LIFE: LOCAL INTRINSIC DIMENSION-GUIDED FEATURE EDITING FOR ONE-TOKEN, ONE-LAYER HALLUCINATION REPAIR IN VISION-LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Large vision-language models (LVLMs) can hallucinate, producing fluent responses with details that the image does not support. Hallucination leaves traces in the model's internal representations, but the decoder layer where these traces are clearest varies across models, benchmarks, and individual inputs, and existing methods inspect or edit a layer that is fixed in advance. We introduce **LIFE**, Local Intrinsic Dimension-Guided Feature Editing, an *inference-time* framework that decides where to inspect, whether to intervene, and how little to change. *LIFE-Detect* uses intrinsic dimension to locate hallucination-sensitive layers without labels, at two scales: the population-level intrinsic dimension of calibration representations identifies a shared layer for each model and benchmark, and the local intrinsic dimension (LID) of an individual input selects a layer for that input. A small MLP probe at the selected layer decides whether the response will likely be a hallucinated one. *LIFE-Mitigate* then edits one hidden state, the representation of a single token at a single decoder layer, by moving it along a direction given by the probe with a step size chosen for that input, and regenerates the response. The LVLM is never fine-tuned and never back-propagated through; the only trained component is the probe, and the correction optimizes one scalar. Across three LVLM backbones and five benchmarks, LIFE reduces the AMBER hallucination rate from 21.3% to 18.8% on LLaVA-1.6 and from 14.2% to 11.0% on Qwen3-VL, and more than halves sentence-level CHAIR on all three backbones, at a per-sample cost dominated by regenerating the response.

1 INTRODUCTION

Large vision-language models (LVLMs) answer questions about images, but they can produce fluent claims that the image does not support, a failure known as visual hallucination (Li et al., 2023; Wang et al., 2023). Training-based remedies add supervision and update LVLM parameters (Sarkar et al., 2025; Zhao et al., 2025; Jiang et al., 2024; Xiao et al., 2025). Inference-time methods keep the model frozen and instead modify decoding (Leng et al., 2024), attention (Huang et al., 2024; Chen et al., 2024b; Liu et al., 2024c; Zhang et al., 2026b), or internal representations (Zhao et al., 2026; Liu et al., 2025). For methods that edit representations, two choices matter: the layer at which the edit is made, and whether an edit is needed at all. An edit at a layer that carries little information about the current hallucination misses the error, and an edit to a response that was already grounded can only do harm.

Hallucination leaves measurable traces inside a model, often more clearly in intermediate layers than in the final one (Park et al., 2025a; Kogilathota et al., 2026; Zhang et al., 2026a). Which layer is most informative, however, varies across models and benchmarks. Wang et al. (2026) show this for text-only LLMs and select a layer automatically from the intrinsic dimension of representations (Levina & Bickel, 2004), an estimate of how many directions are needed to describe their variation. Two questions follow: *does intrinsic dimension locate such layers in LVLMs, and, since population-level intrinsic dimension assigns one layer to every input, can a local, per-input estimate of the same quantity select a layer for each image-question pair?*Figure 1 (top) gives a first answer on AMBER: the population-level criterion picks the layer with the best detection AUROC or the one next to it,

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

and the mean trajectory of local intrinsic dimension (LID) across inputs peaks at the same layer. The layers chosen per input (Figure 4) are centered on that layer but spread over several layers around it.

Research Question

Can a label-free, per-input signal ***identify the layer*** at which an LVLM's hallucination is visible, and does correcting one hidden state at that layer reduce hallucination during ***inference***?

To answer this question we introduce LIFE: Local Intrinsic Dimension-Guided Feature Editing for One-Token, One-Layer Hallucination Repair. LIFE makes three decisions for each input: **where to inspect, whether to intervene, and how little to change**. *LIFE-Detect* applies the First Effective Peak of Intrinsic Dimension (FEPoID) criterion (Wang et al., 2026) to calibration representations to obtain a dataset-level anchor layer without labels, and also uses local intrinsic dimensionality (LID) (Ma et al., 2018; Yin et al., 2024) to select a layer separately for each input based on the local geometry of its representation; a small MLP probe (Kogilathota et al., 2026) at that layer decides whether the response is hallucinated. *LIFE-Mitigate* then edits one hidden state, the representation of a single token position at a single decoder layer, along a direction given by the probe with a scalar step chosen for that input. One state is enough to reach the whole continuation: the edited state feeds every later layer at that position, and every subsequent token attends to it. The LVLM is never fine-tuned and never back-propagated through; the probe is trained once on representations of the frozen LVLM and is frozen at inference, where LIFE optimizes a single scalar. Our main setting uses the final token of the first generated sentence (FST) (Wang et al., 2026); Appendix D uses the final query token before generation (QT) (Kogilathota et al., 2026). We evaluate LIFE on LLaVA-1.6-7B (Liu et al., 2024b), Qwen3-VL-8B (Bai et al., 2025), and InternVL3.5-8B (Wang et al., 2025) across five hallucination benchmarks, with an additional comparison on LLaVA-1.5-7B (Liu et al., 2024a). Our contributions are:

➤ **Label-free localization at two scales.** We show that intrinsic dimension locates hallucination-sensitive layers in LVLMs: across twelve model-benchmark pairs, the FEPoID layer lies within two layers of the layer with the best probe AUROC. We then use LID, the per-input counterpart of the same quantity, to select a layer for each input, and characterize when this refinement helps.

➤ **A minimal, gated repair.** When a response is flagged, LIFE edits one hidden state by optimizing one scalar. No LVLM parameter changes, no gradient passes through the LVLM, no second image pass, and no intervention at any other token or layer; the correction costs a small fraction of one decoding step, and the overhead is dominated by regenerating the response.

➤ **Results.** LIFE cuts the AMBER hallucination rate from 21.3% to 18.8% (LLaVA-1.6), 14.24% to 11.0% (Qwen3-VL), and 22.43% to 14.2% (InternVL3.5), more than halves CHAIR_S on all three backbones, improves POPE and MME over the unmodified models, and beats recent inference-time methods at lower latency than VCD, VTI, and OPERA.

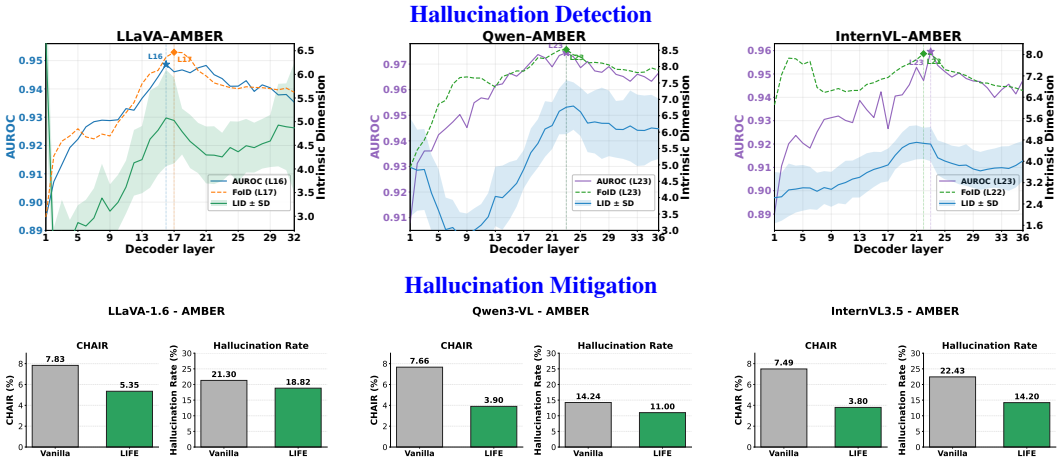


Figure 1: **Automatic layer selection and hallucination mitigation on AMBER. Top:** layer-wise FST-probe AUROC, population-level intrinsic dimension with the FEPoID layer, and the mean LID across inputs ($\pm$ one standard deviation). The population-level and mean-LID curves are different estimators of intrinsic dimension, so their shapes and peak locations are comparable but their values are not. **Bottom:** CHAIR and hallucination rate for LIFE and the corresponding vanilla models on LLaVA-1.6, Qwen3-VL, and InternVL3.5.

Table 1: Conceptual comparison of inference-time hallucination mitigation methods.

Method	Mitigation mechanism	Intervention space	Layer selection	Footprint per input
VCD	Original vs. distorted image	Output logits	N/A	Every decoding step; second image pass
OPERA	Attention penalty and rollback	Attention/logits	N/A	Every decoding step; beam rollback
HALC	Region contrastive decoding	Output logits	N/A	Per token; multiple visual crops
PAI	Visual attention amplification	Attention/logits	Predefined	All layers, every decoding step
SAVAA	Stepwise attention amplification	Attention	Predefined	Every decoding step
VTI	Visual-text steering	Hidden states	Predefined	All tokens, multiple layers
ClearSight	Visual feature enhancement	Hidden states	Predefined	Selected layers, every step
LIFE (ours)	**LID-guided feature editing**	**One hidden state**	**Per input**	**One token, one layer; flagged inputs only**

2 Related Work

Hallucination detection and localization. Output-level methods identify unsupported content or hallucinated tokens (Gunjal et al., 2024; Park et al., 2025a), while internal-state approaches show that hallucination signals are often concentrated in intermediate LLM layers (Azaria & Mitchell, 2023; Chen et al., 2024a; Orgad et al., 2025). FEPoID selects one dataset-level layer using intrinsic dimension (Wang et al., 2026). In LVLMs, HALP compares representations across decoder layers (Kogilathota et al., 2026), and VIB-Probe identifies informative internal features and attention heads (Zhang et al., 2026a). These methods use a layer or representation shared across samples; LIFE selects a layer for each input.

Intrinsic-dimensional geometry. Intrinsic dimension (ID) characterizes the geometry of a collection of representations (Levina & Bickel, 2004; Facco et al., 2017; Ansuini et al., 2019); LID describes the neighborhood of an individual point (Houle, 2017; Amsaleg et al., 2015), and elevated LID marks abnormal inputs, adversarial examples (Ma et al., 2018) and untruthful LLM generations (Yin et al., 2024), compared across inputs at a fixed layer. LIFE uses ID for a shared layer and LID for a per-input layer (Section 3.2).

Training-time mitigation. Training-based methods improve visual grounding through reinforcement learning (Sun et al., 2024), fine-grained human feedback (Yu et al., 2024b), data refinement (Yu et al., 2024a), contrastive learning (Jiang et al., 2024), preference optimization (Zhao et al., 2025), or phrase-level alignment (Sarkar et al., 2025). Although effective, they require additional supervision and LVLM parameter updates. LIFE keeps the pretrained LVLM frozen and performs detection-gated correction at inference time.

Inference-time mitigation. Existing methods reduce hallucination by modifying decoding (Leng et al., 2024; Huang et al., 2024; Chen et al., 2024b), visual attention (Liu et al., 2024c; Zhang et al., 2026b), or internal representations (Liu et al., 2025; Yin et al., 2025). As Table 1 summarizes, they act at predefined locations, with shared rules, and on every input and usually every decoding step. LIFE decides where and whether to intervene for each input, and when it does, it changes one hidden state.

3 Proposed Method

LIFE is a selective inference-time framework for localizing, detecting, and correcting hallucinations in LVLMs. Hallucination-related information is encoded in internal representations and is often most discriminative at intermediate layers (Azaria & Mitchell, 2023; Chen et al., 2024a; Orgad et al., 2025; Park et al., 2025b; Kogilathota et al., 2026; Wang et al., 2026), but the most informative depth varies across architectures, benchmarks, and inputs. As Figure 2 shows, *LIFE-Detect* locates a layer at two scales and decides with a probe whether correction is needed, and *LIFE-Mitigate*, run only for flagged inputs, edits one token representation at the selected layer with one input-specific scalar. The LVLM and the probes remain frozen throughout.

3.1 Problem Formulation

Let a frozen LVLM f_θ receive an image-question pair $x_i = (v_i, q_i)$ and generate $y_i \sim p_\theta(\cdot \mid v_i, q_i)$, where v_i is an image, q_i its textual query, and y_i the response. Suppose the language decoder has L layers, and denote the token-wise hidden states at layer ℓ by $\mathbf{H}_i^{(\ell)} \in \mathbb{R}^{T_i \times d}$, where T_i is the sequence length and d the hidden dimension. LIFE uses one token representation per decoder layer. In the main setting this is the final token of the first generated sentence (FST), $\mathbf{h}_i^{(\ell)} = \mathbf{H}_{i,t_i^{\mathrm{FST}}}^{(\ell)} \in \mathbb{R}^d$, $\ell \in \{1, \ldots, L\}$, where t_i^{FST} is the position of that token. Following First-Sentence Truncation (Wang et al., 2026), this representation captures early response-level information while avoiding

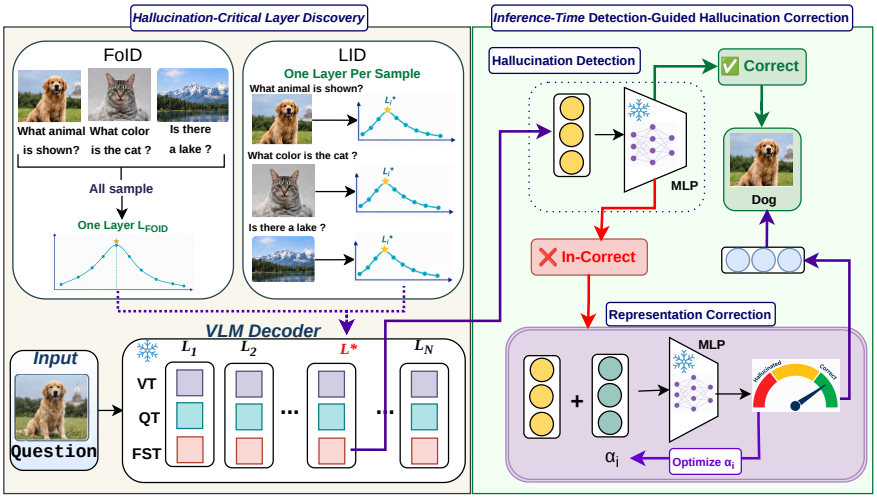


Figure 2: Overview of LIFE. (1) *LIFE-Detect* uses the intrinsic dimension of calibration representations (FoID) to identify a dataset-level layer and the local intrinsic dimension (LID) of each input to select a layer for that input. An MLP probe at the selected layer decides whether the response is hallucinated. (2) If it is, *LIFE-Mitigate* edits the single hidden state at the selected layer and regenerates the response.

the repetition and semantic drift that accumulate later in a response. For each input x_i the goal is to select a layer ℓ_i^*, estimate hallucination risk from $\mathbf{h}_i^{(\ell_i^*)}$, and correct that single state only when necessary. We also evaluate the final query-token representation (QT) as a pre-generation alternative (Appendix D).

3.2 LIFE-Detect: Automatic Localization and Detection

Two Questions LIFE-Detect Answers

Where should the input be inspected, and does its representation indicate hallucination?
Intrinsic geometry answers the first; a lightweight MLP answers the second.

Population-level localization. Given N unlabeled samples $\mathcal{X} = \{x_i\}_{i=1}^N$ from the current benchmark, we extract the selected token representation at every decoder layer; at layer ℓ these form $\mathbf{H}^{(\ell)} = [\mathbf{h}_1^{(\ell)}, \cdots, \mathbf{h}_N^{(\ell)}]^\top \in \mathbb{R}^{N \times d}$. The samples characterize the geometry of the representation space only; their hallucination labels are not used. We estimate the intrinsic dimension $D^{(\ell)}$ of $\mathbf{H}^{(\ell)}$ with TwoNN (Facco et al., 2017), which uses the ratio $\mu_i^{(\ell)} = r_{i,2}^{(\ell)}/r_{i,1}^{(\ell)}$ of each sample's second- to first-nearest-neighbor distance. Repeating this across the decoder gives the trajectory $\mathcal{D} = (D^{(1)}, \ldots, D^{(L)})$. Rather than taking its global maximum, which can sit at a late layer specialized for next-token prediction, we apply the First Effective Peak of Intrinsic Dimension (FEPoID) criterion (Wang et al., 2026), $\ell_{\mathrm{FEP}} = \mathrm{FEPoID}(D^{(1)}, \ldots, D^{(L)}; w)$, which scans the trajectory from shallow to deep and selects the first stable peak, rejecting a candidate if the trajectory keeps rising within a forward horizon of w layers. We use $w = 7$ throughout, except for InternVL3.5 on HaloQuest, where $w = 3$. The resulting ℓ_{FEP} indicates where hallucination-sensitive geometry emerges without any supervised layer-wise comparison.

Sample-specific localization with LID. Population-level intrinsic dimension is a property of a set of representations: it assigns one layer to every input and has no value for an individual input. To carry the same signal to a single input, we use its local counterpart. Let $\mathcal{X}_{\mathrm{inf}} = \{x_i\}_{i=1}^M$ be a reference collection of unlabeled inputs; in our experiments it is the inference set of the benchmark (1,004 inputs for AMBER, 874 for MME), with the current input excluded from its own neighborhood; any fixed set of precomputed representations, such as the calibration set, can be used instead. For every input x_i and layer ℓ, let $r_{i,1}^{(\ell)} \le r_{i,2}^{(\ell)} \le \cdots \le r_{i,k}^{(\ell)}$ be the distances from $\mathbf{h}_i^{(\ell)}$ to its k nearest neighbors among the other representations in $\mathcal{X}_{\mathrm{inf}}$ at that layer. The LID of x_i at layer ℓ is the

maximum-likelihood estimate (Amsaleg et al., 2015; Ma et al., 2018)

$$\widehat{\mathrm{LID}}_i^{(\ell)} = -\left[\frac{1}{k}\sum_{j=1}^{k}\log\left(\frac{r_{i,j}^{(\ell)}}{r_{i,k}^{(\ell)}}\right)\right]^{-1}. \tag{1}$$

LID measures how quickly the neighborhood around $\mathbf{h}_i^{(\ell)}$ fills space (Houle, 2017): a larger value means the representation sits in a locally higher-dimensional region. It uses no labels, and, up to the choice of estimator, the population-level intrinsic dimension of a layer is the average of the per-sample LIDs at that layer, so the two stages of *LIFE-Detect* measure the same quantity at two scales. LIFE selects the layer at which the input's LID is largest,

$$\ell_i^* = \mathrm{argmax}_{\ell\in\mathcal{M}}\ \widehat{\mathrm{LID}}_i^{(\ell)}, \tag{2}$$

where $\mathcal{M}$ is the set of candidate layers; we use all decoder layers. We set $k = 25$ and study $k \in \{10, 25, 40, 60\}$ in Appendix E.

Why LID. An input's LID at layer ℓ combines two parts: the dimension of that layer's representation space, shared by all inputs, and how far this input departs from the typical local geometry of that layer. Prior work shows that abnormal inputs have elevated LID relative to normal ones at the same layer, for adversarial examples in image classifiers (Ma et al., 2018) and for untruthful generations in LLMs (Yin et al., 2024), and FEPoID shows that the shared part peaks at layers where hallucination is detectable (Wang et al., 2026). For a hallucinated input both parts therefore point to the same region, and the layer with the largest LID is the one where the input departs most from the population geometry, which is where we inspect it. For a reliable input the departure is small at every layer and the rule falls back to the population peak. In the three LVLMs we study, the first effective peak coincides with the global maximum of the mean LID trajectory (Figure 1), so the argmax over all layers lands where FEPoID does (Appendix B discusses the general case). We treat this rule as a working hypothesis and test it in Section 4. Figure 4 shows the outcome: the most common selected layer coincides with ℓ_{FEP} in every model-benchmark pair, while individual inputs spread over several layers around it.

MLP-based hallucination detection. We train a lightweight MLP probe g_{ϕ_ℓ} at each decoder layer to classify its token representation as hallucinated or reliable, using binary cross-entropy on labeled representations of the frozen LVLM (labels and data splits are given in Appendix A). At inference, only the probe of the selected layer is applied: $p_i = g_{\phi_{\ell_i^*}}(\mathbf{h}_i^{(\ell_i^*)})$ and $\widehat{z}_i = \mathbb{I}[p_i > \tau]$ with $\tau = 0.5$, where $\widehat{z}_i = 1$ flags hallucination. LID thus decides *where* to inspect and the probe decides *whether* to intervene; only flagged inputs reach *LIFE-Mitigate*. Algorithm 1 summarizes the procedure.

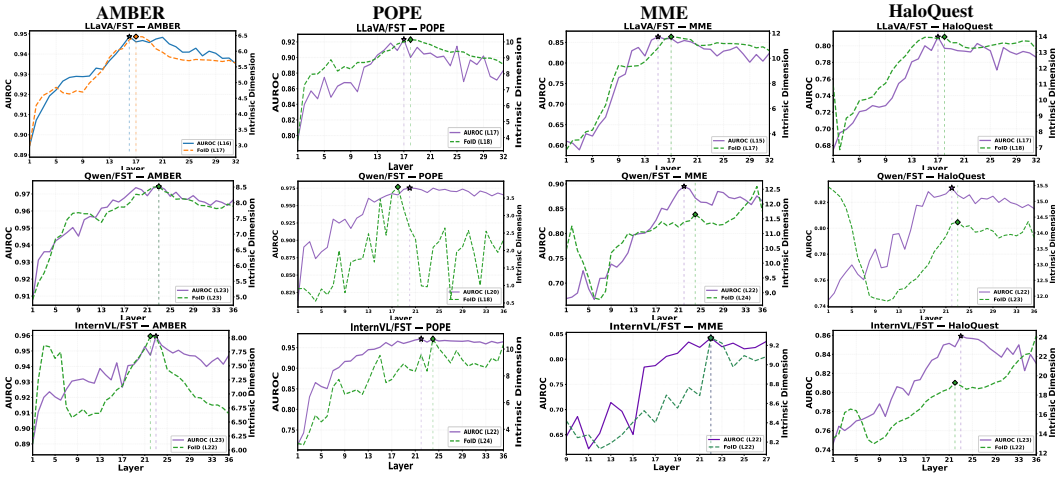


Figure 3: **FEPoID-based layer localization across models and benchmarks.** Layer-wise probe AUROC and population-level intrinsic dimension computed from first-sentence token (FST) representations for LLaVA-1.6-7B, Qwen3-VL-8B, and InternVL3.5-8B on AMBER, POPE, MME, and HaloQuest. Columns are benchmarks and rows are models; the legend of each panel gives the best-AUROC layer and the FEPoID layer. Enlarged panels are in Appendix H.

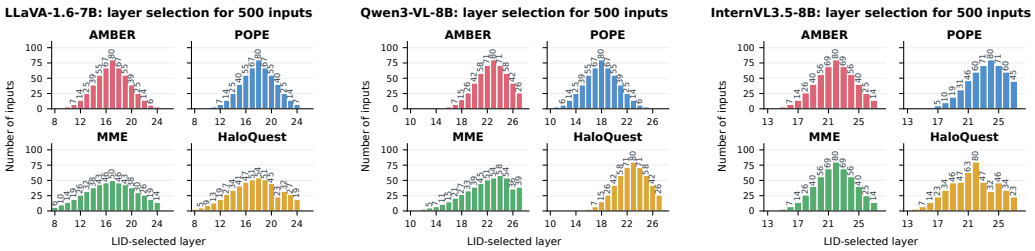


Figure 4: Distribution of LID-selected layers for LLaVA-1.6-7B, Qwen3-VL-8B, and InternVL3.5-8B. Each panel contains 500 inputs; the x-axis shows the selected decoder layer and the y-axis the number of assigned inputs. The modal layer lies within two layers of the FEPoID layer in Figure 3, while individual selections spread across nearby layers.

3.3 LIFE-Mitigate: Selective Feature Correction

Question LIFE-Mitigate Answers

How should a flagged representation be corrected? LIFE-Mitigate applies a probe-guided, input-specific feature edit.

LIFE-Mitigate runs only for inputs the probe flags. For a flagged input i, LIFE edits the single hidden state $\mathbf{h}_i^{(\ell_i^*)}$ at the selected layer; the LVLM and the probe stay frozen. The gradient of the probe's hallucination probability with respect to its input, $\mathbf{g}_i = \nabla_{\mathbf{h}}\, g_{\phi_{\ell_i^*}}(\mathbf{h}_i^{(\ell_i^*)})$, is computed once, and its negative normalized value $\mathbf{d}_i = -\mathbf{g}_i/(\|\mathbf{g}_i\|_2 + \epsilon)$, with a small $\epsilon > 0$, is the correction direction. This gradient is taken with respect to the probe's input, not through the LVLM. The edited state is

$$\widetilde{\mathbf{h}}_i^{(\ell_i^*)} = \mathbf{h}_i^{(\ell_i^*)} + \alpha_i \mathbf{d}_i, \tag{3}$$

where the nonnegative scalar α_i is the only quantity optimized. We choose it by minimizing

$$\mathcal{L}_i(\alpha_i) = \mathrm{BCE}\Big(g_{\phi_{\ell_i^*}}\big(\widetilde{\mathbf{h}}_i^{(\ell_i^*)}\big),\, 0\Big) + \lambda \alpha_i^2, \tag{4}$$

the cross-entropy toward the reliable label plus a penalty on the step size, so that the correction is the smallest one that satisfies the probe. Starting from $\alpha_i^{(0)} = 0$ we take projected first-order steps,

$$\alpha_i^{(s+1)} = \left[\alpha_i^{(s)} - \eta\, \frac{\partial \mathcal{L}_i}{\partial \alpha_i}\right]_+, \tag{5}$$

with $[z]_+ = \max(0, z)$ (Adam with the same nonnegativity clipping in practice; Appendix A), and stop when the probe's probability on the edited state falls below δ or after $S_{\max}$ steps. Algorithm 2 summarizes the procedure.

Regeneration and footprint. In FST, the first sentence is kept: LIFE replaces $\mathbf{h}_i^{(\ell_i^*)}$ at t_i^{FST}, recomputes the remaining layers at that position, and regenerates only the continuation. Because later tokens attend to the edited state, the edit propagates through the continuation; hallucinations within the first sentence are not corrected, whereas QT (Appendix D) edits before generation. Per flagged input, LIFE edits one hidden state and optimizes one scalar, with no LVLM parameter update, LVLM backpropagation, second image pass, or other token/layer edits. The only added LVLM cost is continuation regeneration; localization and correction add only a small fraction of one decoding step (Section 4).

Algorithm 1: LIFE-Detect

Input: Layer states $\{\mathbf{h}_i^{(\ell)}\}_{\ell=1}^{L}$; frozen probes $\{g_{\phi_\ell}\}_{\ell=1}^{L}$; k, τ
Output: Selected layer ℓ_i^* and hallucination prediction $\widehat{z}_i$

for each input x_i **do**
  for $\ell = 1, \ldots, L$ **do**
    Find k-NN and compute $\widehat{\mathrm{LID}}_i^{(\ell)}$ (Eq. (1))
  end for
  $\ell_i^* \leftarrow \arg\max_\ell \widehat{\mathrm{LID}}_i^{(\ell)}$
  $p_i \leftarrow g_{\phi_{\ell_i^*}}\big(\mathbf{h}_i^{(\ell_i^*)}\big)$
  $\widehat{z}_i \leftarrow \mathbb{I}[p_i > \tau]$ ▷ 1: hallucinated; 0: reliable
end for
return $\ell_i^*, \widehat{z}_i, \mathbf{h}_i^{(\ell_i^*)}$

Algorithm 2: LIFE-Mitigate

Input: Flagged state $\mathbf{h}_i^{(\ell_i^*)}$; frozen probe $g_{\phi_{\ell_i^*}}$; η, λ, δ, $S_{\max}$
Output: Corrected response $\widetilde{y}_i$

$\mathbf{d}_i \leftarrow -\mathbf{g}_i/(\|\mathbf{g}_i\|_2 + \epsilon)$ with $\mathbf{g}_i = \nabla_{\mathbf{h}}\, g_{\phi_{\ell_i^*}}(\mathbf{h}_i^{(\ell_i^*)})$ ▷ computed once
$\alpha_i \leftarrow 0, \quad s \leftarrow 0, \quad p_i \leftarrow g_{\phi_{\ell_i^*}}(\mathbf{h}_i^{(\ell_i^*)})$
while $p_i > \delta$ and $s < S_{\max}$ **do**
  Compute $\widetilde{\mathbf{h}}_i^{(\ell_i^*)}$ (Eq. (3)) and $\mathcal{L}_i$ (Eq. (4))
  Update α_i (Eq. (5))
  Recompute $\widetilde{\mathbf{h}}_i^{(\ell_i^*)}$ and p_i
  $s \leftarrow s + 1$
end while
Regenerate the continuation from $\widetilde{\mathbf{h}}_i^{(\ell_i^*)}$ and **return** $\widetilde{y}_i$

4 EXPERIMENTAL EVALUATION

We ask: (1) does intrinsic dimension locate hallucination-sensitive layers across architectures and benchmarks, and does the per-input rule agree with it; (2) does the intervention layer matter, and when does per-input selection help; (3) does LIFE outperform existing inference-time methods while preserving reliable outputs; and (4) what does it cost.

4.1 EXPERIMENTAL SETUP

Models. We evaluate LIFE on LLaVA-1.6-7B (Liu et al., 2024b), Qwen3-VL-8B (Bai et al., 2025), and InternVL3.5-8B (Wang et al., 2025), which span different multimodal architectures and representation spaces, and additionally on LLaVA-1.5-7B (Liu et al., 2024a) for the comparison with VCD, VTI, and OPERA.

Benchmarks and metrics. AMBER (Wang et al., 2023) with CHAIR and hallucination rate; POPE (Li et al., 2023) with accuracy and F1; MME (Fu et al., 2023) and HaloQuest (Wang et al., 2024) with accuracy; and CHAIR (Rohrbach et al., 2018) with CHAIR_S and CHAIR_I. Lower hallucination scores and higher accuracy and F1 are better.

Baselines. We compare with the unmodified LVLM and seven inference-time methods: VCD (Leng et al., 2024), OPERA (Huang et al., 2024), VTI (Liu et al., 2025), PAI (Liu et al., 2024c), VAF (Yin et al., 2025), VGA (Zhao et al., 2026), and SAVAA (Zhang et al., 2026b). To isolate the effect of the intervention layer, we also apply the same detection and correction at $L/2$, $3L/4$, the final layer, and the FoID-selected layer.

Probe data. We train layer-specific probes on FST (or QT) representations from the frozen LVLM. An LLM judge (gpt-5-mini) labels each vanilla response by comparing it with the ground truth. Table 6 reports the number of labeled examples in each probe pool before splitting. We use 70% for probe training, 10% for validation, and 20% for held-out detection testing; all detection AUROCs are computed on the test portion. For LIFE mitigation, we evaluate 1,004 AMBER generative examples, 874 MME questions, 2,910 HaloQuest questions, and 500 CHAIR images. The AMBER probe pool includes both discriminative and generative examples. For POPE, we train the probes on 9000 questions and evaluate mitigation on a separate set of 27,000 questions spanning MSCOCO, A-OKVQA, and GQA under Random, Popular, and Adversarial sampling. The mitigation examples for AMBER, MME, HaloQuest, and CHAIR were selected from the corresponding probe-training splits. Unlike the training-free baselines, LIFE requires labeled in-domain data for probe training; we discuss this requirement in the limitations.

4.2 HALLUCINATION LOCALIZATION AND DETECTION

Figures 3 and 4 report the FST-based results; QT results are in Appendix D. Three observations. First, hallucination-related information concentrates in intermediate layers, but the best layer varies: probe AUROC peaks between layers 15 and 17 for LLaVA-1.6 and between 20 and 23 for Qwen3-VL and InternVL3.5, and the gap between the best and worst layer is small on AMBER (0.89-0.95 for LLaVA) and large on MME and HaloQuest. Second, population-level intrinsic dimension finds this region without labels: in all twelve model-benchmark pairs the FEPoID layer lies within two layers of the best-AUROC layer, and in eight it is within one. Third, the per-input rule agrees with the population rule on aggregate and adds variation on top: the most common LID-selected layer coincides with the FEPoID layer in most pairs and is within two layers of it in all of them, while individual inputs spread over roughly ± 6 layers. Appendix G.1 shows that per-input layer selection performs comparably to the fixed FEPoID layer for hallucination detection, exceeding its AUROC in six of twelve settings.

4.3 HALLUCINATION MITIGATION RESULTS

Does the intervention layer matter? Table 2 applies the same detection and correction at four fixed layers, at L_{FoID}, and at the per-input layer L_i^*. Two patterns stand out. On AMBER, every choice of layer gives nearly the same hallucination rate (18.8-19.3 for LLaVA, 11.0-11.3 for Qwen, 14.2-14.3 for InternVL), so the large reduction relative to the vanilla model comes from the gated edit itself rather than from where it is applied. On MME and HaloQuest the layer matters and no single choice wins: L_i^* is best or tied in four of the twelve rows, and the two automatic choices split the twelve rows evenly: L_i^* beats L_{FoID} by 8.5 points on LLaVA HaloQuest and trails it on Qwen MME and HaloQuest and on InternVL POPE and MME. Both automatic choices remove the need to tune the layer by hand, which is the practical point; the per-input rule is a refinement whose value depends on the model and benchmark.

Table 2: Effect of the intervention layer on mitigation, with the same detection and correction procedure at every layer. Green marks the best layer in each row; underline marks the better of the two automatic choices.

Model	Benchmark / Metric	Vanilla	$L/2$	$3L/4$	L_{last}	LIFE[L_{FoID}]	LIFE[L_i^*]
LLaVA-1.6	AMBER Hal. ↓	21.3	**18.8**	19.1	19.3	19.0	18.8
	POPE F1 ↑	91.5	92.4	**93.2**	92.3	92.4	92.5
	MME Acc. ↑	78.7	**88.9**	81.6	87.5	85.7	85.3
	HaloQuest Acc. ↑	42.5	**52.5**	51.4	44.2	42.9	51.4
Qwen3-VL	AMBER Hal. ↓	14.24	11.3	11.2	11.1	11.3	**11.0**
	POPE F1 ↑	93.0	94.5	94.8	94.5	94.5	**95.0**
	MME Acc. ↑	88.5	94.9	92.5	91.5	**97.6**	94.2
	HaloQuest Acc. ↑	65.6	74.2	80.3	66.8	**86.8**	74.7
InternVL3.5	AMBER Hal. ↓	22.43	14.2	14.2	14.2	14.3	**14.2**
	POPE F1 ↑	91.0	94.5	**95.2**	94.4	95.1	91.8
	MME Acc. ↑	87.1	89.1	89.9	90.5	**94.2**	90.0
	HaloQuest Acc. ↑	57.7	**61.6**	59.2	60.1	60.1	58.4

Comparison with inference-time methods. Table 3 compares LIFE with PAI, VAF, VGA, and SAVAA. On all three backbones, LIFE has the lowest CHAIR_S and CHAIR_I among the evaluated methods: CHAIR_S falls from 33.80 to 13.20 for LLaVA, 58.80 to 10.40 for Qwen, and 41.40 to 17.20 for InternVL. On POPE, LIFE improves accuracy and F1 over the vanilla model on every backbone and has the highest accuracy among the compared methods, while VGA reaches a slightly higher F1 on InternVL. Qualitative examples are in Appendix F. Appendix G.2 compares response length and object coverage on AMBER and CHAIR. Vanilla and LIFE responses average 64 words across all three backbones. The hallucination reductions therefore do not arise from shorter descriptions, although AMBER Cover decreases by 2.8-4.8 percentage points and CHAIR object recall by 2.0-2.3 points.

Table 3: Comparison with inference-time hallucination mitigation methods. Results are reported across three LVLM backbones and two benchmarks.

Model	Method	POPE		CHAIR	
		Acc. ↑	F1 ↑	CHAIR_S ↓	CHAIR_I ↓
LLaVA-1.6	Vanilla	91.34	91.57	33.80	8.46
	PAI	90.36	90.85	39.60	10.06
	VAF	90.24	90.80	36.40	9.14
	VGA	90.83	91.08	32.40	9.90
	SAVAA	91.33	91.55	28.80	7.66
	LIFE L_i^*	**92.53**	**92.59**	**13.20**	**6.84**
Qwen3-VL-8B	Vanilla	93.24	93.09	58.80	10.57
	PAI	93.23	93.17	53.60	10.47
	VAF	93.10	93.09	53.60	9.75
	VGA	92.82	92.68	58.40	10.50
	SAVAA	93.16	92.90	46.00	8.30
	LIFE L_i^*	**95.20**	**95.00**	**10.40**	**6.30**
InternVL3.5-8B	Vanilla	90.96	91.08	41.40	10.80
	PAI	90.17	90.51	45.60	11.81
	VAF	91.70	91.99	42.20	11.00
	VGA	92.32	**92.48**	44.20	11.57
	SAVAA	91.26	91.59	34.40	9.34
	LIFE L_i^*	**92.40**	**92.18**	**17.20**	**6.90**

Correction outcomes. Table 4 reports the number of outputs remaining incorrect after mitigation (Inc.) and the number of initially incorrect outputs corrected (C) for each model and benchmark. Both the dataset-level FoID and the sample-specific LIFE layer selection correct outputs across all four benchmarks, although their relative performance varies. For LLaVA-1.6, LIFE corrects substantially more HaloQuest outputs than FoID (172 versus 8), while the two methods have similar correction counts on AMBER, MME, and POPE. On POPE, LIFE corrects 999 of the 1,211 initially incorrect Qwen3-VL answers, leaving 212 incorrect; FoID corrects 961 and leaves 250 incorrect. For InternVL3.5, FoID corrects more POPE outputs (1,078 versus 817), whereas LIFE corrects more AMBER outputs (16 versus 9). These counts characterize correction of initially incorrect outputs.

Table 4: Correction outcomes across four benchmarks. Inc. is the number of outputs remaining incorrect after mitigation, and C is the number of incorrect-to-correct transitions.

Model	Method	AMBER ($N = 1{,}004$)		MME ($N = 874$)		POPE ($N = 4{,}500$)		HaloQuest ($N = 2{,}910$)	
		Inc.	C	Inc.	C	Inc.	C	Inc.	C
LLaVA-1.6	FoID	357	94/+94	125	61/+61	512	68/+68	1,096	8/+8
	LIFE L_i^*	**350**	**101/+101**	**128**	**58/+58**	**508**	**72/+72**	**932**	**172/+172**
Qwen3-VL	FoID	59	52/+52	21	79/+79	250	961/+961	253	406/+406
	LIFE L_i^*	**58**	**53/+53**	**50**	**50/+50**	**212**	**999/+999**	**484**	**175/+175**
InternVL3.5	FoID	238	9/+9	50	62/+62	415	1,078/+1,078	766	45/+45
	LIFE L_i^*	**231**	**16/+16**	**87**	**25/+25**	**676**	**817/+817**	**798**	**13/+13**

Inference cost. Figure 5 reports mean single-sample latency on POPE with LLaVA-1.5-7B on one NVIDIA H100: 0.401 s with L_{FoID} and 0.448 s with L_i^*, so per-input localization adds 0.047 s, and both are faster than VCD (1.100 s), VTI (0.544 s), and OPERA (2.229 s). The arithmetic explains why: the probe has about 2.3M multiply-adds for $d = 4096$, so twenty correction steps with forward and backward passes cost under 0.3 GFLOP. The LID search over 32 layers against $1{,}000$ precomputed representations costs about 0.26 GFLOP. For comparison, one decoding step of a 7B decoder requires roughly 14 GFLOP. Localization and correction together are about 4% of one generated token; the measured overhead is regenerating the continuation.

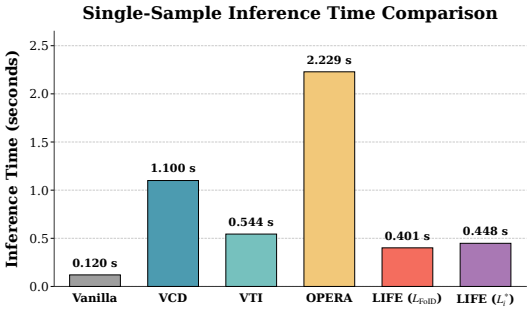


Figure 5: Single-sample inference latency on POPE.

Additional comparison on LLaVA-1.5-7B. We compare LIFE with VTI Chen et al. (2024b), OPERA Huang et al. (2024), and VCD Leng et al. (2024) on the same 2,910 POPE-Random questions. As shown in Table 5, per-input layer selection achieves the highest accuracy (92.68%), recall (96.00%), and F1 (93.11%). Relative to the unmodified model, these are gains of 9.19 accuracy points and 10.75 F1 points. Per-input selection also improves F1 by 1.95 points over dataset-level FoID selection. VTI has the highest precision (94.38%), whereas LIFE achieves substantially higher recall.

Table 5: POPE-Random performance on 2,910 questions using LLaVA-1.5-7B. All metrics are percentages.

Method	Acc.↑	Prec.↑	Rec.↑	F1↑
LLaVA-1.5-7B	83.49	88.84	76.76	82.36
VTI	89.50	**94.38**	84.00	88.89
OPERA	89.18	93.28	85.13	89.02
VCD	86.84	87.15	86.68	86.91
LIFE (ℓ_{FoID})	90.76	89.89	92.47	91.16
LIFE (ℓ_i^*)	**92.68**	90.40	**96.00**	**93.11**

5 Conclusion

We introduced LIFE, an inference-time framework that uses intrinsic dimension at two scales to locate hallucination-sensitive layers without labels, a small probe to decide whether a response needs repair, and an edit of one hidden state, one direction, one scalar, to repair it. Across three LVLM backbones and five benchmarks, the gated edit reduces hallucination substantially while the LVLM stays frozen and is never back-propagated through.

Limitations. The probes need labeled in-domain data for each benchmark, and probe errors bound what mitigation can fix. The per-input layer rule is a heuristic validated empirically: it matches the population-level choice on aggregate but is not uniformly better than it (Table 2). The FST setting cannot repair a hallucination in the first sentence. Label-efficient probes and a principled per-input criterion are future work.

ETHICS STATEMENT

This work studies hallucination detection and mitigation in large vision-language models using existing public benchmarks. Although LIFE reduces measured hallucination in these evaluations, it does not guarantee that corrected responses are accurate or safe. Errors may remain consequential in high-stakes applications, which require additional evaluation and appropriate human oversight.

REPRODUCIBILITY STATEMENT

Section 3 describes the population-level and per-input layer selection, the MLP probes, and the feature-editing procedure, including the threshold, the loss, and the stopping rule. Section 4 lists the models, benchmarks, baselines, and metrics. Appendix A gives probe architecture and training settings, the labeling protocol, data counts and splits, and the correction hyperparameters. The supplementary material includes the implementation code and a `requirements.txt` file listing software dependencies.

AI USE STATEMENT

In this work, we used generative AI tools for two purposes. First, an LLM judge (gpt-5-mini) labeled the responses used to train and evaluate the hallucination probes by comparing each vanilla model response with the benchmark ground truth. Second, we used generative AI tools to improve the manuscript's clarity, readability, and consistency. We have not used generative AI tools to generate research ideas, experiments, or results. We have reviewed all AI-assisted work and take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

REFERENCES

Laurent Amsaleg, Oussama Chelly, Teddy Furon, Stéphane Girard, Michael E Houle, Ken-ichi Kawarabayashi, and Michael Nett. Estimating local intrinsic dimensionality. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 29–38, 2015.

Alessio Ansuini, Alessandro Laio, Jakob H Macke, and Davide Zoccolan. Intrinsic dimension of data representations in deep neural networks. *Advances in Neural Information Processing Systems*, 32, 2019.

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*, pp. 967–976, 2023.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, Wenbin Ge, Zhifang Guo, Qidong Huang, Jie Huang, Fei Huang, Binyuan Hui, Shutong Jiang, Zhaohai Li, Mingsheng Li, Mei Li, Kaixin Li, Zicheng Lin, Junyang Lin, Xuejing Liu, Jiawei Liu, Chenglong Liu, Yang Liu, Dayiheng Liu, Shixuan Liu, Dunjie Lu, Ruilin Luo, Chenxu Lv, Rui Men, Lingchen Meng, Xuancheng Ren, Xingzhang Ren, Sibo Song, Yuchong Sun, Jun Tang, Jianhong Tu, Jianqiang Wan, Peng Wang, Pengfei Wang, Qiuyue Wang, Yuxuan Wang, Tianbao Xie, Yiheng Xu, Haiyang Xu, Jin Xu, Zhibo Yang, Mingkun Yang, Jianxin Yang, An Yang, Bowen Yu, Fei Zhang, Hang Zhang, Xi Zhang, Bo Zheng, Humen Zhong, Jingren Zhou, Fan Zhou, Jing Zhou, Yuanzhi Zhu, and Ke Zhu. Qwen3-vl technical report. *CoRR*, abs/2511.21631, November 2025. URL `https://doi.org/10.48550/arXiv.2511.21631`.

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. In *International Conference on Learning Representations*, volume 2024, pp. 3056–3076, 2024a.

Zhaorun Chen, Zhuokai Zhao, Hongyin Luo, Huaxiu Yao, Bo Li, and Jiawei Zhou. HALC: Object hallucination reduction via adaptive focal-contrast decoding. *arXiv preprint arXiv:2403.00425*, 2024b.

Elena Facco, Maria d'Errico, Alex Rodriguez, and Alessandro Laio. Estimating the intrinsic dimension of datasets by a minimal neighborhood information. *Scientific reports*, 7(1):12140, 2017.

Chaoyou Fu, Peixian Chen, Yunhang Shen, Yulei Qin, Mengdan Zhang, Xu Lin, Jinrui Yang, Xiawu Zheng, Ke Li, Xing Sun, et al. MME: A comprehensive evaluation benchmark for multimodal large language models. *arXiv preprint arXiv:2306.13394*, 2023.

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, pp. 18135–18143, 2024.

Michael E Houle. Local intrinsic dimensionality i: an extreme-value-theoretic foundation for similarity applications. In *International Conference on Similarity Search and Applications*, pp. 64–79. Springer, 2017.

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 *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13418–13427, 2024.

Chaoya Jiang, Haiyang Xu, Mengfan Dong, Jiaxing Chen, Wei Ye, Ming Yan, Qinghao Ye, Ji Zhang, Fei Huang, and Shikun Zhang. Hallucination augmented contrastive learning for multimodal large language model. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 27026–27036. IEEE, 2024.

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

Sicong Leng, Hang Zhang, Guanzheng Chen, Xin Li, Shijian Lu, Chunyan Miao, and Lidong Bing. Mitigating object hallucinations in large vision-language models through visual contrastive decoding. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13872–13882. IEEE, 2024.

Elizaveta Levina and Peter Bickel. Maximum likelihood estimation of intrinsic dimension. *Advances in neural information processing systems*, 17, 2004.

Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, 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*, pp. 292–305, 2023.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 26286–26296. IEEE, 2024a.

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

Sheng Liu, Haotian Ye, and James Y Zou. Reducing hallucinations in large vision-language models via latent space steering. In *International Conference on Learning Representations*, volume 2025, pp. 72402–72419, 2025.

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

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. URL `https://openreview.net/forum?id=Bkg6RiCqY7`.

Xingjun Ma, Bo Li, Yisen Wang, Sarah M Erfani, Sudanthi Wijewickrema, Grant Schoenebeck, Dawn Song, Michael E Houle, and James Bailey. Characterizing adversarial subspaces using local intrinsic dimensionality. *arXiv preprint arXiv:1801.02613*, 2018.

Hadas Orgad, Michael Toker, Zorik Gekhman, Roi Reichart, Idan Szpektor, Hadas Kotek, and Yonatan Belinkov. Llms know more than they show: On the intrinsic representation of llm hallucinations. In *International Conference on Learning Representations*, volume 2025, pp. 66880–66913, 2025.

Eunkyu Park, Minyeong Kim, and Gunhee Kim. Halloc: Token-level localization of hallucinations for vision language models. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 29893–29903. IEEE, 2025a.

Seongheon Park, Xuefeng Du, Min-Hsuan Yeh, Haobo Wang, and Yixuan Li. Steer LLM latents for hallucination detection. In *Forty-second International Conference on Machine Learning*, 2025b. URL https://openreview.net/forum?id=UMqNQEPNT3.

Anna Rohrbach, Lisa Anne Hendricks, Kaylee Burns, Trevor Darrell, and Kate Saenko. Object hallucination in image captioning. In *Proceedings of the 2018 conference on empirical methods in natural language processing*, pp. 4035–4045, 2018.

Pritam Sarkar, Sayna Ebrahimi, Ali Etemad, Ahmad Beirami, Sercan Arik, and Tomas Pfister. Mitigating object hallucination in mllms via data-augmented phrase-level alignment. In *International Conference on Learning Representations*, volume 2025, pp. 73796–73846, 2025.

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*, pp. 13088–13110, 2024.

Junyang Wang, Yuhang Wang, Guohai Xu, Jing Zhang, Yukai Gu, Haitao Jia, Ming Yan, Ji Zhang, and Jitao Sang. AMBER: An LLM-free multi-dimensional benchmark for MLLMs hallucination evaluation. *arXiv preprint arXiv:2311.07397*, 2023.

Weiyun Wang, Zhangwei Gao, Lixin Gu, Hengjun Pu, Long Cui, Xingguang Wei, Zhaoyang Liu, Linglin Jing, Shenglong Ye, Jie Shao, Zhaokai Wang, Zhe Chen, Hongjie Zhang, Ganlin Yang, Haomin Wang, Qi Wei, Jinhui Yin, Wenhao Li, Erfei Cui, Guanzhou Chen, Zichen Ding, Changyao Tian, Zhenyu Wu, JingJing Xie, Zehao Li, Bowen Yang, Yuchen Duan, Xuehui Wang, Zhi Hou, Haoran Hao, Tianyi Zhang, Songze Li, Xiangyu Zhao, Haodong Duan, Nianchen Deng, Bin Fu, Yinan He, Yi Wang, Conghui He, Botian Shi, Junjun He, Yingtong Xiong, Han Lv, Lijun Wu, Wenqi Shao, Kaipeng Zhang, Huipeng Deng, Biqing Qi, Jiaye Ge, Qipeng Guo, Wenwei Zhang, Songyang Zhang, Maosong Cao, Junyao Lin, Kexian Tang, Jianfei Gao, Haian Huang, Yuzhe Gu, Chengqi Lyu, Huanze Tang, Rui Wang, Haijun Lv, Wanli Ouyang, Limin Wang, Min Dou, Xizhou Zhu, Tong Lu, Dahua Lin, Jifeng Dai, Weijie Su, Bowen Zhou, Kai Chen, Yu Qiao, Wenhai Wang, and Gen Luo. Internvl3.5: Advancing open-source multimodal models in versatility, reasoning, and efficiency. *CoRR*, abs/2508.18265, August 2025. URL https://doi.org/10.48550/arXiv.2508.18265.

Xinpeng Wang, William Cao, Andrew Gordon Wilson, and Zhe Zeng. Automatic layer selection for hallucination detection. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=srAuLCB3Ca.

Zhecan Wang, Garrett Bingham, Adams Wei Yu, Quoc V Le, Thang Luong, and Golnaz Ghiasi. HaloQuest: A visual hallucination dataset for advancing multimodal reasoning. In *European Conference on Computer Vision*, pp. 288–304. Springer, 2024.

Wenyi Xiao, Ziwei Huang, Leilei Gan, Wanggui He, Haoyuan Li, Zhelun Yu, Fangxun Shu, Hao Jiang, and Linchao Zhu. Detecting and mitigating hallucination in large vision language models via fine-grained ai feedback. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 25543–25551, 2025.

Fan Yin, Jayanth Srinivasa, and Kai-Wei Chang. Characterizing truthfulness in large language model generations with local intrinsic dimension. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=7DbIyQlfaO.

Hao Yin, Guangzong Si, and Zilei Wang. Clearsight: Visual signal enhancement for object hallucination mitigation in multimodal large language models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 14625–14634, 2025.

Qifan Yu, Juncheng Li, Longhui Wei, Liang Pang, Wentao Ye, Bosheng Qin, Siliang Tang, Qi Tian, and Yueting Zhuang. Hallucidoctor: Mitigating hallucinatory toxicity in visual instruction data. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12944–12953. IEEE, 2024a.

Tianyu Yu, Yuan Yao, Haoye Zhang, Taiwen He, Yifeng Han, Ganqu Cui, Jinyi Hu, Zhiyuan Liu, Hai-Tao Zheng, Maosong Sun, et al. Rlhf-v: Towards trustworthy mllms via behavior alignment from fine-grained correctional human feedback. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13807–13816, 2024b.

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

Jiacheng Zhang, Feng Liu, Chao Du, and Tianyu Pang. Savaa: Mitigating hallucinations in lvlms via step-wise adaptive visual attention amplification. *arXiv preprint arXiv:2602.13600*, 2026b.

Jianfei Zhao, Feng Zhang, Xin Sun, Chong Feng, and Zhixing Tan. Tell model where to look: Mitigating hallucinations in mllms by vision-guided attention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 32582–32591, 2026.

Zhiyuan Zhao, Bin Wang, Linke Ouyang, Xiaoyi Dong, Jiaqi Wang, and Conghui He. Beyond multimodal hallucinations: enhancing lvlms through hallucination-aware direct preference optimization. In *2025 IEEE International Conference on Multimedia and Expo (ICME)*, pp. 1–6. IEEE, 2025.

APPENDIX: ADDITIONAL TECHNICAL DETAILS, DATASETS, AND RESULTS

This appendix provides implementation details and additional results supporting the main paper. Appendix A describes the labeling protocol, data splits, training of the MLP probes, and the optimization of the correction magnitude. Appendix B gives the geometric interpretation of LID-guided layer selection. Appendix C reports complete POPE results across Random, Popular, and Adversarial sampling. Appendix D presents experiments with the query-token representation. Appendix E studies sensitivity to the neighborhood size k. Appendix F provides qualitative HaloQuest examples. Appendix G reports detection performance at the per-input selected layer and analyzes response length and object coverage. Finally, Appendix H provides enlarged versions of the layer-localization plots from the main paper.

A PROBE TRAINING AND CORRECTION OPTIMIZATION

A.1 LABELS AND DATA SPLITS

For each benchmark and model, we run the vanilla LVLM and record the selected-token representation (FST or QT) at every decoder layer. An LLM judge (gpt-5-mini) compares each response with the benchmark ground truth and assigns a response-level label: 1 for hallucinated and 0 for reliable. Table 6 reports the total labeled examples in each probe pool before splitting. We use 70% for training the layer-specific MLPs, 10% for validation, and 20% for held-out detection testing. The AUROCs in Figures 3 and 6 are computed on examples excluded from probe training. For mitigation, we select 1,004 AMBER generative examples, 874 MME questions, 2,910 HaloQuest questions, and 500 CHAIR images from the corresponding probe-training splits. The AMBER probe pool contains both discriminative and generative examples, whereas its mitigation evaluation uses generative examples. POPE mitigation uses 27,000 questions separate from its 9,000-question probe pool.

A.2 LAYER-SPECIFIC MLP PROBES

LIFE trains one lightweight MLP probe per decoder layer to classify that layer's token representation as hallucinated or reliable. The LVLM remains frozen throughout; only the probe parameters are optimized. At inference, only the probe of the layer selected for the current input is applied. Let d_m denote the hidden-state dimension of the language decoder of model m. The input layer of each probe is sized to the model, and the rest of the architecture is fixed: $d_m \to 512 \to 256 \to 128 \to 1$, with ReLU hidden activations and a sigmoid output giving the hallucination probability. Each probe is trained for 50 epochs with binary cross-entropy and AdamW (Loshchilov & Hutter, 2019), learning rate 10^{-3}, batch size 32, and weight decay 10^{-4}. The decision threshold is $\tau = 0.5$ on the probe's probability.

Table 6: Labeled samples per benchmark used for the layer-specific MLP probes (before the train/held-out split).

Benchmark	Labeled samples
AMBER (Wang et al., 2023)	15,000
POPE (Li et al., 2023)	9,000
MME (Fu et al., 2023)	2375
HaloQuest (Wang et al., 2024)	7000
CHAIR (Rohrbach et al., 2018)	5,000

Correction-magnitude optimization. For an input the probe flags, LIFE computes the gradient of the probe's hallucination probability with respect to the selected token representation. Its negative normalized value is the correction direction $\mathbf{d}_i$, computed once and fixed. LIFE then optimizes only the nonnegative scalar α_i in $\tilde{\mathbf{h}}_i^{(\ell_i^*)} = \mathbf{h}_i^{(\ell_i^*)} + \alpha_i \mathbf{d}_i$ by minimizing Eq. (4): the binary cross-entropy of the probe's output on the edited state against the reliable label, plus $\lambda \alpha_i^2$, which penalizes unnecessarily large corrections. We initialize $\alpha_i^{(0)} = 0$ and use Adam with learning rate 0.05, clipping α_i at zero after every step, with $\lambda = 10^{-3}$, stopping probability $\delta = 0.1$, and at most $S_{\max} = 20$ steps. These values are fixed across all models and benchmarks. Both the LVLM and the probe remain frozen; only α_i is updated.

Hyperparameter summary. Table 7 summarizes the probe-training and correction settings.

Table 7: Probe and correction-optimization hyperparameters.

Part	Hyperparameter	Value
MLP probe	Input dimension	d_m
	Hidden dimensions	$512 \to 256 \to 128$
	Activation	ReLU; sigmoid output
	Epochs	50
	Batch size	32
	Optimizer	AdamW
	Learning rate	10^{-3}
	Weight decay	10^{-4}
	Decision threshold	$\tau = 0.5$
Correction	Optimized parameter	α_i
	Initialization	$\alpha_i^{(0)} = 0$
	Optimizer	Adam, clipped at $\alpha_i \geq 0$
	Learning rate	0.05
	Regularization	$\lambda = 10^{-3}$
	Stop / max. updates	$p_i \leq 0.1$ / 20
	LID neighborhood	$k = 25$

B Geometric Interpretation of LID-Guided Layer Selection

Local intrinsic dimensionality describes how quickly the neighborhood around an individual point fills space as the distance from it grows (Houle, 2017). Consider the representation $\mathbf{h}_i^{(\ell)}$ of input i at layer ℓ, and let $F_i^{(\ell)}(r)$ be the probability that another representation at that layer lies within distance r of it. The local growth rate of this probability is

$$\mathrm{LID}_i^{(\ell)}(r) = \frac{d \log F_i^{(\ell)}(r)}{d \log r}, \tag{6}$$

and the LID at $\mathbf{h}_i^{(\ell)}$ is its limit as $r \to 0$. If the neighborhood grows as $F_i^{(\ell)}(r) \propto r^m$, Eq. (6) equals m, the dimension of the submanifold on which the point locally lies. The distribution $F_i^{(\ell)}$ is unknown, and Eq. (1) is its maximum-likelihood estimate from the k nearest-neighbor distances (Amsaleg et al., 2015). Averaged over inputs, the per-sample estimates give a population-level estimate of the intrinsic dimension of the layer, which is why the two stages of *LIFE-Detect* can be read as the same quantity measured at two scales.

What prior work shows. Ma et al. (2018) find that adversarial examples have markedly higher LID than normal examples at the same layer of an image classifier, and explain it by the adversarial point lying off the data submanifold, so that its neighbors are drawn from more than one manifold and together span a higher-dimensional region. Yin et al. (2024) find the same direction for LLM generations: untruthful responses have higher LID than truthful ones. In both cases LID is compared across inputs at a fixed layer.

What LIFE assumes. LIFE compares the LID of one input across layers. Writing an input's LID at layer ℓ as the population intrinsic dimension of that layer plus the input's own departure from the typical local geometry there, the first term is what FEPoID tracks and peaks at hallucination-sensitive layers (Wang et al., 2026), and the second is the quantity that prior work shows is elevated for abnormal inputs. For a hallucinated input both terms favor the same region, and the layer that maximizes their sum is the one where the input departs most from the population geometry; for a reliable input the second term is small everywhere and the argmax returns the population peak. Two caveats follow. Because raw LID inherits the population term, an argmax over all layers can drift toward a late global maximum when that maximum is not the first effective peak; in the models we study the two coincide (Figure 1), and restricting $\mathcal{M}$ to a window around ℓ_{FEP}, or subtracting the population term, would be the safeguard otherwise. And LID says where the input is unusual, not whether it is hallucinated; the probe at the selected layer makes that decision, and Section 4 and Appendix G.1 test whether the choice is useful.

Neighborhood size. The number of neighbors k sets the scale of the estimate. A small k uses only the closest representations and is sensitive to individual distances; a larger k includes more distant ones, which can change the ranking of layers for an input. Larger k therefore need not improve downstream performance. We use $k = 25$ and report the sensitivity in Appendix E.

C Complete POPE Evaluation

Table 8 reports accuracy, recall, and F1 on POPE under Random, Popular, and Adversarial sampling, averaged over MSCOCO, A-OKVQA, and GQA. Under Random sampling, LIFE improves accuracy and F1 over the unmodified model for all three LVLMs. The gains are modest for LLaVA-1.6 (accuracy 91.34% to 92.53%, F1 91.57 to 92.59) and larger for Qwen3-VL (accuracy 93.24% to 95.29%, F1 93.09 to 95.09). InternVL3.5 also improves over vanilla, although VGA reaches a slightly higher F1 in this setting (92.48 versus 92.18). Under Popular sampling, LIFE improves accuracy and F1 for each model; the largest gain is for Qwen3-VL (accuracy 88.48% to 94.23%, F1 88.80 to 94.06). The Adversarial setting shows the same pattern: accuracy rises from 80.18% to 86.53% for LLaVA-1.6, 84.11% to 92.63% for Qwen3-VL, and 78.68% to 85.36% for InternVL3.5, with F1 improving for all three. Recall is lower than vanilla in several settings, so higher accuracy and F1 do not mean that every positive example is recovered. Overall, the gains hold across all three sampling strategies and are largest on the harder Popular and Adversarial questions.

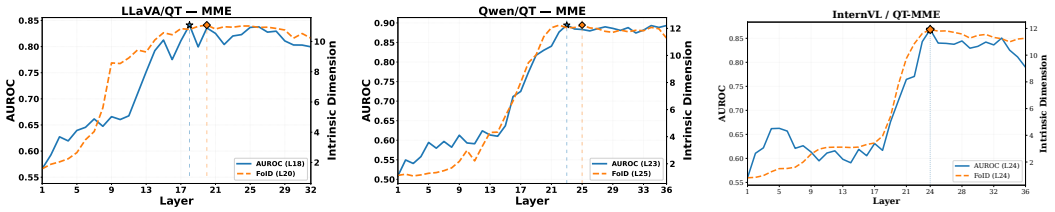
Table 8: Results on POPE, averaged across MSCOCO, A-OKVQA, and GQA for each sampling setting. All metrics are percentages; higher is better.

Method	**LLaVA-1.6-7B**			**Qwen3-VL-8B**			**InternVL3.5-8B**		
	Acc.↑	Rec.↑	F1↑	Acc.↑	Rec.↑	F1↑	Acc.↑	Rec.↑	F1↑
Random									
Vanilla	91.34	93.80	91.57	93.24	91.37	93.09	90.96	92.64	91.08
PAI	90.36	95.13	90.85	93.23	92.62	93.17	90.17	93.62	90.51
VAF	90.24	**95.89**	90.80	93.10	**93.22**	93.09	91.70	95.17	91.99
VGA	90.83	93.29	91.08	92.82	91.29	92.68	92.32	94.53	92.48
SAVAA	91.33	93.78	91.55	93.16	91.06	92.99	91.26	94.97	91.59
LIFE L_i^* (Ours)	**92.53**	**93.33**	**92.59**	**95.29**	**91.80**	**95.09**	**92.49**	**88.53**	**92.18**
Popular									
Vanilla	86.01	93.82	87.15	88.48	91.35	88.80	82.85	92.59	84.35
PAI	84.09	95.13	85.88	88.45	92.60	88.92	81.28	93.53	83.31
VAF	84.64	**95.89**	86.33	87.14	**93.11**	87.88	82.84	95.17	84.74
VGA	83.45	93.22	85.12	87.46	91.29	87.90	83.31	94.50	85.00
SAVAA	86.01	93.80	87.14	88.35	91.04	88.64	82.64	94.91	84.55
LIFE L_i^* (Ours)	**88.71**	**86.10**	**88.41**	**94.23**	**91.80**	**94.06**	**88.24**	**88.53**	**88.27**
Adversarial									
Vanilla	80.18	93.82	82.72	84.11	91.35	85.24	78.68	92.59	81.30
PAI	77.23	95.13	80.93	83.55	92.60	85.00	77.22	93.53	80.44
VAF	77.81	95.89	81.36	82.39	93.11	84.17	78.48	95.17	81.61
VGA	77.46	93.22	80.77	83.05	91.29	84.40	78.71	94.50	81.66
SAVAA	80.15	93.80	82.70	84.09	91.04	85.17	78.46	94.91	81.54
LIFE L_i^* (Ours)	**86.53**	**86.53**	**86.53**	**92.63**	**91.80**	**92.55**	**85.36**	**88.53**	**85.81**

D Experiments with Query-Token Representations

We further evaluate whether hallucination can be detected and corrected before response generation using the query-token (QT) representation. For each input, we extract the hidden state of the final query token, which summarizes the fused visual and textual context immediately before decoding, and apply the same layer localization, detection, and correction as in the main experiments; here the edit is made before any token is generated and the whole response is generated from the edited state.

Hallucination detection. Figure 6 reports layer-wise detection AUROC and intrinsic dimension using QT representations on MME for LLaVA-1.6-7B, Qwen3-VL-8B, and InternVL3.5-8B. Detection performance varies across decoder layers and architectures, so no single fixed layer is best for all LVLMs. In each model, the intrinsic-dimension criterion selects a layer within or near the high-AUROC region without using detection labels. QT representations therefore carry detectable hallucination-related information, and intrinsic dimension can identify an informative layer before answer generation.



(a) LLaVA-1.6-7B (b) Qwen3-VL-8B (c) InternVL3.5-8B

Figure 6: Layer-wise hallucination detection AUROC and automatic layer selection on MME using the QT representation.

Hallucination mitigation. Table 9 reports the corresponding mitigation results. Applying LIFE to QT representations improves MME accuracy over the unmodified LVLM for all three model families, and the transition counts show that the gains come from converting incorrect predictions into correct ones. In this setting the dataset-level layer L_{FoID} and the last layer are stronger than the per-input choice L_i^* for all three models, consistent with Section 4: the per-input rule is not uniformly better than the population-level one. The QT results show that LIFE is not restricted to post-generation signals and can act before decoding begins; generation-conditioned FST representations remain the main setting because they capture the model's own output.

Table 9: MME correction outcomes using the query-token (QT) representation. C is the number of initially incorrect answers that become correct; *Incorrect* is the number that remain incorrect after correction.

Model	Layer	Acc.↑	N	Incorrect↓	C↑
LLaVA-1.6-7B	Vanilla	78.8	874	185	-
	$L/2$	89.7	874	90	95
	$3L/4$	92.1	874	69	116
	Last	92.0	874	70	115
	FoID L_{FoID}	**93.4**	874	58	**127**
	LIFE L_i^* (Ours)	**89.8**	874	89	96
Qwen3-VL-8B	Vanilla	88.6	874	100	-
	$L/2$	97.4	874	23	77
	$3L/4$	97.6	874	21	79
	Last	**97.9**	874	**18**	**82**
	FoID L_{FoID}	**97.9**	874	**18**	**82**
	LIFE L_i^* (Ours)	**96.5**	874	31	69
InternVL3.5-8B	Vanilla	87.2	874	112	-
	$L/2$	93.7	874	55	57
	$3L/4$	97.4	874	23	89
	Last	**97.6**	874	**21**	**91**
	FoID L_{FoID}	97.4	874	23	89
	LIFE L_i^* (Ours)	**96.0**	874	35	77

E SENSITIVITY TO THE NEIGHBORHOOD SIZE k

Figure 7 studies the sensitivity of LIFE to the neighborhood size k used for LID estimation. Across $k \in \{10, 25, 40, 60\}$, performance varies only moderately for all three LVLMs, with $k = 25$ consistently providing a favorable operating point. On POPE, $k = 25$ yields the highest F1 and error-correction rate for all three models, while larger values lead to only small degradations. A similar trend appears on CHAIR, where $k = 25$ gives the lowest CHAIR_I and the highest error-correction rate. This behavior is consistent with the role of k in controlling the locality of the LID estimate: very small neighborhoods can be sensitive to individual nearest-neighbor distances, whereas larger neighborhoods incorporate more distant representations and may smooth the local geometric signal. Overall, the limited variation across the tested range indicates that LIFE is not highly sensitive to the precise choice of k. We therefore use $k = 25$ in all main experiments.

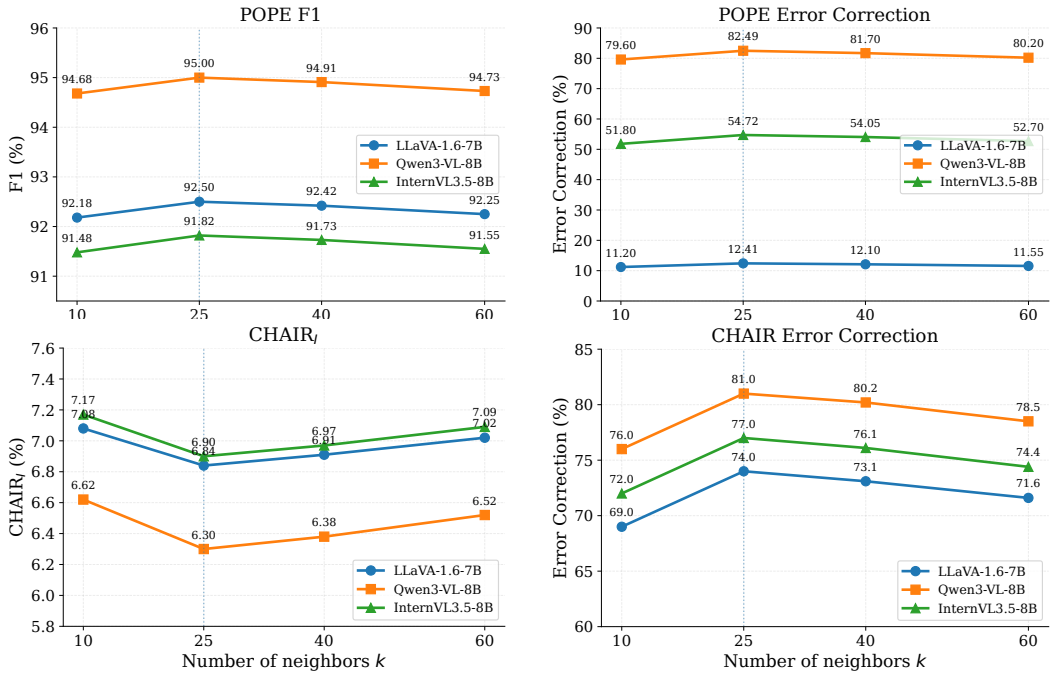


Figure 7: **Sensitivity of LIFE to LID neighborhood size k.** POPE F1 and error correction (top); CHAIR$_I$ and error correction (bottom). Results for three LVLMs at $k \in \{10, 25, 40, 60\}$.

F Qualitative Examples

Figure 8 shows AMBER and MME examples for the three LVLM families. On AMBER, the vanilla models introduce unsupported objects, attributes, or colors; LIFE removes them while keeping the image-grounded content. On MME, LIFE corrects wrong binary answers. Figure 9 compares vanilla and LIFE responses to the same HaloQuest questions across LLaVA, Qwen, and InternVL: unsupported claims in the vanilla responses are removed after intervention, while the corrected responses stay grounded in the visible image content. The aggregate correction results are in Table 4.

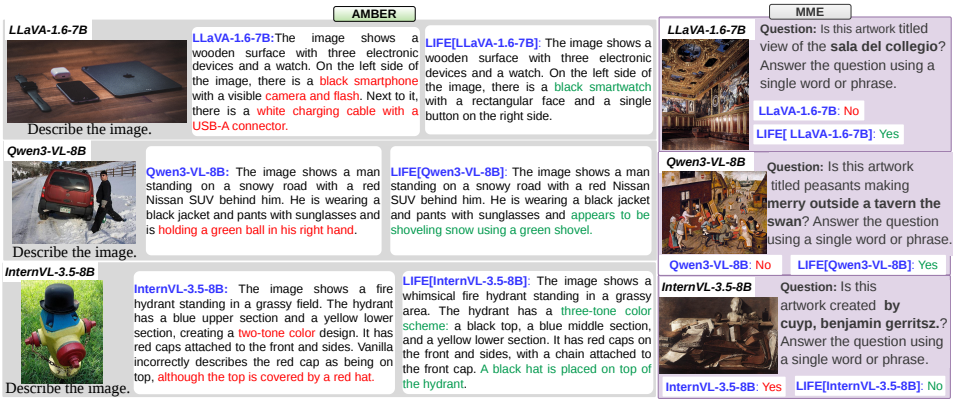


Figure 8: **Qualitative results on AMBER and MME.** Across LLaVA-1.6-7B, Qwen3-VL-8B, and InternVL3.5-8B, LIFE removes unsupported content (red) and produces visually grounded corrections (green).

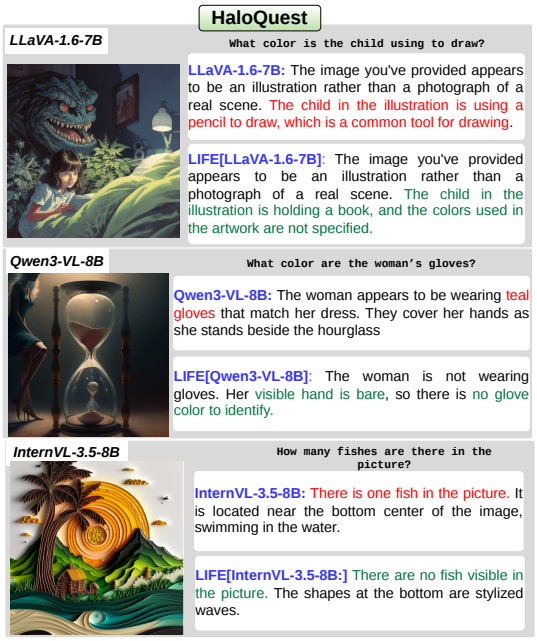


Figure 9: Qualitative results on HaloQuest . Red highlights unsupported claims; green highlights image-grounded answers.

G Additional Analyses

G.1 Detection at the Selected Layer

This is the direct test of the per-input rule. For each held-out sample i we record the probability of the probe at its selected layer, $p_i = g_{\phi_{\ell_i^*}}(\mathbf{h}_i^{(\ell_i^*)})$, and compute one AUROC over all samples from these probabilities and the judge labels. We compare it with the AUROC of the single probe at ℓ_{FEP} and at the best fixed layer (the maximum over layers, which uses the labels to pick the layer and is therefore an upper reference rather than a method). Table 10 shows that selecting a layer for each input yields detection AUROC close to using the fixed FEPoID layer: it is higher in six of twelve settings and within one percentage point in nine. The best fixed layer has the highest AUROC in every setting, but selecting that layer requires held-out labels.

Table 10: Probe AUROC (%) on the held-out split at the per-input selected layer ℓ_i^*, the FEPoID layer ℓ_{FEP}, and the best fixed layer. The best fixed layer is selected using held-out labels and serves as an upper reference for fixed layers.

Model	Benchmark	ℓ_i^*	ℓ_{FEP}	Best fixed layer
LLaVA-1.6-7B	AMBER	94.5	94.6	94.8
	POPE	91.8	90.0	92.2
	MME	85.4	86.0	86.3
	HaloQuest	80.7	79.7	81.0
Qwen3-VL-8B	AMBER	97.2	97.5	97.5
	POPE	97.3	96.8	97.5
	MME	84.1	87.2	89.5
	HaloQuest	82.7	82.6	83.1
InternVL3.5-8B	AMBER	95.5	94.7	96.0
	POPE	95.6	96.8	97.1
	MME	83.7	84.2	84.2
	HaloQuest	85.5	84.8	85.9

G.2 Response Length and Coverage

A lower hallucination rate can arise when a model describes fewer objects. Table 11 examines this possibility by comparing mean response length and object coverage for the vanilla model and LIFE. Across all three models, both methods produce responses averaging 64 words, yet LIFE substantially reduces the AMBER hallucination rate and CHAIR_S. These gains are accompanied by modest reductions in coverage: AMBER Cover decreases by 2.8-4.8 percentage points and CHAIR object recall by 2.0-2.3 percentage points. although it mentions slightly fewer real objects than the vanilla model.

Table 11: Mean response length (words), AMBER Cover (%), and CHAIR recall (%) for the vanilla model and LIFE.

Model	Method	AMBER		CHAIR		Coverage	
		Length	Hal.↓	Length	CHAIR_S ↓	AMBER Cover↑	CHAIR Recall↑
LLaVA-1.6-7B	Vanilla	64	21.3	64	33.80	43.1	42.0
	LIFE	64	18.82	64	13.20	39.4	40.0
Qwen3-VL-8B	Vanilla	64	14.24	64	58.80	36.9	48.2
	LIFE	64	11.00	64	10.40	34.1	46.1
InternVL3.5-8B	Vanilla	64	22.43	64	41.40	42.6	45.3
	LIFE	64	14.20	64	17.20	37.8	43.0

H Enlarged Localization Plots

Figure 10 repeats the twelve panels of Figure 3 at a readable size.

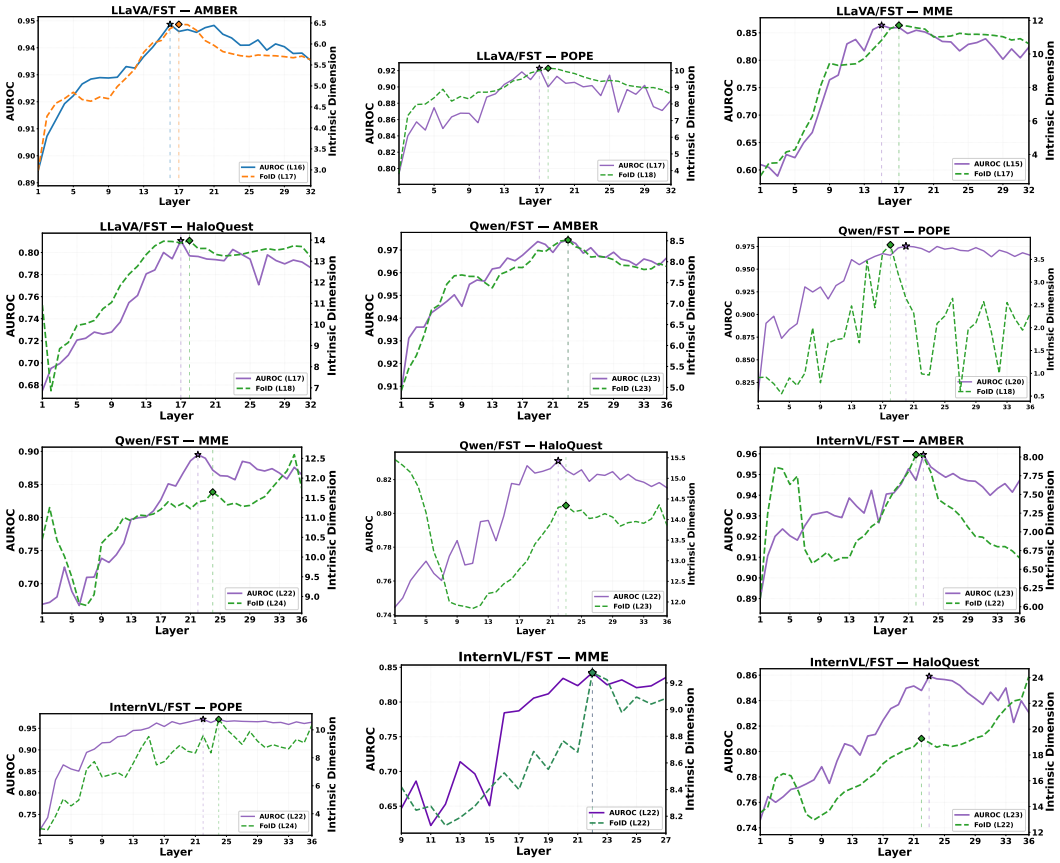


Figure 10: Layer-wise probe AUROC and population-level intrinsic dimension (FST representations), enlarged. Row-major order: LLaVA-1.6-7B on AMBER, POPE, MME, HaloQuest; Qwen3-VL-8B on the same four; InternVL3.5-8B on the same four.