine distance $D_i = \sum_{k=2}^{8}[1 - \cos(h_i^{l_{k-1}}, h_i^{l_k})]$. Table 13 reports the hallucinated-minus-factual token-weighted mean within category-tagged positive responses. Categories are assigned at the response level and may overlap, so the differences describe associations rather than causal attributions.

Across all four categories, annotated hallucination tokens have a larger mean total adjacent-layer cosine distance than factual tokens. Object and attribute differences are more diffuse and peak later; relationship and scene differences are more concentrated and peak earlier. Table 4 supports

Table 13: Category-conditioned differences in total adjacent-layer cosine distance.

Category	Difference in D_i	Pattern / strongest interval
Object	+0.021	Diffuse, later / $L_6 \to L_7$
Attribute	+0.022	Diffuse, later / $L_6 \to L_7$
Relationship	+0.092	Concentrated, earlier / $L_1 \to L_2$
Scene	+0.026	Concentrated, earlier / $L_1 \to L_2$

complementary roles for the three edge types, while region-level visual edges and head-specific variants remain open directions.

Empirically, across 1,000 Qwen3-VL graphs ($K = 8$, median $T = 14$), the median graph had 120 nodes and 384.5 retained edges (98 evolution, 174.5 context, 112 cross-modal). Worst-case context scaling remains $O(KT^2)$.

A.6.7 Latency and Memory Protocol

We profile 73 length-stratified responses (6–109 response tokens) at batch size one, with three repetitions per path on an otherwise-idle NVIDIA RTX 3090 GPU. TRACE captures only the states and attention regions needed for graph construction. "Reported" includes the post-response stage for external verifiers and the generator plus detector for VADE and TRACE; "native added" measures paired incremental computation. Resident GPU denotes additional model weights, working GPU denotes temporary allocation, and total GPU is their sum. Latency and memory are reported as per-response p50/p95 values.

Table 14: Same-hardware detector profile. Values are p50/p95 except resident GPU.

Pipeline	Reported (ms)	Native added (ms)	Resident (MiB)	Working (MiB)	Total added (MiB)
CLIP probe	24.5/36.1	24.5/36.1	824.6	17.4/17.4	842.0/842.0
BLIP + NLI	163.9/207.4	163.9/207.4	1274.1	128.5/130.4	1402.5/1404.5
Qwen2.5-VL-7B judge	217.9/255.9	217.9/255.9	15865.5	87.6/103.1	15953.2/15968.6
Qwen3-VL-8B judge	174.5/205.7	174.5/205.7	0	70.2/87.2	70.2/87.2
VADE	204.5/238.2	4.0/5.6	19.1	12.9/17.9	32.1/37.0
TRACE	204.0/248.6	15.2/21.0	20.3	8.3/33.5	28.6/53.8

For Qwen3-VL-8B, VLM-only p50/p95 latency is 190.0/225.6 ms, capture plus graph construction takes 193.9/227.2 ms (paired overhead 3.0/6.4 ms), and the full detector path takes 204.0/248.6 ms (paired addition 15.2/21.0 ms). For LLaVA-1.5-7B, VLM-only latency is 204.9/241.9 ms and capture plus graph construction takes 206.3/242.4 ms (paired overhead 2.8/5.2 ms); full detector inference was not profiled. These measurements cover responses up to 109 tokens. Offline graph serialization adds storage cost, while online inference need not retain every graph.

A.7 Compared External Methods

Table 2 compares four output-level verifiers on ViGoR. Frozen CLIP similarity and BLIP-generated image evidence followed by DeBERTa-NLI use separate models without accessing the generator's internal states. The supervised CLIP interaction probe adds learned parameters, while the Qwen3-VL-8B verifier reuses the generator family for an additional forward pass. These approaches differ in training requirements and inference cost.

A.8 Additional Visualization Cases

In this section, we present additional results to demonstrate the hallucination localization capabilities of our framework (Figure 6 and 7). For each sample, to intuitively illustrate the dual-granularity outputs of our framework, we report the macroscopic Sentence Hallucination Probability using a percentage bar, followed by the generated text where Response Token Scores are mapped to a color gradient. Tokens with higher predicted hallucination risks are highlighted in darker shades of red. This intuitive visualization confirms that TRACE successfully distills quite precise token-level anomaly onsets directly from coarse-grained supervision.

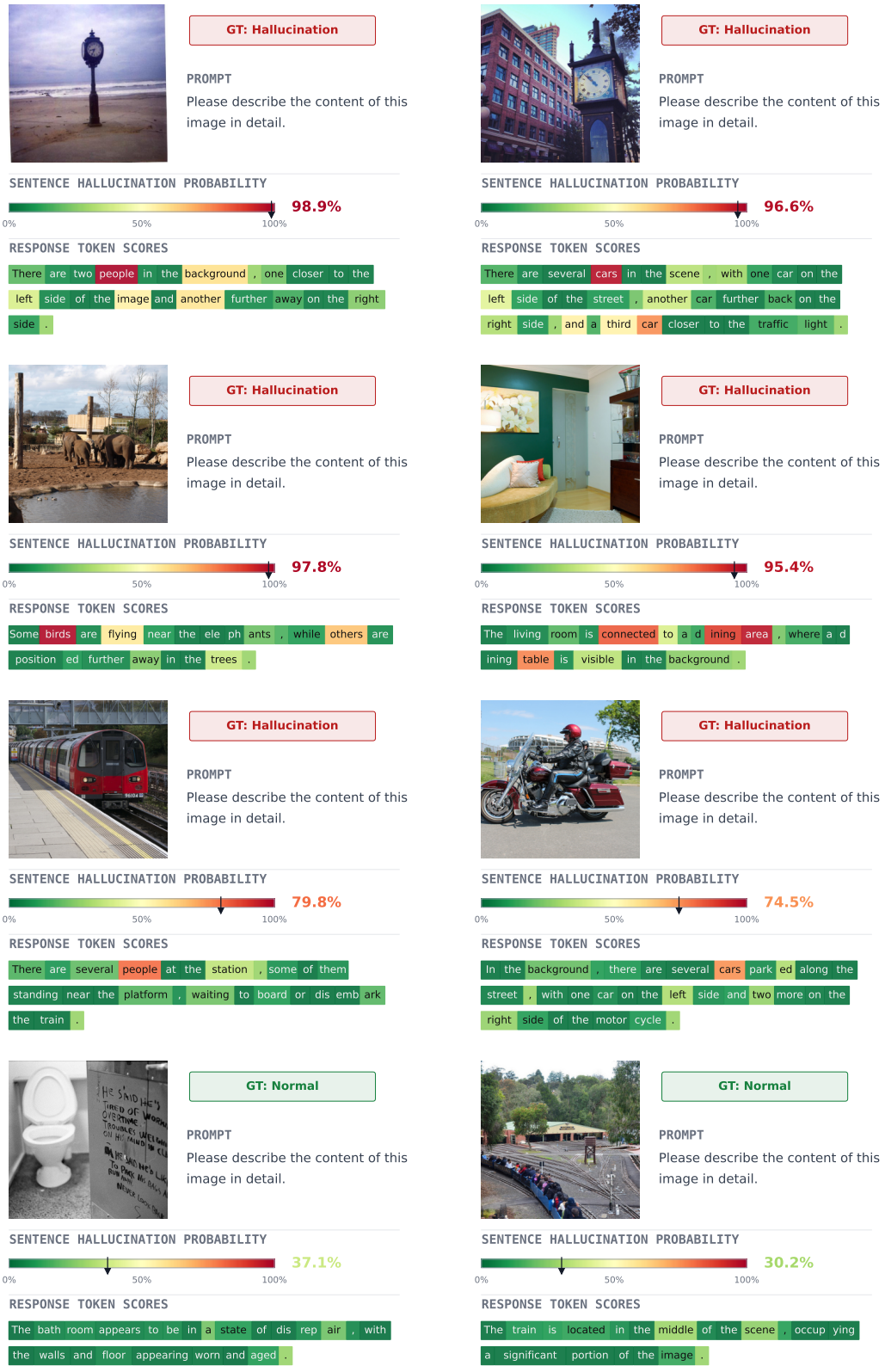


Figure 6: **Results Visualization on LLaVA-1.5.** Additional hallucination detection and localization samples utilizing the LLaVA-1.5 backbone. TRACE effectively captures trajectory drifts, precisely highlighting isolated fabricated tokens (e.g., objects or attributes) without requiring dense token-level annotations.

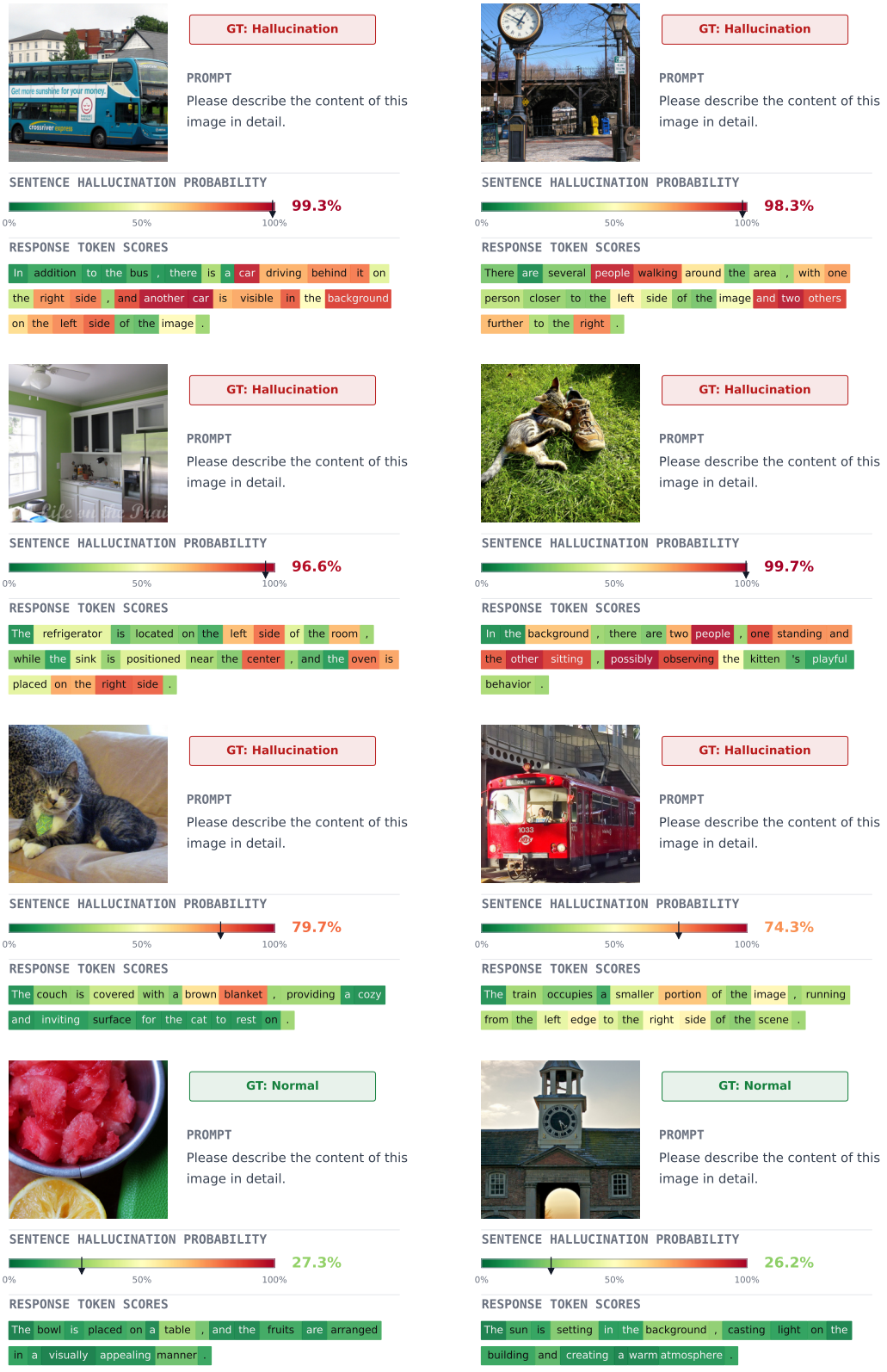


Figure 7: **Results Visualization on Qwen3-VL.** TRACE maintains high localization accuracy on the advanced Qwen3-VL backbone. Notably, it tends to highlight contiguous hallucinated spans rather than isolated words as LLaVA-1.5, which naturally aligns with the semantic ambiguity of defining exact hallucination boundaries.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction explicitly state our core contributions: modeling hallucination detection via spatio-temporal graphs (TRACE) and achieving token-level localization via a weakly-supervised MIL readout. These claims perfectly align with our experimental validation across diverse VLM architectures.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Section 5 explicitly discusses the limitations of our framework, noting its reliance on coarse sentence-level supervision and its current scope as a post-hoc detector. We also outline how future work will address these constraints by integrating real-time mitigation pipelines.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: We explicitly state the theoretical premise in Appendix A.4 as Assumption 1. This is accompanied by a gradient-based proof sketch to provide intuition for how our MIL formulation converges to accurate weakly-supervised localization.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We provide comprehensive details regarding datasets, VLM backbones, baseline implementations, and the hyperparameters in the Experiments section and Appendix A.3.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: We use the publicly available M-HalDetect, ViGoR, and HalLoc datasets/benchmarks. Our code, together with instructions for data preparation, training, and evaluation, is publicly available at `https://github.com/Shaoxuan-Li/TRACE`.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Section 4.1 and Appendix A.3 provide detailed settings, including hyperparameters and experimental conditions, to understand the results. Details of dataset processing are also provided in the Experiments section and Appendix A.1.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

Justification: Due to the substantial computational cost of extracting dense hidden states, our main results are reported based on a single training run with a fixed random seed. However, the consistent performance across multiple benchmarks and diverse architectures substantiates the robustness of our framework.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We detail the exact computational resources in the implementation details, specifying the use of four NVIDIA RTX 3090 GPUs (24GB VRAM each).

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We followed the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: The positive societal impact (enhancing VLM trustworthiness in safety-critical domains) is discussed in Sections 1 and 6. Potential negative impacts, including the risks of automation bias, are explicitly mentioned in Section 6.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: Our work proposes a diagnostic probing framework for detecting hallucinations in existing Vision-Language Models. We do not release pre-trained generative models, image generators, or newly scraped web datasets that inherently carry a high risk of dual-use or require specific access safeguards.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: All the datasets and models used in this work are properly credited with appropriate references, and their licenses and terms of use have been fully respected.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [N/A]

Justification: The paper does not release new assets.

Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [N/A]

Justification: The paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [N/A]

Justification: The paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: The core methodology of this research does not involve LLMs as an important, original, or non-standard component. Our use of LLMs was strictly limited to text editing, formatting, and generating standard auxiliary scripts for visualization, which does not require declaration under the current policy.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.