

Gradient-Stable Attention Heads Signal LLM Correctness

Sophie Ostmeier^{1,†}, Brian Axelrod², Maya Varma³, Asad Aali⁴, Yabin Zhang⁴,
Magdalini Paschali⁴, Sanmi Koyejo³, Curtis Langlotz^{4,*}, Akshay Chaudhari^{4,*}

¹ETH AI Center, ETH Zürich, Zurich, Switzerland, ²Microsoft AI,

³Department of Computer Science, Stanford University, Stanford, CA 94305, USA,

⁴Department of Radiology, Stanford University, Stanford, CA 94305, USA

[†]Work done while at Stanford University. ^{*}Equal contribution.

Correspondence: sophie.ostmeier@ai.ethz.ch

Abstract

Large language models (LLMs) often generate plausible yet incorrect answers, posing risks in safety-critical settings such as medicine. Human evaluation is expensive, and LLM-as-judge approaches risk introducing hidden errors. Recent single-pass white-box methods detect contextual hallucinations using linear probes over model internals, but their generalization to out-of-domain, out-of-task, and training-free settings remains poorly understood. We introduce HEADENTROPY, a method that predicts answer correctness by measuring how susceptible each attention head’s pattern is to modification under further gradient updates. The key insight is that the trace of the softmax Jacobian is a monotonic function of 2-Rényi entropy, linking attention spread to gradient stability during training, even without the reference annotation. With no training, HEADENTROPY reaches 0.736 AUROC, outperforms every training-free baseline in paired comparison across 5 models and 5 datasets, and matches a hidden-state probe transferred across domains, at under 1% of inference cost. We evaluate broadly across 5 instruction-tuned LLMs and 5 datasets spanning general knowledge, multi-hop reasoning, mathematical reasoning, claim verification, and medicine. The code is available at <https://github.com/StanfordMIMI/HeadEntropy>

1 Introduction

Large language models (LLMs) are increasingly being deployed in safety-critical settings, ranging from medical question answering (QA) and clinical decision support, to applications in law and scientific discovery (Singhal et al., 2022, 2025; Tu et al., 2024; Yang et al., 2024; Tanno et al., 2025; Zambrano Chaves et al., 2025). However, even state-of-the-art systems can generate fluent and confident, yet *incorrect* responses (Wang et al., 2024; Sivarajkumar et al., 2024; Chang et al., 2025).

For example:

Parent: “I have 500 mg aspirin tablets. Can I give one for a fever at a body weight of 24 kg?”

LLM: “Yes, but use half a tablet. The standard antipyretic dose is 10–15 mg/kg, so at 24 kg roughly 250 mg is appropriate.”

The dose calculation is correct, and the model even corrects the amount the parent proposed, but the answer is not: 24 kg indicates a child, and aspirin is avoided in febrile children because of its well-known link to Reye’s syndrome (Pugliese et al., 2008). Verifying the output with an expert may not be promptly feasible and is impractical at scale, while relying on Judge-LLMs for verification could introduce unpredictable biases or hidden errors unless the validation is also reviewed by experts (Chen et al., 2024b; Thakur et al., 2025).

Therefore, our work focuses on estimating whether an LLM has answered correctly using signals from *inside the model itself*, computed during the same forward pass that generates the answer. We introduce HEADENTROPY, a scalable and interpretable white-box method that uses the attention spread inside the model itself to determine correctness while generating the answer. In the transformer self-attention mechanism, each attention head computes attention weights by taking the inner product between query and key vectors—a measure of their directional alignment—and then normalizing these scores across all positions via softmax to produce a set of token weights. We find that certain attention patterns are stable during training, while others are unstable and tend to disappear through the course of training. We hypothesize that the presence of unstable attention patterns at inference time correlates with a generation not grounded in the training set—otherwise the training gradients would have modified the attention pattern.

For example: The LLM above reads the weight token sharply enough to compute a dose from it, since

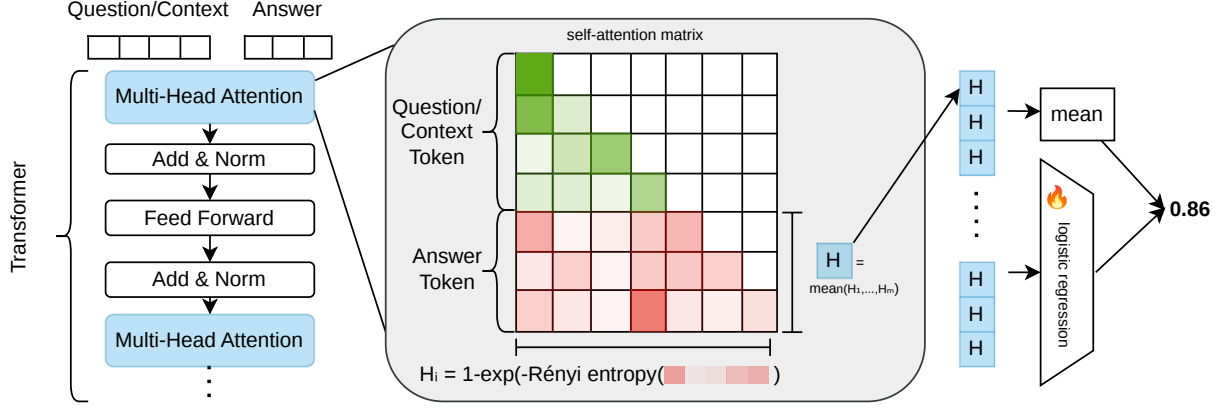


Figure 1: HEADENTROPY overview. The correctness estimate is computed directly from the spread of the attention patterns, measured as the mean trace of the softmax Jacobian.

weight-based dosing is a frequent pattern used in pediatric training data and training has settled the heads that carry it. The attention that would tie the same token to the pediatric contraindication has not settled in the same way: it remains diffuse, in the regime where the attention distribution is still sensitive to perturbation and training pressure would still move it.

Recent methods leave several questions open here. Features are selected empirically, by ablation over layers, heads, or token positions, so it is unclear when the signal should hold; transfer is reported mostly within one model family; and the features are usually paired with a trained probe. Training-free scores such as LLM-Check and the direct ICR Score exist, but have been evaluated on few models and datasets. HEADENTROPY addresses these questions in turn: its signal has a theoretically predicted direction, it requires no trained probe, and we evaluate it across tasks and model families.

HEADENTROPY makes the following contributions:

- **Attention entropy tracks gradient stability.** The trace of the softmax Jacobian equals $1 - \exp(-H_2)$, where H_2 is the 2-Rényi entropy of the attention distribution (Appendix N). Sharp (low-entropy) heads receive small gradients and remain stable, while diffuse (high-entropy) heads stay sensitive to training pressure. We investigate this relationship empirically across pre-training checkpoints (Section 3.1).
- **A correctness predictor built on this signal.** HEADENTROPY represents each generation by its per-head attention entropies. High entropy at inference time indicates attention patterns that training pressure would still change, and thus

a generation that is not grounded in the training data. This gives the feature a fixed direction: higher entropy means a higher chance of an incorrect answer. As a result, HEADENTROPY which predicts answer correctness most consistently across dataset and model families.

- **The raw signal predicts correctness even without training.** Averaged over heads with no probe at all, HEADENTROPY outperforms every competing training-free score by +8.9% to +75.6% relative ROC-AUC in paired comparisons across 5 models and 5 datasets (Figure 3). The untrained mean is statistically indistinguishable from a HEADENTROPY probe transferred across datasets (−0.2%), so under domain shift a trained probe adds nothing over the raw signal.

2 Background

Here, we provide a brief summary of how the Jacobian relates to Gradient Dynamics.

2.1 The Jacobian and Gradient Dynamics

For a vector valued function $f : \mathcal{R}^n \rightarrow \mathcal{R}^m$, the Jacobian matrix of $f(x)$ is the matrix of partial derivatives

$$\mathbf{J}_f = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial x_1} & \cdots & \frac{\partial f_m}{\partial x_n} \end{bmatrix}.$$

The Jacobian can be used as a measure of how sensitive a particular set of activations or parameters are. A "large" Jacobian indicates that an identical gradient flow will change the activations greatly.

We use this to identify stable versus unstable attention patterns, where we define "stable" attention patterns as those that do not change substantially with a gradient update and "unstable" attention pattern as those that change a large amount every time they appear in training. In this view, observing many unstable attention patterns may be an indicator that the current example is not like the training data.

There is no single definition of what makes a matrix such as a Jacobian "large". In this work, we use the trace of the Jacobian, noting that the softmax Jacobian is positive semidefinite (all non-negative eigenvalues), so its trace coincides with its nuclear norm (appendix Section N).

3 Method

3.1 Motivation

Hypothesis. A correct answer rests on associations that training has reinforced. We hypothesise that the attention patterns carrying such associations are ones training has settled: further gradient updates would barely move them. A pattern that continued training would still reshape is one training has not settled, and a generation resting on such patterns is less likely to be grounded in the training distribution. We want to tell the two apart at inference time.

Testing this directly would need the pretraining corpus: checking, for each generation, whether the associations it uses are attested in the data. That corpus is unavailable for most deployed models, and searching it once per generation would be prohibitive in any case. We need a quantity computable from the model alone.

Stability as a Jacobian trace. Each head produces a self-attention matrix, where row p applies a softmax to scores x . The Jacobian

$$J = \frac{\partial p}{\partial x} = \text{diag}(p) - pp^\top \quad (1)$$

says how a change in the scores propagates to the attention weights, and so how far an update displaces the pattern. J is positive semidefinite, so its trace sums its eigenvalues and gives a single scalar for how movable the pattern is. Large $\text{tr}(J)$: an update would readily move this head. Small: training has settled it.

The identity. The trace has a closed form in the 2-Rényi entropy $H_2(p) = -\log \sum_j p_j^2$:

$$\text{tr}(J) = 1 - \exp(-H_2(p)), \quad (2)$$

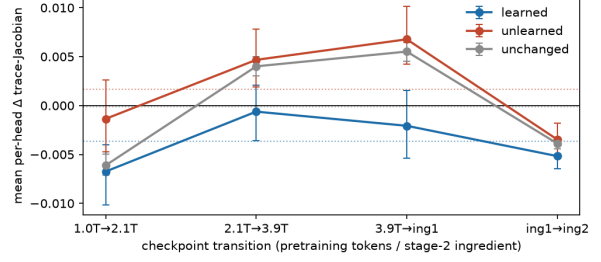


Figure 2: Mean per-head change in $\text{tr}(J)$ across four OLMo-2-1B checkpoint transitions, with examples grouped by the change in $\log P(\text{gold})$ over each transition. Error bars are 95% bootstrap intervals over examples; dotted lines mark the pooled means. Wilcoxon $p = 1.8 \times 10^{-12}$, while the unlearned and control groups track one another closely.

derived in Appendix N. Thus, a head’s gradient stability is fixed entirely by the spread of its attention. We can therefore read it off the forward pass without a backward pass, access to training data or additional generations.

Does training settle attention? If training settles the heads that carry an association, then learning an example should lower $\text{tr}(J)$ on that example, and failing to learn it should not. We test this directly on five OLMo-2-1B pretraining checkpoints. At each checkpoint we run one forward pass over each random 1000 TriviaQA prompts with its gold answer and record $\log P(\text{gold})$ together with per-head $\text{tr}(J)$. Across each of the four checkpoint transitions we rank examples by how much $\log P(\text{gold})$ changed and call the top decile learned, the bottom decile unlearned, and the middle 80% the control.

Figure 2 shows the mean change in $\text{tr}(J)$ for each group. On learned examples $\text{tr}(J)$ falls below the control at every transition (pooled -0.0035 , Wilcoxon $p = 1.8 \times 10^{-12}$), whereas unlearned examples track the control, and the more an example is learned the further its $\text{tr}(J)$ falls. Learning therefore lowers $\text{tr}(J)$ during training, and HEAD-ENTROPY reads the same quantity at inference, where a low value indicates an answer that rests on associations training has already settled.

3.2 Computing Head Entropy

HEADENTROPY encompasses two components: 1) computing the per-head attention (Eq. 9), 2.1) mean over heads (Eq. 10) or for comparison to prior methods 2.2) training a logistic regression model to find predictive patterns (Eq. 12).

Motivated by question-answering using instruction tuned LLMs, we consider an autoregressive transformer, LLM , with causal masking. The LLM has m attention heads in total, across all layers. We index heads by $k \in \{1, \dots, m\}$. Let T be the total sequence length (i.e. question and answer tokens).

Given an input sequence x (a tokenized question), the model outputs a prediction $\hat{y}$ (a tokenized answer) at nearly deterministic decoding (temperature ≈ 0):

$$\hat{y} = LLM(x). \quad (3)$$

For a given labeled question answering dataset, $\{(x_i, y_i)\}_{i=1}^N$, we define

$$z_i := \mathbb{I}[\hat{y}_i = y_i] \in \{0, 1\}, \quad (4)$$

where equality is measured with an evaluation metric (exact match and F1 score in our experiments). Each attention head k produces

$$A_k = \text{softmax}(QK^\top) \in \mathbb{R}^{T \times T}, \quad (5)$$

where each row of A_k is a probability distribution (i.e., rows sum to 1). The full tensor shape for all m heads is $A \in \mathbb{R}^{m \times T \times T}$. For each head k and token t , we compute the α -Rényi entropy ($\alpha = 2$):

$$H_k^{(t)} = -\log \sum_{j=1}^T \left(A_k^{(t,j)} \right)^2 \quad (6)$$

and then compute the trace of the Jacobian (see appendix N for the derivation).

$$\text{trace}(J)_k^{(t)} = 1 - \exp \left(-H_k^{(t)} \right) \quad (7)$$

Equation 7 will become a feature used to predict answer correctness.

We then average over the tokens of a semantic section (e.g., question, thinking, or answer) of length T_{section} :

$$\bar{H}_k = \frac{1}{T_{\text{section}}} \sum_{t=1}^{T_{\text{section}}} \text{trace}(J)_k^{(t)}, \quad (8)$$

where each entry, $\bar{H}_k$, corresponds to the HEADENTROPY of one attention head, where $k \in \{1, \dots, m\}$. For batched inference, we mask the rows and columns of the padding tokens. See empirical testing of aggregation techniques in the appendix K.

This yields a fixed-dimensional feature vector for each example i ,

$$\bar{H}_i = [\bar{H}_{1,i}, \dots, \bar{H}_{m,i}]^\top \in \mathbb{R}^m. \quad (9)$$

In addition to improving interpretability, section-level aggregation reduces memory and compute overhead compared to token-level entropy analysis. This makes large-scale or real-time analysis more practical while preserving alignment with tasks' semantic structures.

3.3 Training-free or supervised setting

In the training-free setting (sometimes in literature "unsupervised"), we compute a scalar per example by taking the mean over the fixed-dimensional feature vector in Eq.9,

$$g(\bar{H}_i) = -\frac{1}{m} \sum_{k=1}^m \bar{H}_{k,i} \in \mathbb{R} \quad (10)$$

where the negative sign orients g as a score for correctness rather than against it, and is fixed a priori by the direction predicted in Section 3.1: higher head entropy indicates a higher chance of an incorrect answer.

In the supervised setting, we use fixed-dimensional feature vector for each example i , $\bar{H}_i \in \mathbb{R}^m$ to train an ℓ_1 -regularized logistic regression classifier

$$f : \mathbb{R}^m \rightarrow [0, 1] \quad (11)$$

to estimate the probability of correctness:

$$\mathbb{P}(z_i = 1 \mid \bar{H}_i) = f(\bar{H}_i) = \sigma(\beta^\top \bar{H}_i + b), \quad (12)$$

where $\beta \in \mathbb{R}^m$ are the coefficients, b is the bias term and σ is the logistic sigmoid. Any classifier can be used here; we chose ℓ_1 -regulated logistic regression due to simplicity and use in prior work. The pseudocode can be found in Appendix M.

3.4 Computational Efficiency and Complexity

As HEADENTROPY essentially uses summary statistics of the existing attention patterns the computational footprint is minimal compared to that of LLM inference. We document the complete complexity in table 1.

Entropy features are computed using quantities already available from the softmax, adding negligible overhead to a standard forward pass. Memory remains $O(m)$ since only per-head aggregates are retained—full attention maps need not be stored. Overall, HEADENTROPY adds less than 1% to inference cost and requires no additional forward passes beyond those typically used for generation.

Table 1: Computational complexity of HEADENTROPY components. **Notation:** m = attention heads, T = sequence length, T_{section} = attention window, N = training set size.

Component	Time	Memory	Notes
LLM forward pass (reference)	$O(mT^2)$	$O(mT^2)$	Reference
Feature generation (entropy)	$O(mT \cdot T_{\text{section}})$	$O(m)$	Negligible overhead
Classification (inference)	$O(m)$	$O(m)$	Sequence-independent
Training (end-to-end)	—	$O(Nm)$	Dominated by LLM forward passes

4 Experimental Setup

Models. We evaluate three model families spanning 1.7B–32B parameters. From the reasoning-oriented Qwen3 family we use Qwen3-1.7B, Qwen3-8B, and Qwen3-32B; from the general-purpose Llama family we use Llama-3.1-8B-Instruct; and from the Gemma family we use Gemma-4-12B-it. All generations use near-greedy decoding ($T < 0.001$) at half-precision floating-point; we also test generalization to $T=1$ (Section 5, Table 10).

Datasets. We use five benchmarks spanning open-domain QA (TriviaQA, Joshi et al., 2017), multi-hop QA with long context (HotpotQA distractor setting, Yang et al., 2018), domain-specific multiple-choice QA (MedMCQA, Pal et al., 2022), fact verification (FEVER, Thorne et al., 2018), and mathematical reasoning (MATH, Hendrycks et al., 2021). For TriviaQA, HotpotQA, and MedMCQA we sample uniformly at random with a fixed seed 30k training examples with a max of 768 generated tokens. Truncated answers were excluded from the training set only. We use the official validation split for evaluation; FEVER and MATH are held out and used only for out-of-distribution evaluation (Table 2. The prompts can be found in Appendix B.3, B.2, B.1). We define binary correctness via the standard metric for each dataset: exact match for TriviaQA, HotpotQA, and MATH; label match for FEVER; option match for MedMCQA (Eq. 4). The overall accuracy of the models on the QA task is given in Table 3.

Baselines. We compare against nine single-forward-pass methods, covering both raw scalar statistics and learned linear probes: (i) token entropy (Kadavath et al., 2022); (ii) attention score, the kernel-similarity statistic from LLM-Check (Sriramanan et al., 2024) (picked layer: $0.8 \times$ total number of layers); (iii) mean hidden states, the average final-layer hidden state used as a raw signal;

(iv) mean Lookback Lens ratio, the average context-vs-generation attention ratio (Chuang et al., 2024); (v) verbalized Uncertainty (see Appendix B.1, B.2, B.3); (vi) mean ICR score, the information-flow statistic of Zhang et al. (2025); (vii) Hidden State LR, an ℓ_1 -regularized logistic regression on final-layer hidden states, following Xie et al. (2024); (viii) Lookback Lens LR, the same probe trained on per-head Lookback Lens features as in (Chuang et al., 2024); (ix) ICR LR, the same probe trained on the per-layer ICR scores. Baselines (i)–(vi) are training-free; (vii)–(ix) use the same logistic regression probe setup as HEADENTROPY. Multi-pass methods such as Semantic Uncertainty (Kuhn et al., 2023) and INSIDE (Chen et al., 2024a) are not directly comparable since they require additional generations or auxiliary models; we report their published TriviaQA AUROC for context in the appendix Table 6.

Metrics. We measure how well each score separates correct from incorrect model answers with AUROC. Scores are reported with their intended orientation, if known from literature, i.e. a higher ratio for Lookback lens indicates an answer being more grounded in the context.

5 Results

With no training at all, the mean of HEADENTROPY over heads reaches 0.736 AUROC across 5 models and 5 datasets, beats every training-free baseline in paired comparison (Figure 3), and matches a hidden-state probe transferred across domains (0.739). We then ask what supervision adds to this raw signal, reporting results as a ladder of increasing supervision: (i) the raw signal with no probe (Section 5.1); (ii) what a trained probe adds over that raw signal, in-domain and under domain shift (Section 5.2); (iii) where a probe is available, how HEADENTROPY compares against probes on other features (Section 5.3). The short answer: a probe helps in-domain and adds nothing under domain shift.

5.1 Training-free correctness prediction

A training-free signal requires no labeled data, no per-domain retraining, and no fitting procedure that could overfit to a particular distribution, properties especially relevant in safety-critical deployments. We compare the mean of per-head HEADENTROPY values (Eq. 10) against six training-free scores: token entropy, attention score (LLM-Check), verbal-

	TriviaQA	HotpotQA	MedMCQA	FEVER	MATH
# Training Examples	Sampled 30k / 130k	Sampled 30k / 98k	Sampled 30k / 183k	-	-
# Validation Examples	17,944	7,405	4,183	20,000	5,000
Task	Open-domain QA	Multi-hop QA	Multiple-choice QA	Fact verification	Mathematical reasoning
Domain	General	General	Biomedical	General	Mathematics
Answer location	Answer not in context	Answer in context	Answer in context, but not all knowledge	Claim verified against evidence	Answer derived via reasoning

Table 2: Datasets used in experiments, with statistics on training/validation splits and model-specific answer lengths.

Model	TriviaQA				HotpotQA				MedMCQA			
	Acc.	Q	Think	Ans	Acc.	Q	Think	Ans	Acc.	Q	Think	Ans
Qwen3-1.7B	0.33	26	310	7	0.38	1243	316	19	0.29	47	519	6
Qwen3-8B	0.58	26	283	7	0.57	1242	291	8	0.39	47	505	6
Qwen3-32B	0.69	26	197	7	-	-	-	-	0.57	47	368	9
Llama3-8B	0.45	25	-	9	0.38	1245	-	48	0.50	46	-	11
Gemma-4-12B	0.59	26	154	10	0.58	1356	233	11	0.43	48	501	12

Table 3: Performance and token lengths per dataset. Accuracy (using exact match) (**Acc.**), Question (**Q**), Think chunk (**Think**), and Answer (**Ans**) token length.

ized uncertainty, mean hidden states, mean Lookback Lens ratio, and the mean ICR score (Table 4).

HEADENTROPY reaches the highest average AUROC, 0.736, ahead of attention score (0.685), verbalized uncertainty (0.655), token entropy (0.646), mean hidden states (0.597), mean Lookback Lens (0.586) and mean ICR (0.491). In the paired comparison (Figure 3) HEADENTROPY is significantly better compared to the baselines, from +0.05 AUC over attention score (+8.9% relative) to +0.25 over mean ICR (+75.6%), and it wins in 17 to 20 of the 24 pairs (Figure 3, top block).

The advantage varies slightly across backbones. HEADENTROPY is the strongest training-free score on the three Qwen3 models (0.815, 0.800 and 0.821 for 1.7B, 32B and 8B), but not on the other two: on Gemma-4-12B mean ICR leads (0.773 vs. 0.733), and on Llama-3.1-8B every training-free score is weak, with HEADENTROPY at 0.536 and token entropy the best at 0.723, as non-thinking model, Llama-3.1-8B is the hardest backbone for all methods (Appendix Table 7).

Across domains, HEADENTROPY leads on MedMCQA (0.806 vs. LLM-Check at 0.721) and on HotpotQA (0.668 vs. 0.657 for token entropy), is level with LLM-Check on MATH (0.879 vs. 0.882), and trails Lookback lens on TriviaQA (0.693 vs. 0.719) and FEVER (0.632 vs. 0.645). FEVER is the hardest domain for all training-free scores (Table 4).

5.2 What the probe buys

Comparing HEADENTROPY against itself across the three regimes isolates the contribution of the trained probe (Figure 3, bottom block). For completeness, in the appendix Figure 4 gives the absolute levels these differences are taken from, for every method and regime.

Under domain shift the HEADENTROPY probe contributes nothing measurable. The training-free mean and the probe transferred from another dataset differ by +0.001 AUC (-0.2% relative). Once the probe is applied outside the domain it was fitted on, it recovers no more of the signal than plain averaging does. Fitting in-domain does buy 0.141 AUC and almost all of that gap is the domain shift itself rather than the fit, since the transferred probe gives up 0.131 AUC relative to the in-domain one, however less compared to 0.176 for Lookback Lens, 0.274 for ICR and 0.299 for hidden states. Hidden states match HEADENTROPY when a probe is fitted (Section 5.3) and lose more than twice as much when it is taken away. This suggests that higher entropy is tied a priori to a less settled attention pattern, the direction of the feature is known before any data is seen, and the probe is left with little to add beyond that direction when tested out-of-distribution.

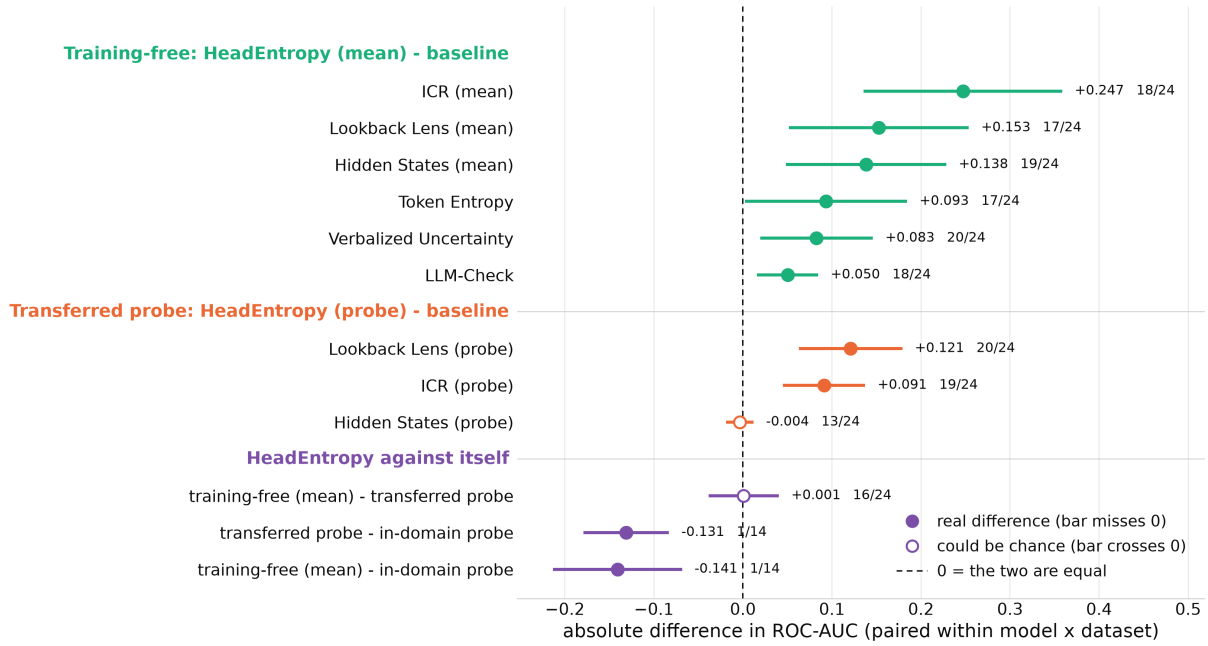


Figure 3: Paired differences in ROC-AUC. Each row is a difference taken within every (model, dataset) cell and then averaged, so model and dataset difficulty cancel instead of inflating the spread. The bar is the 95% interval of those per-cell differences, taken from the Student’s t quantile at $n - 1$ degrees of freedom; For Hidden States, ICR and Lookback Lens in the training-free setting (only there), $1 - \text{AUC}$ was applied after seeing the results; the orientation of HEADENTROPY is fixed a priori by Eq. 10.

Method	TriviaQA	HotpotQA	MedMCQA	FEVER	MATH	Average
Token Entropy	0.609	<u>0.657</u>	0.684	0.578	0.700	0.646
Verbalized Uncertainty	0.657	0.621	0.689	0.619	0.688	0.655
Attention Score	0.659	0.616	<u>0.721</u>	0.549	0.882	<u>0.685</u>
Hidden States (mean)	0.579	0.516	0.601	0.575	0.712	0.597
Lookback Lens (mean)	0.719	0.596	0.596	0.645	0.376	0.586
ICR (mean)	0.482	0.491	0.500	0.509	0.474	0.491
HEADENTROPY (mean)	<u>0.693</u>	0.668	0.806	<u>0.632</u>	<u>0.879</u>	0.736

Table 4: Training-free performance (AUROC) across evaluation datasets, averaged over the five models

5.3 Supervised correctness prediction

In-distribution. Trained and evaluated on the same dataset, a HEADENTROPY probe averages 0.867 AUROC across models and datasets, statistically indistinguishable from Hidden States LR (0.868; paired difference -0.000 , and ahead of Lookback Lens LR (0.818; $+0.051$), and the ICR probe (0.767; $+0.102$); see Appendix F, Figure 4.

Comparison with prior work. Table 6 places HEADENTROPY’s TriviaQA AUROC on Llama-3.1-8B (0.879) next to numbers reported by prior hallucination-detection and uncertainty-estimation methods. Cross-paper comparison is loose, since setups, backbones and evaluation protocols differ.

HEADENTROPY is in the range of reported single-pass numbers (LapEigvals 0.889, Kadavath et al. 0.864) and above the multi-pass Semantic Uncertainty (0.828) and Semantic Energy (0.748).

5.4 Ablations

Appendix ablations confirm consistent performance at 20% of the training data and at regularization strength $c=0.1$ (Appendices J.1, J.2), and justify mean aggregation over min/max (Appendix K, Table 9). Per-head Shapley analysis and a layer ablation (Appendices G, I) show that middle layers carry the most predictive signal. Appendix Table 10 reports AUROC for a classifier trained on deterministic generations and evaluated

on generations under temperature $T=1$. On TriviaQA and HotpotQA, AUROC moves by at most 0.05 in either direction for both Llama-3.1-8B and Qwen3-8B. The exception is MedMCQA, where AUROC drops by 0.20–0.24; in the same setting the underlying model accuracy falls to 0.18 and 0.12 (below the 0.25 chance level for four-option MCQ), indicating that the generations themselves are no longer reliably correct rather than that the signal has degraded.

6 Related Work

Two fundamental goals motivate this study: (i) *uncertainty estimation* and (ii) *hallucination detection*. Uncertainty estimation provides a measure of how much we can trust a model’s output. Hallucination detection, on the contrary, aims to identify statements in generated text that are unsupported or contradict established evidence. Our work translates broader uncertainty estimation and hallucination detection into the more concrete objective of estimating answer-level correctness, that is, the probability that an answer produced by an LLM given an input is correct under a task metric (e.g., exact match, F1 score) (Eq. 4).

Prior work differs in how much of the LLM’s internal state is used to fulfill these goals: (i) black-box methods rely only on outputs, often requiring multiple passes or auxiliary models; (ii) gray-box methods access shallow signals such as final-layer logits; and (iii) white-box methods leverage deeper representations, such as hidden states or attention maps.

Black-box methods. These methods rely solely on the LLM’s input and outputs and often require multiple forward passes or auxiliary models, making them expensive and sensitive to domain shift. SelfCheckGPT (Manakul et al., 2023) detects uncertainty via disagreement across sampled completions, while Chain-of-Verification (Dhuliawala et al., 2024) probes consistency through follow-up questions. Other approaches, such as semantic uncertainty (Kuhn et al., 2023), use embedding models to group paraphrases, introducing the reliance on well-aligned external tools. LLM-as-a-Judge methods (Chen et al., 2024b; Thakur et al., 2025) delegate evaluation to separate LLMs, potentially introducing biases and weak calibration guarantees.

Gray-box methods. Gray-box methods occupy a middle ground, drawing on limited internal signals such as token log probabilities or final hid-

den states. By doing so, they enable single-pass confidence estimation. For example, predictors trained over logits can temper the well-documented tendency of LLMs toward overconfidence (Groot and Valdenegro-Toro, 2024), while filters based on final-layer statistics can anticipate distributional shifts that undermine reliability (Pouget et al., 2025). Likewise, conformal prediction methods extend log-probability signals into calibrated sets of plausible completions (Angelopoulos and Bates, 2021), offering a principled notion of coverage. Despite these strengths, gray-box approaches remain constrained by their reliance on final-layer information, which often calibrates poorly across domains and reduces interpretability to heuristics like “low log-probability” (Geng et al., 2024).

White-box methods. These methods leverage richer internal signals such as layer-wise hidden states, attention maps, or neuron activations, often improving accuracy while introducing additional design choices. Several works (Azaria and Mitchell, 2023; Kadavath et al., 2022) have demonstrated that an LLM’s hidden states encapsulate latent knowledge about hallucinations. Chen et al. (2024a) sample multiple responses and use the log-determinant of their embedding covariance as a correctness signal. Snyder et al. (2024) employ gated recurrent units (GRUs) and multilayer perceptrons (MLPs) over internal representations to predict correctness. For more details on this line of work, please see Appendix D. LLM-Check (Sriramanan et al., 2024) combines multiple internal signals, including attention statistics, to construct a correctness metric. Distinct from the works mentioned above, a small group of methods focuses only on the attention heads of a single sample. Lookback lens detects contextual hallucinations by measuring the location of the attention mass by the ratio of attention weights on the context versus newly generated tokens (Chuang et al., 2024). While AggTruth focuses on contextual hallucination detection in RAG settings, analyzing how attention is distributed across passage tokens using multiple aggregation techniques (Matys et al., 2025). HEADENTROPY targets final-answer correctness rather than contextual hallucination, and uses the spread of the full attention distribution rather than the location of attention mass.

7 Conclusion

HEADENTROPY predicts answer correctness with no training, no calibration, and a direction fixed before any data is seen. Its untrained mean reaches 0.736 AUROC, the level of a hidden-state probe transferred across domains, at under 1% of inference cost. On thinking models the signal is strongest (0.80 to 0.82 AUROC); on the non-thinking Llama-3.1-8B it weakens, which we leave as an open question.

Limitations

The connection between attention entropy and gradient dynamics is a motivation for the feature. $\text{tr}(J)$ says how far a pattern moves under a perturbation of its own scores; the size of a real update also depends on the gradient reaching that head, which we do not measure. HEADENTROPY supports any binary correctness label (including GPT-as-judge), but we restrict to deterministic metrics for reproducibility. Open-ended tasks like summarization and dialogue lack this binary notion and represent a complementary direction, as does code generation where automated verification is available (Chuang et al., 2024; Matys et al., 2025). HEADENTROPY uses the full context (average 26 tokens on TriviaQA to 1242 on HotpotQA). A sliding-window variant in the style of Lookback Lens is a natural extension but not tested here.

References

- Anastasios N Angelopoulos and Stephen Bates. 2021. A gentle introduction to conformal prediction and distribution-free uncertainty quantification. *arXiv preprint arXiv:2107.07511*.
- Amos Azaria and Tom Mitchell. 2023. [The internal state of an llm knows when it’s lying](#). *Preprint*, arXiv:2304.13734.
- Mohammad Beigi, Ying Shen, Runing Yang, Zihao Lin, Qifan Wang, Ankith Mohan, Jianfeng He, Ming Jin, Chang-Tien Lu, and Lifu Huang. 2024. Internalinspector i2: Robust confidence estimation in llms through internal states. In *Findings of the association for computational linguistics: EMNLP 2024*, pages 12847–12865.
- Jakub Binkowski, Denis Janiak, Albert Sawczyn, Bogdan Gabrys, and Tomasz Jan Kajdanowicz. 2025. Hallucination detection in llms using spectral features of attention maps. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 24365–24396.
- Iván Vicente Moreno Cencerrado, Arnau Padrés Masdemont, Anton Gonzalvez Hawthorne, David Demitri Africa, and Lorenzo Pacchiardi. 2025. No answer needed: Predicting llm answer accuracy from question-only linear probes. *arXiv preprint arXiv:2509.10625*.
- Crystal T Chang, Hodan Farah, Haiwen Gui, Shawheen Justin Rezaei, Charbel Bou-Khalil, Ye-Jean Park, Akshay Swaminathan, Jesutofunmi A Omiye, Akaash Kolluri, Akash Chaurasia, and 1 others. 2025. Red teaming chatgpt in medicine to yield real-world insights on model behavior. *npj Digital Medicine*, 8(1):149.
- Chao Chen, Kai Liu, Ze Chen, Yi Gu, Yue Wu, Mingyuan Tao, Zhihang Fu, and Jieping Ye. 2024a. [INSIDE: LLMs’ internal states retain the power of hallucination detection](#). In *The Twelfth International Conference on Learning Representations*.
- Guiming Chen, Shunian Chen, Ziche Liu, Feng Jiang, and Benyou Wang. 2024b. Humans or llms as the judge? a study on judgement bias. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 8301–8327.
- Yung-Sung Chuang, Linlu Qiu, Cheng-Yu Hsieh, Ranjay Krishna, Yoon Kim, and James R. Glass. 2024. [Lookback lens: Detecting and mitigating contextual hallucinations in large language models using only attention maps](#). In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 1419–1436, Miami, Florida, USA. Association for Computational Linguistics.
- Shehzaad Dhuliawala, Mojtaba Komeili, Jing Xu, Roberta Raileanu, Xian Li, Asli Celikyilmaz, and Jason Weston. 2024. Chain-of-verification reduces hallucination in large language models. In *Findings of the association for computational linguistics: ACL 2024*, pages 3563–3578.
- Jiahui Geng, Fengyu Cai, Yuxia Wang, Heinz Koeppel, Preslav Nakov, and Iryna Gurevych. 2024. A survey of confidence estimation and calibration in large language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 6577–6595.
- Tobias Groot and Matias Valdenegro-Toro. 2024. Overconfidence is key: Verbalized uncertainty evaluation in large language and vision-language models. *arXiv preprint arXiv:2405.02917*.
- Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*.
- Jing Huang, Junyi Tao, Thomas Icard, Diyi Yang, and Christopher Potts. 2025. Internal causal mechanisms robustly predict language model out-of-distribution behaviors. *arXiv preprint arXiv:2505.11770*.

- Ziwei Ji, Delong Chen, Etsuko Ishii, Samuel Cahyawijaya, Yejin Bang, Bryan Wilie, and Pascale Fung. 2024. Llm internal states reveal hallucination risk faced with a query. In *Proceedings of the 7th Black-boxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*, pages 88–104.
- Mandar Joshi, Eunsol Choi, Daniel S Weld, and Luke Zettlemoyer. 2017. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. *arXiv preprint arXiv:1705.03551*.
- Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, and 1 others. 2022. Language models (mostly) know what they know. *arXiv preprint arXiv:2207.05221*.
- Jannik Kossen, Jiatong Han, Muhammed Razzak, Lisa Schut, Shreshth Malik, and Yarin Gal. 2024. Semantic entropy probes: Robust and cheap hallucination detection in llms. *arXiv preprint arXiv:2406.15927*.
- Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. 2023. Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation. *arXiv preprint arXiv:2302.09664*.
- Scott M Lundberg and Su-In Lee. 2017. A unified approach to interpreting model predictions. *Advances in neural information processing systems*, 30.
- Huan Ma, Jiadong Pan, Jing Liu, Yan Chen, Joey Tianyi Zhou, Guangyu Wang, Qinghua Hu, Hua Wu, Changqing Zhang, and Haifeng Wang. 2025. Semantic energy: Detecting llm hallucination beyond entropy. *arXiv preprint arXiv:2508.14496*.
- Potsawee Manakul, Adian Liusie, and Mark Gales. 2023. 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*, pages 9004–9017.
- Piotr Matys, Jan Eliaasz, Konrad Kielczyński, Mikołaj Langner, Teddy Ferdinan, Jan Kocoń, and Przemysław Kazienko. 2025. Aggtruth: Contextual hallucination detection using aggregated attention scores in llms. In *International Conference on Computational Science*, pages 227–243. Springer.
- Ankit Pal, Logesh Kumar Umapathi, and Malaikannan Sankarasubbu. 2022. *Medmcqa: A large-scale multi-subject multi-choice dataset for medical domain question answering*. In *Proceedings of the Conference on Health, Inference, and Learning*, volume 174 of *Proceedings of Machine Learning Research*, pages 248–260. PMLR.
- Angéline Pouget, Mohammad Yaghini, Stephan Rabanser, and Nicolas Papernot. 2025. Suitability filter: A statistical framework for classifier evaluation in real-world deployment settings. *arXiv preprint arXiv:2505.22356*.
- Agostino Pugliese, Tiziana Beltramo, and Donato Torre. 2008. Reye’s and reye’s-like syndromes. *Cell Biochemistry and Function: Cellular biochemistry and its modulation by active agents or disease*, 26(7):741–746.
- Lloyd S Shapley and 1 others. 1953. *A value for n-person games*. Princeton University Press Princeton.
- Karan Singhal, Shekoofeh Azizi, Tao Tu, S Sara Mahdavi, Jason Wei, Hyung Won Chung, Nathan Scales, Ajay Tanwani, Heather Cole-Lewis, Stephen Pfohl, and 1 others. 2022. Large language models encode clinical knowledge. *arXiv preprint arXiv:2212.13138*.
- Karan Singhal, Tao Tu, Juraj Gottweis, Rory Sayres, Ellery Wulczyn, Mohamed Amin, Le Hou, Kevin Clark, Stephen R Pfohl, Heather Cole-Lewis, and 1 others. 2025. Toward expert-level medical question answering with large language models. *Nature Medicine*, 31(3):943–950.
- Sonish Sivarajkumar, Mark Kelley, Alyssa Samolyk-Mazzanti, Shyam Visweswaran, and Yanshan Wang. 2024. An empirical evaluation of prompting strategies for large language models in zero-shot clinical natural language processing: algorithm development and validation study. *JMIR Medical Informatics*, 12:e55318.
- Ben Snyder, Marius Moisesescu, and Muhammad Bilal Zafar. 2024. On early detection of hallucinations in factual question answering. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 2721–2732.
- Gaurang Sriramanan, Siddhant Bharti, Vinu Sankar Sadasivan, Shoumik Saha, Priyatham Kattakinda, and Soheil Feizi. 2024. Llm-check: Investigating detection of hallucinations in large language models. *Advances in Neural Information Processing Systems*, 37:34188–34216.
- Ryutaro Tanno, David GT Barrett, Andrew Sellergren, Sumedh Ghaisas, Sumanth Dathathri, Abigail See, Johannes Welbl, Charles Lau, Tao Tu, Shekoofeh Azizi, and 1 others. 2025. Collaboration between clinicians and vision-language models in radiology report generation. *Nature Medicine*, 31(2):599–608.
- Aman Singh Thakur, Kartik Choudhary, Venkat Srinik Ramayapally, Sankaran Vaidyanathan, and Dieuwke Hupkes. 2025. Judging the judges: Evaluating alignment and vulnerabilities in llms-as-judges. In *Proceedings of the Fourth Workshop on Generation, Evaluation and Metrics (GEM²)*, pages 404–430.
- James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018. Fever: a large-scale dataset for fact extraction and verification. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 809–819.

Tao Tu, Shekoofeh Azizi, Danny Driess, Mike Schaekermann, Mohamed Amin, Pi-Chuan Chang, Andrew Carroll, Charles Lau, Ryutaro Tanno, Ira Ktena, and 1 others. 2024. Towards generalist biomedical ai. *Nejm Ai*, 1(3):A10a2300138.

Li Wang, Xi Chen, XiangWen Deng, Hao Wen, MingKe You, WeiZhi Liu, Qi Li, and Jian Li. 2024. Prompt engineering in consistency and reliability with the evidence-based guideline for llms. *NPJ digital medicine*, 7(1):41.

Johnathan Xie, Annie S Chen, Yoonho Lee, Eric Mitchell, and Chelsea Finn. 2024. Calibrating language models with adaptive temperature scaling. *arXiv preprint arXiv:2409.19817*.

Lin Yang, Shawn Xu, Andrew Sellergren, Timo Kohlberger, Yuchen Zhou, Ira Ktena, Atilla Kiraly, Faruk Ahmed, Farhad Hormozdiari, Tiam Jaroensri, and 1 others. 2024. Advancing multimodal medical capabilities of gemini. *arXiv preprint arXiv:2405.03162*.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D Manning. 2018. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 conference on empirical methods in natural language processing*, pages 2369–2380.

Juan Manuel Zambrano Chaves, Shih-Cheng Huang, Yanbo Xu, Hanwen Xu, Naoto Usuyama, Sheng Zhang, Fei Wang, Yujia Xie, Mahmoud Khademi, Ziyi Yang, and 1 others. 2025. A clinically accessible small multimodal radiology model and evaluation metric for chest x-ray findings. *Nature Communications*, 16(1):3108.

Zhenliang Zhang, Xinyu Hu, Huixuan Zhang, Junzhe Zhang, and Xiaojun Wan. 2025. 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)*, pages 17986–18002.

A Use of LLM

We use LLMs to check grammar and typos.

B Prompts

B.1 TriviaQA Prompt

```
1 def build_triviaqa_prompt(item):
2     question = item["question"]
3
4     system_message = (
5         "You are a trivia expert.
6         Please answer questions
7         in exactly this format:\n
8         "
9         "Answer: [1-3 words only]\n"
10        "Certainty: [0-100]\n\n"
11    )
12    user_prompt = f"Question: {
13        question}"
14
15    return system_message,
16           user_prompt
```

B.2 MedMCQA Prompt

```
1 def build_ind_prompt(item):
2
3     system_message = (
4         "You are a medical expert.
5         Please answer questions
6         in exactly this format:\n
7         "
8         "Answer: [repeat correct
9         option]\n"
10        "Certainty: [0-100]\n\n"
11    )
12    user_prompt = (
13        "Question:"
14        + item["question"].split(". "
15        )[-1]
16        + "\n"
17        + "Options:"
18        + " ".join(str(item[op]) + "\n"
19        for i, op in
20        enumerate(["
21        opa", "opb", "
22        opc", "opd"]))
23    )
24    return system_message,
25           user_prompt
```

B.3 HotpotQA Prompt

```
1 def build_hotpot_prompt(item):
2     system_message = (
3         "You are a helpful assistant.
4         "
5         "Answer the question using
6         the information in the
7         provided passages."
8         "Please answer questions in
9         exactly this format:\n"
10        "Answer: [1-5 words only]\n"
11    )
```

```

7         "Certainty: [0-100]\n\n"
8     )
9     list_sentences = []
10    for sentence_list in item["
11        context"]["sentences"]:
12        list_sentences.extend(
13            sentence_list)
14
15    context_text = "\n".join(sentence
16        for sentence in
17        list_sentences)
18
19    user_prompt = (
20        "Context: "
21        f"{context_text}\n\n"
22        f"Question: {item['question
23            '']}\n"
24    )
25
26    return system_message,
27        user_prompt

```

C Model dimensions

Table 5: Feature dimensionality per model. d : hidden size (dimension of a mean-pooled hidden-state feature); h : attention heads per layer; L : layers; $m = hL$: total heads, the dimension of the HEADENTROPY feature vector. The Qwen3 and Llama models use $d_{\text{head}} = 128$ and Gemma-4-12B uses $d_{\text{head}} = 256$; Qwen3-32B and Gemma-4-12B are the two where $h \cdot d_{\text{head}} \neq d$. Values from each model’s config.json on Huggingface.

Model	d	h	L	m	d/m
Qwen3-1.7B	2048	16	28	448	4.6
Llama-3.1-8B	4096	32	32	1024	4.0
Qwen3-8B	4096	32	36	1152	3.6
Qwen3-32B	5120	64	64	4096	1.3
Gemma-4-12B	3840	16	48	768	5.0

D Related work on hidden states as features for hallucination detection

A growing line of work uses hidden states as features for predicting ‘hallucinations’, a term used loosely across this literature to cover varied failure modes, including factual errors, inconsistencies, and errors flagged by LLM judges. In contrast, Head Entropy uses only the attention distribution: it is lower-dimensional, interpretable, automated across layers, and requires a single forward pass at inference time.

Kossen et al. (2024)’s SEPs trains hidden state probes to predict semantic entropy, while HEAD ENTROPY uses attention entropy to predict answer correctness directly under standard QA metrics. HEAD ENTROPY offers four advantages over a

SEPs-style hidden state probe for the given task: First, we provide a formal motivation connecting 2-Rényi attention entropy to training gradient dynamics via the trace (equivalently, nuclear norm) of the softmax Jacobian. SEPs demonstrates empirically that hidden states encode semantic uncertainty, while HEAD ENTROPY explains why attention entropy relates to gradient stability. Second, HEAD ENTROPY avoids manual layer and token position selection entirely. Attention entropy is available at every layer and head, and the sparse classifier automatically identifies informative heads, whereas SEPs require ablations to find optimal layer and token positions. Third, the feature vector has dimensionality m (number of attention heads, typically for open source models 16-64) versus hidden state dimensionality d (typically 1024-8192). Supervising HEAD ENTROPY on binary semantic entropy labels (analogous to Kossen et al.) is an interesting future direction. For the scope of this work we focus on exact match and F1 as correctness metrics.

Cencerrado et al. (2025) focuses on predicting answer correctness before generation using linear probes on hidden states with a heuristic motivation. Although our work emphasizes the attention distribution using a formal motivation, the task is comparable (Section 6.3). Similarities: the best-performing assessor (trained on embeddings from GPT) is logistic regression, signal increases with larger models, and middle layers perform slightly better. Key differences: their core claims focus on OOD generalization for predicting answer correctness from question-only hidden states, a distinct goal from ours. Results are reported on a subsampled training set, making direct comparison non-trivial as open-source models are almost certainly trained on it. They use careful per model layer selection, whereas HEAD ENTROPY automates this process.

Ji et al. (2024) addresses a similar task; however, it focuses on self-awareness of uncertainty by examining how internal states indicate whether the model has seen the query during training, a question related to our paper. We focus on a formal motivation through the lens of training stability and connect feature generation directly to training dynamics, with their hidden-state approach serving as one of our baselines. They employ a more complex multilayer perceptron (MLP) variant.

Huang et al. (2025) investigates whether internal causal mechanisms can robustly predict model correctness in out-of-distribution settings, covering

tasks such as symbol manipulation, knowledge retrieval, and instruction following. While related to our goal, we examine more closely the contribution of each attention head using a logistic regression model and note that generalization across tasks exists, but with some task-dependent component.

Beigi et al. (2024) is in the same field as the studies mentioned above. One focus here is predictor model selection and contrastive loss construction to cope with high-dimensional inputs from hidden, activation, and attention states across all layers.

Unlike methods analyzing static, high-dimensional hidden states, Zhang et al. (2025) focuses on the dynamic update process of the residual stream, which proves more effective at identifying when a model deviates from factual accuracy across layers. The ICR Probe outperforms prior hidden-state baselines and provides improved interpretability for understanding hallucination mechanisms.

Table 6: TriviaQA in-distribution correctness-prediction AUROC across hallucination detection methods. All numbers except HEADENTROPY are reported from the original papers and in-distribution; setups and models differ across rows and numbers are not strictly comparable. [†]Reimplemented result; the original paper does not report TriviaQA.

Method	Model	Train Gen.	Test Gen.	Training Free	TriviaQA AUROC
Semantic Uncertainty (Kuhn et al., 2023)	OPT-30B	—	>1	✓	0.828
Semantic Energy (Ma et al., 2025)	Qwen3-8B	—	>1	✓	0.748
Kadavath et al. (Kadavath et al., 2022)	52B (P(IK))	>1	1	×	0.864
SEP (Kossen et al., 2024)	Llama-2-7B	>1	1	×	<0.85
Snyder et al. (Snyder et al., 2024)	OpenLlaMA-7B	1	1	×	0.71
LapEigvals (Binkowski et al., 2025)	Llama-3.1-8B	1	1	×	0.889
ICR Probe (Zhang et al., 2025)	Llama-3	1	1	×	0.7325
LLM-Check (Sriramanan et al., 2024)	Llama-2-7B	—	1	✓	0.742 [†]
LookbackLens	Llama-3.1-8B	1	1	×	0.883 [†]
Hidden State LR	Llama-3.1-8B	1	1	×	0.858 [†]
HEADENTROPY	Llama-3.1-8B	1	1	×	0.879

E Training-free performance per model

F Supervised performance, per dataset and per model

Method	Llama-3.1-8B	Gemma-4-12B	Qwen3-1.7B	Qwen3-32B	Qwen3-8B	Average
Token Entropy	0.723	0.706	0.580	0.576	0.628	0.642
Verbalized Uncertainty	0.543	0.645	0.708	0.613	0.762	0.654
Attention Score	0.501	0.740	0.769	0.691	0.743	0.688
Hidden States (mean)	0.517	0.730	0.346	0.671	0.750	0.603
Lookback Lens (mean)	0.516	0.499	0.656	0.611	0.652	0.587
ICR (mean)	0.522	0.773	0.271	0.465	0.421	0.490
HEADENTROPY (mean)	0.536	0.733	0.815	0.800	0.821	0.741

Table 7: Training-free performance (AUROC) across models, averaged over the five evaluation datasets.

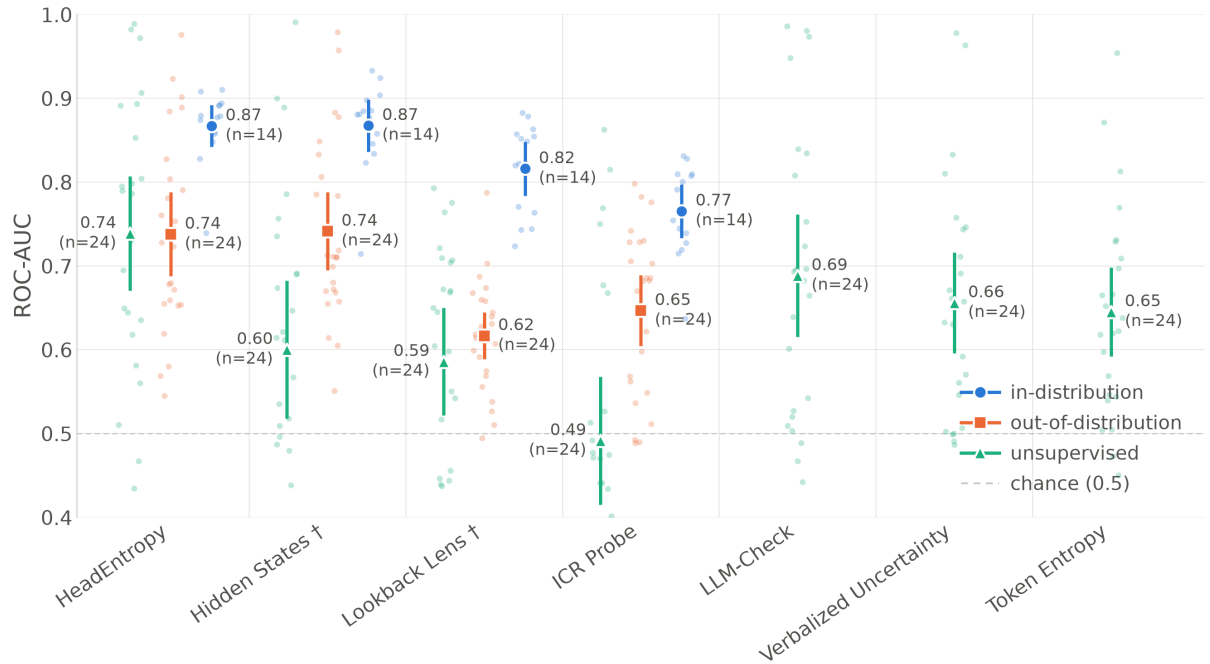


Figure 4: ROC-AUC by method and supervision regime. Each marker is the mean over (model, dataset) cells and the bar the 95% interval of those cells, taken from the Student’s t quantile at $n - 1$ degrees of freedom; the faint dots are the individual cells. In-distribution covers 14 cells and the other two regimes 24, since no probe is trained on MATH or FEVER. LLM-Check, verbalized uncertainty and token entropy exist only in the training-free regime and so carry a single marker. Rows marked † are plotted as $1 - \text{AUC}$: those features are anti-correlated with correctness in the training-free setting, and the flip was applied after seeing the results, whereas the orientation of HEADENTROPY is fixed a priori by Eq. 10. Figure 3 gives the same comparison as within-cell paired differences.

G Explaining Predictions with Shapley Values

To attribute predictive performance to specific heads, we compute Shapley values (Shapley et al., 1953; Lundberg and Lee, 2017) on the trained classifier:

$$\phi_k(f) = \frac{1}{m!} \sum_{\pi} [f(P_k^{\pi} \cup \{k\}) - f(P_k^{\pi})], \quad (13)$$

where π ranges over all feature orderings, P_k^{π} is the set of features preceding k in π , and $\phi_k(f)$ quantifies the contribution of head $k \in \{1, \dots, m\}$. Using Shapley values, we can construct a heat map over heads to visualize which heads contribute positively (red) or negatively (blue) to the prediction of being correct (Appendix Fig. I).

We found that there are negative weighted heads and that there are positive SHAP values (Figure 5).

Negative coefficients are expected by construction: high entropy should push down correctness probability. Positive coefficients are mainly a standardization artifact (from sklearn.preprocessing import StandardScaler): standard feature normalization (for feature) before training shifts some heads to appear positive while encoding the same relationship, just shifted. We confirmed this empirically with the Qwen3 8B and Llama3.1 8B models over all three datasets, finding <3% positive weights on average when using no standardization.

The positive SHAP values show a positive contribution of a head relative to the baseline (mean of training set predictions), explaining red colored heads. The fact that MLP and XGBoost offer no improvement over logistic regression further confirms the entropy-correctness relationship is monotonic.

H Do predictive heads cluster in consistent depth ranges?

Our per-head Shapley analysis shows middle layers consistently carry higher-magnitude values (see below), consistent with the layer ablation in Appendix I where test AUROC plateaus well before all layers are included. We use L1 logistic regression as a data-driven substitute for structured head selection.

I Contributions of Entropy across Layers

We conducted a layer ablation study, beginning with the final layer and progressively adding five layers sat a time. The results indicate that predictive value is not limited to the last layer alone.

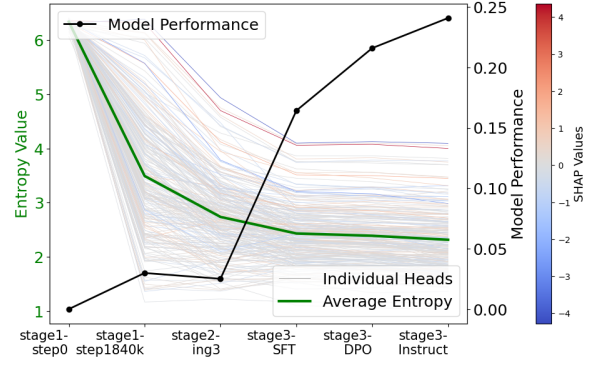


Figure 5: 2-Renyi values (green) over training of Olmo2-1B model. Each of the fine lines is a head’s entropy values with color coding of the shapley value (red means positive and blue means negative, more in appendix G) of the final predictive logistic regression model. The green line represents the average entropy of the heads. Model Performance (black) measured in percentage of exact match examples for the validation set of TriviaQA.

Table 8: Average layer index of above-mean SHAP value layers, normalized by total number of layers.

Model	TriviaQA	HotpotQA	MedMCQA
Qwen3-1.7B	0.532	0.491	0.552
Qwen3 8B	0.519	0.464	0.565
Llama 3.1 8B	0.327	0.469	0.354
Qwen3 32B	0.513	—	0.489

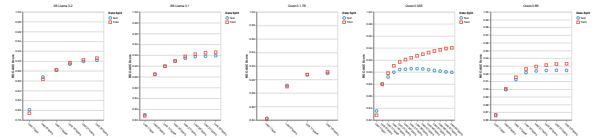


Figure 6: Trivia QA: Layer ablation, where layer 1 is the last layer and then adding more layers

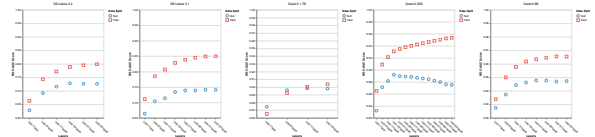


Figure 7: MedMC QA: Layer ablation, where layer 1 is the last layer and then adding more layers

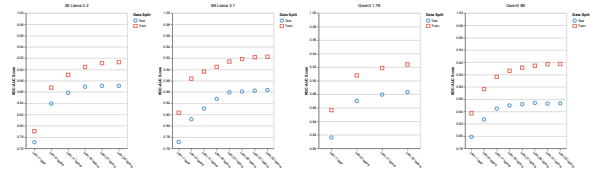


Figure 8: Hotpot QA: Layer ablation, where layer 1 is the last layer and then adding more layers

J Ablations on regularization and data size

J.1 Regularization Ablation

Furthermore, we made the best effort to make the hyperparameter choice as standard as possible, as our main focus is methodological not empirical hyperparameter tuning. We set lambda to be the default value of 1. Please see the ablation in Figure J.1.

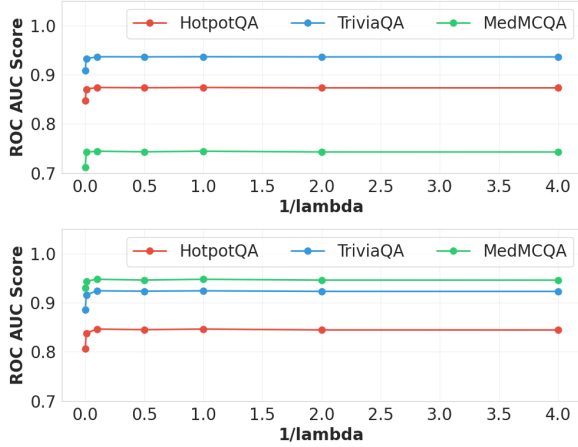


Figure 9: Performance vs. L1 regularization for Qwen3-8B (upper) and Llama 3.1 8B (lower), where for all datasets the performance converges at 0.1. Lower values correspond to more regularization strength.

J.2 Data Ablation

Please see the curves showing performance vs. training set size to demonstrate data efficiency Figure 10.

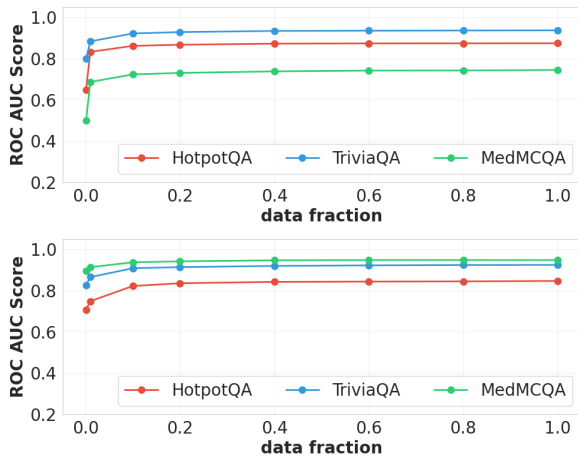


Figure 10: Performance vs. training set size for Qwen3-8B (upper) and Llama 3.1 8B (lower), where 1 data fraction corresponds to 50k examples. For all datasets the performance converges at around 0.2.

Experiments suggest HEADENTROPY maintains strong performance even with 20-30% of training data.

K Aggregation across tokens

We aggregate per-token head entropies into a single per-head feature by averaging over the tokens of a semantic section (Eq. 8). We compared the mean against the per-token minimum and maximum on Llama-3.1-8B and Qwen3-8B across TriviaQA, HotpotQA, and MedMCQA (Table 9). The mean is consistently the best or tied-best, followed closely by the maximum (within roughly 0.02 AUROC), while the minimum is the weakest. We therefore use the mean throughout.

Table 9: Empirically testing average, minimum and maximum aggregation over the rows of the attention matrix. Average showing consistently slightly higher AUROC, followed by maximum, ranging within 2 AUROC points.

Dataset	Llama-3.1-8B			Qwen3-8B		
	Avg	Max	Min	Avg	Max	Min
HotpotQA	0.87	0.86	0.85	0.85	0.83	0.82
MedMCQA	0.74	0.73	0.73	0.95	0.87	0.73
TriviaQA	0.94	0.92	0.92	0.92	0.92	0.92

L Robustness to decoding temperature

Table 10 reports AUROC for a HEADENTROPY classifier trained at $T < 0.0001$ and evaluated on generations sampled at $T = 1$.

Table 10: Performance under different temperature at inference time remains comparable. For the MedMCQA dataset the overall performance drops below random (0.25), likely responsible for the performance drop in AUROC of HEADENTROPY.

Model	Dataset	AUROC of HEADENTROPY		Overall Model Accuracy	
		$T < 0.0001$	$T = 1$	$T < 0.0001$	$T = 1$
Llama-3.1-8B	TriviaQA	0.93	0.94	0.45	0.40
	MedMCQA	0.74	0.54	0.50	0.18
	HotpotQA	0.88	0.89	0.38	0.36
Qwen3-8B	TriviaQA	0.92	0.91	0.58	0.53
	MedMCQA	0.94	0.70	0.39	0.12
	HotpotQA	0.80	0.85	0.58	0.56

M Pseudocode

Algorithm 1 summarizes the full HEADENTROPY pipeline: for each example, we compute the 2-Rényi entropy of every attention head at every token, convert it to the trace of the softmax Jacobian

(Eq. 7), average over the tokens of a semantic section, stack into a per-example feature vector, and train an ℓ_1 -regularized logistic regression on the binary correctness label.

Algorithm 1 HEADENTROPY

Require: Dataset $\{(x_i, y_i)\}_{i=1}^N$; transformer with m heads; sequence length T of x_i

for each (x_i, y_i) **do**

 Obtain $\hat{y}_i$ and attention tensor $A_i \in \mathbb{R}^{m \times T \times T}$ (Eq. 5)

for each head k and token t **do**

 Compute 2-Rényi entropy $H_{k,i}^{(t)} = -\log \sum_j \left(A_{k,i}^{(t,j)}\right)^2$

 Compute trace $(J)_{k,i}^{(t)} = 1 - \exp(-H_{k,i}^{(t)})$ (Eq. 7)

end for

 Average over section tokens to get $\bar{H}_{k,i} \in \mathbb{R}$ (Eq. 8)

 Stack into feature vector $\bar{H}_i = [\bar{H}_{1,i}, \dots, \bar{H}_{m,i}]^\top \in \mathbb{R}^m$

end for

Stack $\{\bar{H}_i\}$ into feature matrix $\bar{H} \in \mathbb{R}^{N \times m}$

Define labels $z_i = \mathbb{I}[\hat{y}_i = y_i]$, where $z \in \mathbb{R}^N$

Take the mean of $\bar{H}$ across m $\bar{H}_{\text{mean}}$

return $\bar{H}_{\text{mean}}, z$.

N Derivation of the Softmax Jacobian

We derive the Jacobian of the softmax function. Let:

$$p_i := \text{softmax}(x)_i = \frac{\exp(x_i)}{\sum_{j=1}^K \exp(x_j)} \quad (14)$$

To compute $\frac{\partial p_i}{\partial x_j}$, we consider two cases.

Case 1: $i = j$. We apply the quotient rule:

$$\frac{\partial p_i}{\partial x_i} = \frac{\exp(x_i) \cdot \sum_k \exp(x_k) - \exp(x_i) \cdot \exp(x_i)}{(\sum_k \exp(x_k))^2} \quad (15)$$

$$= \frac{\exp(x_i)}{\sum_k \exp(x_k)} - \frac{\exp(x_i)^2}{(\sum_k \exp(x_k))^2} \quad (16)$$

$$= p_i - p_i^2 = p_i(1 - p_i) \quad (17)$$

Case 2: $i \neq j$. The numerator $\exp(x_i)$ does not depend on x_j , so:

$$\frac{\partial p_i}{\partial x_j} = \frac{0 - \exp(x_i) \cdot \exp(x_j)}{(\sum_k \exp(x_k))^2} = -p_i p_j \quad (18)$$

Combined. We can write the Jacobian compactly as:

$$\mathcal{J}_{\text{softmax}} = \text{diag}(p) - pp^\top \quad (19)$$

We can now see that the trace of the Jacobian is monotonically related to the 2-Rényi entropy.

Recall that 2-Rényi entropy is $H_2(p) := -\log \sum_i p_i^2$. Using $J = \text{diag}(p) - pp^\top$:

$$\begin{aligned} \text{Tr}(J) &= \text{Tr}(\text{diag}(p)) - \text{Tr}(pp^\top) \\ &= \sum_i p_i - \|p\|^2 \\ &= 1 - \|p\|^2 \\ &= 1 - \exp(-H_2(p)) \end{aligned}$$

O LLM contributions

In this work, we used LLMs to help improve the readability of the paper and create figures.