DLIPS: DETECTING HALLUCINATIONS IN LARGE LANGUAGE MODELS VIA TEMPORAL LIPSCHITZ DYNAMICS

Anonymous authors
Paper under double-blind review

ABSTRACT

Hallucinations undermine the reliability of large language models (LLMs). Existing hallucination detectors often rely on answer representations, token-level scores, or agreement across sampled responses, but such result-centric signals can obscure transient computational anomalies that emerge during generation and are no longer apparent in the final answer state. We therefore investigate whether hallucinations can be identified from abnormal dynamics in the model's local computational sensitivity, rather than solely from static or aggregated evidence after generation. We propose DLIPS (Dynamic LIPSchitz-based hallucination detection), a process-centric framework that uses temporal Lipschitz dynamics as a signal for response-level hallucination detection. DLIPS tracks local perturbation amplification across prefixes of a single candidate answer by estimating decoder-block Jacobian spectral norms, and summarizes their temporal and cross-layer variation for lightweight supervised classification. Experiments on HaluEval, RAGTruth, and MedHallu show that DLIPS outperforms the compared baselines, including an approximately 11% relative improvement in AUROC over the strongest baseline on HaluEval. Ablation studies show that removing temporal features causes the largest F1 decrease among the evaluated feature groups, indicating that temporal Lipschitz dynamics provide information beyond overall sensitivity magnitude. Additional evaluations on different models extend the framework across model scales and architectures. These results support temporal Lipschitz dynamics as a complementary signal to representation- and consistency-based approaches for hallucination detection.

DLIPS: `https://anonymous.4open.science/r/DLIPS-2EB4`

1 INTRODUCTION

Large language models (LLMs) continue to advance, yet they still produce hallucinations, which are fluent outputs that contain factual or logical errors (Ji et al., 2023; Huang et al., 2025; Wang et al., 2024; Srey et al., 2025; Cao et al., 2026). Such errors can arise at any stage of the LLM life cycle (Luo et al., 2025; Augenstein et al., 2024; Feng et al., 2024) and are especially harmful in high-stakes deployments. Reliable response-level hallucination detection is therefore essential for trustworthy LLMs (Azaria & Mitchell, 2023).

Among existing hallucination detection methods for LLMs, sampling-based consistency checks (Manakul et al., 2023; Farquhar et al., 2024) and multi-agent cross-examination (Feng et al., 2024) estimate reliability by comparing multiple responses, and they remain applicable in black-box settings. However, sampled responses or agents may confidently repeat the same incorrect claim, in which case their agreement reveals little about correctness. Internal probing offers an additional source of evidence, since hidden representations can encode factuality information that output confidence does not reliably reflect (Azaria & Mitchell, 2023). Detectors can therefore exploit hidden-state and attention patterns to distinguish faithful answers from hallucinated ones (Sriramanan et al., 2024; Park et al., 2025; Bazarova et al., 2026).

Nevertheless, a probe applied to the final state or a summary computed over the whole answer may fail to retain evidence from earlier generation steps. Because autoregressive decoding conditions

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

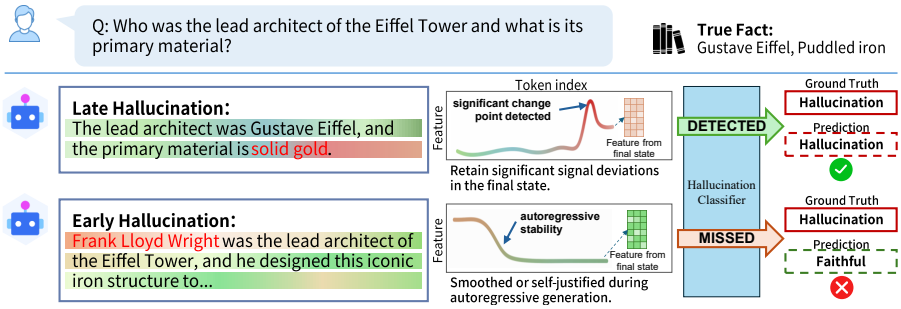


Figure 1: **Temporal signal dilution in final-state probing.** A late hallucination can remain detectable in the final state (top), whereas an early anomaly can be obscured by subsequent autoregressive generation (bottom). The schematic curves and detection outcomes illustrate how a final-state probe may miss evidence that is present at earlier prefixes.

each new token on the existing prefix, an early error becomes part of the context for all subsequent predictions (Azaria & Mitchell, 2023). In a confident hallucination, later tokens can thus be highly probable under this erroneous context, and their confidence reflects a coherent continuation rather than a verification of the initial claim. As later states increasingly reflect this continuation, an early anomaly may become less salient at the endpoint, and averaging over subsequent states can further reduce its contribution. This temporal signal dilution effect is illustrated in detail in Figure 1. To mitigate this temporal information loss, recent process-aware methods track layer-wise logits (He et al., 2025) or hidden-state transitions (Zhang et al., 2025; Zhu et al., 2024). However, such signals only describe how realized states or predictions evolve, and their variation is entangled with semantic progression and context accumulation. More importantly, a smooth trajectory does not imply locally stable computation, as the model may change little along the generated path yet remain highly sensitive to nearby perturbations. Transient instabilities can thus be missed even when intermediate logits or hidden states appear well behaved.

Motivated by the use of Lipschitz analysis in GAN training stability (Arjovsky et al., 2017; Gulrajani et al., 2017) and neural network robustness (Gouk et al., 2021), we extend local sensitivity analysis from static or globally aggregated statistics to its temporal evolution during generation. At a fixed answer prefix, the spectral norm of a decoder block's Jacobian measures the largest first-order amplification of a perturbation to the hidden state of the current token. Tracking this quantity across prefixes yields temporal Lipschitz dynamics, a record of how local sensitivity changes during generation. Since transient sensitivity excursions remain explicit in this trajectory even after later measurements subside, the dynamics preserve information that final-state or globally aggregated assessments may overlook.

Building on this perspective, we propose **DLIPS** (Dynamic LIPSchitz-based hallucination detection), which leverages temporal Lipschitz dynamics for response-level hallucination detection. DLIPS tracks local decoder-block sensitivity across the prefixes of a single observed answer and summarizes the resulting temporal and cross-layer patterns for a lightweight supervised classifier. Experiments on HaluEval (Li et al., 2023a), RAGTruth (Niu et al., 2024), and MedHallu (Pandit et al., 2025) show that DLIPS consistently outperforms the compared methods. Ablation studies identify temporal variation as the most informative feature group, and additional evaluations on larger backbones show that DLIPS applies across model scales and architectures.

Our main contributions are summarized as follows.

- **Predictive value of temporal sensitivity dynamics.** We show that the temporal evolution of local Lipschitz sensitivity provides predictive information beyond layer-wise and globally aggregated sensitivity statistics, supported by feature-group ablations.
- **Single-answer detection framework.** We formulate local sensitivity for complete decoder blocks and develop DLIPS, using matrix-free spectral estimation and trajectory aggregation to classify a single answer with a frozen backbone.
- **Cross-domain evaluation.** We demonstrate improvements on three benchmarks, consistent gains across general QA, RAG, and medical QA, and validate applicability on larger hybrid and full-attention backbones.

2 Related Work

Hallucination definitions and evaluation. Surveys distinguish hallucinations by factuality, consistency with supplied information, and task setting (Ji et al., 2023; Huang et al., 2025; Wang et al., 2024; Li et al., 2024). Analyses of prompting, model behavior, and error categories further show that hallucination evaluation depends on the context and the type of claim being assessed (van Deemter, 2024; Mishra et al., 2024; Anh-Hoang et al., 2025). Benchmarks also differ in annotation granularity; Mu-SHROOM, for example, studies multilingual hallucination spans (Vazquez et al., 2025). Our task is response-level classification of a supplied candidate answer, using the dataset label as the supervision target.

Output and internal-state signals. SelfCheckGPT uses agreement across sampled answers (Manakul et al., 2023), while Semantic Entropy estimates uncertainty over semantic clusters of responses (Farquhar et al., 2024). Internal-state approaches use hidden-state covariance in INSIDE/EigenScore (Chen et al., 2024), latent-space steering in TSV (Park et al., 2025), and topological divergence on attention graphs in TOHA (Bazarova et al., 2026). Other work studies factuality-related information in hidden representations and inference-time interventions (Orgad et al., 2025; Li et al., 2023b). Like these internal-state methods, DLIPS requires white-box access; its distinctive input is the local response of decoder blocks to perturbations.

Process-aware and combined approaches. ICR Probe tracks cross-layer hidden-state updates (Zhang et al., 2025), PoLLMGraph studies state-transition dynamics (Zhu et al., 2024), and DeLTa uses logit-trajectory prediction to modify decoding (He et al., 2025). Recent work has also related hallucination risk to perturbation sensitivity through NTK-based spectral statistics (Zeng et al., 2026), but summarizes such sensitivity using response-level or globally aggregated quantities rather than its prefix-wise evolution. Other strategies combine signals through small probing models, modular evaluation, or sequential testing (Tan et al., 2025; Muhammed et al., 2026; Wang et al., 2023). DLIPS contributes a different process measurement: a sequence of local Jacobian norms conditioned on observed answer prefixes. The comparison is therefore between different dynamic signals, rather than between dynamic analysis and an exclusively static prior literature.

3 Preliminaries

3.1 Problem Formulation

Given a prompt p and a candidate answer a, we predict whether the answer contains hallucinated content (Azaria & Mitchell, 2023). We learn a response-level classifier

$$f : \mathcal{P} \times \mathcal{A} \to \{0, 1\} \tag{1}$$

where $\mathcal{P}$ and $\mathcal{A}$ are the prompt and answer spaces. A prediction of $f(p, a) = 1$ denotes hallucinated content, and $f(p, a) = 0$ denotes the faithful class under the dataset's annotation scheme.

Our working hypothesis is that local sensitivity trajectories carry information about these labels. In application, DLIPS samples internal states at checkpoints during autoregressive generation and aggregates their sensitivity measurements for response-level classification. The dataset evaluation protocol is specified in Section 5.1.

3.2 Lipschitz Stability Analysis

A Lipschitz bound controls how much a function's output can change relative to its input. For a map $h : \mathcal{X} \to \mathcal{Y}$ between metric spaces, $L \geq 0$ is a valid bound if, for every $x_1, x_2 \in \mathcal{X}$,

$$d_{\mathcal{Y}}(h(x_1), h(x_2)) \leq L\, d_{\mathcal{X}}(x_1, x_2). \tag{2}$$

The smallest such bound is the global Lipschitz constant. Lipschitz control is used in Wasserstein GANs, with gradient penalties encouraging the critic to satisfy the desired constraint (Arjovsky et al., 2017; Gulrajani et al., 2017). It also supports analyses of robustness and generalization and motivates spectral normalization and Lipschitz-constrained architectures (Yin et al., 2019; Miyato et al., 2018; Qi et al., 2023; Gouk et al., 2021). These uses concern bounds over a domain or constraints imposed during training.

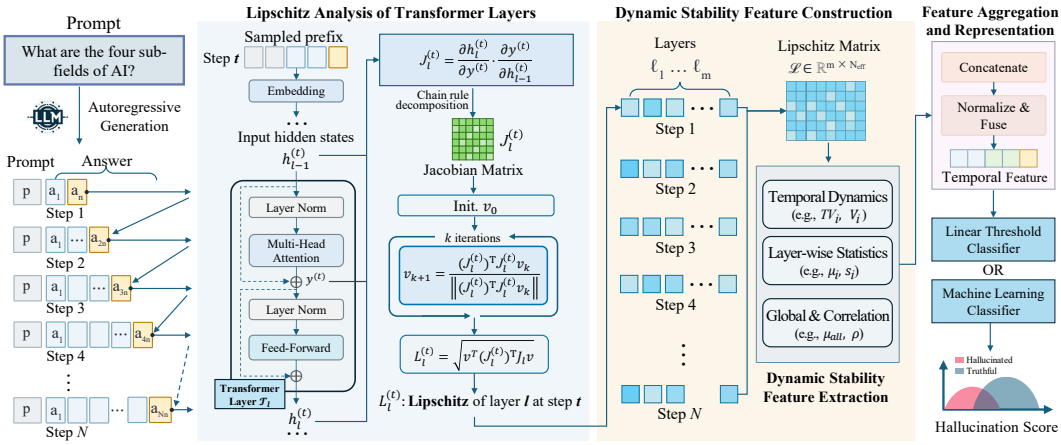


Figure 2: **DLIPS during generation.** (a) Growing answer prefixes are sampled at generation checkpoints. (b) Matrix-free JVP/VJP power iteration estimates the current-token Jacobian norm $\widehat{L}_l^{(q_n)}$ of each monitored block, with earlier prefix inputs fixed. (c) Estimates across layers and checkpoints form a sensitivity matrix, summarized into 21 dynamic features. (d) After generation, these features and three token counts are standardized and scored by a weighted-threshold detector or a learned classifier. Arrays are schematic.

For self-attention, Kim et al. (2021) show that standard dot-product attention is not globally Lipschitz on an unbounded input domain. Further work analyzes attention smoothness under specific input and architectural conditions (Castin et al., 2024). Such global results do not determine the sensitivity at a particular observed hidden state.

DLIPS instead measures the Jacobian spectral norm at each evaluated state. This norm gives maximal infinitesimal amplification at that point; a finite-neighborhood Lipschitz bound would require controlling the Jacobian throughout the neighborhood. We use *temporal Lipschitz dynamics* to refer to the resulting sequence of local sensitivity measurements and test its association with hallucination labels empirically.

4 Dynamic Lipschitz-based Hallucination Detection

DLIPS samples internal states as an LLM generates an answer (Figure 2). At selected checkpoints, it estimates local sensitivity for the monitored decoder blocks. The resulting layer-by-checkpoint matrix is summarized into a fixed-size feature vector for response-level scoring after the answer is complete. Table 3 lists the notation, and Algorithm 1 specifies the sensitivity extraction and detection procedure for an observed answer.

4.1 Dynamic State Sampling

The uniform checkpoint rule used for dataset evaluation is defined for a prompt p and a completed nonempty answer $a = (a_1, \ldots, a_T)$. We request $N \geq 1$ checkpoints and use $N_{\text{eff}} = \min(N, T)$ distinct prefixes. This length-dependent evaluation rule samples positions across the answer while accommodating short responses. For $N_{\text{eff}} > 1$, the answer-token positions are

$$q_n = \operatorname{rint}(\xi_n), \qquad (\xi_1, \ldots, \xi_{N_{\text{eff}}}) = \operatorname{linspace}(1, T, N_{\text{eff}}). \tag{3}$$

Here linspace includes both endpoints, and rint rounds to the nearest integer with ties to even. The checkpoints are distinct; when $N_{\text{eff}} = 1$, we set $q_1 = T$. Reproducing these length-dependent positions from live generation requires retaining the relevant states or replaying prefixes after completion.

Let D be the backbone depth and $\mathcal{S} = \{\ell_1, \ldots, \ell_m\} \subseteq \{0, \ldots, D-1\}$ the ordered set of monitored blocks. At checkpoint q_n, block l receives $H_l^{\text{in},(q_n)} \in \mathbb{R}^{r_n \times d}$, where r_n is the total prompt-plus-

prefix length and d is the hidden size. The complete block mapping F_l includes normalization, the token mixer, both residual additions, and the feed-forward network.

Using column vectors and one-based row indexing, let $z_{l,n} = H_l^{\mathrm{in},(q_n)}[r_n, :]^\top$. Let $\mathcal{R}_{l,n}(z)$ replace the final input row by $z^\top$, keeping earlier rows fixed. The corresponding output and Jacobian are

$$g_{l,n}(z) = [F_l(\mathcal{R}_{l,n}(z); c_{l,n})]^\top_{r_n,:}, \qquad J_l^{(q_n)} = D_z g_{l,n}(z_{l,n}) \in \mathbb{R}^{d\times d}. \tag{4}$$

The conditioning $c_{l,n}$ fixes the model weights, masks, and positional information. All computational paths from the current input row to the current output row remain differentiable. Earlier input rows supply context but are not perturbed in this local measurement.

4.2 Lipschitz Analysis of Transformer Layers

For the pre-normalized Transformer layer illustrated in Figure 2, suppressing layer and checkpoint indices gives

$$u = N_1(z), \qquad b = z + B(u; c^-), \qquad g(z) = b + P(N_2(b)), \tag{5}$$

where N_1, N_2 are normalization maps, B is the token mixer (multi-head attention in the illustration), P is the feed-forward network, and c^- represents fixed prefix information. With $v = N_2(b)$, the chain rule gives

$$J = \big(I + D_v P(v) D_b N_2(b)\big)\big(I + D_u B(u; c^-) D_z N_1(z)\big). \tag{6}$$

The attention derivative $D_u B$ includes the current query, key, and value paths. Equation (6) also retains both normalization stages, residual additions, and the feed-forward branch. Adaptations to the experimental backbones are specified separately in Appendix A.

We measure the infinitesimal local sensitivity

$$L_l^{(q_n)} = \|J_l^{(q_n)}\|_2 = \sup_{\|v\|_2=1} \|J_l^{(q_n)} v\|_2. \tag{7}$$

At a differentiable evaluation point, $\|g(z+\delta) - g(z)\|_2 \le L_l^{(q_n)} \|\delta\|_2 + o(\|\delta\|_2)$. Thus, $L_l^{(q_n)}$ quantifies first-order amplification of a current-token block-input perturbation, conditional on the prefix. Joint perturbations to earlier tokens or the full recurrent trajectory define different Jacobians.

To estimate $L_l^{(q_n)}$, let $J = J_l^{(q_n)}$ and initialize a unit vector v_0. Power iteration on $J^\top J$ uses

$$u_k = J v_k, \qquad w_k = J^\top u_k, \qquad \sigma_k = \|u_k\|_2, \tag{8}$$

$$\mu_k = \sigma_k^2, \qquad v_{k+1} = \frac{w_k}{\|w_k\|_2}. \tag{9}$$

Automatic differentiation computes Jacobian–vector products (JVPs) and vector–Jacobian products (VJPs) without storing J. Both use the same fixed mapping and evaluation point, with u_k treated as a fixed cotangent in the VJP. We stop when both the relative change in σ_k and the normalized eigen-residual satisfy their tolerances, or the iteration budget is exhausted. After K updates, we recompute

$$\widehat{L}_l^{(q_n)} = \|J v_K\|_2, \tag{10}$$

using the final direction. Zero or nonfinite normalization inputs are flagged; failed or unconverged estimates trigger the FP32 fallback described in Appendix D. A small eigen-residual does not by itself certify the largest singular value, so we also compare estimates with exact FP32 SVD.

4.3 Dynamic Stability Feature Construction

The estimates form $\boldsymbol{\mathcal{L}} \in \mathbb{R}^{m\times N_{\mathrm{eff}}}$, with $\mathcal{L}_{i,n} = \widehat{L}_{\ell_i}^{(q_n)}$. Distinguishing row index i from layer index ℓ_i permits any monitored subset. We extract a fixed-length representation $\mathbf{x}_{\mathrm{dyn}} = \phi(\boldsymbol{\mathcal{L}}) \in \mathbb{R}^{21}$ from three feature groups.

Temporal dynamics (10 features). For each layer, we compute monotonicity, first-difference variability, interval-averaged total variation, second-difference variability (termed smoothness), and dynamic range. The mean and standard deviation of each statistic across layers yield ten features. Differences are taken between sampled checkpoints without division by token spacing; larger second-difference variability indicates less smooth trajectories.

Layer-wise statistics (4 features). Each trajectory supplies a temporal mean and standard deviation. Their respective means and standard deviations across layers describe sensitivity magnitude and dispersion. This group is labeled Spatial in the ablation table.

Global and cross-layer statistics (7 features). These comprise the matrix mean, standard deviation, maximum, minimum, mean pairwise trajectory correlation, normalized entropy of layer means, and coefficient of variation of layer ranges.

Appendix B specifies the statistics and conventions for short or constant trajectories. The headline configuration appends $(T_p, T, T_p + T)$, where T_p and T count prompt and answer tokens, giving $\mathbf{x} \in \mathbb{R}^{24}$. These counts are length metadata, and the total count is redundant with the first two. The reported feature dimensions describe the implemented schema rather than a set of independent quantities.

4.4 Hallucination Detection

We standardize each feature using its training-set mean and standard deviation, assigning unit scale to a feature with zero training variance. For standardized input $\widetilde{\mathbf{x}}$, the preset weighted-threshold detector predicts

$$S(\widetilde{\mathbf{x}}) = \mathbf{w}^\top \widetilde{\mathbf{x}}, \qquad \widehat{y} = \mathbb{I}\{S(\widetilde{\mathbf{x}}) > \tau\}, \tag{11}$$

where $\widehat{y} = 1$ denotes hallucination. This variant uses preset weights and a preset threshold; its feature standardization still uses training data. We also fit logistic regression, support vector machines (SVMs), and random forests (RFs) to the same features, keeping the backbone frozen. For learned classifiers, hyperparameters and decision thresholds are selected on training and validation data, with F1 used for threshold selection. Section 5.1 and Appendix E describe the evaluation protocol.

A linear score combines trajectory features according to their label association. If $\boldsymbol{\mu}_y$ and Σ_y are the class-conditional mean and covariance of $\widetilde{\mathbf{x}}$, then $S = \mathbf{w}^\top \widetilde{\mathbf{x}}$ satisfies $\mathbb{E}[S \mid y] = \mathbf{w}^\top \boldsymbol{\mu}_y$ and $\mathrm{Var}(S \mid y) = \mathbf{w}^\top \Sigma_y \mathbf{w}$. These identities describe the projected mean separation and within-class variation. Learning $\mathbf{w}$ can adjust the relative contribution of informative, variable, or redundant features; whether this improves detection is evaluated in Section 6.

5 Experiments

5.1 Experimental Setup

Datasets and response protocol. We evaluate HaluEval (Li et al., 2023a), RAGTruth (Niu et al., 2024), and MedHallu (Pandit et al., 2025), covering general question answering, retrieval-augmented generation, and medicine. Each dataset is split into training, validation, and test subsets at a ratio of 60%/20%/20%, with related answers grouped within the same split to avoid leakage. Tables 1 and 2 report results on the held-out 20% test set. For evaluation on labeled candidate answers, we replay answer prefixes through the detector backbone using teacher forcing at the specified checkpoints, while keeping the candidate answers fixed. Retrieval context, when available, is included in the prompt. Sampling-based baselines additionally generate responses for consistency estimation. Further protocol and implementation details are provided in Appendix E.

Backbones and comparisons. Direct baseline comparisons use Qwen3.5-0.8B. Additional experiments evaluate DLIPS variants on Qwen3.5-9B and Llama-3.1-8B, keeping the candidate answers fixed. Qwen3.5 uses hybrid decoder blocks, whereas Llama-3.1 uses full-attention blocks.[1][2] Sensitivity estimation follows each backbone's complete block structure (Appendix A). Classifiers are trained separately for each backbone and dataset, so these experiments assess applicability across backbones rather than transfer of a fixed detector.

Configuration and metrics. Qwen3.5-0.8B has 24 decoder blocks: 18 Gated DeltaNet blocks and six full-attention blocks at zero-based indices $3, 7, 11, 15, 19, 23$. The headline configuration monitors blocks 8–15 with $N = 20$ requested checkpoints and 24 classifier features. We report accuracy,

[1] Qwen Team. *Qwen3.5*. `https://qwen.ai/blog?id=qwen3.5`.

[2] Meta AI. *Llama3.1*. `https://huggingface.co/collections/meta-llama/llama-31`.

Table 1: Response-level detection across three backbones. Baselines use Qwen3.5-0.8B; larger-backbone rows evaluate DLIPS variants. Acc denotes accuracy and AUC denotes AUROC. Bold and underlined values mark the largest and second-largest point estimates for each dataset and metric within a backbone, including ties.

Method	HaluEval			RAGTruth			MedHallu		
	Acc ↑	F1 ↑	AUC ↑	Acc ↑	F1 ↑	AUC ↑	Acc ↑	F1 ↑	AUC ↑
Qwen3.5-0.8B									
PPL	0.588	0.590	0.563	0.615	0.647	0.625	0.600	0.602	0.595
SelfCheckGPT	0.628	0.545	0.641	0.560	0.661	0.535	0.685	0.668	0.736
SemEnt	0.715	0.536	0.733	0.685	0.598	0.738	0.455	0.506	0.573
LLMCheck(HS)	0.570	0.555	0.596	0.671	0.631	0.708	0.520	0.378	0.519
LLMCheck(Attn)	0.775	0.765	0.847	0.669	0.639	0.765	0.631	0.620	0.716
TOHA	0.701	0.523	0.720	0.695	0.601	0.715	0.665	0.637	0.715
TSV	0.809	0.817	0.845	0.606	0.593	0.622	0.656	0.664	0.637
ICR Probe	0.578	0.581	0.624	0.745	0.714	0.762	0.688	0.694	0.741
PoLLMGraph	0.521	0.438	0.531	0.612	0.638	0.571	0.523	0.463	0.520
DLIPS-Threshold	0.746	0.760	0.808	0.675	0.667	0.573	0.515	0.670	0.441
DLIPS-RF	0.856	0.861	**0.937**	0.731	0.719	0.780	0.633	0.651	0.683
DLIPS-SVM	0.831	0.834	0.929	0.744	0.725	0.796	0.635	0.665	0.703
DLIPS-Logistic	**0.867**	**0.864**	0.928	**0.758**	**0.737**	**0.805**	**0.694**	**0.710**	**0.788**
Qwen3.5-9B									
DLIPS-Threshold	0.770	0.783	0.837	0.513	0.664	0.454	0.619	**0.697**	0.662
DLIPS-RF	0.900	0.899	0.953	**0.796**	0.787	0.838	0.655	0.657	**0.736**
DLIPS-SVM	0.896	0.896	0.954	0.771	0.762	0.850	**0.664**	0.670	0.709
DLIPS-Logistic	**0.913**	**0.912**	**0.965**	0.794	**0.789**	**0.873**	0.642	0.646	0.695
Llama-3.1-8B									
DLIPS-Threshold	0.761	0.776	0.772	0.498	0.639	0.513	0.483	0.634	0.457
DLIPS-RF	0.906	0.905	0.968	**0.819**	**0.813**	**0.869**	0.623	0.611	0.656
DLIPS-SVM	0.905	0.905	0.971	0.797	0.787	0.859	0.661	**0.690**	0.672
DLIPS-Logistic	**0.927**	**0.927**	**0.975**	0.791	0.787	0.859	**0.663**	0.676	**0.677**

F1 for the hallucination class, and AUROC as point estimates. Appendix E provides implementation and evaluation details. Appendix D reports numerical calibration on separate cohorts.

6 Results and Analysis

6.1 Main Results

Table 1 reports two comparisons: DLIPS versus baseline detectors on Qwen3.5-0.8B, and DLIPS classifier variants across three backbones. The larger-backbone rows assess DLIPS behavior without establishing a baseline ranking on those models.

Consistent gains over all baselines. DLIPS-Logistic achieves the highest F1 on all three datasets (0.864 on HaluEval, 0.737 on RAGTruth, and 0.710 on MedHallu), and its AUROC exceeds all non-DLIPS baselines. The best F1 and AUROC need not come from the same classifier: on HaluEval, DLIPS-RF attains the highest AUROC (10.6% relative gain). Section 6.2 analyzes which feature groups drive these gains.

Cross-backbone applicability. For DLIPS-Logistic, HaluEval AUROC rises from 0.928 (Qwen3.5-0.8B) to 0.965 (Qwen3.5-9B) and 0.975 (Llama-3.1-8B). RAGTruth shows a comparable trend, improving from 0.805 (Qwen3.5-0.8B) to 0.873 (Qwen3.5-9B), with the cross-family Llama-3.1-8B also achieving a high AUROC of 0.859. In contrast, MedHallu AUROC falls from 0.788 to 0.695 and 0.677. Local-sensitivity features thus remain effective across the evaluated architectures, but larger backbones do not uniformly improve detection. The best backbone and classifier depend on the domain and knowledge demands of the data.

Table 2: Feature and layer ablations using Qwen3.5-0.8B/DLIPS-Logistic. Feature deltas use Full Features (middle layers, 24 inputs); layer deltas use Full Layers (all 24 blocks). Spatial denotes the four layer-wise statistics; Global includes seven sensitivity summaries and three token counts. Bold marks the largest point estimate within each group.

Type	Configuration	HaluEval		RAGTruth		MedHallu	
		F1 ↑	AUC ↑	F1 ↑	AUC ↑	F1 ↑	AUC ↑
Feature	**Full Features**	**0.864**	**0.928**	**0.737**	0.805	**0.710**	**0.788**
	w/o Temporal	0.747 $_{\downarrow 0.117}$	0.871 $_{\downarrow 0.057}$	0.698 $_{\downarrow 0.039}$	0.750 $_{\downarrow 0.055}$	0.666 $_{\downarrow 0.044}$	0.758 $_{\downarrow 0.030}$
	w/o Spatial	0.846 $_{\downarrow 0.018}$	0.919 $_{\downarrow 0.009}$	0.730 $_{\downarrow 0.007}$	**0.808** $_{\uparrow 0.003}$	0.679 $_{\downarrow 0.031}$	0.737 $_{\downarrow 0.051}$
	w/o Global	0.833 $_{\downarrow 0.031}$	0.910 $_{\downarrow 0.018}$	0.735 $_{\downarrow 0.002}$	0.802 $_{\downarrow 0.003}$	0.707 $_{\downarrow 0.003}$	0.774 $_{\downarrow 0.014}$
Layer	**Full Layers**	0.856	0.925	0.729	0.800	0.704	0.783
	Bottom Layers	0.513 $_{\downarrow 0.343}$	0.522 $_{\downarrow 0.403}$	0.515 $_{\downarrow 0.214}$	0.540 $_{\downarrow 0.260}$	0.504 $_{\downarrow 0.200}$	0.545 $_{\downarrow 0.238}$
	Middle Layers	**0.864** $_{\uparrow 0.008}$	**0.928** $_{\uparrow 0.003}$	**0.737** $_{\uparrow 0.008}$	**0.805** $_{\uparrow 0.005}$	**0.710** $_{\uparrow 0.006}$	**0.788** $_{\uparrow 0.005}$
	Top Layers	0.805 $_{\downarrow 0.051}$	0.888 $_{\downarrow 0.037}$	0.678 $_{\downarrow 0.051}$	0.759 $_{\downarrow 0.041}$	0.654 $_{\downarrow 0.050}$	0.741 $_{\downarrow 0.042}$

Simple learned classifiers suffice. Logistic regression achieves the highest HaluEval F1 on all three backbones. RF and SVM lead in some other settings, so added classifier complexity is not uniformly beneficial. Learned classifiers generally outperform the preset weighted score, which supports data-dependent feature weighting. This advantage is particularly evident on Qwen3.5-0.8B/MedHallu, where logistic regression achieves 0.710 F1 and 0.788 AUROC, showing a more pronounced improvement across both metrics than the other detector variants.

6.2 Ablation Study

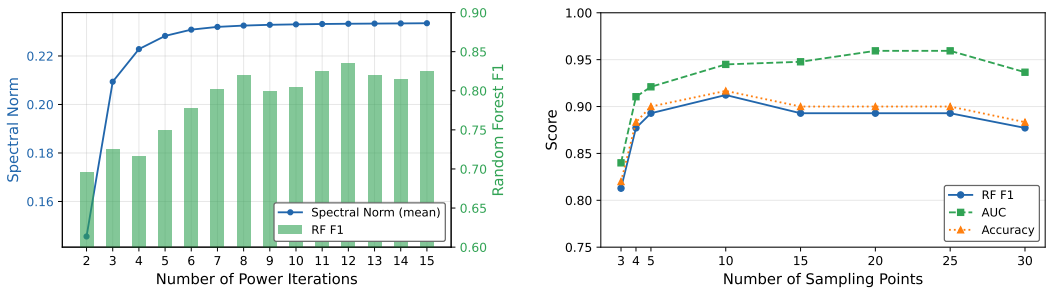
Feature groups. Removing temporal features causes the largest F1 drop on every dataset: 11.7, 3.9, and 4.4 points on HaluEval, RAGTruth, and MedHallu, respectively (Table 2). Both w/o Temporal and w/o Global retain 14 inputs, with larger F1 and AUROC drops for the former. Full Features and w/o Temporal both retain the three token counts, indicating additional predictive value from temporal features beyond these explicit length inputs. Removing Spatial slightly raises RAGTruth AUROC (0.805 to 0.808) but causes the largest MedHallu AUROC drop (0.051), so the contribution of the other groups depends on metric and domain. Removing Global drops seven sensitivity statistics and three token counts together, leaving 14 sensitivity features without explicit length inputs. It therefore measures their joint effect rather than isolating length, and the remaining features may still correlate with answer length.

Layer depth. The middle range (zero-based blocks 8–15) has the highest F1 and AUROC point estimates among the evaluated ranges, exceeding all 24 blocks by 0.8, 0.8, and 0.6 F1 percentage points across the three datasets. Bottom (0–7), middle (8–15), and top (16–23) each contain six Gated DeltaNet and two full-attention blocks, holding the block-type proportions fixed. The middle-range configuration evaluates eight blocks instead of 24 and is also used by Full Features.

6.3 Hyperparameter Analysis

Power iterations. Figure 3 reports validation sweeps with an auxiliary attention-module extractor on the main benchmark datasets. Its measurement differs from the complete residual-block map in Eq. (4). Figure 3a shows that the spectral estimate rises rapidly with the iteration count and largely stabilizes after about $K = 6$–7. RF F1 exhibits a similar trend, remaining relatively stable thereafter with only minor fluctuations. Numerical validation against exact FP32 SVD further gives a median relative error of 0.387% at $K = 10$. The complete-block extractor therefore uses adaptive stopping capped at 50 updates.

Sampling points. In Figure 3b, RF F1 and accuracy peak at $N = 10$, while F1 is unchanged from $N = 15$ to $N = 25$. AUROC is highest at $N = 20$ and $N = 25$, and all three metrics decrease at $N = 30$. The headline choice $N = 20$ corresponds to the higher-AUROC region of this sweep with fewer checkpoints than $N = 25$. The sweep therefore illustrates a trade-off between sampling density and the selected detection metric.



(a) (b)

Figure 3: Hyperparameter sweeps for RF detection: (a) spectral estimate and F1 versus power-iteration count; (b) F1, accuracy, and AUROC versus requested checkpoint count. The spectral-estimate curve is distinct from the exact-SVD error calibration in Appendix D.

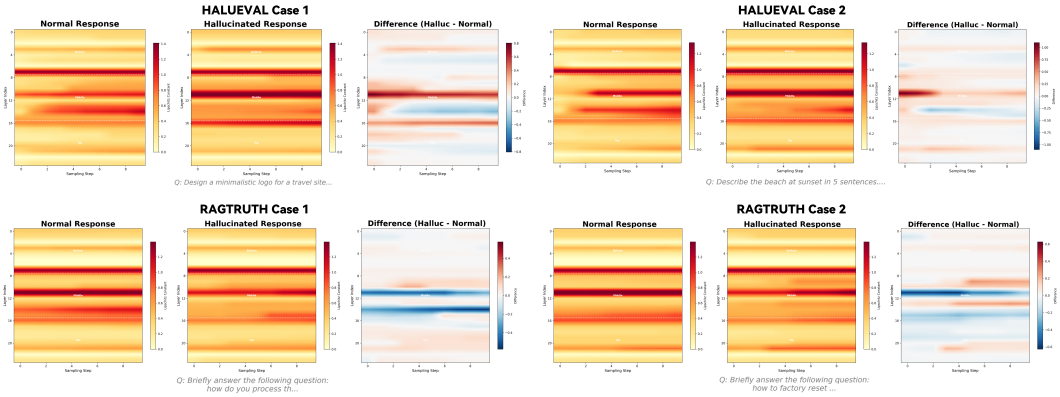


Figure 4: Displayed sensitivity trajectories for HaluEval and RAGTruth cases. Each triplet contains a faithful-answer heatmap, a hallucinated-answer heatmap, and their difference (hallucinated minus faithful). Rows index layers; columns index sampled positions within each answer, rather than matched token identities.

6.4 Case Visualization

Figure 4 visualizes local sensitivity trajectories for faithful and hallucinated answers, together with their differences across layers and sampled checkpoints. The displayed contrasts vary across both dimensions, illustrating the layer-wise and temporal variation summarized by DLIPS. Extended multi-scenario visualizations are provided in Appendix H.

7 Conclusion

DLIPS measures local decoder-block sensitivity across answer prefixes and summarizes its temporal and cross-layer structure for response-level hallucination detection. Combined with token-length metadata, these features yield strong reported results on HaluEval, RAGTruth, and MedHallu, with DLIPS-Logistic leading the listed F1 comparisons on Qwen3.5-0.8B. Temporal-feature removal causes the largest F1 decrease among the evaluated feature groups, and the middle-layer subset improves point estimates while evaluating fewer blocks. The central finding is that trajectories of local computational sensitivity provide useful features for detecting hallucinations in observed answers.

Limitations. DLIPS requires hidden-state access and differentiable block computations. Future work can investigate cheaper layer and checkpoint selection and whether sensitivity-based interventions can reduce hallucinations during generation.

Reproducibility Statement

The appendix specifies the conditioned block mapping, matrix-free estimator, feature definitions, numerical checks, and the reported evaluation protocol. Tables 1 and 2 report detection results. Appendix D describes numerical validation, and Appendix F clarifies the scope of the baseline comparison. The project code is available at `https://anonymous.4open.science/r/DLIPS-2EB4`

AI Use Statement

In this work, we used artificial intelligence (AI) tools to assist with language polishing and to improve the readability of the English manuscript. We did not use generative AI tools for other tasks requiring disclosure, including generating synthetic data, developing theoretical models or conceptual frameworks, formulating or communicating mathematical claims, proposing hypotheses, designing research methods or experiments, providing feedback on research methods or experiments, implementing methods, or interpreting results. In addition, we reviewed all AI-assisted text and verified that the technical content, claims, and reported numerical results were derived from our own experiments and analyses. We take full responsibility for the final content of this work, including any text, statements, or artifacts produced with the assistance of generative AI.

Ethics Statement

We confirm that our work has been conducted in accordance with the ICLR Code of Ethics. This study does not involve human subjects, the collection of sensitive personal data, or experiments that are expected to cause harm to individuals or groups. All datasets used in our experiments are publicly available and, to the best of our knowledge, do not contain personally identifiable information relevant to our study. We have considered potential issues related to fairness, bias, privacy, and misuse, and have conducted and reported our research in accordance with applicable ethical and research-integrity standards.

Our evaluation includes multiple domains, including medical hallucination detection on MedHallu. DLIPS is designed as a research method for detecting hallucinations rather than as a clinically validated diagnostic or decision-support system. In high-stakes medical applications, its predictions should therefore not be used as the sole basis for clinical decisions or as the only safety mechanism. Practical deployment would require domain-specific validation, appropriate calibration, and human oversight.

References

Dang Anh-Hoang, Vu Tran, and Le-Minh Nguyen. Survey and analysis of hallucinations in large language models: attribution to prompting strategies or model behavior. *Frontiers in Artificial Intelligence*, Volume 8 - 2025, 2025.

Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, pp. 214–223, 2017.

Isabelle Augenstein, Timothy Baldwin, Meeyoung Cha, Tanmoy Chakraborty, Giovanni Luca Ciampaglia, David Corney, Renee DiResta, Emilio Ferrara, Scott Hale, Alon Halevy, Eduard Hovy, Heng Ji, Filippo Menczer, Ruben Miguez, Preslav Nakov, Dietram Scheufele, Shivam Sharma, and Giovanni Zagni. Factuality challenges in the era of large language models and opportunities for fact-checking. *Nature Machine Intelligence*, 6(8):852–863, 2024.

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, December 2023.

Alexandra Bazarova, Andrei Volodichev, Aleksandr Yugay, Andrey Shulga, Alina Ermilova, Konstantin Polev, Julia Belikova, Rauf Parchiev, Dmitry Simakov, Maxim Savchenko, Andrey Savchenko, Serguei Barannikov, and Alexey Zaytsev. Hallucination detection in LLMs with topological divergence on attention graphs. pp. 15449–15470, July 2026.

Zouying Cao, Yifei Yang, Xiaojing Li, and Hai Zhao. Autohall: Automated factuality hallucination dataset generation for large language models. *IEEE Transactions on Audio, Speech and Language Processing*, 34:184–195, 2026.

Valérie Castin, Pierre Ablin, and Gabriel Peyré. How smooth is attention? In *Proceedings of the 41st International Conference on Machine Learning*, 2024.

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 *Proceedings of the Twelfth International Conference on Learning Representations*, 2024.

Sebastian Farquhar, Jannik Kossen, Lorenz Kuhn, and Yarin Gal. Detecting hallucinations in large language models using semantic entropy. *Nature*, 630(8017):625–630, 2024.

Shangbin Feng, Weijia Shi, Yike Wang, Wenxuan Ding, Vidhisha Balachandran, and Yulia Tsvetkov. Don't hallucinate, abstain: Identifying LLM knowledge gaps via multi-LLM collaboration. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 14664–14690, August 2024.

Henry Gouk, Eibe Frank, Bernhard Pfahringer, and Michael J. Cree. Regularisation of neural networks by enforcing lipschitz continuity. *Machine Learning*, 110(2):393–416, 2021.

Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron Courville. Improved training of wasserstein gans. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pp. 5769–5779, 2017.

Yunzhen He, Yusuke Takase, Yoichi Ishibashi, and Hidetoshi Shimodaira. DeLTa: A decoding strategy based on logit trajectory prediction improves factuality and reasoning ability. In *Proceedings of the 2nd Workshop on Uncertainty-Aware NLP (UncertaiNLP 2025)*, pp. 309–321, November 2025.

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, and Ting Liu. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ACM Transactions on Information Systems*, 43(2), January 2025. ISSN 1046-8188.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. Survey of hallucination in natural language generation. *ACM Comput. Surv.*, 55(12), March 2023.

Hyunjik Kim, George Papamakarios, and Andriy Mnih. The lipschitz constant of self-attention. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139, pp. 5562–5571, 18–24 Jul 2021.

Junyi Li, Xiaoxue Cheng, Xin Zhao, Jian-Yun Nie, and Ji-Rong Wen. HaluEval: A large-scale hallucination evaluation benchmark for large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 6449–6464, December 2023a.

Junyi Li, Jie Chen, Ruiyang Ren, Xiaoxue Cheng, Xin Zhao, Jian-Yun Nie, and Ji-Rong Wen. The dawn after the dark: An empirical study on factuality hallucination in large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 10879–10899, August 2024.

Kenneth Li, Oam Patel, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. Inference-time intervention: eliciting truthful answers from a language model. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, 2023b.

Yun Luo, Zhen Yang, Fandong Meng, Yafu Li, Jie Zhou, and Yue Zhang. An empirical study of catastrophic forgetting in large language models during continual fine-tuning. *IEEE Transactions on Audio, Speech and Language Processing*, 33:3776–3786, 2025.

Potsawee Manakul, Adian Liusie, and Mark Gales. SelfCheckGPT: Zero-resource black-box hallucination detection for generative large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 9004–9017, December 2023.

Abhika Mishra, Akari Asai, Vidhisha Balachandran, Yizhong Wang, Graham Neubig, Yulia Tsvetkov, and Hannaneh Hajishirzi. Fine-grained hallucination detection and editing for language models. In *Proceedings of the First Conference on Language Modeling*, 2024.

Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. In *Proceedings of the International Conference on Learning Representations*, 2018.

Diyana Muhammed, Giusy Giulia Tuccari, Gollam Rabby, Sören Auer, and Sahar Vahdati. Selfcheck-eval: A multi-module framework for zero-resource hallucination detection in large language models, 2026.

Cheng Niu, Yuanhao Wu, Juno Zhu, Siliang Xu, KaShun Shum, Randy Zhong, Juntong Song, and Tong Zhang. RAGTruth: A hallucination corpus for developing trustworthy retrieval-augmented language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 10862–10878, August 2024.

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 *Proceedings of the Thirteenth International Conference on Learning Representations*, 2025.

Shrey Pandit, Jiawei Xu, Junyuan Hong, Zhangyang Wang, Tianlong Chen, Kaidi Xu, and Ying Ding. MedHallu: A comprehensive benchmark for detecting medical hallucinations in large language models. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 2858–2873, November 2025.

Seongheon Park, Xuefeng Du, Min-Hsuan Yeh, Haobo Wang, and Yixuan Li. Steer LLM latents for hallucination detection. In *Proceedings of the Forty-second International Conference on Machine Learning*, 2025.

Xianbiao Qi, Jianan Wang, Yihao Chen, Yukai Shi, and Lei Zhang. Lipsformer: Introducing lipschitz continuity to vision transformers. In *Proceedings of the Eleventh International Conference on Learning Representations*, 2023.

Ponhvoan Srey, Xiaobao Wu, and Anh Tuan Luu. Unsupervised hallucination detection by inspecting reasoning processes. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 22117–22129, November 2025.

Gaurang Sriramanan, Siddhant Bharti, Vinu Sankar Sadasivan, Shoumik Saha, Priyatham Kattakinda, and Soheil Feizi. Llm-check: Investigating detection of hallucinations in large language models. In *Advances in Neural Information Processing Systems*, volume 37, pp. 34188–34216, 2024.

Hexiang Tan, Fei Sun, Sha Liu, Du Su, Qi Cao, Xin Chen, Jingang Wang, Xunliang Cai, Yuanzhuo Wang, Huawei Shen, and Xueqi Cheng. Too consistent to detect: A study of self-consistent errors in LLMs. pp. 4755–4765, November 2025.

Kees van Deemter. The pitfalls of defining hallucination. *Computational Linguistics*, 50(2):807–816, June 2024.

Raul Vazquez, Timothee Mickus, Elaine Zosa, Teemu Vahtola, Jörg Tiedemann, Aman Sinha, Vincent Segonne, Fernando Sanchez-Vega, Alessandro Raganato, Jindřich Libovický, Jussi Karlgren, Shaoxiong Ji, Jindřich Helcl, Liane Guillou, Ona De Gibert, Jaione Bengoetxea, Joseph Attieh, and Marianna Apidianaki. SemEval-2025 task 3: Mu-SHROOM, the multilingual shared-task on hallucinations and related observable overgeneration mistakes. In *Proceedings of the 19th International Workshop on Semantic Evaluation (SemEval-2025)*, pp. 2472–2497, July 2025.

Xiaohua Wang, Yuliang Yan, Longtao Huang, Xiaoqing Zheng, and Xuanjing Huang. Hallucination detection for generative large language models by Bayesian sequential estimation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 15361–15371, December 2023.

Yuxia Wang, Minghan Wang, Muhammad Arslan Manzoor, Fei Liu, Georgi Nenkov Georgiev, Rocktim Jyoti Das, and Preslav Nakov. Factuality of large language models: A survey. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 19519–19529, November 2024.

Songlin Yang, Jan Kautz, and Ali Hatamizadeh. Gated delta networks: Improving mamba2 with delta rule. In *Proceedings of the Thirteenth International Conference on Learning Representations*, 2025.

Dong Yin, Ramchandran Kannan, and Peter Bartlett. Rademacher complexity for adversarially robust generalization. In *Proceedings of the International Conference on Machine Learning*, pp. 7085–7094, 2019.

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 *Proceedings of the Fourteenth International Conference on Learning Representations*, 2026.

Zhenliang Zhang, Xinyu Hu, Huixuan Zhang, Junzhe Zhang, and Xiaojun Wan. ICR probe: Tracking hidden state dynamics for reliable hallucination detection in LLMs. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 17986–18002, July 2025.

Derui Zhu, Dingfan Chen, Qing Li, Zongxiong Chen, Lei Ma, Jens Grossklags, and Mario Fritz. PoLLMgraph: Unraveling hallucinations in large language models via state transition dynamics. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pp. 4737–4751, June 2024.

A Complete-Block Derivatives and Sensitivity Estimation

A.1 Notation and the conditioned mapping

Table 3 collects the notation. At each checkpoint, the measured variable is the current input row of one decoder block. The earlier rows, model parameters, masks, and positions are conditioning variables. Causality makes the incoming recurrent and convolution states independent of that current-row perturbation. The block's current state update remains differentiable. More formally, let $H = H_l^{\mathrm{in},(q_n)}$, $h = z_{l,n} = H[r_n, :]^\top \in \mathbb{R}^d$, and let $e_{r_n} \in \mathbb{R}^{r_n}$ be the last coordinate vector. Define the affine embedding and final-row projection by

$$\mathcal{R}_{l,n}(z) = H + e_{r_n}(z - h)^\top, \qquad \pi_{r_n}(X) = X[r_n, :]^\top.$$

Thus $\mathcal{R}_{l,n} : \mathbb{R}^d \to \mathbb{R}^{r_n \times d}$ has derivative $D\mathcal{R}_{l,n}(z)[\xi] = e_{r_n}\xi^\top$, and Eq. (4) means $g_{l,n} = \pi_{r_n} \circ F_l(\cdot; c_{l,n}) \circ \mathcal{R}_{l,n}$ under this column-vector convention. In particular, $Dg_{l,n}(h)[\xi] = \pi_{r_n}\big(D_H F_l(H; c_{l,n})[e_{r_n}\xi^\top]\big)$. The embedding perturbs only the current row; differentiability of the measured map depends on the block computation at the evaluated state, not on any additional assumption about row replacement.

A.2 Residual chain rule

Suppress the layer and checkpoint indices and write

$$u = N_1(z), \qquad b = z + B(u; c^-), \qquad v = N_2(b), \qquad g(z) = b + P(v). \tag{12}$$

For a direction $\xi \in \mathbb{R}^d$, successive differentials give

$$du = D_z N_1(z)\xi, \qquad db = \xi + D_u B(u; c^-)du, \tag{13}$$

$$dv = D_b N_2(b)db, \qquad dg = db + D_v P(v)dv. \tag{14}$$

Set $A = D_u B(u; c^-)D_z N_1(z)$ and $C = D_v P(v)D_b N_2(b)$. Substitution yields

$$J = D_z g(z) = (I + C)(I + A) = I + A + C + CA. \tag{15}$$

The order follows the forward composition: the mixer residual is differentiated first and the feed-forward residual second. No commutativity of A and C is assumed; reversing the factors would produce AC instead of the required CA. The factorization records these two residual stages and yields a norm bound, while matrix-free estimation differentiates the complete map g rather than evaluating the bound as the sensitivity feature.

Table 3: Notation for sensitivity extraction and detection

Notation	Description
Indices and Dimensions	
T, T_p	Numbers of analyzed answer tokens and prompt tokens.
N, N_{eff}	Requested and effective numbers of sampled checkpoints, where $N_{\text{eff}} = \min(N, T)$.
D	Total number of decoder blocks in the backbone.
$\mathcal{S}, m$	Ordered monitored layer set and its size; ℓ_i is the layer at matrix row i.
q_n	The n-th sampled answer-prefix length, $q_n \in [1, T]$.
l	Decoder-layer index, $l \in \mathcal{S}$.
d	Hidden state dimension.
States and Matrices	
$H_l^{\text{in},(q_n)}$	Full-prefix hidden-state input to block l at checkpoint q_n.
$F_l(\cdot; c_{l,n})$	Complete decoder-block map under fixed mask, positions, parameters, and other conditioning $c_{l,n}$.
$J_l^{(q_n)}$	State-conditioned, current-token Jacobian of the complete block at checkpoint q_n.
$\boldsymbol{\mathcal{L}}$	Dynamic sensitivity matrix, $\boldsymbol{\mathcal{L}} \in \mathbb{R}^{m \times N_{\text{eff}}}$.
$L_l^{(q_n)}, \widehat{L}_l^{(q_n)}$	Exact local Jacobian norm and its matrix-free estimate.
Features and Classifiers	
$\Delta_{i,n}$	Difference $\mathcal{L}_{i,n+1} - \mathcal{L}_{i,n}$ between adjacent checkpoints.
TV_i, V_i	Total variation and first-difference variability for monitored layer ℓ_i.
μ_i, s_i	Checkpoint mean and standard deviation for monitored layer ℓ_i.
ρ, H	Mean pairwise correlation and normalized entropy of layer means.
$\mathbf{x}_{\text{dyn}}$	Extracted 21-dimensional dynamic stability feature vector.
$\hat{y}$	Predicted hallucination label, $\hat{y} \in \{0, 1\}$.
$f_\theta(\cdot)$	Supervised hallucination classifier parameterized by θ.

A.3 Normalization and feed-forward derivatives

For RMS normalization, let $r = (d^{-1}x^\top x + \epsilon_{\text{norm}})^{1/2}$, with $\epsilon_{\text{norm}} > 0$, and let Γ be the diagonal matrix of effective learned scales. Then

$$N_{\text{RMS}}(x) = \Gamma x/r, \qquad DN_{\text{RMS}}(x) = \Gamma\left(\frac{I}{r} - \frac{xx^\top}{dr^3}\right). \tag{16}$$

This follows from $\mathrm{d}r = \frac{x^\top \mathrm{d}x}{d \cdot r}$. An additive bias has zero derivative. A parameterization with scale $1 + w$ uses $\Gamma = \operatorname{diag}(1 + w)$.

For centered LayerNorm, define $\Pi = I - \mathbf{1}\mathbf{1}^\top/d$, $c = \Pi x$, and $r_c = (d^{-1}c^\top c + \epsilon_{\text{norm}})^{1/2}$. Since $c^\top \Pi = c^\top$,

$$DN_{\text{LN}}(x) = \Gamma\left(\frac{\Pi}{r_c} - \frac{cc^\top}{dr_c^3}\right). \tag{17}$$

Thus, normalization is an input-dependent part of the Jacobian even with fixed learned parameters.

For a gated feed-forward network, let $a = W_g v + b_g$, $b' = W_u v + b_u$, and $P(v) = W_d[\text{SiLU}(a) \odot b'] + b_d$. Its derivative is

$$D_v P(v) = W_d\left[\operatorname{diag}(b' \odot \text{SiLU}'(a))W_g + \operatorname{diag}(\text{SiLU}(a))W_u\right], \tag{18}$$

where $\mathrm{SiLU}'(t) = s(t) + t\,s(t)(1 - s(t))$ and s is the sigmoid. For an output vector $\mathbf{o}$ and gate vector $\mathbf{g}$, gated RMS output normalization has the form $\mathcal{G}(\mathbf{o}, \mathbf{g}) = N_{\mathrm{RMS}}(\mathbf{o}) \odot \mathrm{SiLU}(\mathbf{g})$, giving

$$d\mathcal{G} = \mathrm{diag}(\mathrm{SiLU}(\mathbf{g}))DN_{\mathrm{RMS}}(\mathbf{o})\,d\mathbf{o} + \mathrm{diag}(N_{\mathrm{RMS}}(\mathbf{o}) \odot \mathrm{SiLU}'(\mathbf{g}))\,d\mathbf{g}. \tag{19}$$

The map $\mathcal{G}$ is the gated RMS output normalization used in the Qwen3.5 Gated DeltaNet readout following the state update in Eq. (26). It is distinct from the elementwise output gate of the full-attention mixer below, whose displayed derivative does not include RMS output normalization. An architecture without either gate omits its corresponding gate path. These formulas expose the normalization and gate paths retained in automatic differentiation.

A.4 Attention and recurrent mixer derivatives

Equation (6) applies to a complete pre-normalized decoder block when its token mixer B is instantiated correctly. The formulas below expand its derivative for full attention and Gated DeltaNet. They hold earlier prefix inputs fixed and differentiate all current-token pathways. Model-specific projections, normalization, position transformations, and gates are included in the corresponding maps.

Full attention. For one attention head, let $\mathbf{q}, \mathbf{k}, \mathbf{v}$ be the current-token query, key, and value after their configured transformations. Stack the keys and values as rows of K and V, including the current token. Earlier rows are constant in the local perturbation. Restricting to unmasked causal positions and holding their additive mask $\mathbf{m}$ fixed, the head output is

$$\mathbf{r} = K\mathbf{q}/\sqrt{d_k} + \mathbf{m}, \qquad \boldsymbol{\alpha} = \mathrm{softmax}(\mathbf{r}), \qquad \mathbf{o} = V^\top \boldsymbol{\alpha}. \tag{20}$$

Directional differentiation gives

$$d\mathbf{r} = ((dK)\mathbf{q} + K\,d\mathbf{q})/\sqrt{d_k}, \tag{21}$$

$$d\boldsymbol{\alpha} = (\mathrm{diag}(\boldsymbol{\alpha}) - \boldsymbol{\alpha}\boldsymbol{\alpha}^\top)d\mathbf{r}, \tag{22}$$

$$d\mathbf{o} = (dV)^\top \boldsymbol{\alpha} + V^\top d\boldsymbol{\alpha}. \tag{23}$$

Only the current rows of dK and dV can be nonzero. Key/value sharing across query heads is retained when the head derivatives are assembled. After concatenating heads, gated attention outputs $B = W_O(\mathbf{g} \odot \mathbf{o})$, so

$$dB = W_O\big(\mathbf{g} \odot d\mathbf{o} + \mathbf{o} \odot d\mathbf{g}\big). \tag{24}$$

Qwen3.5 uses a learned output gate; for an ungated attention block, $\mathbf{g} = \mathbf{1}$ and $d\mathbf{g} = 0$. Query/key normalization and rotary-position transformations remain inside the differentiable projection maps.

Gated DeltaNet. For a recurrent mixer, write its state update and output as $s^+ = U(s^-, u)$ and $B(u; s^-) = O(s^+, u)$. The current-input derivative is

$$D_u B = D_2 O(s^+, u) + D_1 O(s^+, u) D_2 U(s^-, u), \tag{25}$$

where D_1 and D_2 differentiate the first and second arguments. Holding the incoming recurrent and convolutional states fixed therefore retains the current state-update pathway. The implementation uses deterministic block calls with mutable caches disabled and recomputes the prefix as needed.

For one value head, use the value-by-key state convention $R^- \in \mathbb{R}^{d_v \times d_k}$. The gated delta update can be written as (Yang et al., 2025)

$$\overline{R} = \alpha R^-, \qquad \mathbf{e} = \mathbf{v} - \overline{R}\mathbf{k}, \qquad R^+ = \overline{R} + \beta \mathbf{e}\mathbf{k}^\top, \qquad \mathbf{o} = R^+\mathbf{q}. \tag{26}$$

The vectors $\mathbf{q}, \mathbf{k}, \mathbf{v}$ include the causal convolution and configured normalization; query scaling is absorbed into $\mathbf{q}$. Decay α, update gate β, and the output gate depend on the current input. Qwen3.5 further applies gated RMS normalization and an output projection. Denote this final map by $B = W_O\mathcal{G}(\mathbf{o}, \mathbf{g})$.

With $dR^- = 0$, the complete current-input derivative is

$$d\overline{R} = (d\alpha)R^-, \qquad d\mathbf{e} = d\mathbf{v} - (d\overline{R})\mathbf{k} - \overline{R}\,d\mathbf{k}, \tag{27}$$

$$dR^+ = d\overline{R} + (d\beta)\mathbf{e}\mathbf{k}^\top + \beta(d\mathbf{e})\mathbf{k}^\top + \beta\mathbf{e}(d\mathbf{k})^\top, \tag{28}$$

$$d\mathbf{o} = (dR^+)\mathbf{q} + R^+ d\mathbf{q}, \qquad dB = W_O\big(D_1\mathcal{G}\,d\mathbf{o} + D_2\mathcal{G}\,d\mathbf{g}\big). \tag{29}$$

Here α is a multiplicative decay factor; if an implementation stores log-decay a, then $\alpha = \exp(a)$ and $d\alpha = \alpha\, da$. The past convolution buffer is also fixed, but convolution terms involving the current input remain differentiable. Equation (29) therefore retains the current state update, readout, gates, and normalization. It is the explicit counterpart of Eq. (25).

Perturbing the full prefix would instead add terms propagated through dR^- and earlier convolution states. That defines a different, rectangular Jacobian from all prefix inputs to the current output. The $d \times d$ estimand in Eq. (4) deliberately conditions on those earlier inputs. Its sequence of evaluations is a trajectory of local measurements, not the Jacobian of the entire generation process.

A.5 Local amplification and residual bounds

At a differentiable point z, $g(z+\delta) - g(z) = J\delta + o(\|\delta\|_2)$. Taking the maximum over unit directions gives the infinitesimal amplification $\|J\|_2$. For a continuously differentiable map on a convex neighborhood $\mathcal{B}$, integration along a line segment gives

$$\|g(z') - g(z)\|_2 \le \left(\sup_{u \in \mathcal{B}} \|Dg(u)\|_2\right) \|z' - z\|_2, \qquad z, z' \in \mathcal{B}, \tag{30}$$

provided the supremum is finite. The pointwise measurement in DLIPS supplies $\|Dg(z)\|_2$, whereas this finite-neighborhood bound requires control along the whole segment.

For every unit vector x, the reverse triangle inequality implies $\|(I+A)x\|_2 \ge 1 - \|A\|_2$. Consequently,

$$\sigma_{\min}(I + A) \ge \max\{0, 1 - \|A\|_2\}, \tag{31}$$

$$\|J\|_2 \le (1 + \|C\|_2)(1 + \|A\|_2). \tag{32}$$

The second inequality follows from $J = (I+C)(I+A)$, submultiplicativity, and $\|I + M\|_2 \le 1 + \|M\|_2$. Equivalently, applying the triangle inequality to $I + A + C + CA$ gives $1 + \|A\|_2 + \|C\|_2 + \|C\|_2\|A\|_2$. The scalar factors commute in this bound, but the matrix factors in J generally do not. Residual connections alone impose no unit lower bound on $\|J\|_2$. Identity normalizations, $B(u) = -u/2$, and $P = 0$ give $J = I/2$. Residual inclusion is therefore checked through the differentiated graph and exact-Jacobian calibration.

A.6 Power iteration: convergence and stopping diagnostics

For the fixed Jacobian J, define $G = J^\top J \succeq 0$. A JVP followed by a VJP with a fixed cotangent computes Gv_k. The recurrence is

$$v_{k+1} = \frac{Gv_k}{\|Gv_k\|_2}, \qquad \widehat{L}_K = \|Jv_K\|_2, \qquad \|v_0\|_2 = 1. \tag{33}$$

The base point and conditioning are held fixed throughout this recurrence.

Proposition 1 (Exact-arithmetic convergence) *Let G have eigenvalues $\eta_1 > \eta_2 \ge \cdots \ge \eta_d \ge 0$, with $\eta_1 > 0$, and orthonormal eigenvectors e_j. Write $v_0 = \sum_j c_j e_j$ with $c_1 \ne 0$. For $K \ge 1$,*

$$v_K = \frac{\sum_j \eta_j^K c_j e_j}{(\sum_j \eta_j^{2K} c_j^2)^{1/2}}, \tag{34}$$

$$0 \le \eta_1 - \widehat{L}_K^2 \le \eta_1 \left(\frac{\eta_2}{\eta_1}\right)^{2K} \frac{1 - c_1^2}{c_1^2}. \tag{35}$$

In particular, $\widehat{L}_K \to \|J\|_2$.

Proof. Repeated substitution gives $v_K = G^K v_0 / \|G^K v_0\|_2$. The Rayleigh quotient is $\widehat{L}_K^2 = v_K^\top G v_K$, so

$$\eta_1 - \widehat{L}_K^2 = \frac{\sum_{j>1} (\eta_1 - \eta_j) \eta_j^{2K} c_j^2}{\sum_j \eta_j^{2K} c_j^2}. \tag{36}$$

Bound the numerator by $\eta_1\eta_2^{2K}(1-c_1^2)$ and the denominator below by $\eta_1^{2K}c_1^2$. This proves Eq. (35). For a repeated top eigenvalue, the analogous argument uses the squared projection onto its eigenspace and the largest eigenvalue outside that space. □

The rate depends on the spectral gap and initialization. A fixed iteration count or an F1 plateau does not determine that gap. If $Gv_k = 0$, normalization is undefined; the implementation flags the coordinate rather than treating it as a certified zero norm. A nonzero matrix can annihilate a particular initial vector.

For a unit vector v and $\mu = v^\top Gv$, expansion in the eigenbasis gives

$$\operatorname{dist}(\mu, \operatorname{spec}(G)) \le \|Gv - \mu v\|_2. \tag{37}$$

Indeed, the squared residual is a weighted average of $(\eta_j - \mu)^2$. This certifies proximity to some eigenvalue. For example, $J = \operatorname{diag}(2,1)$ and $v = (0,1)^\top$ give zero residual and estimate 1, while $\|J\|_2 = 2$. The estimator therefore combines stopping diagnostics with the exact-SVD calibration in Appendix D.

A.7 Complete extraction and scoring procedure

Algorithm 1: DLIPS response scoring under the completed-answer evaluation protocol

Input : Prompt p, nonempty observed answer a, frozen backbone, ordered monitored layers $\mathcal{S} = (\ell_1, \dots, \ell_m)$, requested checkpoints $N \ge 1$, iteration cap $K_{\max}$, tolerances $\epsilon_\sigma, \epsilon_r$, numerical safeguard $\varepsilon > 0$, fitted preprocessing and detector

Output: Label $\widehat{y}$, or an explicit extraction-failure status

Set $N_{\text{eff}} \leftarrow \min(N, T)$ and compute q_n using Eq. (3);
Allocate $\boldsymbol{\mathcal{L}} \in \mathbb{R}^{m \times N_{\text{eff}}}$ with entries initially unassigned;
for $n = 1, \dots, N_{\text{eff}}$ **do**
    Evaluate $p \oplus a_{1:q_n}$ in evaluation mode with mutable caches disabled;
    for $i = 1, \dots, m$ **do**
        Capture the complete block ℓ_i and construct $g_{\ell_i,n}$ from Eq. (4);
        Set $J = Dg_{\ell_i,n}(z_{\ell_i,n})$ implicitly; initialize a seeded random unit vector v_0; set $K \leftarrow 0$ and *converged*←false;
        for $k = 0, \dots, K_{\max} - 1$ **do**
            $u_k \leftarrow Jv_k$ by JVP; $w_k \leftarrow J^\top u_k$ by VJP with fixed cotangent u_k;
            if *a product is nonfinite or* $\|w_k\|_2 = 0$ **then**
                Flag this coordinate for fallback and leave the iteration loop;
            $\sigma_k \leftarrow \|u_k\|_2$; $\mu_k \leftarrow \sigma_k^2$; $v_{k+1} \leftarrow w_k/\|w_k\|_2$; $K \leftarrow k+1$;
            $r_k \leftarrow \|w_k - \mu_k v_k\|_2/(\|w_k\|_2 + \varepsilon)$;
            if $k > 0$ *and* $|\sigma_k - \sigma_{k-1}|/(|\sigma_{k-1}| + \varepsilon) \le \epsilon_\sigma$ *and* $r_k \le \epsilon_r$ **then**
                Set *converged*←true and leave the iteration loop;
        if *no arithmetic failure occurred* **then**
            Recompute $\widehat{L}_{\ell_i}^{(q_n)} \leftarrow \|Jv_K\|_2$ and store a finite result in $\mathcal{L}_{i,n}$;
        Retain convergence diagnostics; flag failed, unassigned, or unconverged coordinates;
for *each monitored layer with a failed or unconverged coordinate* **do**
    Recompute all its checkpoints in FP32 with the documented tighter tolerances;
    if *any coordinate remains invalid or unconverged* **then**
        return explicit extraction-failure status;
Compute five temporal statistics M_i, V_i, TV_i, S_i, R_i and their cross-layer means and standard deviations (10 features);
Compute means and standard deviations of μ_i, s_i across layers (4 features);
Compute $\mu_{\text{all}}, s_{\text{all}}, \max_{i,n} \mathcal{L}_{i,n}, \min_{i,n} \mathcal{L}_{i,n}, \rho, H, C_R$ (7 features);
Assemble the 21-feature vector using the saved schema in Table 4; append $(T_p, T, T_p + T)$ for the headline 24-feature setting;
Apply the fitted training-only preprocessing and classifier or weighted threshold;
return $\widehat{y}$;

B Feature Definitions and Implementation Conventions

B.1 Matrix notation and aggregation

Let $B = N_{\text{eff}} \geq 1$ and $m \geq 1$ and write $\lambda_{i,n} = \mathcal{L}_{i,n}$ for the estimated sensitivity of monitored layer ℓ_i at checkpoint q_n. A valid matrix is finite and nonnegative. All means and standard deviations below use population denominators. Define

$$\operatorname{mean}_i c_i = \frac{1}{m}\sum_{i=1}^{m} c_i, \qquad \operatorname{std}_i c_i = \sqrt{\frac{1}{m}\sum_{i=1}^{m}(c_i - \operatorname{mean}_j c_j)^2}. \tag{38}$$

The same conventions apply over checkpoints and matrix entries.

Temporal dynamics: ten features. For $B > 1$, let $\Delta_{i,n} = \lambda_{i,n+1} - \lambda_{i,n}$ and $\overline{\Delta}_i = (B-1)^{-1}\sum_{n=1}^{B-1}\Delta_{i,n}$. The five trajectory statistics are

$$M_i = \frac{\max\left(\sum_n \mathbb{I}\{\Delta_{i,n} > 0\}, \sum_n \mathbb{I}\{\Delta_{i,n} < 0\}\right)}{B-1}, \tag{39}$$

$$V_i = \sqrt{\frac{1}{B-1}\sum_{n=1}^{B-1}(\Delta_{i,n} - \overline{\Delta}_i)^2}, \qquad TV_i = \frac{1}{B-1}\sum_{n=1}^{B-1}|\Delta_{i,n}|, \tag{40}$$

$$S_i = \operatorname{std}_{n=1,\ldots,B-2}(\Delta_{i,n+1} - \Delta_{i,n}), \qquad R_i = \max_n \lambda_{i,n} - \min_n \lambda_{i,n}. \tag{41}$$

We set M_i, V_i, TV_i, S_i to zero when $B = 1$, and $S_i = 0$ when $B = 2$. The total-variation statistic averages absolute changes over the $B-1$ observed intervals, matching the implementation for variable-length trajectories. Larger S_i denotes greater variation in second differences. When $B = 3$, the single second difference also has population standard deviation zero. A constant nonzero second difference gives $S_i = 0$ at any length, so this statistic measures variation of curvature rather than curvature magnitude. The implementation names it "smoothness". Differences are taken over sampled checkpoints, without division by their token spacing; they describe the sampled trajectory rather than a derivative per token.

Layer-wise statistics: four features. For each monitored layer, define

$$\mu_i = \frac{1}{B}\sum_{n=1}^{B}\lambda_{i,n}, \qquad s_i = \sqrt{\frac{1}{B}\sum_{n=1}^{B}(\lambda_{i,n} - \mu_i)^2}. \tag{42}$$

Their cross-layer means and standard deviations contribute four features describing sensitivity magnitude and dispersion.

Global and cross-layer statistics: seven features. The matrix mean and standard deviation are

$$\mu_{\text{all}} = \frac{1}{mB}\sum_{i,n}\lambda_{i,n}, \qquad s_{\text{all}} = \sqrt{\frac{1}{mB}\sum_{i,n}(\lambda_{i,n} - \mu_{\text{all}})^2}. \tag{43}$$

For $m > 1$, the headline feature definition averages correlations over all layer pairs,

$$\rho = \frac{2}{m(m-1)}\sum_{i<j} c_{ij}, \qquad c_{ij} = \begin{cases} \operatorname{Corr}(\lambda_{i,:}, \lambda_{j,:}), & s_i s_j > 0, \\ 0, & \text{otherwise.} \end{cases} \tag{44}$$

We set $\rho = 0$ when $m = 1$. With $p_i = \mu_i / \sum_j \mu_j$, normalized layer-mean entropy and the coefficient of variation of ranges are

$$H = -\frac{\sum_{i:p_i>0} p_i \log p_i}{\log m}, \qquad C_R = \frac{\operatorname{std}_i R_i}{\operatorname{mean}_i R_i}. \tag{45}$$

Set $H = 0$ when $m = 1$ or $\sum_i \mu_i = 0$, and $C_R = 0$ when the mean range is zero. The entropy summarizes how total mean sensitivity is distributed across layers; it is distinct from the mutual information with labels discussed in Appendix C.

Table 4: Complete feature specification. For a vector c over monitored layers, $\mathcal{A}(c) = (\mathrm{mean}_i\, c_i, \mathrm{std}_i\, c_i)$. The classifier input contains 21 dynamic features, or 24 features when the headline length metadata are appended.

Group	Count	Features
Temporal	10	$\mathcal{A}(M), \mathcal{A}(V), \mathcal{A}(TV), \mathcal{A}(S), \mathcal{A}(R)$
Layer-wise	4	$\mathcal{A}(\mu), \mathcal{A}(s)$
Global and cross-layer	7	$\mu_{\mathrm{all}}, s_{\mathrm{all}}, \max_{i,n} \lambda_{i,n}, \min_{i,n} \lambda_{i,n}, \rho, H, C_R$
Text length (headline)	3	$T_p, T, T_p + T$

The global dispersion feature s_{all} is a standard deviation. Since $\mu_{\mathrm{all}} = \mathrm{mean}_i\, \mu_i$ on a rectangular matrix, the 21 entries are not all linearly independent; their count is a feature-schema size, not an intrinsic dimension.

Feature ordering. Table 4 groups features by interpretation. The stored representation places the four matrix-wide statistics before the four layer-wise aggregates. This ordering is part of the fitted detector and is preserved between training and evaluation.

Length metadata and ablation scope. The headline input appends separately counted prompt and answer tokens and their sum, so its third length feature is exactly redundant by construction. The dynamic representation is $\mathbf{x}_{\mathrm{dyn}} = \phi(\boldsymbol{\mathcal{L}})$. In Table 2, removing Global drops seven sensitivity summaries together with the three token counts. This comparison measures their joint contribution; it does not isolate the effect of token length. Preprocessing is fitted using training data only, and each feature configuration is fixed before its test-fold evaluation.

C Statistical Interpretation and Generalization Conditions

C.1 Feature separation and Bayes error

Let $Y \in \{0, 1\}$ be the response label and $X = \phi(\boldsymbol{\mathcal{L}}) \in \mathbb{R}^{21}$ the dynamic representation. The headline representation appends three token counts and has 24 entries. Differences between the label-conditional feature distributions are an empirical property of the data, backbone, and extractor.

Proposition 2 (Detectability conditional on feature separation) *Let P_0 and P_1 be the conditional laws of X given $Y = 0$ and $Y = 1$, with equal class priors. Define total variation distance by $\mathrm{TV}(P_0, P_1) = \sup_A |P_0(A) - P_1(A)|$. The optimal error among all measurable classifiers is*

$$R_X^* = \frac{1}{2}\left(1 - \mathrm{TV}(P_0, P_1)\right). \tag{46}$$

Thus $R_X^* < 1/2$ precisely when the two feature distributions differ.

Proof. Use the dominating measure $\nu = P_0 + P_1$ and densities $p_y = dP_y/d\nu$. Pointwise minimization chooses the smaller of the two weighted error densities, giving

$$R_X^* = \frac{1}{2}\int \min(p_0, p_1)\, d\nu = \frac{1}{2} - \frac{1}{4}\int |p_1 - p_0|\, d\nu. \tag{47}$$

The last integral equals $2\,\mathrm{TV}(P_0, P_1)$. □

For unequal priors π_0, π_1, the expression becomes $R_X^* = \int \min(\pi_0 p_0, \pi_1 p_1)\, d\nu$. This optimal risk concerns unrestricted classifiers. A linear or tree-based detector may have additional approximation and estimation error, and a decision threshold selected for F1 need not minimize classification error.

C.2 What the feature map preserves

Let $\mathbb{P}_y$ denote the class-conditional law of the full sensitivity matrix. For any measurable deterministic feature map,

$$\mathrm{TV}(\phi_{\#}\mathbb{P}_0, \phi_{\#}\mathbb{P}_1) \le \mathrm{TV}(\mathbb{P}_0, \mathbb{P}_1), \tag{48}$$

where $\phi_{\#}\mathbb{P}_y$ is the distribution after applying ϕ. Every feature-space event has a matrix-space preimage, so its probability difference is bounded by the supremum over matrix-space events. A distribution shift in the matrix can therefore disappear under aggregation.

Because N_{eff} varies with answer length, the matrix sample space in Eq. (48) is the disjoint union $\bigsqcup_{b=1}^{N} \mathbb{R}_{+}^{m\times b}$, equipped with the corresponding measurable structure. The feature map is defined on each component using the short-trajectory conventions in Appendix B. Thus the contraction statement also applies to populations with different answer lengths.

The temporal statistics depend on differences between adjacent checkpoints and are therefore sensitive to temporal arrangement. They are invariant to reversing the checkpoint order of every layer, summarizing trajectory variation without encoding its temporal orientation. These summaries serve as predictive features for response-level hallucination detection.

C.3 Information retained about the label

Because X is a deterministic function of $\boldsymbol{\mathcal{L}}$, the appropriate conditional-independence relation is $Y \longrightarrow \boldsymbol{\mathcal{L}} \longrightarrow X$. The chain rule for mutual information gives

$$I(Y;\boldsymbol{\mathcal{L}}) = I(Y;X) + I(Y;\boldsymbol{\mathcal{L}} \mid X), \qquad I(Y;X) \le I(Y;\boldsymbol{\mathcal{L}}). \tag{49}$$

Equality holds exactly when Y and $\boldsymbol{\mathcal{L}}$ are conditionally independent given X, namely when X is sufficient for the label relative to the matrix. Positive label information and sufficiency require properties of the joint distribution; feature dimension alone establishes neither.

For the headline representation, the source variable is $(\boldsymbol{\mathcal{L}}, T_p, T)$, since the token counts need not be recoverable from the matrix. The same data-processing argument applies to this augmented source. Furthermore, deterministic compression of continuous variables need not give finite $I(X;\boldsymbol{\mathcal{L}})$; if X has a non-atomic distribution, this mutual information can be infinite. An information-bottleneck optimization would need a specified stochastic encoder or quantization and an explicit objective. DLIPS uses fixed statistics and supervised classification.

C.4 A conditional PAC-Bayes risk bound

Consider $n_{\text{tr}} \ge 8$ independent training observations drawn from one distribution, a fixed hypothesis space $\mathcal{F}$, a sample-independent prior P, and a possibly data-dependent posterior $Q \ll P$. Each hypothesis includes any training-selected layer set, preprocessing, and classifier and incurs a loss $\ell \in [0,1]$. The candidate backbone and extraction operations must either be fixed before the sample is drawn or included in this hypothesis space. Write $R(Q) = \mathbb{E}_{f\sim Q,Z}[\ell(f,Z)]$ and $\widehat{R}(Q) = n_{\text{tr}}^{-1}\sum_i \mathbb{E}_{f\sim Q}\ell(f,Z_i)$. The PAC-Bayes-kl bound states that, with probability at least $1-\delta$, simultaneously for all such Q,

$$\mathrm{kl}(\widehat{R}(Q)\|R(Q)) \le \frac{\mathrm{KL}(Q\|P) + \log(2\sqrt{n_{\text{tr}}}/\delta)}{n_{\text{tr}}}, \tag{50}$$

where $\mathrm{kl}(a\|b) = a\log(a/b) + (1-a)\log((1-a)/(1-b))$. The inequality $\mathrm{kl}(a\|b) \ge 2(a-b)^2$ yields

$$R(Q) \le \min\left\{1, \widehat{R}(Q) + \sqrt{\frac{\mathrm{KL}(Q\|P) + \log(2\sqrt{n_{\text{tr}}}/\delta)}{2n_{\text{tr}}}}\right\}. \tag{51}$$

This is a bound for the posterior-averaged loss. A deterministic fitted classifier requires a justified choice of P, Q or an additional conversion argument.

Derivation of the bound. For each fixed f, let $h_f = n_{\text{tr}}\,\mathrm{kl}(\widehat{R}(f)\|R(f))$. For independent losses in $[0,1]$, convexity reduces the exponential moment to the Bernoulli envelope with the same mean. The resulting binomial sum is at most $2\sqrt{n_{\text{tr}}}$, giving $\mathbb{E}_S \exp(h_f) \le 2\sqrt{n_{\text{tr}}}$. Averaging over the sample-independent prior and applying Markov's inequality yields $\mathbb{E}_P \exp(h_f) \le 2\sqrt{n_{\text{tr}}}/\delta$ on an event of probability at least $1-\delta$. On this same event, convexity of binary relative entropy and the change-of-measure inequality give

$$n_{\text{tr}}\,\mathrm{kl}(\widehat{R}(Q)\|R(Q)) \le \mathbb{E}_Q h_f \le \mathrm{KL}(Q\|P) + \log \mathbb{E}_P e^{h_f}. \tag{52}$$

This proves Eq. (50); the binary relative-entropy inequality then gives Eq. (51). For the change-of-measure step, apply Jensen's inequality to $\mathbb{E}_Q[h_f - \log(dQ/dP)]$, with the usual absolute-continuity convention. This proof requires the loss family and prior to be fixed before observing the sample.

Low feature dimension does not determine the complexity term. For example, with $P = \mathcal{N}(0, \tau^2 I_r)$ and $Q = \mathcal{N}(w, \sigma^2 I_r)$ over r parameters,

$$\mathrm{KL}(Q\|P) = \frac{1}{2}\left[\frac{\|w\|_2^2}{\tau^2} + r\left(\frac{\sigma^2}{\tau^2} - 1 + \log\frac{\tau^2}{\sigma^2}\right)\right]. \tag{53}$$

The formula follows by taking the Q-expectation of the Gaussian log-density ratio. At fixed r, moving w away from the prior or shrinking σ can make the KL divergence arbitrarily large. Feature compression therefore does not establish a small generalization gap.

The bound assumes independent sampling from the target distribution. For paired responses, an independent question/source group can serve as one observation if the loss is defined as a bounded group average; n_{tr} then counts independent groups. This targets group-average risk rather than an arbitrary response-weighted risk. Training-dependent standardization belongs inside the hypothesis, or must be fitted on independent data before invoking a fixed-feature argument. A prior cannot be selected from the same evaluation sample without additional justification. The reported experiments do not specify P, Q or evaluate the bound numerically, so their generalization evidence comes from the stated evaluation protocols.

D Numerical Validation and Computational Cost

D.1 Exact Jacobians and sensitivity distributions

Eight explicitly materialized FP32 Jacobians span five Gated DeltaNet and three full-attention coordinates. Their exact spectral norms range from 2.773 to 7.351 (median 3.913). Across the three 200-response complete-block validation runs, all 71,952 final coordinates are finite and converged after the documented FP32 fallback. With 24 monitored blocks and five requested checkpoints, the nominal total is 72,000; the HaluEval run contributes 23,952 rather than 24,000 coordinates because its effective checkpoint count is reduced for short answers. Table 5 reports their distributions. These validation runs are separate from the headline detector comparison.

Table 5: Observed complete-block spectral-norm distributions in the numerical validation runs. "Exact panel" uses explicitly materialized FP32 Jacobians; the other rows use validated matrix-free estimates over all 24 blocks and five answer-prefix checkpoints.

Source	Coordinates	Min.	Median	95th pct.	Max.
Exact FP32 panel	8	2.773	3.913	7.332	7.351
HaluEval validation	23,952	1.742	3.531	8.000	39.000
RAGTruth validation	24,000	1.742	3.563	8.188	38.250
MedHallu validation	24,000	1.727	3.469	7.500	37.500

D.2 Convergence and precision audit

With $\sigma_k = \|Jv_k\|_2$ and $\mu_k = \sigma_k^2$, the stopping diagnostics are

$$e_k = \frac{|\sigma_k - \sigma_{k-1}|}{|\sigma_{k-1}| + \varepsilon}, \qquad r_k = \frac{\|w_k - \mu_k v_k\|_2}{\|w_k\|_2 + \varepsilon}. \tag{54}$$

Both $e_k \le \epsilon_\sigma$ and $r_k \le \epsilon_r$ must hold; e_k is evaluated only from the second iteration onward. The safeguard ε is the floating-point machine epsilon of the relevant computed quantity. The convergence flag uses diagnostics at the pre-update direction v_k, whereas the returned value is recomputed at v_{k+1}; the logged residual is therefore not a residual evaluated at the returned direction. We also record $\min(\|v_{k+1} - v_k\|_2, \|v_{k+1} + v_k\|_2)$ as a sign-invariant diagnostic.

The convergence conditions and interpretation of the eigen-residual are derived in Appendix A. Approximation error is measured against exact FP32 SVD. Table 6 reports this calibration.

The implementation imposes a finite iteration cap and records convergence status. It flags zero normalization denominators and nonfinite products instead of replacing them with zero sensitivity. Any failed or unconverged coordinate triggers FP32 recomputation of the whole sample–layer cell. The validated extraction uses $K_{\max} = 50$, $\epsilon_\sigma = 10^{-3}$, and the default $\epsilon_r = \sqrt{\epsilon_\sigma}$; fallback runs use their saved tighter tolerances. A remaining failure is reported explicitly, and no feature vector is formed from an invalid matrix. For K valid updates, the final norm evaluation requires $K + 1$ JVPs and K VJPs, in addition to prefix and block evaluation.

On 864 independently audited coordinates, the BF16–FP32 absolute relative difference has median 0.300%, 95th percentile 1.136%, and maximum 10.81%, with flattened Spearman correlation 0.9994. These precision comparisons assess the numerical estimator; they do not measure hallucination-detection accuracy.

Table 6: Relative error (%) of matrix-free power iteration against exact FP32 SVD on seven calibration coordinates.

K	**Median**	**95th pct.**	**Max.**
1	50.2100	52.0890	52.6107
2	19.7483	28.7456	30.2079
5	3.8019	16.7222	20.2835
10	0.3869	5.6673	7.3996
20	0.00047	4.1133	5.8710
50	0.00047	0.00187	0.00236

D.3 Computational Cost

Matrix-free estimation avoids explicitly materializing the dense $d \times d$ Jacobian and therefore eliminates its $O(d^2)$ memory cost, while repeated differentiation remains the main computational overhead. For a common K valid updates per coordinate, extraction requires $mN_{\text{eff}}(K + 1)$ JVPs and $mN_{\text{eff}}K$ VJPs, in addition to prefix replay. With adaptive stopping, these counts are summed over the actual updates at each coordinate, with occasional fallback introducing additional work. If $C_{l,n}$ denotes the cost of one differentiated block call, the leading extraction cost scales with $\sum_{l,n}(K_{l,n} + 1)C_{l,n}$ plus prefix evaluation, up to the relative costs of JVPs and VJPs. In practice, the small numbers of sampled coordinates and power iterations keep this overhead manageable; on HaluEval, the complete DLIPS pipeline adds approximately 1.1 second per sample on average. Because the Lipschitz estimates can be computed alongside autoregressive decoding, part of this cost can be overlapped with generation. Memory usage is dominated by retained or recomputed activations and architecture-specific states rather than an explicit Jacobian.

E Experimental Setup and Reproducibility

E.1 Datasets and response protocol

HaluEval (Li et al., 2023a), RAGTruth (Niu et al., 2024), and MedHallu (Pandit et al., 2025) cover general question answering, retrieval-augmented generation, and medical question answering. We use response-level faithful/hallucinated labels and represent each example as a prompt paired with a candidate answer. When retrieved evidence is available, it is included in the prompt. The original corpus sizes and annotation units are as follows.

HaluEval (QA). The QA subset contains 10,000 records derived from HotpotQA, each pairing a correct answer with a hallucinated answer. Including both answers from every record yields 10,000 correct and 10,000 hallucinated response-level instances. The full HaluEval collection also contains 10,000 dialogue records, 10,000 summarization records, and 5,000 general queries.

RAGTruth. The dataset contains 2,965 source instances and 17,790 generated, annotated responses: 5,934 for question answering, 6,198 for data-to-text generation, and 5,658 for summarization. Of these responses, 7,664 (43.1%) contain at least one annotated hallucination span and 10,126 contain none. These span annotations support response-level labels based on the presence of a hallucination span.

MedHallu. The dataset contains 10,000 medical QA records, each pairing a correct answer with a generated hallucinated answer. Of these records, 1,000 come from PubMedQA's `pqa_labeled` subset and 9,000 from `pqa_artificial`. Among the hallucinated answers, the type counts are 7,600 for question misinterpretation, 1,222 for incomplete information, 1,129 for incorrect mechanism attribution, and 49 for evidence fabrication.

The evaluation examples from each dataset are divided into training, validation, and test subsets using a 60%/20%/20% split. Splitting is performed by question or source group so that related responses, including faithful/hallucinated pairs, remain in the same subset and do not leak across splits.

For labeled-answer evaluation, candidate answers are teacher-forced through Qwen3.5-0.8B, Qwen3.5-9B, or Llama-3.1-8B. At checkpoint q_n, the backbone receives the prompt together with the first q_n answer tokens. Candidate text and labels remain fixed across detector backbones. Prompt and answer token IDs are constructed separately to preserve an explicit boundary, and token counts may therefore differ across tokenizers.

The application workflow in Figure 2 instead collects states during autoregressive generation. The length-dependent checkpoint rule in Eq. (3) defines checkpoint placement for completed-answer evaluation; during online generation, checkpoints are selected using the currently available prefix or from stored states after completion. Response-level features, including answer length, are assembled once the candidate answer is complete.

SelfCheckGPT and Semantic Entropy additionally generate responses required for their consistency-based measurements.

E.2 Backbones and extraction settings

The headline Qwen3.5-0.8B experiments use one NVIDIA A100 (80GB) GPU and PyTorch. This backbone has hidden size 1024 and 24 decoder blocks, comprising 18 Gated DeltaNet and six full-attention blocks. The headline configuration monitors zero-based layers 8–15 and requests $N = 20$ checkpoints, reduced to $N_{\text{eff}} = \min(N, T)$ for short nonempty answers. Table 2 compares this middle-layer range with the bottom, top, and full-layer configurations. All ablations use the Qwen3.5-0.8B/DLIPS-Logistic configuration in Table 1, varying only the indicated feature groups or monitored layers.

The complete-block extractor retains each backbone's normalization, mixer, gates, residual additions, and feed-forward paths. The current-token map in Eq. (4) corresponds to the implementation's `instantaneous` replay mode. It disables mutable caches and uses differentiable eager operations or an unfused linear-attention fallback when fused kernels do not support JVP/VJP computation. The validated settings are BF16, $K_{\max} = 50$, $\epsilon_\sigma = 10^{-3}$, and $\epsilon_r = \sqrt{\epsilon_\sigma} \simeq 0.0316$. Exact-Jacobian calibration and failed-cell fallback use FP32; fallback tolerances are taken from the corresponding run configuration. Appendix A derives the complete-block Jacobian, and Appendix D reports its calibration.

The numerical validation cohorts contain 200 responses per dataset and monitor all 24 blocks at five requested checkpoints. These settings differ from the headline configuration. The Qwen3.5-9B and Llama-3.1-8B results in Table 1 use separately fitted detectors on the same candidate-answer cohorts; they evaluate sensitivity features on each backbone rather than transfer of a classifier between backbones.

E.3 Comparison methods and model access

Output-based signals include candidate-answer perplexity, SelfCheckGPT's sampled-response consistency (Manakul et al., 2023), and Semantic Entropy over semantic clusters (Farquhar et al., 2024). Perplexity and probability-weighted uncertainty require token probabilities; text-only consistency variants can use generated text alone.

Internal-state methods include LLM-Check (Sriramanan et al., 2024), attention-graph topology in TOHA (Bazarova et al., 2026), cross-layer updates in ICR Probe (Zhang et al., 2025), state transitions in PoLLMGraph (Zhu et al., 2024), and hidden-state covariance in INSIDE/EigenScore (Chen et al., 2024).

We reproduce the baselines using the implementation details and hyperparameter settings reported in their original papers. The direct headline baseline comparison uses Qwen3.5-0.8B. The larger-backbone experiments evaluate DLIPS variants. Sampling-based methods retain the auxiliary scorers required by their procedures. The candidate-answer comparisons follow the grouped training/validation/test protocol in Section 5.1. Layer selection, preprocessing, hyperparameters, and score orientation use training/validation data. Appendix F identifies the scope of the reported comparison and the adaptation details that are not specified here.

E.4 Metrics and uncertainty

The headline table reports Accuracy, F1, and AUROC. AUPRC, balanced accuracy, precision, recall, Matthews correlation coefficient (MCC), Brier score, and expected calibration error (ECE) are not reported in the present manuscript; Brier score and ECE would additionally require probability-valued predictions. Continuous scores retain a positive direction fixed without test-set inspection; classification metrics use a prespecified or training/validation-selected threshold.

F Scope of Supplementary Baseline Comparison

The baseline rows in Table 1 use Qwen3.5-0.8B and the fixed candidate-answer evaluation described in Section 5.1. ICR Probe measures cross-layer hidden-state updates, and PoLLMGraph measures state-transition patterns. Their rows are evaluated against the same response-level labels and grouped split as DLIPS.

INSIDE/EigenScore is discussed in Related Work but has no row in Table 1. No separate generated-answer comparison or uncertainty table is reported here. The headline results therefore support comparisons only with the methods and metrics displayed in Table 1.

G Scope of the Measurement and Application

DLIPS requires white-box access to differentiable decoder blocks. The local Jacobian measures how the current block output responds to its current input under a fixed prefix. It does not differentiate discrete token sampling, perturb earlier prefix states jointly, or identify the causal origin of a factual error. Answer-level labels supervise the resulting trajectory summaries.

Sampling states during generation and producing a final response score are distinct operations. The current features summarize multiple checkpoints and include completed-answer length. Prefix-conditioned scores or token-level interventions would require a separately specified feature construction, training protocol, and evaluation.

The ablations compare feature groups and monitored layer ranges. They describe predictive performance in the reported cohorts; they do not localize knowledge storage or prove that instability causes hallucinations. Equations (46)–(49) identify the distributional separation needed for detection, and Eqs. (50)–(51) state additional assumptions for a generalization bound. The numerical calibration tests the spectral estimator, while held-out detection metrics test the usefulness of its features.

H MULTI-SCENARIO CASE VISUALIZATIONS

Figures 5–7 display selected sensitivity matrices from HaluEval, RAGTruth, and MedHallu. Each triplet juxtaposes a faithful and a hallucinated answer and their pointwise difference at the displayed layer and checkpoint indices. The displayed pairs have strong layer bands in both label classes, while their signed differences vary with layer and checkpoint. The HaluEval pairs contain localized positive and negative deviations; the RAGTruth pairs show prominent negative differences around some middle layers; the MedHallu pairs show narrower deviations at different depths. These examples illustrate variation in the measured trajectories.

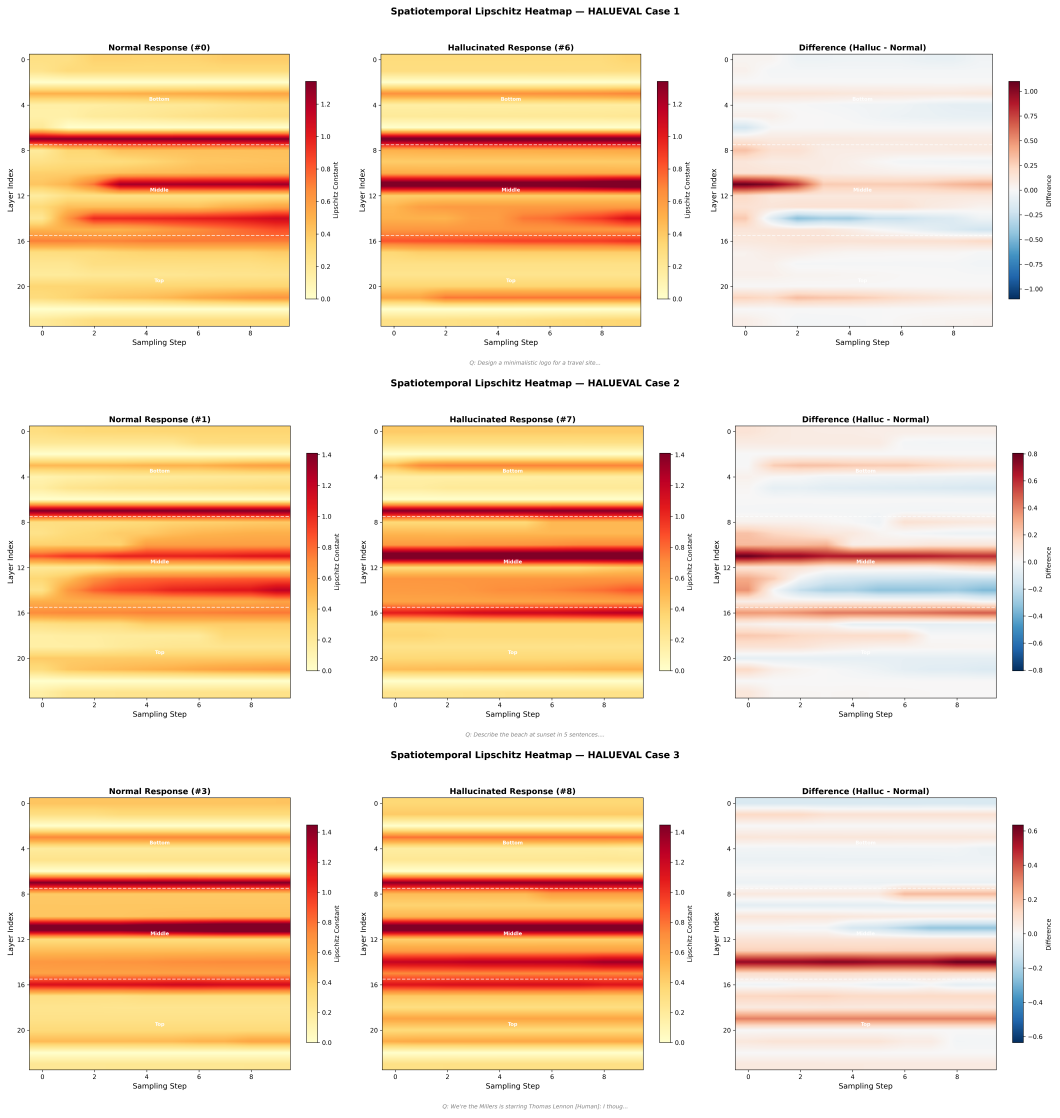


Figure 5: HaluEval sensitivity trajectories. Each triplet shows a faithful answer, a hallucinated answer, and their difference (hallucinated minus faithful). Rows are decoder layers and columns are sampled checkpoint indices; equal indices do not identify matched tokens.

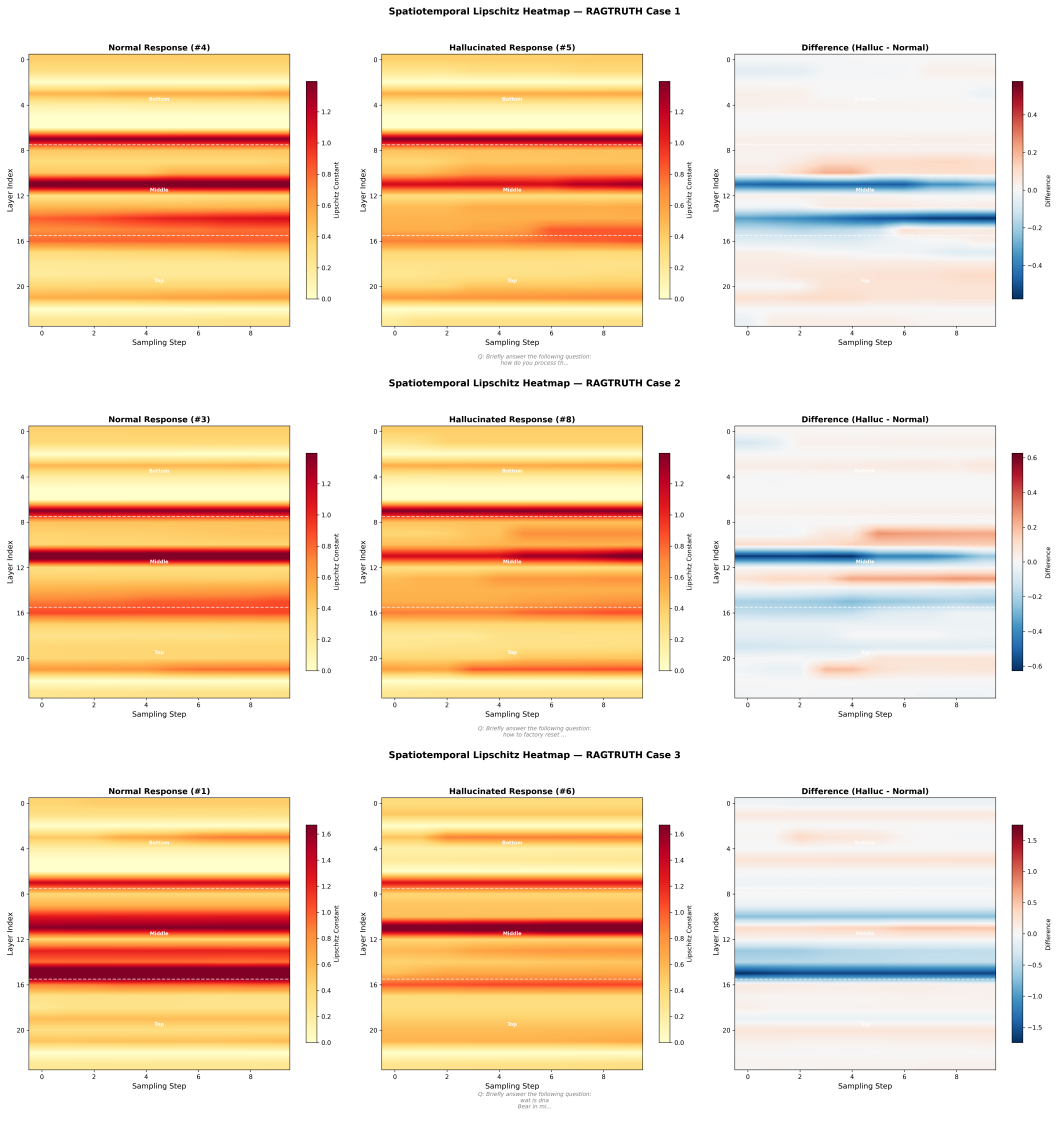


Figure 6: RAGTruth sensitivity trajectories, shown with the same panel definitions as Figure 5. Each difference panel compares distinct answers at corresponding checkpoint indices.

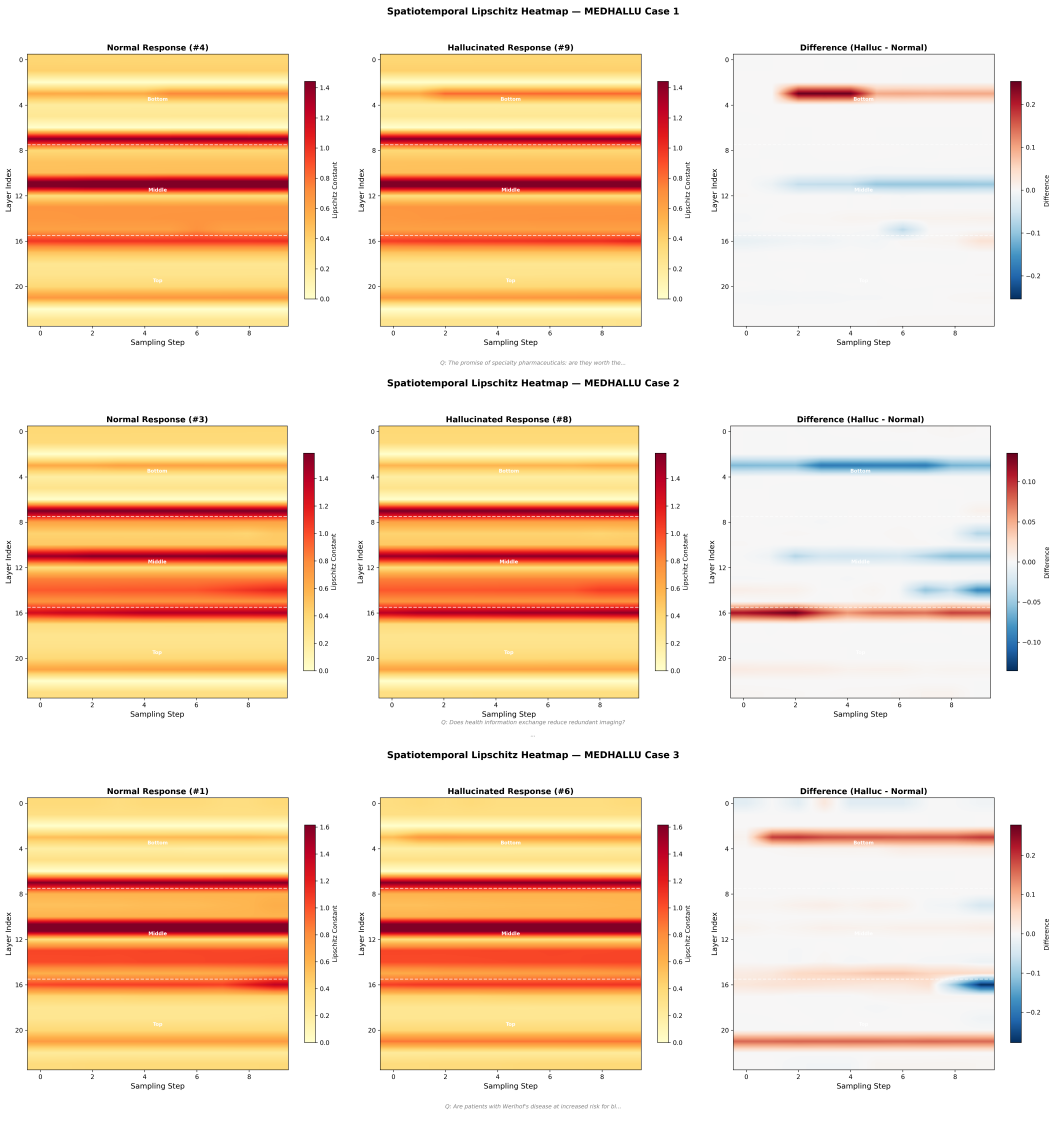


Figure 7: MedHallu sensitivity trajectories, shown with the same panel definitions as Figure 5. The difference panels describe sensitivity contrasts between examples, not matched token effects.