DETECTING CONTEXTUAL HALLUCINATIONS BY POOLING AND PROBING ATTENTION-HEAD OUTPUTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Retrieval-augmented language models can produce claims that are unsupported by their supplied context, motivating methods for detecting contextual hallucinations. Prior work has shown that truthfulness and hallucination can be predicted from internal model representations, suggesting that these failures leave detectable signatures within the model. Building on linear probing methods for truthfulness, we pool each attention head's current-value contributions across generated positions into a response-level representation and train a linear probe to learn a hallucination-discriminating axis. Across multiple RAG hallucination benchmarks and model families, HEADAXIS consistently matches or exceeds strong white-box hallucination detectors. Analysis shows that the axes capture a hallucination signal that generalizes across datasets and can be decomposed across response tokens without additional supervision, while ablations show that the representation is robust across alternative design choices.

1 INTRODUCTION

Retrieval-augmented generation (RAG) seeks to improve the reliability of language-model outputs by grounding generation in external evidence (Lewis et al., 2021; Shuster et al., 2021). Yet access to relevant evidence does not guarantee that a model will use it faithfully: RAG systems can still produce claims that are unsupported by or contradictory to the supplied context (Niu et al., 2024; Chuang et al., 2024). Detecting these contextual hallucinations therefore remains important even when the appropriate evidence has already been retrieved.

A growing body of work suggests that hallucination-related information is present in model internals, including attention patterns (Chuang et al., 2024), attention-head activations (Ho et al., 2025) and feed-forward activations (Sun et al., 2025). One way to extract this information is through activation probing. Li et al. (2023) use linear probes on individual attention heads to identify representations associated with truthfulness, while CH-Wang et al. (2024) use linear probes over hidden states to detect contextual hallucinations.

Building on these approaches, we ask whether contextual hallucination can be detected from response-level representations of individual attention heads. We fit linear probes to representations constructed from the value spaces of attention heads to learn a hallucination-discriminating axis and combine the resulting scores. Across multiple RAG hallucination benchmarks and model families, HEADAXIS matches or exceeds existing hallucination detectors. We further analyze the detector to understand which representation choices drive its performance, how individual response positions contribute to its predictions, and how well it transfers to unseen datasets. Our contributions are:

- We construct a response-level representation for contextual hallucination detection by aggregating attention-head value contributions across the generated response.
- We train headwise linear probes on these representations and combine their scores into a lightweight response-level hallucination detector requiring only a single forward pass.
- We evaluate HEADAXIS against existing hallucination detectors across multiple RAG benchmarks and model families, where it consistently achieves strong performance.
- We analyze our detector through representation ablations and within-response attribution, show that it transfers to unseen datasets without retraining, and find that the same representation also improves truthfulness detection on TruthfulQA.

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.

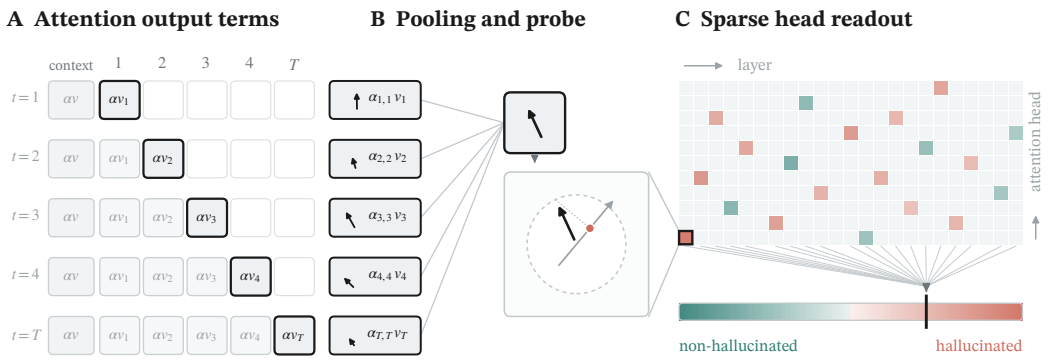


Figure 1: Overview of HEADAXIS. We construct response-level representations from attention-head value contributions, fit a linear probe to each head, and combine the resulting scores for contextual hallucination detection.

2 BACKGROUND AND RELATED WORK

Hallucination detectors can broadly be divided into black-box methods, which operate on model inputs and outputs, and white-box methods, which use internal model representations. We focus on white-box detectors and on work using linear probes to study truthfulness and hallucination.

Truthfulness probing and steering. Several studies show that truthfulness can be decoded from internal model representations using simple classifiers. SAPLMA (Azaria & Mitchell, 2023) trains classifiers on hidden activations to distinguish true from false statements, while Bürger et al. (2024) study how well truth-related representations generalize across datasets and model settings. Marks & Tegmark (2024) identify linear structure separating representations of true and false statements and use class-conditional mean differences to define truth directions. Inference-Time Intervention (Li et al., 2023) applies linear probing at the level of individual attention heads and uses the highest-performing heads for truthfulness steering. Concurrent work on HalluTracer (Guo et al., 2026) instead aggregates truthfulness signals across model depth. These results motivate our use of simple linear probes over internal representations.

Hallucination signals in internal representations. CH-Wang et al. (2024) train linear probes on transformer sublayer states for token- and response-level hallucination detection in grounded generation, showing that hallucination can be detected directly from internal states. INSIDE (Chen et al., 2024) measures consistency across sampled hidden-state representations, while Semantic Entropy Probes (Kossen et al., 2024) predict semantic uncertainty from the hidden states of a single generation. HIDE (Chatterjee et al., 2025) measures the dependence between representations of the supplied context and generated response, and ICR Probe (Zhang et al., 2025) tracks how hallucination-related information changes across layers. Concurrent work on CORTEX (Furumai et al., 2026) detects token-level RAG hallucinations by comparing internal representations with and without retrieved context. Together, these methods show that contextual hallucination is reflected in internal model representations at both response and token level.

Attention and component-level detectors. Other methods use signals from more specific transformer components. Lookback Lens (Chuang et al., 2024) uses per-head ratios between attention to the supplied context and attention to previously generated tokens. NoVo (Ho et al., 2025) uses the norms of selected attention-head outputs, while ReDeEP (Sun et al., 2025) combines retrieval-oriented attention signals with feed-forward components associated with parametric knowledge. TPA (Lu et al., 2026) attributes next-token probability to different information sources. These methods show that hallucination-related signals can be traced to individual transformer components rather than only to the full hidden state.

3 FROM TRUTHFULNESS PROBING TO RESPONSE-LEVEL FAITHFULNESS

Li et al. (2023) train linear probes on attention-head activations at the terminal generated-token position to identify heads whose representations distinguish truthful from false answers. They rank heads by validation probe accuracy and use the highest-performing heads for interventions that steer generation toward truthfulness. Our goal is related but distinct: rather than using probe performance to identify heads for intervention, we use the information encoded in individual heads for response-level detection. In particular, we consider contextual hallucination detection, where the objective is to determine whether a generated response is supported by its supplied context.

Although ITI was not introduced as a hallucination detector, its terminal-head representation provides a strong starting point. As Table 1 shows, linear probes on terminal attention-head activations are competitive with dedicated white-box hallucination detectors. However, unlike the shorter truthful statements considered by ITI, RAG responses are typically longer and contain multiple claims, so evidence of hallucination may arise well before the final response position. A single terminal activation may therefore provide an incomplete representation of the response for this task.

Related work by CH-Wang et al. (2024) shows that probes over feed-forward and self-attention output states can detect hallucinations in grounded generation, including response-level probes that use attention pooling across the generated response. This motivates moving from terminal head representations to response-level representations while retaining the ability to probe individual attention heads.

Building on these findings, HeadAxis constructs a response-level representation within each attention head and learns a linear hallucination direction from it. Rather than relying on a single terminal activation, the representation aggregates information across the generated response, allowing evidence from earlier positions to contribute directly to response-level detection.

3.1 ATTENTION HEAD OUTPUTS

For a given attention head, let v_j denote its value vector at sequence position j and $\alpha_{t,j}$ the attention weight from position t to position j. Let s denote the position of the first generated token and T the position of the final state used to predict a generated token. For any generated position $t \in \{s, \dots, T\}$, let o_t denote the head's pre-output-projection activation. Each head's output can be decomposed into prompt and generated contributions:

$$o_t = \underbrace{\sum_{p=1}^{s-1} \alpha_{t,p} v_p}_{\text{prompt values}} + \underbrace{\sum_{g=s}^{t} \alpha_{t,g} v_g}_{\text{generated values}} , \tag{1}$$

where p indexes prompt positions and g indexes generated positions.

The head's contribution to the residual stream is $W_O^{(h)} o_t$, where $W_O^{(h)}$ denotes the block of the attention output projection corresponding to head h. Li et al. probe the complete head activation at the terminal generated-token position. Consequently, the representation supplied to their probe contains both prompt and generated value vectors, weighted according to the attention distribution of the terminal query. This is suitable for identifying informative heads, but for response-level detection we instead seek to aggregate information represented throughout generation.

3.2 POOLING HEAD OUTPUTS

A natural way to incorporate all response positions is to average the attention-head activations across generated positions. Let $n = T - s + 1$ denote the number of positions from s to T. Rearranging this average reveals two properties of naive output pooling:

$$\bar{o} = \frac{1}{n} \sum_{t=s}^{T} o_t = \frac{1}{n} \left[\underbrace{\sum_{p=1}^{s-1} \left(\sum_{t=s}^{T} \alpha_{t,p} \right) v_p}_{\text{prompt values}} + \underbrace{\sum_{g=s}^{T} \left(\sum_{t=g}^{T} \alpha_{t,g} \right) v_g}_{\text{generated values}} \right] . \tag{2}$$

First, the pooled representation directly incorporates prompt value vectors. Since our objective is to detect a state associated with the generated response, we remove these direct prompt contributions. This does not make the representation independent of the prompt: generated-position representations remain conditioned on the preceding context.

Second, each generated value v_g can contribute to every subsequent head activation. Its effective weight therefore accumulates attention across positions $g, \ldots, T$, giving earlier generated values more opportunities to influence the pooled representation simply because they occur earlier in the response.

We remove both effects by retaining only each generated token's current-position contribution, $\alpha_{g,g} v_g$, and averaging these contributions across the response:

$$r = \frac{1}{n} \sum_{g=s}^{T} \alpha_{g,g} v_g. \tag{3}$$

This representation excludes direct prompt-value contributions and includes each value vector from s to T exactly once, weighted by its contribution to the head activation at its own position. Empirically, this weighting performs best among the pooling schemes we evaluate on the training set; see Appendix G.1 for a post-hoc analysis. We use value space because $\alpha_{g,g} v_g$ is the natural per-token term in the decomposition of Eq. 1; Section 6 compares alternative projections.

3.3 Directional Response Representations

For each head h, we normalize the pooled vector to obtain one directional representation per response:

$$q_h = \frac{r_h}{\|r_h\|_2}. \tag{4}$$

The order of operations is important: normalizing each current-position contribution separately would cancel its positive attention coefficient,

$$\frac{\alpha_{t,t} v_t}{\|\alpha_{t,t} v_t\|_2} = \frac{v_t}{\|v_t\|_2}, \qquad \alpha_{t,t} > 0. \tag{5}$$

Pooling before normalization therefore allows the relative attention weights across generated positions to affect the direction of the response representation.

3.4 Headwise Linear Axes

Following the linear probing approach of Li et al., we fit an ℓ_2-regularized binary logistic probe to each head's response representation. Let $y = 1$ denote a hallucinated response and $y = 0$ a faithful response. For each head h,

$$s_h = b_h + w_h^\top q_h, \qquad p_h(y = 1 \mid q_h) = \sigma(s_h). \tag{6}$$

The vector w_h defines the normal to a learned linear boundary separating hallucinated and faithful responses in the head representation. Its score s_h measures evidence for hallucination within that head's response representation. In practice, each head's response representation is standardized featurewise using training-set statistics before fitting the probe; the exact implementation details are given in Appendix B.

3.5 Combining Heads

Different heads can provide complementary or redundant evidence. We combine their scores using a class-balanced sparse logistic regression:

$$p(y = 1 \mid x) = \sigma\left(b + \sum_h \beta_h s_h\right). \tag{7}$$

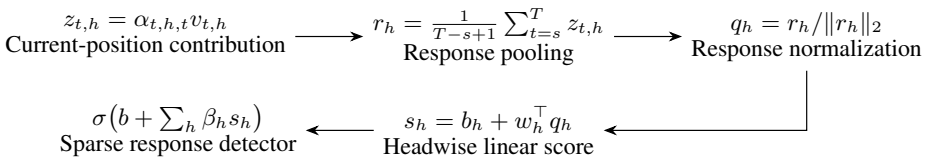


Figure 2: HeadAxis construction. Current-position value contributions are pooled across the generated response and then normalized. A linear probe scores each head's response representation, and a sparse logistic readout combines the scores.

An ℓ_1 penalty on β_h selects a compact subset of informative heads. Probe scores are standardized using training-set statistics before fitting the sparse reader. For clarity, this standardization is again omitted from the notation; Appendix B gives the complete fitted form and implementation details.

The method thus adapts linear truthfulness probing from a mechanism for identifying intervention heads into a response-level detector. Rather than probing a single terminal activation, it constructs each head's representation from current-position value contributions distributed throughout the generated response.

4 Experimental Setup

Datasets and models. We evaluate contextual hallucination detection across three model families: LLaMA-2-7B, Mistral-7B, and Qwen3-4B. When possible, we evaluate a model on responses that it generated itself. RAGTruth (Niu et al., 2024) and RAGognize (Ridder et al., 2026) contain responses generated by LLaMA-2-7B and Mistral-7B, allowing these models to be evaluated on their own generations. For Qwen3-4B, which is not a response generator in these datasets, we use the LLaMA-2-7B responses and extract Qwen3-4B activations by teacher forcing.

For RAGBench (Friel et al., 2025) and HDM-Bench (Paudel et al., 2025), the supplied responses are treated as externally generated for all three evaluated models, and activations are therefore extracted by teacher forcing each context–response pair through the model.

We evaluate on the provided test split of each dataset where available. For HDM-Context, which does not provide a conventional train/test split, we use 5-fold cross-validation over the whole dataset. Exact sample counts for all evaluation splits and label distributions as well as details on dataset selection and processing are provided in Appendix A.

Baselines. We compare HeadAxis with ReDeEP (Sun et al., 2025), NoVo (Ho et al., 2025), TPA (Lu et al., 2026), TSV (Park et al., 2025), Lookback Lens (Chuang et al., 2024), SAPLMA (Azaria & Mitchell, 2023), RAGLens (Xiong et al., 2026), ITI (Li et al., 2023), and PoolingE (CH-Wang et al., 2024). All methods are evaluated on the same fixed response sets and evaluation partitions. We use the authors' implementations and configurations when available and otherwise reproduce the published method.

Because ITI was introduced for intervention rather than response-level hallucination detection, we define a detector adaptation, which we denote *ITI-Detect*. Following its original representation, ITI-Detect trains one linear probe per head on the terminal pre-W_O attention-head activation, selects the 48 heads with the strongest validation performance, standardizes their probe scores, and combines them with equal weights. Baseline-specific adaptations are detailed in Appendix C.

We separately evaluate two additional baselines in the appendix: RAGognizer (Ridder et al., 2026) in Appendix C.10 and MiniCheck (Tang et al., 2024) in Appendix D. RAGognizer jointly fine-tunes the language model together with a hallucination-detection head, so we report it separately from the white-box baselines. MiniCheck is a black-box verifier that operates on the supplied context and generated response, and we include it as a separate black-box comparison.

HeadAxis pools current-position value contributions across the response, normalizes the resulting representation for each head, and fits an ℓ_2-regularized logistic probe to every layer–head pair. The head scores are then combined with a class-balanced ℓ_1-regularized logistic classifier. To fit this classifier without reusing in-sample probe scores, we generate grouped out-of-fold scores: within

	RAGTruth						RAGognize					
	LLaMA-2-7B		Mistral-7B		Qwen3-4B		LLaMA-2-7B		Mistral-7B		Qwen3-4B	
ReDeEP (Sun et al.)	.736	.706	.805	.828	.740	.740	.667	.689	.608	.415	.707	.728
NoVo (Ho et al.)	.753	.755	.834	.848	.772	.807	.737	.762	.820	.683	.892	.913
TPA (Lu et al.)	.782	.783	.853	.874	.779	.799	.737	.766	.770	.601	.831	.853
TSV (Park et al.)	.706	.741	.795	.800	.701	.680	.647	.681	.606	.397	.735	.737
Lookback Lens (Chuang et al.)	.813	.824	.886	.894	.833	.847	.836	.872	.860	.778	.909	.929
RAGLens (Xiong et al.)	.828	.846	.864	.889	.841	.860	.852	.876	.865	.766	.931	.949
Activation probing												
SAPLMA (Azaria & Mitchell)	.770	.791	.851	.866	.796	.808	.785	.811	.880	.800	.915	.937
ITI-Detect (Li et al.)	.793	.814	.882	.888	.834	.852	.808	.833	.869	.774	.929	.946
PoolingE (CH-Wang et al.)	.821	.806	.888	.885	.850	.836	.829	.846	.814	.740	.931	.942
HEADAXIS (Ours)	**.841**	**.850**	**.906**	**.907**	**.856**	**.873**	**.869**	**.898**	**.890**	**.825**	**.946**	**.959**

	RAGBench						HDM-Context					
	LLaMA-2-7B		Mistral-7B		Qwen3-4B		LLaMA-2-7B		Mistral-7B		Qwen3-4B	
ReDeEP (Sun et al.)	.523	.154	.527	.147	.533	.155	.747	.868	.700	.856	.730	.874
NoVo (Ho et al.)	.740	.340	.737	.315	.818	.426	.851	.938	.845	.928	.885	.950
TPA (Lu et al.)	.734	.364	.753	.385	.765	.376	.843	.930	.844	.937	.838	.931
TSV (Park et al.)	.652	.232	.650	.227	.738	.273	.709	.869	.720	.866	.729	.859
Lookback Lens (Chuang et al.)	.791	.442	.788	.440	.828	.493	.866	.938	.863	.941	.880	.949
RAGLens (Xiong et al.)	.785	.436	.782	.430	.848	.502	.877	**.953**	.869	.946	.884	.951
Activation probing												
SAPLMA (Azaria & Mitchell)	.759	.369	.778	.396	.846	.468	.870	.947	.852	.940	.885	.953
ITI-Detect (Li et al.)	.801	.430	.803	.441	.853	.514	.877	.950	.878	.949	.904	.963
PoolingE (CH-Wang et al.)	.555	.232	.562	.226	.652	.358	.829	.905	.846	.923	.849	.923
HEADAXIS (Ours)	**.816**	**.485**	**.814**	**.479**	**.868**	**.551**	**.878**	.952	**.883**	**.951**	**.906**	**.963**

Table 1: Response-level hallucination detection. **Top:** Datasets for which the model that produced each response is known; **Bottom:** datasets with responses generated by unspecified model. Columns indicate the model used for scoring; models are evaluated on their own responses when available. Qwen3-4B is always evaluated on responses generated by a different model. Each entry reports AUROC and AUPRC; higher is better. Bold and underlined values indicate the **best** and second-best scores based on unrounded metrics.

each fold, the response representations are standardized and the headwise probes are fitted on the remaining training data, then applied to the held-out examples. The resulting out-of-fold head scores are standardized using training-derived statistics and used to fit the final classifier. For evaluation, the representation standardization and headwise probes are refitted on the full training set; test examples are excluded from all fitting and model-selection steps. Full implementation details and hyperparameters are provided in Appendix B. We additionally release the source code anonymously.[1]

Metrics and statistical testing. We report AUROC and AUPRC separately for each model–dataset combination. Differences between HeadAxis and the strongest baseline are evaluated for significance using paired bootstrap tests in Appendix E.

5 RESULTS

Table 1 reports response-level hallucination detection across four datasets and three model families. HeadAxis achieves the highest AUROC in all 12 model–dataset settings and the highest or tied-highest AUPRC in 11 of 12.

On RAGTruth and RAGognize, HeadAxis achieves the highest AUROC and AUPRC for LLaMA-2-7B and Mistral-7B when each model is evaluated on its own responses. It also achieves the highest scores with Qwen3-4B, which is evaluated on LLaMA-generated responses. Across these six settings, HeadAxis consistently improves over both ITI-Detect and PoolingE.

[1] https://github.com/careful-pineapple-42/headaxis

On RAGBench, HeadAxis achieves the highest AUROC and AUPRC with all three evaluated models. On HDM-Context, HeadAxis achieves the highest AUROC with all three models. It also achieves the highest AUPRC with Mistral-7B, ties ITI-Detect with Qwen3-4B, and is slightly below RAGLens with LLaMA-2-7B. Overall, the detector remains effective when applied to responses produced by a different or unspecified model. Statistical comparisons are reported in Appendix E.

5.1 Cross-Dataset Transfer

We evaluate whether a detector trained on one dataset transfers to unseen datasets. For each model, we fit the complete HeadAxis detector on the training split of RAGTruth, RAGognize, or RAGBench and apply it unchanged to the remaining datasets. All fitted components remain fixed during transfer, and target labels are used only for evaluation. HDM-Context is used only as a target because it does not provide an independent training split.

Model	Training data	RAGTruth	RAGognize	RAGBench	HDM-Context
LLaMA-2-7B	RAGTruth	*.840/.850*	.756/.810	.562/.200	.796/.915
	RAGognize	.796/.811	*.869/.898*	.743/.341	.817/.920
	RAGBench	.742/.715	.777/.807	*.816/.485*	.854/.938
Mistral-7B	RAGTruth	*.906/.907*	.662/.504	.525/.204	.848/.934
	RAGognize	.753/.754	*.890/.825*	.738/.333	.835/.927
	RAGBench	.778/.783	.753/.599	*.814/.479*	.842/.934
Qwen3-4B	RAGTruth	*.856/.873*	.893/.921	.719/.292	.840/.936
	RAGognize	.783/.771	*.946/.959*	.810/.402	.879/.952
	RAGBench	.708/.686	.897/.916	*.868/.551*	.883/.952

Table 2: **Cross-dataset transfer.** Each cell reports AUROC/AUPRC; columns are target datasets. Italic cells are same-dataset references.

Table 2 shows that the learned signal generally transfers across datasets, sometimes strongly, with a mean AUROC of .777 across the 27 cross-dataset evaluations. Transfer is particularly strong to HDM-Context. Performance nevertheless depends on the source dataset, indicating that the learned representations are not fully dataset invariant.

Overall, these results suggest that HeadAxis captures hallucination-related information that extends beyond the dataset on which the detector is trained, while still being affected by differences in task and response distributions.

6 Analysis and Discussion

To better understand the role of the representation, we compare the complete pre-W_O attention-head output used by ITI with the current-value contribution used by HeadAxis, evaluated at the terminal response position, pooled across the response, or pooled across the prompt and response. All conditions use the same normalization, headwise probing methodology, and final classifier.

	Extraction / pooling region		
Head representation	Terminal	Pooled response	Pooled prompt + response
Complete head output	0.8910 / 0.8986	0.8790 / 0.8909	0.8430 / 0.8640
Current-value contribution	0.8658 / 0.8604	**0.9061 / 0.9071**	0.8586 / 0.8777

Table 3: **Representation and pooling ablation on RAGTruth/Mistral.** Each cell reports AUROC / AUPRC on the same test responses, using the same probing and sparse supervision setup as HeadAxis. Complete outputs are the full pre-W_O attention-head outputs; current contributions are $\alpha_{t,h,t}v_{t,h}$. Pooled representations are computed as representation means.

Table 3 shows that current-value contributions are weaker than complete head outputs at the terminal position, but perform best when pooled across the response. Consistent with the analysis in Section 3.2, pooling does not improve complete head outputs. Including prompt positions lowers

AUROC for both pooled representations. These results support combining current-value contributions with response pooling rather than treating either choice as independently superior.

Appendix G.5 extends this matched comparison to all twelve settings. With the fitting procedure held fixed, the response-pooled representation achieves higher AUPRC in all twelve settings and higher AUROC in eleven, with the remaining AUROC tied. The differences are largest on RAGTruth and RAGognize, while the two representations are closer on RAGBench and HDM-Overall. For this control, RAGBench uses the full available training set and HDM uses the overall hallucination label, so these rows are not directly comparable to the corresponding main-table results. Additional ablations examine temporal attention weighting, normalization, prompt-value inclusion, and a fixed-answer context ablation in Appendix G.

Our controls do not establish that the hallucination signal is specific to only the value space. Matched comparisons across query, key, and value projections perform similarly (Appendix G.4), and a fitted low-dimensional projection of the hidden state performs comparably to native values, while the full hidden state performs worse. This suggests that the majority of the benefit comes from probing a low-dimensional representation of the hidden state rather than from value space in particular. We choose the value space because the current-position value sum is a natural summary of the pooled decomposed attention outputs; furthermore, the learned value-space reduction is available directly from the trained model, with no additional fitting required.

6.1 Attribution Within a Response

Because both HeadAxis and ITI-Detect use linear scores, their response-level predictions can be decomposed into contributions from individual positions. Appendix H.2 gives the full decomposition. Figure 3 shows these contributions for one RAGTruth response and one RAGognize response.

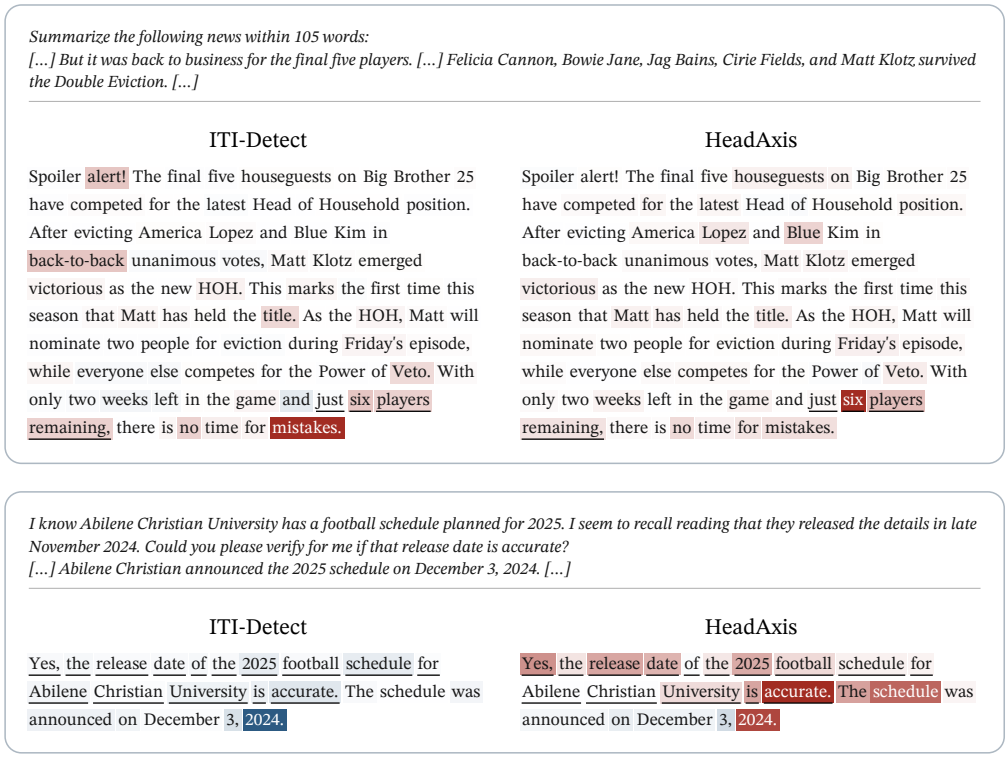


Figure 3: **Position-level contributions to the hallucination score.** Red marks positions that push the detector toward hallucination, while blue marks positions that push it toward a faithful prediction. HeadAxis decomposes its score across generated positions, while ITI-Detect decomposes its terminal head output across the positions attended to by the final query. The displayed source excerpts are shortened; both methods use the full input.

In the RAGTruth example, the unsupported claim is that six players remain. HeadAxis assigns its largest increase in hallucination score to "six," while ITI-Detect's score is raised by several positions near the end of the response. In the RAGognize example, the response first incorrectly confirms a November release date and later gives the correct December 3, 2024 date. The incorrect confirmation raises the HeadAxis hallucination score, while it contributes little to ITI-Detect. Instead, the correct ending lowers ITI-Detect's hallucination score.

These examples show how the two representations distribute evidence differently within a response. The terminal state-based discriminator used by ITI is influenced more strongly by positions near the end of the response and by earlier positions relevant to the terminal output. HeadAxis assigns contributions directly across generated positions, allowing an earlier unsupported claim to raise the hallucination score even when it is less relevant to the final token.

6.2 Truthfulness Detection on TruthfulQA

We also evaluate whether the HeadAxis representation is useful for truthfulness detection beyond contextual hallucination. On TruthfulQA, we classify candidate answers using LLaMA-2-7B-Chat with five question-disjoint folds. Both representations use the same headwise ℓ_2 logistic probes and sparse outer readout used for contextual hallucination detection.

Representation	AUROC	AUPRC
ITI terminal complete-head output	.9140	.9024
HeadAxis response representation	**.9339**	**.9206**

Table 4: **Truthfulness detection on TruthfulQA.** Metrics are averaged over five question-disjoint test folds. The ITI row uses its terminal attention-head representation in the same response-level detection pipeline as HeadAxis.

HeadAxis improves both AUROC and AUPRC in all five folds (Table 4), with paired confidence intervals for both improvements excluding zero. This indicates that the response representation captures information relevant to truthfulness as well as contextual faithfulness, despite the two tasks having different objectives.

We also explore whether the HeadAxis representation can be used for steering, with mixed results. The experiments and results are reported in Appendix J.

7 Conclusion

We introduced HeadAxis, a response-level contextual hallucination detector built from linear probes over attention-head value representations. HeadAxis aggregates current-position value contributions across the generated response, allowing each head to capture information distributed across multiple response positions. Across four RAG hallucination benchmarks and three model families, HeadAxis performs strongly against existing white-box detectors, including both head-wise probing and response-pooling baselines.

Our analyses show that the learned signal extends beyond the setting in which it is fitted. Detectors retain predictive performance when transferred to unseen datasets, and position-level score decompositions show how unsupported spans within a response can contribute to the final hallucination score. The same response representation also improves truthfulness detection on TruthfulQA.

Ablations further show that the benefit comes from the response-level representation rather than value space alone: current-position contributions work best when pooled across the response, while other low-dimensional projections contain similar hallucination-related information. Value space provides a natural representation for decomposing attention-head outputs without requiring an additional learned projection. Together, these results show that contextual hallucination is reflected in linearly accessible internal representations that can support accurate and interpretable response-level detection.

AI USE STATEMENT

We used generative AI tools to assist with research execution, including data processing, code generation, and experiment execution. We also used generative AI tools for literature discovery and reference checking. During manuscript preparation, generative AI was used to draft and revise text at the sentence and paragraph level. AI-generated text was reviewed and edited by the authors before inclusion in the paper.

All methodological and mathematical suggestions were checked against the implemented method, and all reported numerical results were produced by our experimental pipeline. Suggested literature and bibliographic information were verified against the original sources.

We did not use generative AI tools for ideation, generating synthetic datasets or proving mathematical claims. We take responsibility for the final content of this work, including text, claims, analyses, and artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

We provide detailed data and evaluation protocols in the appendix, including dataset construction, evaluation populations, label definitions, and cross-validation procedures. The appendix also specifies the HeadAxis extraction and fitting procedure, classifier settings, and representation ablations. We release anonymized source code for feature extraction, model fitting, and reproduction of the HeadAxis results. The paper identifies the models and public datasets used in our experiments.

REFERENCES

Amos Azaria and Tom Mitchell. The internal state of an LLM knows when it's lying. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 967–976, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-emnlp.68. URL https://aclanthology.org/2023.findings-emnlp.68/.

Lennart Bürger, Fred Hamprecht, and Boaz Nadler. Truth is universal: Robust detection of lies in llms. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 138393–138431. Curran Associates, Inc., 2024. doi: 10.52202/079017-4392. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/f9f54762cbb4fe4dbffdd4f792c31221-Paper-Conference.pdf.

Sky CH-Wang, Benjamin Van Durme, Jason Eisner, and Chris Kedzie. Do androids know they're only dreaming of electric sheep? In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 4401–4420, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-acl.260. URL https://aclanthology.org/2024.findings-acl.260/.

Anwoy Chatterjee, Yash Goel, and Tanmoy Chakraborty. Hide and seek: Detecting hallucinations in language models via decoupled representations, 2025. URL https://arxiv.org/abs/2506.17748.

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, 2024. URL https://arxiv.org/abs/2402.03744.

Yung-Sung Chuang, Linlu Qiu, Cheng-Yu Hsieh, Ranjay Krishna, Yoon Kim, and James R. Glass. Lookback lens: Detecting and mitigating contextual hallucinations in large language models using only attention maps. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 1419–1436, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.84. URL https://aclanthology.org/2024.emnlp-main.84/.

Robert Friel, Masha Belyi, and Atindriyo Sanyal. Ragbench: Explainable benchmark for retrieval-augmented generation systems, 2025. URL https://arxiv.org/abs/2407.11005.

Kazuaki Furumai, Shuichiro Haruta, Kazunori Matsumoto, and Daisuke Kamisaka. Cortex: Token-level hallucination detection in rag via comparative internal representations, 2026. URL https://arxiv.org/abs/2606.31033.

Zhihao Guo, Zonghan Wu, Huan Huo, DaYong Ye, Junwei Zhang, Weiran Yao, Zhiwei Liu, Qingsong Wen, and Yilei Shao. Hallutracer: Pre-decoding truthfulness prediction via depth-averaged probe-logit, 2026. URL https://arxiv.org/abs/2608.16353.

Zheng Yi Ho, Siyuan Liang, Sen Zhang, Yibing Zhan, and Dacheng Tao. Novo: Norm voting off hallucinations with attention heads in large language models. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=yaOe2xBcLC.

Jannik Kossen, Jiatong Han, Muhammed Razzak, Lisa Schut, Shreshth Malik, and Yarin Gal. Semantic entropy probes: Robust and cheap hallucination detection in llms, 2024. URL https://arxiv.org/abs/2406.15927.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive nlp tasks, 2021. URL https://arxiv.org/abs/2005.11401.

Kenneth Li, Oam Patel, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. Inference-time intervention: Eliciting truthful answers from a language model. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 41451–41530. Curran Associates, Inc., 2023. doi: 10.52202/075280-1797. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/81b8390039b7302c909cb769f8b6cd93-Paper-Conference.pdf.

Pengqian Lu, Jie Lu, Anjin Liu, and Guangquan Zhang. TPA: Next token probability attribution for detecting hallucinations in RAG. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 25273–25292, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-390-6. doi: 10.18653/v1/2026.acl-long.1159. URL https://aclanthology.org/2026.acl-long.1159/.

Samuel Marks and Max Tegmark. The geometry of truth: Emergent linear structure in large language model representations of true/false datasets. In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=aajyHYjjsk.

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 Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 10862–10878, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.585. URL https://aclanthology.org/2024.acl-long.585/.

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

Bibek Paudel, Alexander Lyzhov, Preetam Joshi, and Puneet Anand. Hallucinot: Hallucination detection through context and common knowledge verification, 2025. URL https://arxiv.org/abs/2504.07069.

Fabian Ridder, Laurin Lessel, and Malte Schilling. Ragognizer: Hallucination-aware fine-tuning via detection head integration, 2026. URL https://arxiv.org/abs/2604.15945.

Kurt Shuster, Spencer Poff, Moya Chen, Douwe Kiela, and Jason Weston. Retrieval augmentation reduces hallucination in conversation. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2021*, pp. 3784–3803, Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.findings-emnlp.320. URL https://aclanthology.org/2021.findings-emnlp.320/.

ZhongXiang Sun, Xiaoxue Zang, Kai Zheng, Jun Xu, Xiao Zhang, Weijie Yu, Yang Song, and Han Li. RedeEP: Detecting hallucination in retrieval-augmented generation via mechanistic interpretability. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=ztzZDzgfrh.

Liyan Tang, Philippe Laban, and Greg Durrett. MiniCheck: Efficient fact-checking of LLMs on grounding documents. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 8818–8847, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.499. URL https://aclanthology.org/2024.emnlp-main.499/.

Guangzhi Xiong, Zhenghao He, Bohan Liu, Sanchit Sinha, and Aidong Zhang. Toward faithful retrieval-augmented generation with sparse autoencoders. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=hgBZP67BkP.

Zhenliang Zhang, Xinyu Hu, Huixuan Zhang, Junzhe Zhang, and Xiaojun Wan. ICR probe: Tracking hidden state dynamics for reliable hallucination detection in LLMs. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 17986–18002, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.880. URL https://aclanthology.org/2025.acl-long.880/.

A Data and Evaluation Protocol

Dataset statistics. Table 5 gives the statistics of the datasets used for response-level detection and cross-dataset transfer. The RAGognize evaluation uses the official training and test splits, including both answerable and unanswerable contexts. Qwen uses the LLaMA-generated responses on RAGTruth and RAGognize. We include RAGBench responses with a valid adherence label and nonempty response from the official splits. To limit the influence of large configurations during fitting, we retain up to 250 training and test responses from each of the 12 configurations. HDM uses the synthetic subset with contextual hallucination labels. We use the larger `synthetic` split of HDM-Bench rather than the smaller `mr` split, which contains 199 examples randomly assigned for human annotation.

Dataset	Model	Train N	Train H	Test N	Test H
RAGTruth	LLaMA-2-7B	2515	1606	450	226
RAGTruth	Mistral	2515	1702	450	251
RAGognize	LLaMA-2-7B	1842	1007	2781	1522
RAGognize	Mistral	1842	624	2781	891
RAGBench	All models	2999	547	2765	513
HDM-Context	All models	5-fold evaluation		1118	835

Table 5: Dataset statistics. H is the number of hallucinated responses.

Label and metric conventions. For RAGBench, we use the response-level `adherence_score` label and treat responses that are not fully supported by the supplied context as hallucinated. For HDM-Bench, we use `is_ctx_hallucination`, which captures contextual unsupportedness rather than overall factual error; we refer to this evaluation as HDM-Context. We report AUROC and

AUPRC, with AUPRC computed as average precision. Main-table significance marks are based on Holm-adjusted paired bootstrap comparisons against the strongest eligible baseline for each metric.

B HeadAxis Implementation

Extraction. HeadAxis pools the states at positions s through T, as defined in Section 3.2. The first included state is at the position of the first generated token, and the last is the state used to predict the final answer token.

For models with grouped-query attention, h indexes query heads rather than distinct value projections. Query heads that share a value projection can still produce different response representations because their diagonal attention weights differ across heads and positions.

Normalization and standardization. For numerical stability, the denominator of the ℓ_2 normalization is lower-bounded at 10^{-6}. Each normalized head representation is then standardized using means and scales estimated from the training data. If m_h and d_h denote these statistics, the representation supplied to the probe is

$$\tilde{q}_h = (q_h - m_h) \oslash d_h, \tag{8}$$

where $\oslash$ denotes element-wise division and scales below 10^{-6} are set to 10^{-6}.

Probe scores are standardized again before they are combined:

$$\tilde{s}_h = \frac{s_h - \mu_h}{\tau_h}, \tag{9}$$

where μ_h and τ_h are estimated from training scores.

Fitting. Each head probe is an unweighted ℓ_2-regularized logistic regression with $C = 1$, fitted with `lbfgs` for at most 1,000 iterations. To fit the final classifier, we generate three-fold grouped out-of-fold head scores, using source or question groups as appropriate. Within each fold, the representation standardizers and head probes are fitted on the remaining folds and applied to the held-out examples.

The resulting out-of-fold scores are used to estimate μ_h and τ_h and to fit the final classifier. This classifier is class-balanced logistic regression with an ℓ_1 penalty, $C = 9/N_{\text{train}}$, fitted with `liblinear` for at most 5,000 iterations. The penalty was selected once on the RAGTruth/LLaMA-2-7B training set and held fixed across all datasets and models.

For evaluation, the representation standardizers and head probes are refitted on the full training set. The score standardizer and final classifier remain those fitted from the out-of-fold training scores. No evaluation examples are used for fitting or standardization.

For HDM-Context, this procedure is repeated within each of five stratified evaluation folds with shuffle seed 0. TruthfulQA uses the separate question-disjoint fitting procedure described in Appendix I.

C Baseline Implementation and Evaluation

We use released implementations when available and otherwise reproduce the published method. Dataset-specific fitting and model selection use only training data; for HDM-Context, these steps are repeated within each evaluation fold.

C.1 ITI-Detect

ITI-Detect adapts the terminal complete pre-W_O attention-head representations of Li et al. (2023) to response classification. We fit an ℓ_2-regularized logistic probe to each head with $C = 1$ and at most 1,000 iterations. Probe fitting and head selection use a group-disjoint training/validation split with seed 0. Heads are ranked by validation accuracy, and the top 48 are retained. Their scores are standardized using training-set mean and standard deviation and averaged with equal weights. Because the probes predict non-hallucination, we negate the average to obtain a hallucination score.

C.2 REDEEP

ReDeEP (Sun et al., 2025) combines external-context similarity (ECS) with a parametric-knowledge score (PKS). We use the released chunk-level functions, a BGE-base encoder, and 256-character chunks with 20-character overlap. We use the JSD divergence as defined by the paper rather than the released divergence implementation. The LLaMA copying-head candidates come from the released list; for Mistral and Qwen, we reproduce the Appendix-B full-vocabulary trace and Gershgorin-bound selection procedure.

Using a fixed group-disjoint training/validation split with seed 0, we rank candidate head and layer features, fit min–max scales, and select the number of heads, number of layers, and $\beta \in \{0.1, 0.2, \ldots, 1.9\}$ by validation AUROC. The response score is obtained by averaging $\mathrm{PKS} - \beta\,\mathrm{ECS}$ across response chunks.

C.3 NOVO: RESPONSE-LEVEL ADAPTATION

NoVo (Ho et al., 2025) uses attention-head output norms. For response-level detection, we extract the Euclidean norm of every terminal complete pre-W_O head output. These features are standardized using training-set statistics and passed to a class-balanced ℓ_1-regularized logistic classifier with $C = 0.1$, fitted using `liblinear` with seed 0 and at most 4,000 iterations. The predicted hallucination probability is used as the response score.

C.4 TPA

TPA (Lu et al., 2026) uses the released next-token probability attribution code together with the authors' part-of-speech aggregation, producing 126 response features. We use the released classifier training procedure with a 50-trial Optuna search and five-fold cross-validation on the training data. For each of five seeds, five XGBoost models are fitted and their predicted probabilities are averaged. Reported metrics are averaged over the five resulting ensembles.

C.5 LOOKBACK LENS

Lookback Lens (Chuang et al., 2024) represents each head by the ratio between its mean attention to the supplied context and its mean attention to the generated prefix. We use the released ratio computation and temporal aggregation over the full response, followed by the released logistic-regression classifier. The classifier predicts non-hallucination, so the response hallucination score is one minus its predicted probability. We retain the prompt formatting and prediction-position convention used by the released implementation.

C.6 SAPLMA: RESPONSE-LEVEL ADAPTATION

Following Azaria & Mitchell (2023), SAPLMA uses terminal residual hidden states for classification. For a model with L layers, we evaluate candidate states at

$$\{\lfloor L/2 \rfloor, L-12, L-8, L-4, L\},$$

removing duplicates. Each candidate layer is evaluated with three MLP fits on a group-disjoint training/validation split with seed 0. The MLP has hidden widths 256, 128, and 64 with ReLU activations and is trained for five epochs with Adam, a learning rate of 10^{-3}, and batch size 64.

We select the layer with the highest mean validation AUROC and then fit three classifiers on the complete training partition. The reported score is the mean of their test metrics.

C.7 RAGLENS

RAGLens (Xiong et al., 2026) uses sparse-autoencoder features together with the released feature encoding, mutual-information feature selection, and Explainable Boosting Machine classifier. We use the released LLaMA SAE at layer 15 and model-specific SAEs at layer 16 for Mistral and layer 22 for Qwen. The SAE parameters remain fixed during detector fitting. Feature selection and classifier training use only the training partition, and response probabilities are obtained from the released prediction procedure.

C.8 POOLINGE

We reconstruct PoolingE from CH-Wang et al. (2024), as an official implementation was not available. We extract post-attention and post-MLP residual states at each response-token position, excluding prompt positions. For each sublayer, a learned-query softmax pooling operation produces a response representation that is classified by a bias-free linear probe.

Training examples are divided into group-disjoint fitting and validation sets in a 7:1 ratio with seed 42. Probes are trained with Adam, no weight decay, batch sizes in $\{20, 50, 100\}$, learning rates in $\{10^{-3}, 10^{-2}, 10^{-1}\}$, and validation-loss early stopping with patience 10. A shared hyperparameter setting is selected by mean validation F1 across sublayers, with mean validation loss used to break ties.

PoolingE combines the frozen probe probabilities using a bias-free logistic classifier trained with the same optimization settings.

C.9 TRUTHFULNESS SEPARATOR VECTOR

Truthfulness Separator Vector (TSV) (Park et al., 2025) learns an additive residual direction while keeping the model parameters fixed. We use decoder layer 9, intervention strength 5, and seed 42. From the training split, 32 labeled examples are reserved as exemplars and 128 as validation examples; the remaining examples are used by the pseudo-labeling procedure.

Both training stages run for 20 epochs with AdamW at a learning rate of 0.005, using optimal-transport pseudo-labeling and moving-average class centroids. The epoch is selected by validation AUROC. We reverse the resulting truthfulness score so that larger values indicate hallucination.

C.10 RAGOGNIZER

RAGognizer (Ridder et al., 2026) jointly trains LoRA adapters and a token-level hallucination classifier. Because the LoRA adapters modify the model representations used for detection, we report it separately from the frozen-model baselines.

We use the released `F4biian/RAGognizer-Qwen3-4B-Instruct-2507` checkpoint without target-dataset refitting. The response score is the maximum hallucination probability across assistant tokens. For RAGTruth and RAGognize, we retain the released chat formatting; RAGBench and HDM-Context use the detector's document/user formatting. RAGognizer uses its released Qwen model, LoRA adapters, and detection classifier for all four datasets.

Table 6 compares RAGognizer with HeadAxis using Qwen3-4B on the same response populations. For RAGTruth and RAGognize, both methods are evaluated on the LLaMA-generated responses used for the Qwen3-4B results in Table 1. RAGBench and HDM-Context use the shared external response sets.

	RAGognizer		HeadAxis (Qwen3-4B)	
Dataset	AUROC	AUPRC	AUROC	AUPRC
RAGTruth	.763	.734	**.856**	**.873**
RAGognize	.931	.945	**.946**	**.959**
RAGBench	.801	.374	**.868**	**.551**
HDM-Context	.904	.960	**.906**	**.963**

Table 6: **Comparison with RAGognizer.** HeadAxis uses Qwen3-4B for scoring. For RAGTruth and RAGognize, both methods are evaluated on the LLaMA-generated response populations used for the Qwen3-4B results in Table 1. RAGBench and HDM-Context use the same external response sets. Bold indicates the higher score within each comparison.

D BLACK-BOX COMPARISON

MiniCheck (Tang et al., 2024) is a fixed external verifier and requires no training on the evaluated datasets. We use `lytang/MiniCheck-Flan-T5-Large`. Each response is split into sentences, and MiniCheck estimates the support of each sentence from the supplied context. When the context is divided into multiple document chunks, we retain the highest support probability for each sentence. The response-level hallucination score is

$$s_{\text{MiniCheck}} = 1 - \min_t p_{\text{support},t}, \tag{10}$$

so a response receives a high hallucination score when at least one sentence has low support from the context.

Dataset	Response population	N	H	AUROC	AUPRC
RAGTruth	LLaMA-2	450	226	.7675	.7408
RAGTruth	Mistral	450	251	.8223	.8502
RAGognize	LLaMA-2	2,781	1,522	.6594	.6844
RAGognize	Mistral	2,781	891	.6351	.4521
RAGBench	Shared responses	2765	513	.6411	.2316
HDM-Context	Shared responses	1,118	835	.8573	.9389

Table 7: **MiniCheck response-level hallucination detection.** H denotes responses labeled as hallucinated. Scores are lower than HeadAxis.

E STATISTICAL SIGNIFICANCE

We compare HeadAxis with the numerical second-best method for each metric in Table 1. Tests use 9,999 paired source- or question-cluster bootstrap resamples, with Holm correction across all 24 dataset–model–metric comparisons.

The largest and most consistent differences occur on RAGBench, where HeadAxis improves over the second-best method for both metrics with all three models. On RAGognize, the differences are statistically resolved for LLaMA-2 and Qwen3, while the Mistral comparison is smaller. The differences on RAGTruth and HDM-Context are not statistically resolved after correction.

F COMPUTATIONAL ANALYSIS

Wall-clock runtime is difficult to compare directly across methods because implementations differ in hardware use, batching, caching, and preprocessing. We therefore summarize the main computation required to score a response and to fit each detector. A model pass denotes one teacher-forced pass over the context and response.

HeadAxis, ITI-Detect, NoVo, SAPLMA, and Lookback Lens obtain the required model quantities from a single pass over the context and response. HeadAxis additionally fits one linear probe per head and a sparse final classifier, but does not require an auxiliary neural model or separately trained representation. ReDeEP, RAGLens, PoolingE, TPA, and TSV require additional preprocessing, model-specific preparation, or more extensive fitting.

G REPRESENTATION ABLATIONS

We study several choices in the HeadAxis representation on RAGTruth/Mistral. Unless stated otherwise, the comparisons use the same response labels, normalization, headwise logistic probes, grouped out-of-fold fitting, and final sparse classifier as the main detector. The HeadAxis configuration was fixed before these ablations were run. They were conducted after the main experiments, are reported on the RAGTruth/Mistral test set for illustration, and were not used for any design or hyperparameter decisions.

		AUROC			AUPRC		
Dataset	Model	Second-best	Δ	Holm p	Second-best	Δ	Holm p
RAGTruth	LLaMA-2	RAGLens	+.0129	1.0000	RAGLens	+.0042	1.0000
RAGTruth	Mistral	PoolingE	+.0187	1.0000	Lookback Lens	+.0129	1.0000
RAGTruth	Qwen3	PoolingE	+.0059	1.0000	RAGLens	+.0127	1.0000
RAGognize	LLaMA-2	RAGLens	**+.0170**	$\mathbf{.0053}^{**}$	RAGLens	**+.0215**	$\mathbf{.0004}^{***}$
RAGognize	Mistral	SAPLMA	+.0097	.8630	SAPLMA	+.0253	.0532
RAGognize	Qwen3	RAGLens	**+.0146**	$< \mathbf{.0001}^{***}$	RAGLens	**+.0109**	$< \mathbf{.0001}^{***}$
RAGBench	LLaMA-2	ITI-Detect	**+.0144**	$\mathbf{.0009}^{***}$	Lookback Lens	**+.0435**	$< \mathbf{.0001}^{***}$
RAGBench	Mistral	ITI-Detect	**+.0112**	$\mathbf{.0142}^{*}$	ITI-Detect	**+.0379**	$< \mathbf{.0001}^{***}$
RAGBench	Qwen3	ITI-Detect	**+.0143**	$\mathbf{.0004}^{***}$	ITI-Detect	**+.0375**	$\mathbf{.0002}^{***}$
HDM-Context	LLaMA-2	RAGLens	+.0009	1.0000	RAGLens	-.0005	1.0000
HDM-Context	Mistral	ITI-Detect	+.0055	1.0000	ITI-Detect	+.0013	1.0000
HDM-Context	Qwen3	ITI-Detect	+.0016	1.0000	ITI-Detect	+.0005	1.0000

Table 8: **Paired tests against the numerical second-best method in each main-table metric column.** Δ is HeadAxis minus the named method. Tests use 9,999 paired source- or question-cluster bootstrap resamples and two-sided Wald tests based on the bootstrap standard error. Holm correction is applied across all 24 dataset–model–metric comparisons. Bold indicates comparisons that remain significant after correction; *, **, and *** denote adjusted $p < .05$, $p < .01$, and $p < .001$.

Method	Response scoring	Fitting and preparation
HeadAxis	One model pass; current-position values and attention weights	One ℓ_2 probe per head and a sparse ℓ_1 classifier
ITI-Detect	One model pass; terminal attention-head outputs	One probe per head; validation selects 48 heads
NoVo	One model pass; terminal head-output norms	One ℓ_1 logistic classifier
SAPLMA	One model pass; terminal residual states	Layer selection and three MLP classifiers
Lookback Lens	One model pass; attention maps across the response	One logistic classifier
ReDeEP	One model pass plus BGE encoding of response chunks	Head/layer selection, β tuning, and copying-head identification
RAGLens	One model pass followed by SAE encoding	Feature selection and EBM classifier; model-specific SAE required
PoolingE	One model pass; post-attention and post-MLP states	Pooling probes across sublayers and a final logistic classifier
TPA	Tokenwise next-token probability attribution	POS aggregation, hyperparameter search, and XGBoost ensembles
TSV	Model inference with the learned residual intervention	Two 20-epoch training stages with pseudo-labeling

Table 9: **Computational requirements of the evaluated white-box detectors.** The table summarizes the main computation required to score a response and the dataset-specific fitting or preparation required by each method.

G.1 Temporal Weighting

HeadAxis weights each current value by its diagonal attention coefficient before pooling. In Table 10, we compare this with uniform pooling and apply the same weighting choices to complete head outputs. Because the pooled vector is subsequently normalized, mean and sum pooling differ only by a positive scalar and are equivalent here.

Diagonal attention weighting improves both representations, with a larger improvement for current values. The response-pooled current-value representation with diagonal weighting and an ℓ_1 final classifier gives the strongest result in this comparison.

Representation	Temporal weight	Final penalty	AUROC	AUPRC
Current value	Uniform	ℓ_1	.8826	.8879
Current value	Diagonal attention	ℓ_1	**.9061**	**.9071**
Current value	Uniform	ℓ_2	.8755	.8785
Current value	Diagonal attention	ℓ_2	.8986	.8960
Complete output	Uniform	ℓ_1	.8792	.8908
Complete output	Diagonal attention	ℓ_1	.8864	.8960
Complete output	Uniform	ℓ_2	.8706	.8848
Complete output	Diagonal attention	ℓ_2	.8838	.8892

Table 10: **Temporal-weighting comparison on RAGTruth/Mistral.** Diagonal attention weighting is applied before pooling.

G.2 Role of Diagonal Attention Weights

We examine whether the diagonal attention weights alone account for the performance of the HeadAxis representation. We evaluate on the same RAGTruth/Mistral split used for the main experiments.

We first compare the full α-weighted value representation with two α-only representations and with value vectors pooled uniformly by replacing $\alpha_{t,t}$ with 1. We then test whether the information in the attention weights can be recovered by combining α-only scores with scores from the uniformly pooled value representation in a single sparse classifier.

Representation	AUROC	AUPRC
α only: mean per head	.8493	.8441
α only: mean, spread, and maximum	.8582	.8566
Uniform value vectors	.8826	.8879
Uniform values + α-only scores	.8827	.8881
Uniform values + α-only scores, tuned	.8846	.8899
α-weighted value vectors	**.9061**	**.9071**

Table 11: **Controls for diagonal attention weighting on RAGTruth/Mistral.** Uniform values replace each diagonal attention coefficient with 1 before pooling. The additive controls combine scores from the uniform-value and α-only representations in a single sparse classifier; the tuned variant selects its regularization strength using training-only validation data.

Attention weights alone are predictive, but they do not account for the full result. The stronger α-only representation reaches .8582 AUROC and .8566 AUPRC, while uniformly pooled value vectors reach .8826 and .8879. Applying the diagonal weights before pooling increases performance further to .9061 AUROC and .9071 AUPRC.

Combining α-only scores with uniformly pooled value scores does not recover this improvement: the additive classifier reaches .8827/.8881, or .8846/.8899 after selecting its regularization strength on training-only validation data. In the fixed-penalty fit, the sparse classifier assigns zero coefficients to the α-only features. These results suggest that weighting the value vectors before pooling captures information that is not reproduced by adding separate attention-weight and value-based scores.

G.3 Normalization and Prompt Inclusion

Representation	AUROC	AUPRC
Response current values, raw	.8084	.8844
Response current values, normalized	**.8150**	**.8905**
Question plus response, normalized	.8063	.8856
Question, context and response, normalized	.7425	.8439

Table 12: **Normalization and prompt-inclusion controls.** Results are reported on a held-out portion of the RAGTruth/LLaMA-2 training set rather than the official test set.

We next examine vector normalization and the inclusion of prompt information in Table 12. Normalization gives a modest improvement on this development split. Adding the question does not improve over the response-only representation, while including the full prompt substantially reduces performance. Normalization removes the overall magnitude of the pooled vector, although the magnitudes of individual contributions can still affect the direction of the sum.

G.4 Alternative Projection Spaces

We compare the value projection used by HeadAxis with query and key projections, a fixed random Gaussian projection, PCA, and the full residual state. All representations use the same attention weighting, response pooling, normalization, standardization, headwise probes, and final classifier. The PCA projection is fitted on the training data only, while the random projection is fixed with seed 0.

Representation	AUROC	AUPRC
Query	.9011	.9037
Key	.9052	.9047
Value	**.9061**	.9071
Random Gaussian	.8981	.9033
PCA	.9059	**.9132**
Full residual	.8853	.8902

Table 13: **Alternative projection spaces on RAGTruth/Mistral.** All representations use the same response pooling and fitting procedure as HeadAxis. Bold indicates the highest score for each metric.

Query, key, value, and PCA representations perform similarly, while the random projection is slightly weaker and the full residual state performs worse. These results indicate that the hallucination signal is not specific to the value projection; we use the value projection because it arises naturally from the attention decomposition and does not require learning an additional projection.

G.5 Matched Detection Pipeline

HeadAxis differs from ITI-Detect in both the representation supplied to each headwise probe and the way the resulting probe scores are combined. To isolate the representation, we apply the same normalization, headwise probes, and grouped out-of-fold sparse classifier used by HeadAxis to ITI's terminal complete pre-W_O head outputs.

For this matched comparison, we use the full available RAGBench training set of 70,799 eligible responses and the 11,155-response full-context test set, with inputs truncated to at most 4,096 tokens. For HDM, we use the benchmark's overall hallucination label (499 positive responses out of 1,118) with five-fold cross validation, providing a broader test of the representation comparison than the contextual-unsupportedness target used in Table 1. These RAGBench and HDM results are therefore not directly comparable to the corresponding rows in the main table.

Table 14 shows that the response-pooled representation achieves higher AUPRC in all 12 settings and higher AUROC in 11 of 12, with the remaining AUROC tied on RAGBench/Qwen3. The largest differences appear on RAGTruth and RAGognize, while the two representations are closer on RAGBench and HDM-Overall. These results show that the response-level representation provides gains even when the fitting procedure is held fixed.

G.6 Fixed-Answer Context Ablation

Finally, we test whether a current-value representation responds to changes in the supporting context while holding the generated response fixed. We select 200 faithful RAGognize/Mistral responses and remove either a supporting sentence or a low-overlap irrelevant sentence from the context before teacher-forcing the same response.

Removing supporting evidence lowers the faithful-oriented score substantially more than removing irrelevant context. This experiment uses a claim-level current-value projection rather than the fi-

Dataset	Model	HeadAxis AUROC	HeadAxis AUPRC	Matched terminal AUROC	Matched terminal AUPRC
RAGTruth	LLaMA-2	**.8405**	**.8504**	.8281	.8236
RAGTruth	Mistral	**.9061**	**.9071**	.8909	.8985
RAGTruth	Qwen3	**.8559**	**.8726**	.8483	.8605
RAGognize	LLaMA-2	**.8693**	**.8979**	.8373	.8645
RAGognize	Mistral	**.8900**	**.8254**	.8711	.7939
RAGognize	Qwen3	**.9459**	**.9594**	.9389	.9547
RAGBench	LLaMA-2	**.8444**	**.5050**	.8354	.4766
RAGBench	Mistral	**.8470**	**.5139**	.8384	.4822
RAGBench	Qwen3	.8891	**.5853**	**.8891**	.5794
HDM-Overall	LLaMA-2	**.7766**	**.7423**	.7531	.7120
HDM-Overall	Mistral	**.7918**	**.7590**	.7774	.7388
HDM-Overall	Qwen3	**.8371**	**.8214**	.8362	.8132

Table 14: **Matched representation comparison.** HeadAxis and the terminal representation use the same normalization, headwise probes, grouped out-of-fold sparse classifier, training folds, labels, and held-out responses within each setting. Bold indicates the higher score within each comparison.

Context modification	Mean faithful-oriented claim score
Remove supporting sentence	$-.7587$
Remove irrelevant sentence	$.1029$
Additional decrease after support removal	$.8616$
95% paired bootstrap interval	$[.8052, .9183]$
Paired bootstrap p	$< .001$

Table 15: **Fixed-answer context ablation on 200 faithful RAGognize/Mistral responses.** The response is held fixed while supporting or irrelevant context is removed. The irrelevant condition averages three separate deletions per response.

nal response-level detector, so it shows sensitivity to changes in the supplied evidence rather than response-level detection performance.

H Qualitative Analysis of the Response Detector

H.1 Layerwise Scores and Head Selection

Figure 4 shows how hallucination-related probe scores are distributed across layers for the RAGTruth/Mistral detector. The left panel accumulates probe probabilities across layers with equal head weights; this is the probe's centered sigmoid score $\sigma - 0.5$. The lower left panel shows the class separation at each layer. The right panel marks the heads assigned nonzero coefficients by the final sparse classifier. These plots describe where the probes distinguish hallucinated and faithful responses; they do not identify where hallucination originates.

Hallucination evidence accumulates smoothly when moving through the model layers; however, the middle layers have the largest separation and are therefore those selected by the sparse regression.

H.2 Attribution Within a Response

Because the final detector is linear in the standardized head scores, its response score can be decomposed into contributions from individual positions.

HeadAxis. Given

$$r_h = \frac{1}{n}\sum_{t=s}^{T} \alpha_{t,h,t} v_{t,h}, \qquad D_h = \max(\|r_h\|_2, \epsilon), \tag{11}$$

Let (m_h, d_h) denote the standardization statistics for the response representation, (w_h, b_h) the headwise probe parameters, (μ_h, τ_h) the standardization statistics for its score, and β_h the coefficient

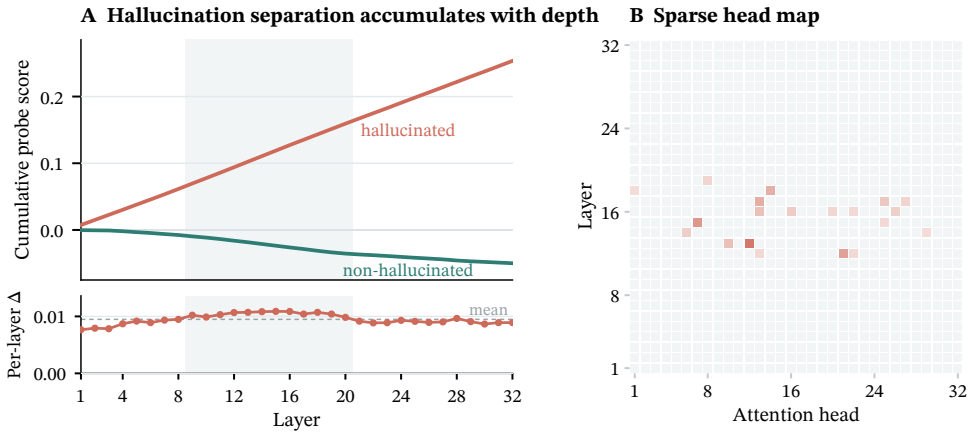


Figure 4: Layerwise probe scores and selected heads for the RAGTruth/Mistral detector over all 450 test responses. The left panel accumulates probe scores with equal head weights, while the right panel shows the heads retained by the final sparse classifier.

assigned to that head by the final classifier. Define

$$\theta_h = \frac{\beta_h}{\tau_h}\left(w_h \oslash d_h\right), \tag{12}$$

and

$$B_{\mathrm{HA}} = b + \sum_h \frac{\beta_h}{\tau_h}\left[b_h - \mu_h - w_h^\top (m_h \oslash d_h)\right]. \tag{13}$$

The response logit can then be written as

$$H_{\mathrm{HA}} = B_{\mathrm{HA}} + \sum_{t=s}^{T} c_t^{\mathrm{HA}}, \tag{14}$$

where

$$c_t^{\mathrm{HA}} = \frac{1}{n}\sum_h \frac{\theta_h^\top\left(\alpha_{t,h,t} v_{t,h}\right)}{D_h}. \tag{15}$$

Each c_t^{HA} is assigned to the token at position t, whose value vector it contains. Because D_h depends on the pooled representation, these contributions decompose the score of the completed response rather than defining an online token-level detector.

ITI-Detect. Let $\mathcal{K}$ denote the K selected heads. For each head $h \in \mathcal{K}$, let (u_h, c_h) denote the parameters of the probe predicting non-hallucination and $(\mu_h^{\mathrm{I}}, \tau_h^{\mathrm{I}})$ its score-standardization statistics. Since ITI-Detect averages the standardized probe scores and negates the result, define

$$\gamma_h = -\frac{u_h}{K\tau_h^{\mathrm{I}}}, \tag{16}$$

and

$$B_{\mathrm{I}} = -\frac{1}{K}\sum_{h\in\mathcal{K}} \frac{c_h - \mu_h^{\mathrm{I}}}{\tau_h^{\mathrm{I}}}. \tag{17}$$

Let T_{term} denote the terminal state after the final answer token. Its complete head output is

$$o_{T_{\mathrm{term}},h} = \sum_{j=1}^{T_{\mathrm{term}}} \alpha_{T_{\mathrm{term}},h,j} v_{j,h}. \tag{18}$$

The ITI-Detect score therefore decomposes as

$$H_{\mathrm{I}} = B_{\mathrm{I}} + \sum_{j=1}^{T_{\mathrm{term}}} c_j^{\mathrm{I}}, \tag{19}$$

Dataset	Representation	Mixed N	AUROC	AUPRC	Enrich@K	Span hit@K
RAGTruth	HeadAxis	251	**.7385**	**.3509**	**3.8721**	**.7961**
	Matched terminal	251	.6675	.2532	2.2968	.6278
RAGognize	HeadAxis	577	**.6091**	**.4545**	**1.6742**	**.9105**
	Matched terminal	577	.6036	.4464	1.3946	.8556

Table 16: **Localization from response-detector contributions on Mistral-7B.** Metrics are averaged across responses containing both annotated token classes. For each response, K is the number of annotated hallucination tokens. Enrich@K is precision@K divided by the prevalence of hallucination tokens, and Span hit@K is the fraction of annotated hallucination spans containing at least one of the top-K positions. No token-level classifier is fitted.

where

$$c_j^{\mathrm{I}} = \sum_{h \in \mathcal{K}} \gamma_h^\top \left(\alpha_{T_{\mathrm{term}},h,j} v_{j,h} \right). \tag{20}$$

Unlike HeadAxis, ITI-Detect includes contributions from values throughout the prompt and response; these contributions are omitted in order to enable a direct comparison.

> Lauren Hill, an 19-year-old woman from Indiana who inspired millions with her fight against brain cancer and her dream of playing college basketball, passed away on Friday. She had been diagnosed with diffuse intrinsic pontine glioma in 2013 and continued to play despite undergoing chemotherapy treatment. Her story went viral when Mount St. Joseph University successfully petitioned the NCAA to move up the opening game of its schedule so she could play. During her brief career, Hill scored 94 points in four games. She also raised $1.4 million for pediatric cancer research through the nonprofit group The Cure Starts Now. Her legacy will live on as a symbol of perseverance and courage.

Figure 5: Additional illustrative attribution in a RAGTruth summary. The source supports an age of 19 and fundraising of $1.4 million, but does not support the generated claim of 94 points in four games. Highlighting shows contributions to the completed response score rather than predictions from a separately fitted token classifier.

In Figure 5, the same digit appears in an unsupported scoring claim and a supported fundraising amount, but receives different contributions to the response score. This illustrates how the detector can distinguish identical surface tokens in different contexts.

The attribution is not always aligned with annotated hallucination spans. Faithful responses can contain positions that increase the hallucination score, and unsupported content can sometimes decrease it. The examples therefore illustrate how the detector distributes its score across a response rather than measuring localization accuracy.

H.3 Token-Level Evaluation from Response-Score Decomposition

We evaluate whether the position-level contributions of the response detector align with annotated hallucination spans. No token-level classifier is trained; the response-level detectors remain fixed, and token labels are used only for evaluation. We compare HeadAxis with the matched terminal representation using the score decompositions described in Appendix H.2.

On RAGTruth, HeadAxis shows substantially stronger localization than the matched terminal representation across all four metrics, including a higher AUPRC (.3509 versus .2532), enrichment (3.8721 versus 2.2968), and span hit rate (.7961 versus .6278). On RAGognize, the differences are smaller, although HeadAxis remains higher on each metric. These results show that the response-score decomposition can recover some token-level structure without token-level training, particularly on RAGTruth.

I TruthfulQA Detection Protocol

Table 17 summarizes the data used for TruthfulQA detection and steering. Detection evaluates the provided candidate answers, while steering evaluates newly generated answers to the same questions. The experiments use different LLaMA checkpoints.

	Detection	Steering
Model	LLaMA-2-7B-Chat	LLaMA-7B
Questions	817	817
Provided candidate answers	5,882	–
Truthful candidates	2,585	–
Untruthful candidates	3,297	–
Generated answers per condition	–	817
Question-disjoint evaluation folds	5	2

Table 17: TruthfulQA data statistics. Steering truthfulness and informativeness are judged on generated answers; they are not the candidate-answer labels used for detection.

The TruthfulQA detection comparison uses LLaMA-2-7B-Chat and 5,882 candidate answers from 817 questions. Five shuffled question-disjoint outer folds use seed 42. Within each development fold, 20% of questions are reserved for regularization selection. Headwise class-balanced L2 probes use $C = 1$ and training-only coordinate standardization. Three-fold question-grouped out-of-fold head scores train the outer classifier; probes used for validation and test are refitted on the fitting subset, excluding validation and test questions. The balanced outer L1 classifier selects C from $\{.25, 1, 4, 9, 16, 64, 256\}/N_{\text{fit}}$ by validation AUROC, breaking exact ties in favor of fewer nonzero heads.

ITI uses raw terminal complete outputs, while HeadAxis uses normalized pooled current values. Both use the same fitting procedure, so the comparison includes the differences in pooling and normalization. Truthful candidates are positive. Metrics are averaged over folds rather than computed from pooled predictions. Paired question-bootstrap intervals for both improvements exclude zero. They condition on the fitted models and do not account for variability from refitting.

J TruthfulQA Steering Protocol and Full Results

The steering experiment uses the original LLaMA-7B checkpoint and the 817-question TruthfulQA validation set, using the same model and question-level two-fold design as ITI. For each held-out fold, the other fold is divided 4:1 for probe fitting and head selection. Both methods fit one logistic probe per head, retain the 48 heads with highest validation accuracy, orient class-mean directions toward truthful answers, and intervene greedily at every generated answer step for at most 50 tokens. Thus no evaluated question is used to estimate its controller. HeadAxis and ITI use their own projected-standard-deviation calibration distributions, so an equal numeric dose is not an equal activation or residual-write norm.

ITI fits terminal complete pre-W_O head outputs and adds its shift to the complete head output. Its multiplier is estimated from the upstream protocol's separate end-of-question distribution, in which each TruthfulQA answer is followed by a randomly sampled question and the complete head output is read at the new question-ending position. HeadAxis fits unit-normalized response pools of diagonal current-value contributions. During generation it applies the resulting headwise shift to each newly computed value at response positions, so the changed value remains in the key–value cache. Prompt values are unchanged. This experiment therefore compares intervention methods, not an isolated representation comparison.

The original Curie-based truth and information judges used by Li et al. (2023) are no longer available. We score all conditions blindly with one fixed Qwen2.5-32B replacement judge using the original binary definitions. Absolute judge rates are consequently not a replication of ITI's published rates, although comparisons within this experiment use the same evaluator. MC1 and MC2 use the standard TruthfulQA candidate likelihoods. Table 18 reports every evaluated dose. The se-

lected operating points maximize judged truthfulness within each displayed curve and are therefore descriptive rather than independently confirmed hyperparameters.

Condition	Truth	Info	Both	MC1	MC2
Unmodified	28.40	51.41	23.26	25.46	40.18
ITI 5	30.23	54.59	26.07	26.56	41.43
ITI 10	29.87	53.49	25.34	27.91	42.84
ITI 15	31.46	51.65	25.83	27.78	44.34
ITI 20	24.60	38.68	17.63	28.76	45.81
ITI 25	14.71	17.77	8.70	28.64	46.83
ITI 30	2.33	3.30	1.47	28.76	48.24
ITI 40	0.00	0.00	0.00	27.54	50.17
HeadAxis 5	29.50	51.16	24.11	25.70	40.48
HeadAxis 10	31.09	52.26	25.21	25.95	40.84
HeadAxis 15	32.80	53.24	26.56	26.07	41.23
HeadAxis 20	35.13	53.00	27.54	26.07	41.60
HeadAxis 25	34.88	52.14	27.05	26.19	41.95
HeadAxis 30	37.21	52.26	27.42	26.81	42.34
HeadAxis 40	37.09	46.27	25.46	28.03	43.36
HeadAxis 50	36.47	39.66	21.42	28.64	44.46
HeadAxis 60	32.31	30.60	15.67	29.13	45.73
HeadAxis 80	19.09	11.26	6.00	30.11	47.93

Table 18: Complete TruthfulQA steering sweep on 817 questions. Truth, Info, Both, MC1, and MC2 are percentages. One ITI-25 judge response was malformed; that row uses 816 valid judge outputs, while all selected-condition comparisons use all 817 paired questions.

At the selected operating points, HeadAxis improves judged truthfulness over ITI in one fold but not the other. The operating points are also selected from the evaluated curves, so the aggregate comparison is exploratory. OpenWebText diagnostics used interventions over complete sequences, which differs from the response-only generation protocol. We do not use them to draw conclusions about generation quality in this experiment.

J.1 Exploratory Truthfulness Steering

Detection and steering are different objectives. An axis that separates truthful from untruthful responses is not necessarily useful as an intervention direction. We nevertheless investigate whether the HeadAxis representation can be adapted for steering in the TruthfulQA setting studied by ITI.

Following ITI, we evaluate free generation with LLaMA-7B. ITI estimates class-mean shifts from terminal complete-head outputs, while the HeadAxis intervention estimates them from response-level current-value representations and applies the resulting shifts during generation.

All generations are evaluated blindly by Qwen2.5-32B-Instruct, replacing the original ITI judge, using the TruthfulQA definitions of truthfulness and informativeness. The same judge and evaluation prompts are used for all methods.

HeadAxis produces more answers judged truthful than ITI at their selected operating points. These points are selected from the evaluated sweeps, and the improvement is not consistent across folds. Informativeness is similar between the two methods, while the difference in jointly truthful-and-informative answers is not significant. HeadAxis also does not improve every TruthfulQA metric, including the likelihood-based multiple-choice scores. We therefore treat this as evidence that the representation can support truthfulness steering, rather than as a general improvement over ITI.

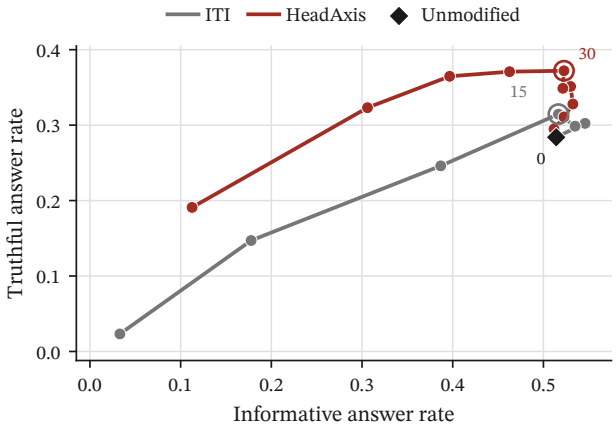


Figure 6: **TruthfulQA free-generation steering.** Truthful and informative answer rates for unmodified LLaMA-7B and dose sweeps of ITI and the HeadAxis-derived intervention. Rings mark each method's highest-truthfulness operating point.

Method	Truth ↑	Info ↑	Truth ∧ Info ↑
Unmodified	28.40	51.41	23.26
ITI	31.46	51.65	25.83
HeadAxis	**37.21**	**52.26**	**27.42**

Table 19: **TruthfulQA steering results.** Values are percentages at each method's highest-truthfulness operating point, selected post hoc from the dose sweeps in Table 18, Appendix J.

K Contextual Hallucination Steering

We evaluate whether the representations learned for detection can also be used to steer generation toward greater contextual faithfulness. The experiments use LLaMA-2-7B with directions fitted separately on RAGTruth and RAGognize. Heads are selected using validation probe performance, and intervention strength is scaled by the projected standard deviation. ITI modifies complete head outputs, while HeadAxis modifies current values that remain in the key–value cache.

K.1 Class-Mean Steering

We first fit class-mean intervention directions from contextual hallucination labels and evaluate them on 128 examples per dataset.

On RAGTruth, both interventions improve over the unmodified model, with ITI achieving the highest scores. On RAGognize, the differences are smaller and neither method shows a clear advantage over the unmodified model. Overall, the detection representation does not directly yield a consistently stronger steering direction.

K.2 Alternative Intervention Directions

We next replace the class-mean direction with the gradient of the response detector. The intervention strength is selected on development prompts and then fixed for evaluation on the remaining test prompts. The improvement seen during development does not carry over to the test set. Modifying all values or using mass-mean directions gives similarly inconsistent results.

K.3 HeadAxis for Head Selection

Finally, we test whether the heads identified by HeadAxis are useful even when the intervention direction itself is unchanged. We use either ITI or HeadAxis to select the heads, but in both cases apply the same dataset- and model-specific ITI class-mean directions to complete head outputs. Intervention strengths are selected on development examples and then fixed for evaluation.

Dataset	Method	Entailment	Informativeness	Utility
RAGTruth	Unmodified	.7445	.5945	.4709
	ITI	**.7789**	**.6344**	**.5247**
	HeadAxis	.7648	.6234	.5064
RAGognize	Unmodified	.6875	.7063	.6296
	ITI	**.7133**	**.7563**	**.6724**
	HeadAxis	.7094	.7250	.6671

Table 20: **Exploratory contextual hallucination steering.** Both methods use class-mean directions fitted on contextual hallucination labels. Results use 128 LLaMA-2-7B examples per dataset.

Model / dataset	Method	Entailment	Completeness	Quality
LLaMA-2 / RAGTruth	Unmodified	3.5400	3.4133	4.5252
	ITI	3.5600	3.3822	4.5222
	HeadAxis-selected ITI	3.5600	3.3711	4.5119
LLaMA-2 / RAGognize	Unmodified	3.8275	3.7922	4.3832
	ITI	3.7398	3.6772	4.3034
	HeadAxis-selected ITI	4.0137	3.9511	4.4989
Mistral / RAGTruth	Unmodified	3.6022	3.4267	4.5393
	ITI	3.5844	3.4089	4.5267
	HeadAxis-selected ITI	3.5933	3.4489	4.5452
Mistral / RAGognize	Unmodified	4.1014	4.0446	4.6180
	ITI	4.0697	4.0043	4.6048
	HeadAxis-selected ITI	4.1042	4.0410	4.6132

Table 21: HeadAxis head selection with ITI intervention directions. RAGTruth uses 450 test examples and RAGognize 1,391 per model. Quality averages coherence, fluency, and relevance.

Using HeadAxis to select heads gives the clearest improvement for LLaMA-2 on RAGognize. The other settings show little difference between the two selection methods. This suggests that the heads identified by HeadAxis can be useful for steering in some cases, but the effect is not consistent across models and datasets.