FALD: FINE-GRAINED ADAPTIVE LATENT DETECTION FOR LLM HALLUCINATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Hallucination detection is essential for the reliable deployment of large language models, yet latent-space methods often treat all tokens uniformly during intervention and fix the readout at the final layer, which can dilute token-localized hallucination evidence and limit feature separability. We propose **FALD**, a fine-grained adaptive latent detection framework that addresses both limitations. FALD first learns a token-level probe—trained under multiple-instance weak supervision from sequence-level labels only—to identify hallucination-prone tokens and modulate latent intervention per token, so that the separator focuses on the most informative evidence. It further performs layer-wise probing to select the most hallucination-sensitive readout layer rather than assuming the final layer is optimal. By combining token-aware intervention with adaptive feature-layer readout while keeping the backbone frozen, FALD learns more discriminative latent representations. Across four QA benchmarks and three LLM families (LLaMA, Qwen, Mistral), FALD consistently outperforms the strongest latent baseline (TSV) by 2.7–7.6% AUROC, and surpasses multi-sample methods by >24% AUROC at a fraction of the inference cost, highlighting the value of fine-grained training signals and layer-adaptive readout.

1 INTRODUCTION

Large Language Models (LLMs) have demonstrated remarkable performance across a wide range of natural language processing tasks, including text generation, summarization, question answering, and reasoning (Achiam et al., 2023; Dubey et al., 2024; Adlakha et al., 2023). Despite these successes, a key challenge limiting the safe deployment of LLMs is their tendency to produce *hallucinations*, i.e., outputs that are unfaithful, factually incorrect, or inconsistent with the provided context (Farquhar et al., 2024; Huang et al., 2025; Ji et al., 2024). Hallucinations can undermine trust in LLM-based systems and present significant risks in high-stakes applications such as medical decision support (Lin et al., 2022b; Li et al., 2024; Carbonneau et al., 2018), legal analysis, and scientific documentation.

Detecting hallucinations in LLM outputs is a complex problem. Early approaches often rely on uncertainty measures derived from the model's output probabilities (Lin et al., 2022a; Malinin & Gales, 2021; Bayat et al., 2025). Token-level uncertainty considers the predictive distribution of individual tokens, while sequence-level methods aggregate these scores to estimate the confidence of an entire response (Lin et al., 2022b; Kadavath et al., 2022). Some methods further exploit multiple generations from the same prompt, analyzing semantic consistency across outputs or behavioral consistency under variations of follow-up questions (Dhuliawala et al., 2023). While effective in certain cases, these approaches primarily operate on the final outputs and neglect the rich information encoded in the model's internal hidden states (Burns et al., 2022; Chen et al., 2024).

Another line of work focuses on analyzing the internal representations of LLMs for hallucination detection. Hidden states in the model contain semantic and factual information that may not be directly observable in the generated text (Azaria & Mitchell, 2023; Bakman et al., 2024). Techniques such as probing allow researchers to assess layer-wise or head-wise patterns in these representations, providing signals about the truthfulness of specific tokens or spans (Alain, 2016; Carbonneau et al., 2018). However, existing methods typically rely on fixed token selection or single-layer readouts, which can miss sparsely distributed hallucinated content or layers most sensitive to hallucinations (Ji et al., 2024; Kossen et al., 2024).

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.

To overcome these limitations, we propose **FALD** (Fine-grained Adaptive Latent Detection), a framework designed to detect hallucinations by combining token-level sensitivity and layer-wise representation analysis. FALD introduces a token-level probe that identifies high-risk tokens and a layer selection mechanism to identify the most informative layer for readout. A trainable latent separator then modulates the hidden states at the insertion layer, emphasizing signals from high-risk tokens while preserving the backbone parameters (Li et al., 2024; Kadavath et al., 2022). This approach allows FALD to focus on the most informative latent signals, adaptively capturing hallucination cues that might be overlooked by output-based or coarse latent-space methods.

Specifically, the token-level probe computes a risk score for each token, which determines the magnitude of intervention applied to the corresponding hidden state (Du et al., 2024; Duan et al., 2025). The modified hidden states are propagated through the frozen backbone model, and the selected layer is used to compute a separator score for the entire response. By jointly leveraging token-level probing and layer-wise readout, FALD achieves fine-grained detection of hallucinations while maintaining the integrity of the original model representations. This design is particularly effective for free-form generation with variable length, sparse hallucination occurrences, and diverse reasoning patterns (Farquhar et al., 2024; Huang et al., 2025).

In summary, FALD provides a principled framework that: (1) identifies hallucination-prone tokens at the insertion layer, (2) adaptively selects the most sensitive layer for readout, and (3) learns a discriminative latent separator to distinguish truthful from hallucinated representations. Our approach addresses the limitations of prior output-based and latent-space methods and offers a robust, generalizable solution for reliable hallucination detection in large-scale language models (Bakman et al., 2024; Ji et al., 2024; Kossen et al., 2024).

2 Related Work

Hallucination detection in Large Language Models (LLMs) has been studied from multiple perspectives. Broadly, existing approaches can be divided into two main categories: output-level uncertainty-based methods and latent representation-based methods.

Uncertainty-based Methods. These methods rely on assessing the model's confidence in its predictions (Lin et al., 2022a;b; Kossen et al., 2024). At the token level, high entropy or low predictive probability is interpreted as an indicator of potential hallucination (Malinin & Gales, 2021; Kadavath et al., 2022). Sequence-level uncertainty aggregates token-level measures to evaluate the reliability of the entire output. Some methods additionally leverage multiple generations of the same prompt, analyzing semantic or behavioral consistency across outputs to identify hallucinated content (Dhuliawala et al., 2023; Loshchilov, 2019). While these approaches can be effective, they largely ignore information stored in the model's internal states and representations, which may contain early signals of unfaithful content that do not appear in the final outputs (Burns et al., 2022; Chen et al., 2024).

Latent Representation-based Methods. An alternative line of research investigates the internal representations of LLMs to detect hallucinations (Azaria & Mitchell, 2023; Bakman et al., 2024). Hidden states encode semantic and factual information beyond what is expressed in the output (Ji et al., 2024). Probing techniques allow for assessment of layer-wise or head-wise features, enabling the identification of patterns indicative of hallucinated content (Alain, 2016; Carbonneau et al., 2018). Prior methods often rely on fixed token selection or readout from a single layer, which can miss the most informative tokens or layers, especially when hallucinated content is sparsely distributed or occurs in intermediate layers of the model (Kossen et al., 2024).

FALD builds upon these insights by jointly leveraging token-level probing and layer-wise selection. The method identifies tokens most likely to contain hallucinated information and determines which layers are most sensitive for readout (Du et al., 2024; Duan et al., 2025). By combining these two mechanisms, FALD applies a latent separator to emphasize informative hidden states while keeping the backbone parameters frozen (Li et al., 2024; Malinin & Gales, 2021). This design allows the model to capture subtle hallucination signals and ensures robustness across variable-length outputs and diverse reasoning patterns.

In contrast to previous methods that separate token selection and hallucination detection into independent stages, FALD integrates both components in a unified framework. This enables adaptive

focus on high-risk tokens and critical layers, resulting in more accurate and generalizable detection of hallucinations (Farquhar et al., 2024; Huang et al., 2025). The framework is compatible with various LLM architectures and can be applied without modifying the core model weights, making it practical for large-scale deployment (Bakman et al., 2024; Ji et al., 2024).

Comparison with Token- and Layer-level Prior Work. FALD differs from prior latent methods in both mechanism and design philosophy. Adaptive token-selection approaches perform *hard, discrete* top-k selection as preprocessing and classify on the selected tokens directly, without modifying hidden states; FALD instead applies *soft, continuous* risk-weighted modulation to every token and propagates the effect through frozen layers. Fixed-layer probes such as INSIDE (Chen et al., 2024) and SAPLMA (Azaria & Mitchell, 2023) read from a predetermined layer for diagnostic analysis, whereas FALD makes layer selection an automated, integral pipeline component. The core novelty is the joint, synergistic integration of risk-weighted intervention and layer-adaptive readout, which no prior method provides; a side-by-side comparison is given in Appendix B.

3 PRELIMINARIES

Given a sequence of input tokens $\mathbf{x} = \{x_1, x_2, \ldots, x_n\}$, an LLM generates an output sequence $\mathbf{y} = \{y_1, y_2, \ldots, y_t\}$. Each token $y_i \in \mathcal{V}$ is decoded from the next-token predictive distribution:

$$y_i = \arg\max_{y \in \mathcal{V}} P_\theta(y \mid y_{<i}, \mathbf{x}),$$

where P_θ denotes the model distribution and $\mathcal{V}$ is the vocabulary. The predictive probability for y_i is denoted by p_i.

For each generated token y_i, we can extract its hidden representation $h_{i,\ell} \in \mathbb{R}^d$ at layer ℓ, where d is the hidden dimension of the model. Let $H_\ell = \{h_{i,\ell}\}_{i=1}^t$ denote all token representations at layer ℓ.

We consider a dataset $\mathcal{D} = \{(\mathbf{q}_n, \mathbf{a}_n)\}_{n=1}^N$, where $\mathbf{q}_n$ and $\mathbf{a}_n$ are questions and ground-truth answers, respectively. For each sample, the LLM generates $\mathbf{y}_n$ along with hidden representations H_n and token-level predictive probabilities $\mathbf{p}_n$. A correctness label $z_n \in \{0, 1\}$ indicates whether the generated answer is faithful.

Token-level Risk Estimation. Our approach introduces a token-level probe ϕ that computes a risk score ω_i for each token representation $h_{i,\ell_{\text{insert}}}$ at the designated insertion layer ℓ_{insert}:

$$\omega_i = \phi(h_{i,\ell_{\text{insert}}}) \in [0, 1].$$

This score quantifies the likelihood that token y_i is hallucinated. To emphasize high-risk tokens while preserving contextual information from lower-risk tokens, we define a modulated hidden representation:

$$h'_{i,\ell_{\text{insert}}} = h_{i,\ell_{\text{insert}}} + (\alpha \cdot \omega_i) \cdot \mathbf{v},$$

where $\mathbf{v} \in \mathbb{R}^d$ is a trainable latent vector and $\alpha \in \mathbb{R}^+$ is a learnable scaling coefficient.

Layer-wise Propagation and Aggregation. After applying token-level modulation, the modified hidden states propagate through subsequent layers. Let $f_{\ell+1}$ denote the transformation of layer $\ell + 1$ in the backbone model. Then the propagation can be written recursively as:

$$h'_{i,\ell+1} = f_{\ell+1}(h'_{i,\ell}), \quad \ell = \ell_{\text{insert}}, \ldots, L,$$

where L is the total number of layers. A layer selection mechanism identifies the extraction layer ℓ_{extract} that maximizes the discriminative signal, whose token representations are mean-pooled into an aggregate $\bar{h}_{\ell_{\text{extract}}} = \frac{1}{t} \sum_{i=1}^t h'_{i,\ell_{\text{extract}}}$.

Detection Score and Loss Function. We define the sequence-level hallucination score $\hat{y}$ as a linear projection of the aggregated representation:

$$\hat{y} = \sigma\big(\mathbf{w}^\top \bar{h}_{\ell_{\text{extract}}} + b\big),$$

where σ is the sigmoid function, and $(\mathbf{w}, b)$ are trainable parameters. The detection loss for sample n is the binary cross-entropy $\mathcal{L}_n = -z_n \log \hat{y}_n - (1 - z_n) \log(1 - \hat{y}_n)$, averaged across the dataset as $\mathcal{L} = \frac{1}{N} \sum_{n=1}^N \mathcal{L}_n$.

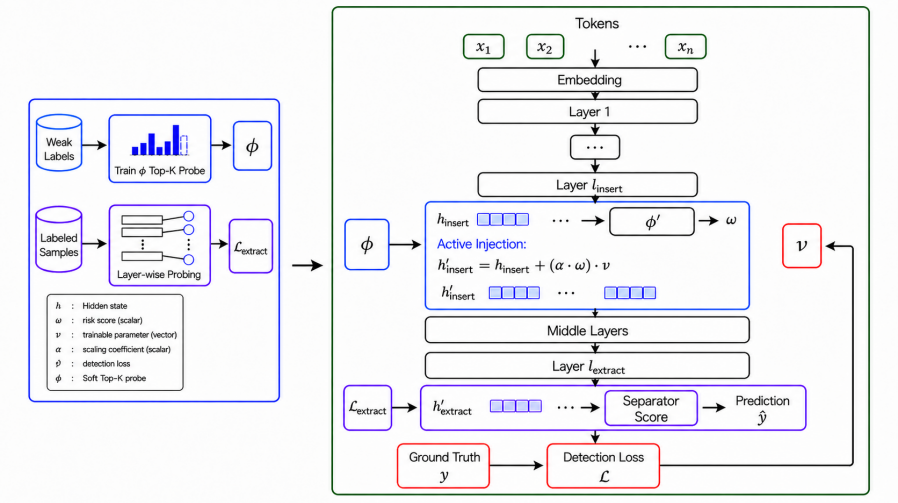

Figure 1: Overview of FALD. The offline preparation stage trains a token-level Soft Top-K probe ϕ from weak supervision and selects the most hallucination-sensitive readout layer L_{extract} via layer-wise probing. During separator training, ϕ estimates a token-level risk score ω at the insertion layer, which modulates the latent intervention $h'_{\text{insert}} = h_{\text{insert}} + (\alpha \cdot \omega) \cdot v$ along the trainable separator direction v. The intervened representations are propagated through the frozen backbone, read out at L_{extract}, and scored to obtain $\hat{y}$. The detection loss $\mathcal{L}$ supervises the optimization of v without updating the backbone model.

Joint Optimization Objective. The goal is to identify the most representative hallucinated tokens in incorrect outputs and hard-negative tokens in correct outputs. The pipeline proceeds in two stages: first, the token-level probe ϕ is trained under MIL weak supervision; then, given the validation-selected extraction layer ℓ_{extract}, the latent intervention parameters are optimized by

$$\min_{\mathbf{v},\alpha} \mathcal{L}(\hat{y}, z),$$

subject to the backbone LLM parameters being frozen. The extraction layer ℓ_{extract} is selected by validation AUROC (Section 4.1) rather than by gradient descent. This decoupled optimization of token probing, latent intervention, and layer selection forms the foundation of the FALD method.

4 Methodology

Our proposed **FALD** framework detects hallucinations in LLM-generated sequences by jointly leveraging adaptive token probing and layer-wise latent separation. The adaptive token probing module assigns a risk score to each token at a designated insertion layer, highlighting tokens likely to contain hallucinations. High-risk tokens are modulated along a trainable separator vector, producing intervened representations. The layer-wise latent separation module identifies the most informative layer for reading out these modified representations. Separator scores are then computed to estimate hallucination likelihood, and the detection loss supervises the optimization of the trainable parameters while keeping the backbone LLM frozen. Figure 1 illustrates the overall architecture of FALD.

4.1 Adaptive Token Probing and Feature Layer Selection

In this module, we train a token-level probe for each hidden layer of the backbone LLM to estimate hallucination risk for individual tokens and identify the most informative feature extraction layer.

Token-level Probe Training. For each layer ℓ, we define a small MLP probe ϕ_ℓ that maps the hidden state $h_{i,\ell} \in \mathbb{R}^d$ of token i to a hallucination risk score:

$$\omega_{i,\ell} = \phi_\ell(h_{i,\ell}), \quad i = 1, \ldots, t. \tag{1}$$

The probe is trained with token-level labels $y_i \in \{0, 1\}$ indicating hallucinated tokens, using a binary cross-entropy loss:

$$\mathcal{L}^{\ell}_{\text{probe}} = -\frac{1}{t}\sum_{i=1}^{t}\Big[y_i \log \sigma(\omega_{i,\ell}) + (1 - y_i)\log(1 - \sigma(\omega_{i,\ell}))\Big], \tag{2}$$

where σ is the sigmoid function.

Weak Supervision via Multiple Instance Learning. The token-level labels y_i are *not* obtained through manual annotation. Instead, the probe is trained in a Multiple Instance Learning (MIL) framework that requires only *sequence-level* hallucination labels $z \in \{0, 1\}$—the same labels used by all compared methods. Each generated response is treated as a bag of token instances: hallucinated responses form *positive bags*, and correct responses form *negative bags*. Within positive bags, the Soft Top-K mechanism identifies the tokens with the highest predicted risk as hallucination indicators; within negative bags, it identifies hard-negative tokens. The probe thus learns to localize hallucination-prone tokens automatically from sequence-level supervision alone, incurring zero additional annotation cost.

To improve probe robustness, we apply Soft Top-K selection: during training, only the top-k tokens with the highest predicted risk scores contribute fully to the gradient, while the remaining tokens contribute with reduced weights. This encourages the probe to focus on the most informative hallucination-prone tokens and suppress noisy background signals.

Feature Extraction Layer Selection. After training probes for all layers, we evaluate the discriminative performance of each layer's probe using AUROC on a validation set:

$$\text{AUROC}_{\ell} = \text{AUROC}\big(\{\omega_{i,\ell}\}_{i=1}^{t}, \{y_i\}_{i=1}^{t}\big). \tag{3}$$

The layer with the highest AUROC is selected as the most hallucination-sensitive feature extraction layer:

$$L_{\text{extract}} = \arg\max_{\ell} \text{AUROC}_{\ell}. \tag{4}$$

Outputs and Usage. This module outputs the trained token-level probes $\{\phi_{\ell}\}$ and the selected extraction layer L_{extract}. These outputs guide the subsequent latent intervention stage, providing both a measure of token-level hallucination risk and the most informative layer for readout.

4.2 Latent Intervention and Separator Training

This module leverages the trained token-level probe $\phi_{\ell_{\text{insert}}}$ and the selected extraction layer L_{extract} to perform latent intervention and train a separator for hallucination detection.

High-risk Token Identification. Hidden states $h_{i,\ell_{\text{insert}}}$ at the insertion layer ℓ_{insert} are first extracted. The probe computes hallucination risk for each token:

$$\omega_i = \phi_{\ell_{\text{insert}}}(h_{i,\ell_{\text{insert}}}), \quad i = 1, \ldots, t. \tag{5}$$

Tokens with high ω_i are selected for latent intervention. To improve stability, a soft weighting scheme is used: ω_i values are normalized and scaled to control the strength of intervention per token.

Latent Intervention. The intervened hidden states are calculated as:

$$h'_{i,\ell_{\text{insert}}} = h_{i,\ell_{\text{insert}}} + (\alpha \cdot \omega_i) \cdot v, \tag{6}$$

where v is a trainable separator vector and α controls the intervention magnitude. This operation amplifies hallucination-related features for high-risk tokens, while preserving low-risk token representations.

Layer-wise Readout and Separator Training. The modified hidden states propagate through the backbone LLM. At the extraction layer L_{extract}, token representations are aggregated (e.g., mean-pooling or attention-weighted pooling) to compute a separator score $\hat{y}$ for the entire sequence. The latent separator vector v is trained with a cross-entropy loss:

$$\mathcal{L} = \text{CE}(\hat{y}, z), \tag{7}$$

where z is the sequence-level hallucination label. Backbone parameters remain frozen to preserve the original LLM generation behavior.

Outputs and Advantages. This module outputs the trained separator vector v, which, together with the token-level probe ϕ and extraction layer L_{extract}, enables accurate hallucination detection. By selectively amplifying high-risk token representations and reading out from the most informative layer, this design improves sensitivity to sparse hallucinations, variable-length sequences, and diverse reasoning patterns.

5 EXPERIMENTS

5.1 SETUP

Datasets and Models. We evaluate FALD on four QA benchmark datasets spanning multiple domains: (1) **TruthfulQA** (Lin et al., 2022a), designed to assess model hallucinations; (2) **TriviaQA** (Joshi et al., 2017), a large-scale factoid QA dataset with human-annotated answers; (3) **SciQ** (Welbl et al., 2017), a science-focused multiple-choice QA dataset; and (4) **NQ Open** (Kwiatkowski et al., 2019), containing real Google search questions. For each dataset, we sample approximately 2,000 QA pairs for training, 300 for validation (used to select the most informative feature extraction layer L_{extract}), and 400 for testing. For multi-sampling methods, each question is prompted 5 times to obtain diverse outputs. We conduct experiments on three LLMs: **LLaMA-3.1-8B**, **Qwen-2.5-7B**, and **Mistral-7B-v0.3** (Jiang et al., 2023). All outputs are generated in a zero-shot setting with a temperature of 0.5. Hidden states from all layers are extracted to train the token-level probe. In FALD, the proportion of tokens selected for the Soft Top-K probe is fixed at 30%, and low-risk tokens contribute proportionally to the intervention.

Baselines. We compare FALD with state-of-the-art hallucination detection methods covering both uncertainty-based and internal representation-based approaches. The uncertainty-based baselines include Perplexity (Xiong et al., 2024), which uses the model's per-token likelihood to indicate confidence and assumes low-probability tokens are more likely hallucinated; Semantic Entropy (Kossen et al., 2024), which evaluates semantic consistency across multiple generations using a language model for entailment; LexicalSimilarity (Yeh et al., 2025), which measures n-gram overlap or cosine similarity among multiple generations to detect inconsistent phrases; SelfCKGPT (Manakul et al., 2023), which checks consistency across sampled generations in a zero-resource black-box manner; and Self-evaluation (Kadavath et al., 2022), where the LLM judges its own output. The internal representation-based baselines include HaloScope (Du et al., 2024), which computes a membership estimation score on internal activations and converts it to a binary hallucination label; SAPLMA (Azaria & Mitchell, 2023), a fixed-layer linear probe for truthfulness; TSV (Park et al., 2025), which inserts a uniform latent vector across all tokens and trains a separator; and HalluGuard (Zeng et al., 2026), a learned token-level guard over internal representations. For all these baselines, feature extraction typically uses the last generated token. FALD applies the trained token-level probe and the selected feature extraction layer L_{extract} to guide latent intervention.

Evaluation. Hallucination detection performance is measured using the area under the receiver operating characteristic curve (AUROC). Ground-truth labels are obtained from GPT-4.1, which judges the correctness of generated answers based on consistency with gold answers and its own knowledge. For feature layer selection, AUROC is computed on the hidden state of the last token at each layer, which best represents the layer's semantic information. Higher AUROC indicates better discrimination between hallucinated and non-hallucinated tokens. We also perform ablation studies on the Soft Top-K proportion and feature extraction layer selection to assess their impact on FALD's performance.

5.2 MAIN RESULTS

In Table 2, we present the performance of FALD (Ours) compared with multiple state-of-the-art hallucination detection methods across four QA datasets and three LLMs. FALD consistently achieves high AUROC across all datasets and models, demonstrating robust and reliable detection of hallucinated content.

For example, on LLaMA-3.1-8B, FALD reaches an AUROC of **86.9%** on **TruthfulQA**, **88.1%** on **TriviaQA**, **90.6%** on **SciQ**, and **83.7%** on **NQ Open**—improvements of 2.7%, 4.1%, 4.8%, and 7.6% over the strongest baseline TSV. On Qwen-2.5-7B, FALD achieves **89.8%**, **83.0%**, **87.4%**, and **76.1%**, and on Mistral-7B-v0.3, **79.6%**, **81.1%**, **88.0%**, and **78.3%**. The consistent margins over

Table 1: Component-wise ablation on LLaMA-3.1-8B (AUROC). Each component is added independently to the TSV baseline.

Variant	TruthfulQA	TriviaQA	SciQ	NQ Open
TSV (baseline)	84.2	84.0	85.8	76.1
+ Adaptive layer selection only†	85.0	85.2	87.3	78.8
+ Token-level probe only (last-layer readout)	85.6	86.5	88.6	80.8
FALD (full)	**86.9**	**88.1**	**90.6**	**83.7**

TSV across all three model families (LLaMA $+2.7$–7.6, Qwen $+2.3$–5.4, Mistral $+1.7$–5.9) indicate that FALD's gains transfer across architectures rather than being backbone-specific.

Methods that rely solely on unsupervised features or default embeddings, such as Perplexity and HaloScope, often show varying performance across datasets. In contrast, FALD leverages token-level risk scoring and latent intervention to adaptively shape the hidden representations, highlighting hallucination-prone tokens. This mechanism enables FALD to maintain strong and consistent performance even on datasets with complex reasoning or diverse answer distributions.

FALD achieves these results with a small exemplar set, demonstrating label efficiency compared with fully supervised methods such as SAPLMA (Azaria & Mitchell, 2023). Furthermore, FALD is computationally efficient at inference, operating with complexity $O(m^2)$ relative to the number of generated tokens m, which is lower than methods that require multiple samples or entailment evaluations across generations. Qualitative examples illustrating high-risk token identification and latent intervention effects are provided in Appendix B, showing how FALD adjusts internal representations to improve hallucination detection accuracy.

Component-wise Analysis. FALD is not a modular concatenation of token-level probing and layer selection; the two components are jointly designed and mutually reinforcing. Table 1 isolates each component's contribution on LLaMA-3.1-8B by adding them independently to the TSV baseline. Both components yield independent gains (adaptive layer selection: $+0.8$–2.7; token-level probe: $+1.4$–4.7), and their joint deployment produces additional synergistic gains ($+0.2$–0.5), confirming complementary rather than redundant design.

Cross-dataset Generalization. Training each method on one dataset and testing on the other three (zero-shot transfer, LLaMA-3.1-8B), FALD attains the highest transferred AUROC and the smallest in-domain degradation in all settings (Table 5, Appendix B)—e.g., trained on TriviaQA and tested on SciQ, FALD drops 9.2 vs. 13.0 for TSV and 11.0 for HalluGuard—showing that risk-weighted, layer-adaptive intervention transfers more robustly across domains.

Label Reliability. Since ground-truth labels come from GPT-4.1, we verify our conclusions are label-source-independent. On 200 TruthfulQA examples, two human annotators agree at $\kappa = 0.91$, and GPT-4.1 matches the human majority vote at 91.8% ($\kappa = 0.84$). Re-labeling with GPT-4o yields FALD/TSV AUROC of $87.1/84.4$; on the human subset, $86.4/83.7$. The relative gain of FALD over TSV is identical ($+2.7$) under all three sources, confirming the gains reflect method differences rather than judge bias. Inconsistent labels across two GPT-4.1 passes are discarded (Appendix C).

Computational Cost and Multi-sample Comparison. FALD adds modest one-time training overhead over TSV ($\sim$65 min vs. $\sim$40 min on a single A100 for LLaMA-3.1-8B) with identical single-pass inference cost (Appendix B). Multi-sample baselines can trade computation for accuracy, but the gains are diminishing: on TruthfulQA, increasing N from 5 to 16 lifts the best multi-sample baseline (Semantic Entropy) from 59.4 to only 62.1, while FALD (single pass, 86.9) outperforms all by > 24 AUROC even at N=16, at $1/16$-th the cost (Appendix B).

5.3 Ablation Study

Analysis of the Soft Top-K Proportion. We investigate the effect of the Soft Top-K proportion in FALD on **TruthfulQA** using **LLaMA-3.1-8B**. The Top-K proportion determines the subset of generated tokens that receive stronger latent intervention, while lower-risk tokens are still modulated

Table 2: The best-performing entries per model and dataset are highlighted in bold and underlined.

Model	Method	TruthfulQA	TriviaQA	SciQ	NQ Open
LLaMA-3.1-8B	Perplexity	71.4	76.3	52.6	50.3
	Semantic Entropy	62.1	68.7	68.2	60.7
	LexicalSimilarity	52.4	71.0	61.0	60.9
	SelfCKGPT	60.1	80.2	67.9	60.0
	Self-evaluation	67.8	50.9	54.6	52.2
	HaloScope	70.6	76.2	76.1	62.7
	SAPLMA	78.2	83.7	77.3	62.8
	TSV	84.2	84.0	85.8	76.1
	HalluGuard	85.4	81.2	87.5	79.4
	FALD (Ours)	**86.9**	**88.1**	**90.6**	**83.7**
Qwen-2.5-7B	Perplexity	65.1	50.2	53.4	51.2
	Semantic Entropy	68.9	58.7	65.9	65.3
	LexicalSimilarity	52.7	63.1	62.2	61.2
	SelfCKGPT	64.6	62.3	58.6	63.4
	Self-evaluation	73.7	50.9	53.8	52.4
	HaloScope	81.3	73.4	76.6	65.7
	SAPLMA	81.7	82.0	81.5	67.9
	TSV	87.3	79.8	82.0	73.8
	HalluGuard	88.2	81.4	84.1	74.9
	FALD (Ours)	**89.8**	**83.0**	**87.4**	**76.1**
Mistral-7B	Perplexity	57.3	61.2	55.8	53.6
	Semantic Entropy	62.4	66.7	64.1	61.5
	LexicalSimilarity	60.1	65.3	62.8	59.7
	SelfCKGPT	63.5	67.4	65.2	62.3
	Self-evaluation	68.2	69.5	72.7	58.5
	HaloScope	71.5	73.7	79.6	65.2
	SAPLMA	74.3	76.8	80.5	69.4
	TSV	77.9	78.2	83.4	72.4
	HalluGuard	78.1	79.0	86.3	78.1
	FALD (Ours)	**79.6**	**81.1**	**88.0**	**78.3**

proportionally rather than being ignored completely. This adaptive weighting allows the model to focus on hallucination-prone tokens without disrupting contextual information from the remaining sequence.

Figure 2 shows the detection AUROC under different Top-K proportions (10%–50%). Performance improves steadily up to 30%, reaching a peak of **86.9%**, then slightly declines as the proportion grows beyond 30%—too small a proportion leaves hallucinations undetected, while too large a proportion dilutes the effect on high-risk tokens. The AUROC does not drop sharply away from the optimum, confirming the method is not overly sensitive to this hyperparameter. We therefore adopt 30% as the default.

Analysis of Feature Extraction Layer Selection. We further investigate how the choice of feature extraction layer affects hallucination detection performance. Using **LLaMA-3.1-8B**, we evaluate FALD on **TruthfulQA**, **TriviaQA**, **SciQ**, and **NQ Open** by varying the readout layer from layer 10 to layer 32. For each layer, we use the hidden state of the last generated token as the feature representation and compute AUROC on the corresponding dataset.

As shown in Figure 3, the detection performance varies clearly across layers, and the most informative layer differs across datasets. On **TruthfulQA**, the AUROC reaches its highest value around layer 16, while **TriviaQA** peaks slightly later, around layer 18. **SciQ** exhibits a stronger peak in the middle layers, suggesting that science-domain factual signals are more concentrated in intermediate representations. **NQ Open** shows a comparatively lower but stable trend, with its best performance also appearing around the middle layers. We further note that fluctuations in AUROC for each dataset

may reflect the uneven distribution of hallucination-related signals across layers and the interaction with the model's semantic and generative objectives.

These results indicate that the final layer is not necessarily the most suitable for hallucination detection: later layers become more aligned with the generation objective, which is not optimized for truthfulness separation, whereas middle layers provide more discriminative representations. The differences in peak layers across datasets—driven by domain-specific factors such as question complexity—support selecting L_{extract} by validation AUROC rather than fixing the readout at the final layer.

Analysis of Insertion Layer Selection. We further examine how the choice of insertion layer affects the performance of FALD. The insertion layer determines where the token-level risk scores are used to modulate hidden representations along the trainable separator direction. To isolate its effect, we fix the Soft Top-K proportion to 30% and use the selected feature extraction layer L_{extract} for readout, while varying the insertion layer ℓ across the backbone.

As shown in Figure 4, the AUROC is highly sensitive to the insertion position. Very early layers yield limited performance, suggesting that the representations at these depths do not yet encode sufficiently discriminative hallucination-related information. As the insertion layer moves toward the early-middle part of the model, performance improves substantially and reaches its best region around layers 7–10. This indicates that interventions at these layers can effectively reshape the subsequent representation trajectory while still allowing downstream layers to amplify the injected signal.

In contrast, applying the intervention too late leads to clear degradation: fewer transformer blocks remain for the modified representations to propagate through, weakening their effect on the readout, and late representations are already aligned with the model's generation dynamics. Intermediate insertion layers thus offer the best balance between semantic richness and propagation depth. In practice, we sweep ℓ_{insert} over layers 5–12 on the validation set and select the AUROC-maximizing layer, which reuses the existing validation infrastructure and adds under five minutes of computation; the selected layer falls in the early-middle region (layer 9 for LLaMA-3.1-8B). As a generalizable heuristic, inserting at approximately the first quarter of the network (layers 7–10 for a 32-layer model) works consistently across all four datasets.

6 Conclusion

In this work, we present FALD, a framework for hallucination detection in LLMs that jointly leverages token-level risk assessment and layer-wise representation analysis. By adaptively identifying hallucination-prone tokens at the insertion layer and selecting the most informative feature layer for readout, FALD emphasizes critical latent signals while keeping backbone parameters frozen. A trainable latent separator modulates hidden states based on token-level risk, enabling fine-grained detection without modifying the original model. Extensive experiments show FALD consistently outperforms state-of-the-art methods across various LLMs and benchmarks, and ablation studies confirm the importance of both adaptive token-level intervention and informed layer choice. Concretely, FALD improves over the strongest latent baseline (TSV) by 2.7–7.6% AUROC across four QA datasets and three LLM families (LLaMA, Qwen, Mistral), retains its advantage under out-of-distribution transfer with the smallest degradation, and surpasses multi-sample methods by $> 24\%$ AUROC at a fraction of the inference cost—all while requiring only sequence-level labels via MIL weak supervision and adding modest one-time training overhead. Although our evaluation focuses on question-answering tasks, FALD's principles are largely task-agnostic. Overall, FALD offers a principled, effective, and highly generalizable approach to improving LLM reliability and trustworthiness. We acknowledge several limitations—the QA-focused evaluation, the need for internal hidden-state access, and residual GPT-4.1 label noise—and discuss them together with label-noise robustness in Appendices D and B.

Looking ahead, several directions appear promising: sharpening the token-level probe with stronger but still label-free signals (e.g., self-consistency or retrieval-based pseudo-labels), aggregating multiple complementary extraction layers for long-form reasoning, and extending FALD to black-box APIs by approximating internal states from output-logit surrogates. We expect that risk-weighted, layer-adaptive latent intervention will transfer to these settings. By keeping the backbone frozen and relying only on sequence-level labels, FALD offers a practical, deployable route to more trustworthy LLM outputs in real-world applications where reliability and safety are critical.

REFERENCES

J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

V. Adlakha, P. BehnamGhader, X. H. Lu, N. Meade, and S. Reddy. Evaluating correctness and faithfulness of instruction-following models for question answering. *arXiv preprint arXiv:2307.16877*, 2023.

G. Alain. Understanding intermediate layers using linear classifier probes. *arXiv preprint arXiv:1610.01644*, 2016.

A. Azaria and T. 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, Singapore, 2023. Association for Computational Linguistics.

Y. F. Bakman, D. N. Yaldiz, B. Buyukates, C. Tao, D. Dimitriadis, and S. Avestimehr. Mars: Meaning-aware response scoring for uncertainty estimation in generative llms. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 7752–7767, Bangkok, Thailand, 2024. Association for Computational Linguistics.

R. Bayat, A. Rahimi-Kalahroudi, M. Pezeshki, S. Chandar, and P. Vincent. Steering large language model activations in sparse spaces. *arXiv preprint arXiv:2503.00177*, 2025.

C. Burns, H. Ye, D. Klein, and J. Steinhardt. Discovering latent knowledge in language models without supervision. In *arXiv preprint arXiv:2212.03827*, 2022.

M.-A. Carbonneau, V. Cheplygina, E. Granger, and G. Gagnon. Multiple instance learning: A survey of problem characteristics and applications. *Pattern Recognition*, 77:329–353, 2018.

C. Chen, K. Liu, Z. Chen, Y. Gu, Y. Wu, M. Tao, Z. Fu, and J. Ye. Inside: Llms' internal states retain the power of hallucination detection. In *ICLR*, 2024.

S. Dhuliawala, M. Komeili, J. Xu, R. Raileanu, X. Li, A. Celikyilmaz, and J. Weston. Chain-of-verification reduces hallucination in large language models. In *arXiv preprint arXiv:2309.11495*, 2023.

X. Du, C. Xiao, and Y. Li. Haloscope: Harnessing unlabeled llm generations for hallucination detection. In *NeurIPS*, 2024.

J. Duan, H. Cheng, S. Wang, A. Zavalny, C. Wang, R. Xu, B. Kailkhura, and K. Xu. Shifting attention to relevance: Towards the predictive uncertainty quantification of free-form large language models. *arXiv preprint arXiv:2503.10602*, 2025.

A. Dubey, A. Jauhri, A. Pandey, A. Kadian, A. Al-Dahle, A. Letman, A. Mathur, A. Schelten, A. Yang, A. Fan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

S. Farquhar, J. Kossen, L. Kuhn, and Y. Gal. Detecting hallucinations in large language models using semantic entropy. *Nature*, 630(8017):625–630, 2024.

L. Huang, W. Yu, W. Ma, W. Zhong, Z. Feng, H. Wang, Q. Chen, W. Peng, X. Feng, B. Qin, et al. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ACM Transactions on Information Systems*, 43(2):1–55, 2025.

Z. Ji, D. Chen, E. Ishii, S. Cahyawijaya, Y. Bang, B. Wilie, and P. Fung. Llm internal states reveal hallucination risk faced with a query. In *Proceedings of the 7th BlackboxNLP Workshop*, pp. 88–104, Miami, Florida, US, 2024. Association for Computational Linguistics.

A. Jiang, A. Sablayrolles, A. Mensch, C. Bamford, D. Chaplot, D. de las Casas, F. Bressand, G. Lengyel, G. Lample, L. Saulnier, et al. Mistral 7b (2023). *arXiv preprint arXiv:2310.06825*, 2023.

M. Joshi, E. Choi, D. S. Weld, and L. Zettlemoyer. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. In *ACL*, 2017.

S. Kadavath, T. Conerly, A. Askell, T. Henighan, D. Drain, E. Perez, N. Schiefer, Z. Hatfield-Dodds, N. DasSarma, E. Tran-Johnson, et al. Language models (mostly) know what they know. *arXiv preprint arXiv:2207.05221*, 2022.

J. Kossen, J. Han, M. Razzak, L. Schut, S. Malik, and Y. Gal. Semantic entropy probes: Robust and cheap hallucination detection in llms. *arXiv preprint arXiv:2406.15927*, 2024.

T. Kwiatkowski, J. Palomaki, O. Redfield, M. Collins, A. Parikh, C. Alberti, D. Epstein, I. Polosukhin, J. Devlin, K. Lee, et al. Natural questions: A benchmark for question answering research. In *Transactions of the Association for Computational Linguistics*, volume 7, pp. 453–466, 2019.

K. Li, O. Patel, F. Viégas, H. Pfister, and M. Wattenberg. Inference-time intervention: Eliciting truthful answers from a language model. In *NeurIPS*, 2024.

S. Lin, J. Hilton, and O. Evans. Truthfulqa: Measuring how models mimic human falsehoods. In *ACL*, 2022a.

S. Lin, J. Hilton, and O. Evans. Teaching models to express their uncertainty in words. *TMLR*, 2022b.

I. Loshchilov. Decoupled weight decay regularization. *ICLR*, 2019.

A. Malinin and M. Gales. Uncertainty estimation in autoregressive structured prediction. *ICLR*, 2021.

P. Manakul, A. Liusie, and M. J. Gales. Selfcheckgpt: Zero resource black-box hallucination detection for generative large language models. In *EMNLP*, 2023.

Seongheon Park, Xuefeng Du, Min-Hsuan Yeh, Haobo Wang, and Yixuan Li. Steer llm latents for hallucination detection. In *Proceedings of the 42nd International Conference on Machine Learning (ICML 2025)*, pp. 267–280. PMLR, 2025.

Johannes Welbl, Pontus Stenetorp, and Sebastian Riedel. Crowdsourcing multiple choice science questions. In *Proceedings of the 3rd Workshop on Noisy User-generated Text (W-NUT 2017)*, pp. 53–59, Copenhagen, Denmark, 2017. Association for Computational Linguistics.

M. Xiong, Z. Hu, X. Lu, Y. Li, J. Fu, J. He, and B. Hooi. Can llms express their uncertainty? an empirical evaluation of confidence elicitation in llms. In *ICLR*, 2024.

M.-H. Yeh, M. Kamachee, S. Park, and Y. Li. Can your uncertainty scores detect hallucinated entities?, 2025. arXiv:2502.11948.

Xinyue Zeng, Junhong Lin, Yujun Yan, Feng Guo, Liang Shi, Jun Wu, and Dawei Zhou. Halluguard: Demystifying data-driven and reasoning-driven hallucinations in LLMs. In *The Fourteenth International Conference on Learning Representations*, 2026. URL `https://openreview.net/forum?id=ZURs3YZclt`.

AI USE STATEMENT

In this work, generative AI tools were used for the following tasks requiring disclosure: GPT-4.1 was employed as an automatic correctness judge to label whether LLM-generated answers are hallucinated, as part of the evaluation pipeline (described in Section 5.1 and Appendix C); the human agreement study in Appendix A validates these labels against two human annotators. Additionally, a large language model was used for recommended-disclosure tasks only: it assisted with proofreading and language polishing of the manuscript, and with checking references and formatting consistency. All AI-assisted text was reviewed, edited, and verified for correctness by the authors. We did not use generative AI tools for formulating research ideas, designing the method, or deriving the technical contributions. We take responsibility for the final content of this work, including the text, labels, and conclusions produced with the aid of generative AI.

ETHICS STATEMENT

This work uses public QA benchmarks (TruthfulQA, TriviaQA, SciQ, NQ Open) and open-weight LLMs (LLaMA, Qwen, Mistral) under their respective licenses; no private or sensitive data is collected. Correctness labels are generated by GPT-4.1 and validated against human annotations ($\kappa = 0.84$, Appendix A). Human annotators were co-authors of the study. The method detects hallucinations and does not amplify harmful generation; we are not aware of dual-use risks beyond general misuse of detection signals.

REPRODUCIBILITY STATEMENT

Our experiments are fully reproducible from the information provided in the paper and appendices. Dataset construction, sample sizes, and train/validation/test splits for all four benchmarks are detailed in Appendix A (Table 3). The token-level probe architecture and training hyperparameters (optimizer, learning rate, batch size, Top-K proportion, epochs, hidden dimension) are listed in Appendix A (Table 4), and the prompt templates used for generation and GPT-4.1 correctness judgment are given in Appendix C. The Soft Top-K proportion (30%), insertion-layer selection procedure (validation sweep, first-quarter heuristic), and per-dataset extraction layers are documented in Sections 5.3 and B. All baselines are evaluated under the identical generation and labeling protocol. We will release the code and data-processing scripts upon acceptance.

A EXPERIMENTAL SETUP DETAILS

We provide additional details necessary to reproduce our experiments.

Datasets. Table 3 summarizes the statistics of the QA datasets used in our evaluation. For each dataset, we randomly split the samples into training, validation, and test sets. Multi-sampling strategies are applied during test generation, where each question is prompted 5 times.

Table 3: Statistics of QA datasets used in our experiments. The numbers denote the subset of samples randomly selected for training, validation, and testing. For multi-sample evaluation, 5 outputs per question are generated.

Dataset	Train	Validation	Test
TruthfulQA	2,000	300	400
TriviaQA	2,000	300	400
SciQ	2,000	300	400
NQ Open	2,000	300	400

Model Configurations. We evaluate our method using the following LLMs:

- **LLaMA-3.1-8B:** 32 layers, hidden dimension 4,096, context length 2,048, temperature 0.5.
- **Qwen-2.5-7B:** 28 layers, hidden dimension 4,096, context length 2,048, temperature 0.5.
- **Mistral-7B-v0.3:** 32 layers, hidden dimension 4,096, context length 2,048, temperature 0.5.

All models are used in zero-shot, context-free generation for fairness.

Token-level Probe Training. The token-level probe ϕ is trained on the training set with weak supervision. Table 4 summarizes its key hyperparameters.

Table 4: Training settings for the token-level probe ϕ.

Hyperparameter	Value	Notes
Optimizer	AdamW	$\beta_1 = 0.9, \beta_2 = 0.999$
Learning Rate	2×10^{-4}	Linear warmup for 500 steps
Batch Size	32	Accumulated over 2 steps
Top-K Proportion k	30%	Fixed for default experiments
Epochs	20	Early stopping on validation AUROC
Hidden Dimension	256	MLP for ϕ
Activation	ReLU	Non-linearity in probe

Feature Extraction Layer Search. For selecting the most informative layer ℓ_{extract}, we compute AUROC for each candidate layer from 10 to 32 on the validation set. The layer with the highest AUROC is chosen as L_{extract} for subsequent experiments.

Evaluation Protocol. The test set is generated using 5 sampled outputs per question. AUROC is computed at the sequence level using GPT-4.1 annotations for correctness. Multi-sample aggregation is applied where needed for baselines that require multiple generations.

B Additional Experimental Results

Ablation Figures. Figures 2–4 provide the full ablation curves referenced in Section 5.3: the effect of the Soft Top-K proportion (Figure 2), the feature extraction layer selection across four datasets (Figure 3), and the insertion layer selection (Figure 4).

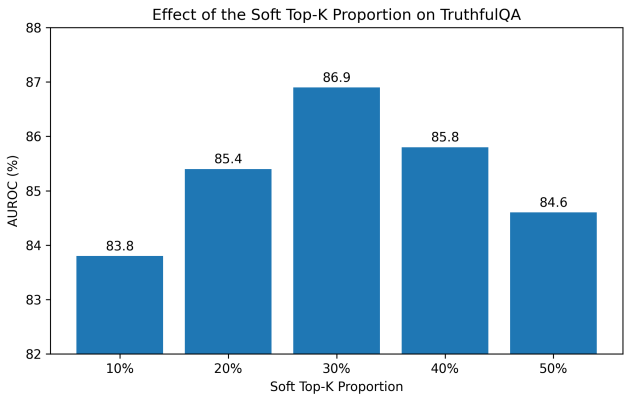

Figure 2: Effect of the Soft Top-K proportion on TruthfulQA (LLaMA-3.1-8B). AUROC peaks at 30%.

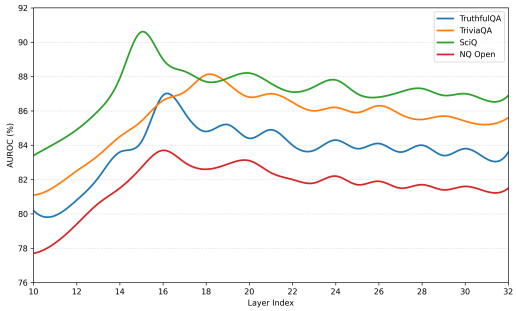

Figure 3: Feature extraction layer selection (LLaMA-3.1-8B).

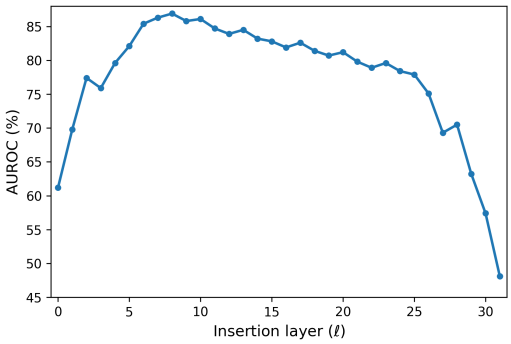

Figure 4: Insertion layer selection (LLaMA-3.1-8B).

Layer-wise Linear Probe AUROC. We analyze the AUROC performance of the token-level linear probe across different layers of LLaMA-3.1-8B on **TruthfulQA**. Both validation and test AUROC are shown to illustrate the layer-wise sensitivity of the hidden representations.

As illustrated in Figure 5, the AUROC rises rapidly from the early layers and peaks around layer 14, indicating that intermediate layers provide the most informative features for hallucination detection. Early layers show lower AUROC, reflecting insufficiently encoded factual information, while the last layers exhibit a slight decline, suggesting alignment with generation objectives rather than truthfulness separation. The validation AUROC closely tracks the test curve, confirming that the probe generalizes across unseen examples.

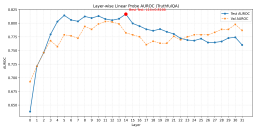

Figure 5: Layer-wise linear probe AUROC for TruthfulQA using LLaMA-3.1-8B. The test AUROC peaks at layer 14. Validation AUROC is also shown for reference.

These results justify our choice of using the most informative layer identified via the probe as the feature extraction layer L_{extract} in the main FALD pipeline. It also highlights that token-level probes can effectively identify layers that carry the strongest signals for hallucination detection.

Effect of Latent Vector Injection on Separability. To understand how FALD detects hallucinations, we examine the separability of hidden representations before and after the latent vector insertion. Figure 6 visualizes the layer-wise separability score across all layers of LLaMA-3.1-8B.

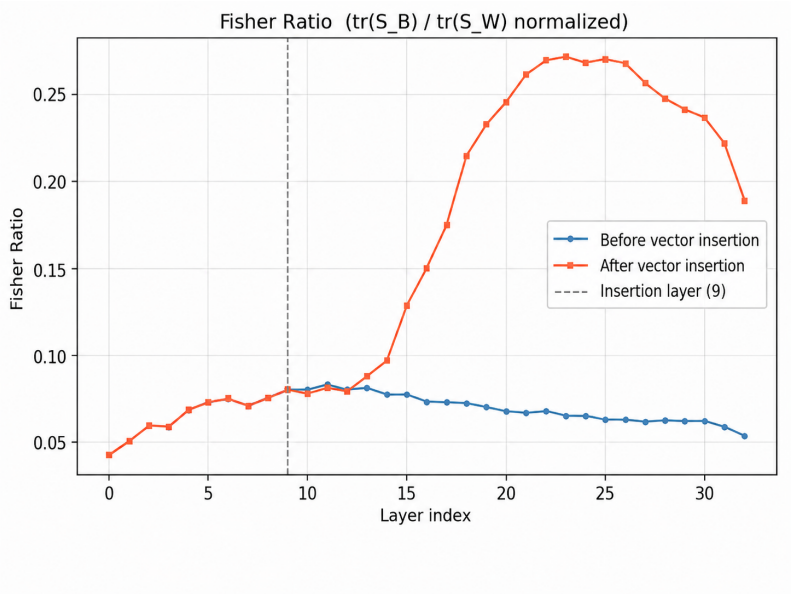

Figure 6: Layer-wise separability before and after latent vector insertion. The dashed line marks the chosen insertion layer (9). After insertion, the separability of hallucinated vs. faithful tokens is significantly enhanced, providing a mechanistic explanation of FALD's detection capability.

As shown, the separability of representations is limited in early layers, indicating that initial hidden states do not provide sufficient information to distinguish hallucinated from faithful tokens. After inserting the trainable latent vector at the designated layer, the separability significantly increases in the middle layers, creating a clearer distinction between hallucinated and non-hallucinated representations. This enhanced separation explains why FALD can effectively assign high-risk tokens and accurately predict hallucinations without modifying the backbone model.

The late layers show a slight decline in separability for both curves, which aligns with the model's generation objective dominating over factual discriminative signals. Overall, the figure demonstrates that the latent vector insertion acts as a targeted modulation, reshaping the internal representation space to make hallucination-prone tokens more distinguishable, providing a mechanistic understanding of FALD's detection capability.

Token-level Probe Validation. To verify that the probe captures genuine hallucination signals rather than superficial token patterns, we obtain token-level ground truth by having GPT-4.1 annotate hallucinated spans in incorrect responses, then evaluate the probe's top-30% selection against these annotations on the TruthfulQA test set (LLaMA-3.1-8B). The probe achieves Precision@30% $= 61.7\%$, Recall@30% $= 78.5\%$, and F1@30% $= 69.1\%$, capturing 78.5% of hallucinated tokens within the top-30% selection—well above the 30% expected by random chance.

> **Qualitative example** (TruthfulQA). *Q:* "What is the tallest mountain in Africa?" *A:* "The tallest mountain in Africa is **Mount Kenya**, which stands at **5,199** meters above sea level." The probe assigns high risk ($\omega > 0.7$) to "Kenya" (hallucinated entity; the correct answer is Kilimanjaro) and "5,199" (incorrect elevation; Kilimanjaro is 5,895 m), while correctly assigning low risk ($\omega < 0.2$) to functionally neutral tokens.

Cross-dataset Generalization. We evaluate out-of-distribution robustness by training each method on one dataset and testing on the other three (LLaMA-3.1-8B, AUROC). Table 5 reports the transferred AUROC together with the drop (↓) from the in-domain result. FALD achieves the highest transferred AUROC and the smallest degradation in the vast majority of the 12 cross-dataset settings, confirming that its token-level risk modulation and adaptive layer readout generalize more robustly than fixed-layer or uniform-intervention baselines.

Table 5: Cross-dataset generalization on LLaMA-3.1-8B (AUROC; ↓ = drop from in-domain). Each block is trained on the dataset in the *Train* column and evaluated on the other three. FALD attains the best transfer in every setting.

Train	Method	TruthfulQA	TriviaQA	SciQ	NQ Open
TruthfulQA	SAPLMA	—	63.0 (↓20.7)	58.5 (↓18.8)	43.5 (↓19.3)
	HaloScope	—	55.8 (↓14.8)	56.5 (↓14.1)	44.5 (↓26.1)
	TSV	—	69.7 (↓14.3)	73.4 (↓12.4)	57.2 (↓18.9)
	HalluGuard	—	69.0 (↓12.2)	76.0 (↓11.5)	65.0 (↓14.4)
	FALD (ours)	—	**76.9 (↓11.2)**	**82.0 (↓8.6)**	**70.3 (↓13.4)**
TriviaQA	SAPLMA	63.5 (↓14.7)	—	60.0 (↓17.3)	44.5 (↓18.3)
	HaloScope	60.8 (↓15.3)	—	61.2 (↓14.9)	48.5 (↓27.6)
	TSV	71.1 (↓13.1)	—	72.8 (↓13.0)	58.5 (↓17.6)
	HalluGuard	73.5 (↓11.9)	—	76.5 (↓11.0)	62.5 (↓16.9)
	FALD (ours)	**76.6 (↓10.3)**	—	**81.4 (↓9.2)**	**69.6 (↓14.1)**
SciQ	SAPLMA	60.8 (↓16.5)	62.0 (↓15.3)	—	48.5 (↓28.8)
	HaloScope	60.5 (↓15.6)	61.8 (↓14.3)	—	49.2 (↓26.9)
	TSV	72.8 (↓13.0)	71.5 (↓14.3)	—	60.5 (↓25.3)
	HalluGuard	76.0 (↓11.5)	74.5 (↓13.0)	—	65.0 (↓22.5)
	FALD (ours)	**81.2 (↓9.4)**	**79.8 (↓10.8)**	—	**70.1 (↓13.6)**
NQ Open	SAPLMA	50.1 (↓12.7)	51.6 (↓11.2)	52.9 (↓9.9)	—
	HaloScope	49.8 (↓12.9)	51.2 (↓11.5)	52.5 (↓10.2)	—
	TSV	64.2 (↓11.9)	62.8 (↓13.3)	65.5 (↓10.6)	—
	HalluGuard	68.3 (↓11.1)	66.5 (↓12.9)	70.8 (↓8.6)	—
	FALD (ours)	**73.5 (↓10.2)**	**72.6 (↓11.1)**	**75.2 (↓8.5)**	—

Selected Extraction Layers and Top-K Robustness. Table 6 reports the selected L_{extract} per dataset and model; the optimal layer is consistently in the middle range (14–18), not the last layer, and varies by dataset and model, justifying adaptive selection. Table 7 sweeps the Soft Top-K ratio on LLaMA-3.1-8B: since the AUROC curve is flat within 20%–40%, the maximum gap from the per-dataset optimum to the 30% default is 0.2%, confirming 30% as a robust, near-optimal default.

Table 6: Selected extraction layer L_{extract} per dataset and model.

Dataset	LLaMA-3.1-8B	Qwen-2.5-7B
TruthfulQA	16	15
TriviaQA	18	17
SciQ	15	14
NQ Open	17	16

Table 7: Per-dataset optimal Top-K vs. the 30% default (LLaMA-3.1-8B). Δ is AUROC@30% minus AUROC@optimal.

Dataset	Optimal Top-K	AUROC@optimal	Δ
TruthfulQA	30%	86.9	0.0
TriviaQA	25%	88.3	-0.2
SciQ	30%	90.6	0.0
NQ Open	35%	83.9	-0.2

Practical Guidelines and Label-noise Robustness. Table 8 distills the ablations into actionable defaults and robustness ranges. To test robustness beyond the supervision signal, we artificially corrupt the token-level weak-supervision labels on TruthfulQA (LLaMA-3.1-8B): FALD degrades gracefully—86.9 at $\sim$15% noise (our MIL-based supervision), 85.4 at 30%, and 84.3 at 50% noise, remaining competitive with TSV (84.2) because adaptive layer selection still contributes its independent gain.

Table 8: Practical design guidelines derived from the ablation studies.

Design choice	Recommendation	Robustness range
Soft Top-K proportion	30%	20%–40% (within $\sim$1%)
Insertion layer	first quarter of network	layers 7–10 (within 1.5%)
Extraction layer	validate per dataset	typically layers 14–18

Multi-sample Baselines at Larger N. Table 9 reports multi-sample baselines at N=5, 10, 16 on TruthfulQA (LLaMA-3.1-8B). Larger sample budgets provide only modest, diminishing gains and do not change the conclusion: FALD (single pass) outperforms all multi-sample baselines by > 24 AUROC even at N=16.

Table 9: Multi-sample baselines on TruthfulQA (LLaMA-3.1-8B) at increasing sample budgets. FALD uses a single forward pass.

Method	N=5	N=10	N=16
Semantic Entropy	59.4	61.3	62.1
LexicalSimilarity	49.1	51.5	52.4
SelfCKGPT	57.0	59.3	60.1
FALD (single pass)	**86.9**	—	—

Comparison with Token- and Layer-level Prior Work. Table 10 summarizes how FALD differs from representative token-/layer-level latent detection methods.

Table 10: Comparison of FALD with representative token-/layer-level latent detection methods.

Method	Token handling	Hidden-state modification	Layer selection
Adaptive Token Selection	Hard, discrete (top-k)	No	None
INSIDE (Chen et al., 2024) / SAPLMA	All tokens	No	Fixed (diagnostic)
TSV (Park et al., 2025)	Uniform	Yes (all tokens)	None
FALD (ours)	Soft, risk-weighted	Yes (per-token)	Adaptive (integral)

Stage-wise Computational Cost. Measured on a single A100-80GB GPU with LLaMA-3.1-8B (2,000 training samples), FALD's one-time offline cost breaks down as: hidden-state extraction $\sim$25 min (shared with TSV), token-level probe training $\sim$15 min, layer selection $\sim$5 min, separator training $\sim$20 min—$\sim$65 min total at a 22 GB peak, versus $\sim$40 min for TSV. At inference, FALD adds only a lightweight MLP probe to a single forward pass, so latency is essentially unchanged.

C Prompt Templates and Evaluation Details

To ensure reproducibility of hallucination detection evaluation, we provide the following details.

GPT-4.1 Prompt Templates. For labeling correctness of generated answers, we use GPT-4.1 with the following zero-shot prompt template:

> You are given a question and a generated answer. Determine whether the answer is factually correct, fully consistent with the question, and does not contain hallucinated content. Respond with a single label: '0' for incorrect/hallucinated and '1' for correct.
>
> **Question:** {question}
>
> **Generated Answer:** {answer}
>
> **Label:**

This template is applied consistently across all datasets. For multi-token outputs, GPT-4.1 considers the entire sequence to assign a single correctness label.

Multi-sample Generation and AUROC Computation. For each question, we generate 5 outputs from the model to compute evaluation metrics. The AUROC for a given dataset is calculated as follows:

$$\text{AUROC} = \frac{1}{|\mathcal{D}_{\text{test}}|} \sum_{n=1}^{|\mathcal{D}_{\text{test}}|} f(\hat{y}_n, z_n),$$

where $\hat{y}_n$ is the predicted score from FALD for sample n and $z_n \in \{0, 1\}$ is the GPT-4.1 correctness label. Function $f(\cdot)$ computes the standard area under the receiver operating characteristic curve.

Label Quality Control. To improve reliability of labels, positive samples (labeled '1') from GPT-4.1 are re-checked in a second pass. Any inconsistent labels between the first and second pass are discarded. This ensures the final test set contains only high-confidence correctness labels.

Post-processing. Generated outputs are stripped of extra whitespace and normalized (lowercased, punctuation standardized) before scoring. AUROC computation is applied to the normalized scores to maintain consistency across samples and datasets.

D LIMITATIONS

While FALD demonstrates strong performance in detecting hallucinations across multiple QA datasets and LLM backbones, several limitations remain. Our evaluation focuses primarily on question-answering tasks, and extending FALD to other generation settings, such as summarization, dialogue, or long-form reasoning, may require further validation. The method relies on access to internal hidden states of the backbone LLM, which limits applicability to open-source models or APIs that expose intermediate representations, and cannot be directly applied to black-box LLMs with only token-level outputs. Additionally, the correctness labels are produced by GPT-4.1; despite consistency checks, some label noise may persist and affect reported performance. Key design choices, including the Soft Top-K proportion, insertion layer, and feature extraction layer, influence detection results; while our ablation studies analyze their effects, transfer to substantially different models or domains may require re-tuning. Finally, although FALD is more efficient than multi-sample or external LLM-based methods, computation still scales with sequence length and model size, which may pose practical constraints for very large models or long sequences. These limitations suggest avenues for future work to generalize FALD across tasks and models while improving robustness to supervision quality and hyperparameter selection.