Learned Hallucination Detection in Black-Box LLMs using Token-level Entropy Production Rate

Charles Moslonka[A,B], Hicham Randrianarivo[A,B], Arthur Garnier[C] and Emmanuel Malherbe[A]
[A]Artefact Research Center, Paris, France
[B]MICS, CentraleSupélec, Université Paris-Saclay, Gif-sur-Yvette, France
[C]Ardian, Paris, France

Hallucinations in Large Language Model (LLM) outputs for Question Answering (QA) tasks can critically undermine their real-world reliability. This paper introduces a methodology for robust, one-shot hallucination detection, specifically designed for scenarios with limited data access, such as interacting with black-box LLM APIs that typically expose only a few top candidate log-probabilities per token. Our approach derives uncertainty indicators directly from these readily available log-probabilities generated during non-greedy decoding. We first derive an Entropy Production Rate (EPR) that offers baseline performance, later augmented with supervised learning. Our learned model leverages the entropic contributions of the accessible top-ranked tokens within a single generated sequence, without multiple re-runs per query. Evaluated across diverse QA datasets and multiple LLMs, this estimator significantly improves token-level hallucination detection over state-of-the-art methods. Crucially, high performance is demonstrated using only the typically small set of available log-probabilities (e.g., top-10 per token), confirming its practical efficiency and suitability for API-constrained deployments. This work provides a lightweight technique to enhance the trustworthiness of LLM responses, at the token level, after a single generation pass, for QA and Retrieval-Augmented Generation (RAG) systems. Our experiments confirmed the performance of our method against existing approaches on public dataset as well as for a financial framework analyzing annual company reports.

Keywords: *Hallucination Detection, Large Language Models, Uncertainty Quantification, Retrieval-Augmented Generation*

1. Introduction

Large Language Models (LLMs) have demonstrated remarkable capabilities across a wide range of tasks [1–3]. However, a significant and persistent challenge is their propensity to generate hallucinations [4, 5]. These are defined as LLM-generated content that is factually incorrect, nonsensical, or unfaithful to a provided source, despite often appearing plausible and coherent [6, 7]. The impact of hallucinations can be severe, particularly in safety-critical applications such as medicine or infrastructure engineering [8, 9], leading to the propagation of misinformation and erosion of user trust. Therefore, detecting and quantifying the uncertainty associated with LLM outputs is paramount to building trust and enabling responsible deployment.

The practical deployment of Uncertainty Quantification (UQ) [10] and hallucination detection methods is often constrained by limited access to LLM internals. These limitations are especially true when interacting with proprietary models through APIs, which may expose only a small number of top-K log-probabilities per token (where K might be around 15[1]). This black-box setting restricts the applicability of techniques requiring access to full logits, hidden states, or architectural mod-

[1]The maximum value is $K = 20$ for the OpenAI API at the time of writing.

Published at the European Conference on Information Retrieval (ECIR) 2026
Corresponding author(s): charles.moslonka@artefact.com
Artefact Research Center, Paris, France

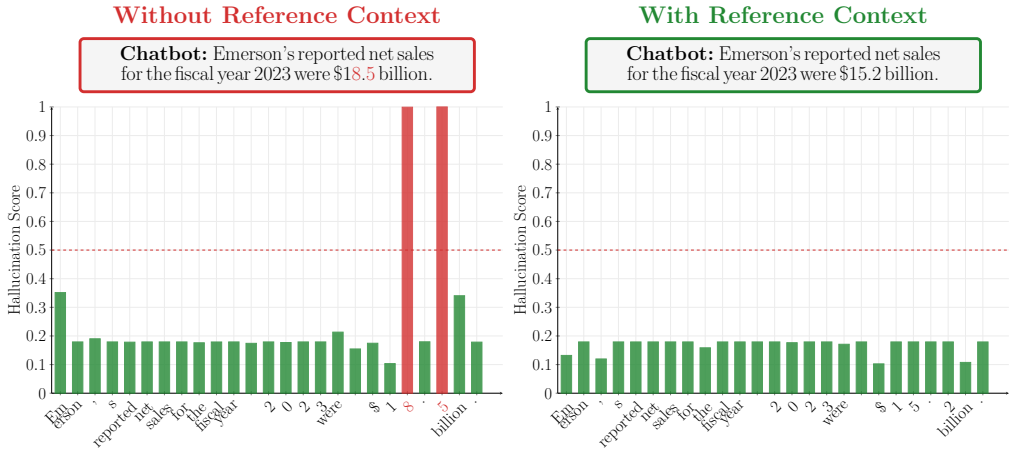


Figure 1 | Illustration of token-level uncertainty detection in a financial RAG pipeline.

ification [11]. Moreover, many real-world applications require one-shot (or "single-round" [12]) detection—the ability to assess the reliability of a single generated sequence without the need for multiple, often costly, model inferences for the same input query to measure output variability.

This paper introduces a UQ methodology rooted in information theory, tailored for these black-box, one-shot scenarios. Our approach leverages the accessible log-probabilities to derive entropic measures of model hesitation during the generation process. We define and evaluate a metric termed the Entropy Production Rate (EPR) of a generated sequence. We show that EPR, calculated as the average entropy of the token probability distributions across the entire sequence, serves as an initial, unsupervised estimator of the degree of hesitation exhibited by the model during inference.

Our second contribution is a supervised learning model that adapts entropy measures to a dataset of annotations. By training on the entropic contributions of top-ranked tokens across the sequence, we learn an estimator that more accurately distinguishes between faithful and hallucinatory responses. This estimator can also highlight high-uncertainty tokens in a generated sequence, as displayed in Figure 1.

We detail the entropic framework, its theoretical and practical underpinnings under limited log-probability access, and its empirical validation. We showcase its potential as a practical tool for improving the reliability of LLM systems, using RAG in a financial scenario as an example. To facilitate adoption and future research, we release our methods (EPR and WEPR) as an open-source Python package available at `https://github.com/artefactory/artefactual/`.

2. Related Work

UQ in Large Language Models. The primary goal of UQ in LLMs is to enable them to signal their confidence — to "know what they do not know" [13, 14]. LLM uncertainty arises from traditional aleatoric (data-inherent) and epistemic (model-based) sources [15], and additionally from their generative characteristics like divergent multi-step reasoning paths and decoding stochasticity [12]. Traditional neural network UQ approaches, such as training modifications for aleatoric uncertainty [16] or dropout-based methods for epistemic uncertainty [17, 18], are often ill-suited or prohibitively costly for LLMs due to training expense or architectural differences. Developing UQ techniques tailored to LLM intricacies remains an active research area [19].

Information-theoretic entropy has emerged as a cornerstone for assessing LLM uncertainty and

hallucinations, based on the hypothesis that higher output distribution entropy often correlates with a greater likelihood of error [20]. Researchers have employed diverse entropy measures to capture different facets of LLM uncertainty [21]. Predictive entropy [22] assesses overall output distribution uncertainty. Semantic Entropy [23] quantifies uncertainty over semantic meaning by clustering multiple generated responses, with Discrete Semantic Entropy [24, 25] using frequency-based cluster probabilities. Answer Entropy [26], common in QA, measures variability in complete answer strings from multiple trials, sometimes enhanced with noise injection. Particularly relevant to our work, token-level entropy [13] measures per-step uncertainty from the next-token probability distribution and has been used to predict errors in QA and mathematical reasoning [11, 27].

Entropy-driven detection techniques and challenges. Direct application of entropy scores, often via thresholding, is a common approach for hallucination detection or calibration [11, 13, 23]. More sophisticated techniques include Semantic Entropy Probes (SEPs) [28]—learned models predicting Semantic Entropy from single-pass hidden states for efficiency—or Shifting Attention to Relevance (SAR) [29], which weights tokens by relevance. However, entropy-based methods face challenges like high-certainty hallucinations [30], where LLMs produce low-entropy incorrect outputs, necessitating strategies beyond simple thresholds. Moreover, the computational cost of multi-sample methods (e.g., for Answer or Semantic Entropy) can be prohibitive, highlighting the need for efficient single-shot techniques such as SEPs [28], Logit-induced Token Uncertainty (LogTokU) [31], and ours.

3. Theoretical Framework for Entropic Uncertainty

Our approach to estimate uncertainty in LLM generations is rooted in information theory [32, 33]. The core idea is to derive an entropic score reflecting the model's hesitation when generating tokens. Unlike methods requiring multiple model runs to assess output variability, our method is a one-shot analysis, leveraging the probabilistic information from a single generated sequence.

3.1. Entropy computation during token generation.

For a standard classification task with N_c classes, given an input q, a model outputs probabilities $p_i = p(\text{class}_i|q)$ for each class $i \in \{1, \ldots, N_c\}$. The uncertainty can be quantified by the Shannon entropy (in bits) of this distribution:

$$H(q) = -\sum_{i=1}^{N_c} p_i \log_2(p_i) \geq 0. \tag{1}$$

This entropy $H(q)$ is minimized to zero when the distribution is sharply peaked (high certainty) and maximized at $H_{\max}(q) = \log_2(N_c)$ for a uniform distribution (maximum uncertainty).

Extending this to LLM sequence generation, let $V = \{v_i\}_{i=1}^{|V|}$ denote the model's vocabulary, with size[2] $|V|$. For a given input query q, the LLM generates a sequence of tokens $\mathcal{T} = \{t_1, t_2, \ldots, t_{|\mathcal{T}|}\}$ $(\forall j, t_j \in V)$. The generated sequence $\mathcal{T}$ depends on q, but we omit this dependency for clarity. At each generation step j, the LLM internally computes a probability distribution $\{p(v_i \mid q, t_{<j})\}_{i=1}^{|V|}$ for the next token t_j, conditioned on the preceding tokens $t_{<j} := \{t_1, \ldots, t_{j-1}\}$ and the input q. The complete entropy for the j-th token's distribution is:

$$H(q, t_{<j}) = -\sum_{i=1}^{|V|} p(v_i \mid q, t_{<j}) \log_2 p(v_i \mid q, t_{<j}). \tag{2}$$

[2] e.g., $|V| = 2^{17} = 131{,}072$ for some contemporary models, such as `Mistral-Small-3.1-24B` [34].

Note that this quantity does not depend on the sampled token t_j.

In black-box scenarios involving proprietary LLMs accessed via APIs, only the log-probabilities (or probabilities) for a small number, denoted by K, of top-ranked candidate tokens are exposed (e.g., $K \leq 20$). Thus, we estimate the per-token entropy based on these top K probabilities. Let $r : \{1, \ldots, |V|\} \to \{1, \ldots, |V|\}$ be the ranking operator on the vocabulary indices, meaning $r(k)$ is the index i of the k-th token in decreasing order of $p(v_i \mid q, t_{<j})$. The estimator can then be written as:

$$\tilde{H}_K(q, t_{<j}) = -\sum_{k=1}^{K} p_{r(k),j} \log_2 \left(p_{r(k),j}\right), \tag{3}$$

where $p_{r(k),j} := p(v_{r(k)}|q, t_{<j})$ is the probability of the k-th ranked token at generation step j.

3.2. Sufficiency of Top-K Log-Probabilities for Entropy Estimation

A critical question is whether $\tilde{H}_K(q, t_{<j})$, calculated from only the top K probabilities, is a reliable proxy for the true entropy $H(q, t_{<j})$. The discrepancy arises from the missing tail of the distribution:

$$\Delta H_K(q, t_{<j}) = H(q, t_{<j}) - \tilde{H}_K(q, t_{<j}) = -\sum_{k=K+1}^{|V|} p_{r(k),j} \log_2 p_{r(k),j}. \tag{4}$$

We can establish an upper bound for this missing entropy by assuming the remaining probability mass $Q_K(q, t_{<j}) = 1 - \sum_{k=1}^{K} p_{r(k),j}$ is distributed uniformly among the remaining tokens considered sampleable. When the LLM generation uses a cutoff parameter $K_{\text{samp}} \ll |V|$ (e.g., $K_{\text{samp}} = 50$), probabilities beyond K_{samp} are set to zero. In that case:

$$\Delta H_{K,\max}(q, t_{<j}) = -Q_K(q, t_{<j}) \left[\log_2(Q_K(q, t_{<j})) - \log_2(K_{\text{samp}} - K)\right] \tag{5}$$

The quality of $\tilde{H}_K(q, t_{<j})$ as an estimator of the entropy within the relevant token pool (either full vocabulary or K_{samp} limited) can be assessed by comparing $\tilde{H}_K(q, t_{<j})$ and $\Delta H_{K,\max}(q, t_{<j})$; a large ratio $\tilde{H}_K / \Delta H_{K,\max}$ suggests that $\tilde{H}_K(q, t_{<j})$ captures most of the relevant uncertainty.

Empirically, we observe a concentration in the log-probabilities of lower-ranked tokens (e.g., for ranks $i \approx 10 - 15$), as illustrated in Figure 2. Differences between successive $p_{r(k),j}$ tend to diminish for larger k within the accessible K. When combined with K_{samp} sampling, this supports the idea that $\tilde{H}_K(q, t_{<j})$ captures the dominant portion of the uncertainty within the set of practically sampleable tokens, especially if K is reasonably close to K_{samp}.

3.3. Influence of Sampling Temperature

In practice, at j-th generation step, a LLM first predicts raw logits $l_{j,i}$ and converts them to probabilities $p_{j,i}$ by a softmax function with a temperature parameter T_{samp}, such that: $p_{j,i}(T_{\text{samp}}) = e^{l_{j,i}/T_{\text{samp}}} / \sum_{m=1}^{|V|} e^{l_{j,m}/T_{\text{samp}}}$.
The per-token information entropy $H(q, t_{<j}; T_{\text{samp}})$ thus depends on T_{samp}, verifying:

- As $T_{\text{samp}} \to 0^+$, $p_{j,i}(T_{\text{samp}})$ approaches 1 for the token with the highest logit (assuming a unique maximum) and 0 for others. Consequently, $H(q, t_{<j}; T_{\text{samp}}) \to 0$, reflecting high certainty.
- As $T_{\text{samp}} \to \infty$, $p_{j,i}(T_{\text{samp}})$ approaches $1/|V|$ for all tokens, leading to a uniform distribution. Therefore, the entropy $H(q, t_{<j}; T_{\text{samp}}) \to \log_2 |V|$, its maximum possible value, reflecting maximum uncertainty.

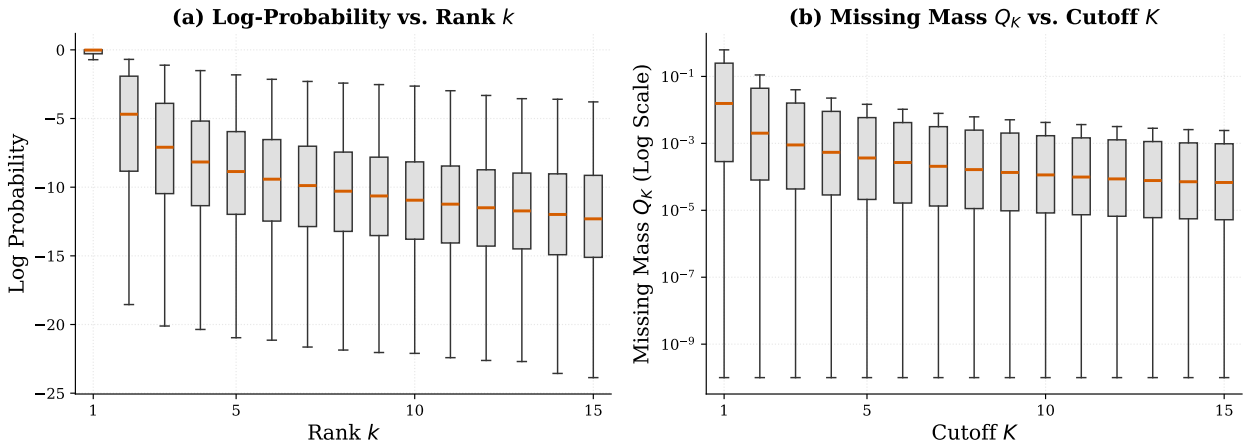


Figure 2 | Analysis of probability mass concentration on 180k tokens generated by `Mistral-Small-3.1-24B` ($T_{\text{samp}} = 1.0$). **(a)** Distribution of log-probabilities for the token at rank k, showing a rapid decay in likelihood. **(b)** Distribution of the missing probability mass Q_K (log scale) when truncating at cutoff K. The median Q_K drops below 10^{-4} by $K = 10$, validating the Top-K approximation.

However, a critical distinction exists between the distribution used for *sampling* tokens and the log-probabilities returned by modern inference engines. Systems like `vllm`, or the OpenAI API, typically expose log-probabilities computed from raw logits (equivalent to fixing $T = 1$ for calculation), regardless of the T_{samp} used to select the token t_j. Consequently, while T_{samp} dictates the trajectory of the generated sequence $\mathcal{T}$, our entropic features (defined in the following sections) measure the model's intrinsic hesitation based on its unscaled raw distribution. This decoupling ensures that our estimator remains mathematically well-defined and stable even if the user employs conservative decoding settings (e.g., low T_{samp}).

3.4. Entropy Production Rate (EPR) of the sequence

Based on the per-token entropy estimate $\tilde{H}_K(q, t_{<j})$, we define the Entropy Production Rate (EPR) for a generated sequence $\mathcal{T}$ of length $|\mathcal{T}|$ in response to query q. It is the average estimated entropy over the top K accessible log-probabilities, across all tokens in the sequence:

$$\begin{aligned} \text{EPR}_K(q, \mathcal{T}) &= \frac{1}{|\mathcal{T}|} \sum_{j=1}^{|\mathcal{T}|} \tilde{H}_K(q, t_{<j}) \\ &= -\frac{1}{|\mathcal{T}|} \sum_{j=1}^{|\mathcal{T}|} \sum_{k=1}^{K} p_{r(k),j} \log_2(p_{r(k),j}). \end{aligned} \tag{6}$$

This EPR_K may serve as a global measure of hesitation or uncertainty for the generated sequence, given the black-box constraints (see Fig. 3) In the following, we improve its alignment with ground-truth annotations via supervised learning.

3.5. Supervision on Entropic Contributions by Rank

We consider a dataset $\mathcal{D}$ where each entry is a query q, the generated sequence $\mathcal{T}$, and an annotation $Y \in \{0, 1\}$ indicating whether $\mathcal{T}$ is a correct (therefore considered non-hallucinatory) answer to q. Note that this annotation depends on the LLM used to generate $\mathcal{T}$. A realistic dataset size is on the order of hundreds to thousands of entries.

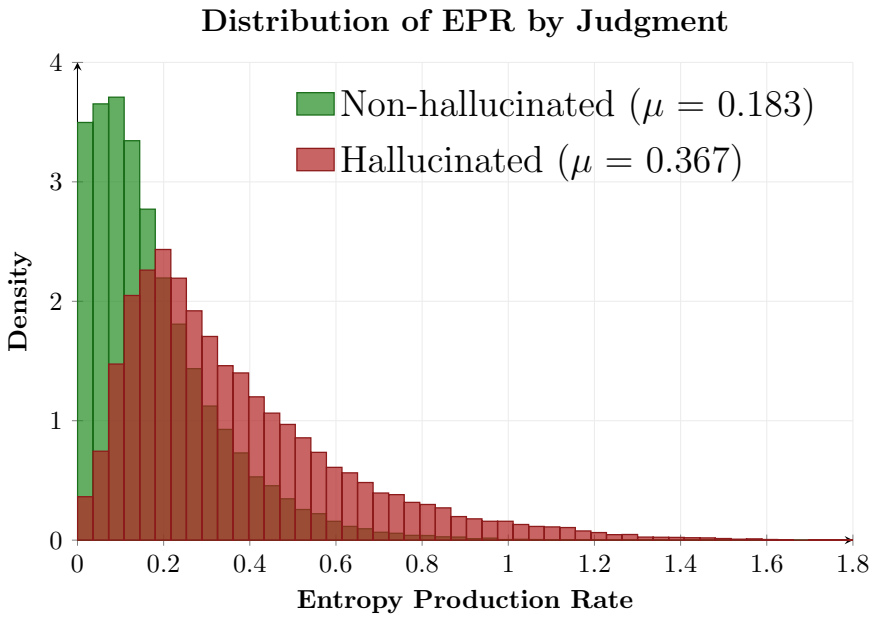


Figure 3 | Distributions of EPR scores on `Falcon-10B` responses. The separation between the *Hallucinated* (red) and *Non-hallucinated* (green) distributions illustrates the discriminative power of the metric.

To build a more nuanced supervised detector, we define features based on the entropic contributions of tokens at specific ranks. For the j-th token generation and given $k \in \{1, \ldots, K\}$, the entropic contribution of the k-th ranked token is

$$s_{k,j} = -p_{r(k),j} \log_2(p_{r(k),j}).$$

Note that the actually sampled token at step j may not be the highest-ranked token ($k = 1$), especially with high T_{samp}. Our estimation leverages the probabilities of the tokens at fixed ranks k, regardless of which token was sampled. We consider weighing each term of $\tilde{H}_K(q, t_{<j})$ (Eq. (3)) according to its rank k:

$$S_\beta(q, t_{<j}) = \beta_0 + \sum_{k=1}^{K} \beta_k s_{k,j}, \tag{7}$$

where $\beta \in \mathbb{R}^{K+1}$ are weights that aim to adapt the entropy measure and are learned using the annotations of $\mathcal{D}$, as described in the following. This token-wise quantity $S_\beta(q, t_{<j})$ is close to $\tilde{H}_K(q, t_{<j})$, but can be negative.

We aggregate the token-wise scores to learn the parameters β at the sequence level. A simple average of $S_\beta(q, t_{<j})$ across the sequence captures the overall model hesitation. However, a single moment of high uncertainty (a "spike") can often be indicative of a hallucination, even if the rest of the sequence is generated confidently, and in such case $s_{k,j}$ values will be high for the uncertain generation, even at high ranks k. To capture these critical tokens, we also consider the maximum entropic contribution $s_{k,j}$ for each rank across the sequence. This leads to our final scoring function, the Weighted Entropy Production Rate (WEPR):

$$\text{WEPR}_{\beta,\gamma}(q, \mathcal{T}) = \frac{1}{|\mathcal{T}|} \sum_{j=1}^{|\mathcal{T}|} S_\beta(q, t_{<j}) + \sum_{k=1}^{K} \gamma_k \max_j s_{k,j} \tag{8}$$

where $\gamma \in \mathbb{R}^K$ are parameters to be learned along with β by maximizing on the annotations of the dataset $\mathcal{D}$:

$$\max_{\beta,\gamma} \sum_{q,\mathcal{T},Y\in\mathcal{D}} Y \log\left[\sigma\left(\mathrm{WEPR}_{\beta,\gamma}(q,\mathcal{T})\right)\right] + (1-Y)\log\left[\sigma\left(1-\mathrm{WEPR}_{\beta,\gamma}(q,\mathcal{T})\right)\right] \tag{9}$$

where σ is the sigmoid function. This corresponds to a logistic regression loss, without regularization, on the averaged contributions $S_\beta(q,t_{<j})$ across tokens and the maximal $s_{k,j}$ values. This way, the signal provided by the entropy along the sequence is aligned to the ground truth annotations.

Beyond this supervision, $\sigma\left(\mathrm{WEPR}_{\beta,\gamma}(q,\mathcal{T})\right) \in [0,1]$ can be interpreted as a confidence score for the generated sequence $\mathcal{T}$ to be valid, with no hallucination. In particular, contrary to $\mathrm{EPR}_K(q)$, this quantity is scaled and can easily be displayed as a normalized score to the user. Moreover, while the annotations are at the sequence level, we can also consider a token-level measure, computed on the entropic contributions features of each token:

$$\sigma\big(S_\beta\left(q,t_{<j}\right)\big) \in [0,1]. \tag{10}$$

This score can be interpreted as the localized risk of hallucination associated with the j-th token, allowing users to identify specific parts of a less reliable response. A significant advantage of this token-level score is its suitability for online estimation: it can be computed and displayed in real-time as the sequence is generated, token by token, without waiting for the full output $\mathcal{T}$.

4. Experiments

4.1. Experimental Setup

4.1.1. Models and Datasets

[3] We evaluated our methods on a suite of contemporary instruction-finetuned LLMs, including `Mistral-Small-3.1-24B-Instruct-2503` [34], `Falcon3-10B-Instruct` [35], `Phi-4` [36], and `Ministral-8B-Instruct-2410` [37]. For Question Answering (QA) tasks, we primarily utilized the TriviaQA [38] (Wikipedia domain) and WebQuestions [39] datasets. We also explored the applicability of our entropic measures in a Retrieval-Augmented Generation (RAG) [40] context using a specialized dataset based on financial reports from the ArGiMi-Ardian Finance dataset [41] to observe how entropy varies with the provision of relevant context (we detail this particular application below). We focused on models generating relatively short answers, typical of non-extended reasoning QA.

4.1.2. Answer Generation and Log-probability Extraction

LLMs were served using the `vllm` inference library [42]. To ensure variability and access to meaningful probability distributions for entropy calculation, we used non-greedy decoding with $T_{\text{samp}} = 1.0$. The `top_p` parameter was set to 1.0, while the sampling cutoff was set to $K_{\text{samp}} = 50$. This truncation allows us to compute the theoretical upper bound on missing entropy defined in Eq. (5). For each generated token, we extracted the accessible top-K log-probabilities (typically $K \leq 15$ or $K \leq 20$ depending on the API/model).

[3] Code and data to reproduce the experiments presented in this paper are available at https://github.com/artefactory/artefactual/releases/tag/ECIR2026.

4.1.3. Hallucination Annotation Protocol

Generated answers for the QA tasks were labeled as hallucinated or non-hallucinated (as displayed in Figure 3) using an LLM-as-a Judge [43–46] approach, employing `Gemma-3-12b-it` [47]. The judge LLM semantically compared the generated answer to the ground-truth answer (and any available aliases). If both answers were judged similar, the generated answer was labeled as non-hallucinated. This automated annotation is scalable and suitable for one-shot analysis. However it may mislabel random correct guesses with high uncertainty or fail to flag confident but incorrect statements (high-certainty hallucinations). To validate its reliability, we conducted a human evaluation. A group of 15 annotators, comprising Ph.D. and graduate students in Machine Learning from our laboratory, hand-labeled a sample set of 1,333 output/ground-truth pairs. The human-machine agreement was exceptionally high: they matched on 1,275 instances (95.7%), yielding a Cohen's Kappa of $\kappa = 0.898$ (see Table 2). This strong alignment confirms that the automated labels used for our large-scale training and evaluation are reliable proxies for human judgment.

4.1.4. RAG Uncertainty Analysis

Beyond general QA, we explored the applicability of our entropic methods for analyzing outputs in RAG pipelines. The premise is that uncertainty, as captured by EPR or WEPR, might increase when an LLM generates an answer without sufficient supporting context. We demonstrated this capability using the `ArGiMi-Ardian Finance 10k` dataset [41]. This open-source resource comprises 52,000 financial annual reports, written in English and extracted from their original PDF format, covering the period from the late 90s to 2024. For our evaluation, we focused on a specific 81-page report from this corpus, from which we manually constructed 50 question/answer/source-page triplets. Answers were generated both with and without the retrieval of the relevant context pages. The task was framed as detecting "missing context" hallucinations—instances where the model hallucinates facts due to the specific absence of the retrieved information—based solely on the entropic signature of the answer.

4.2. Evaluation Protocol

4.2.1. Metrics

The primary metrics used to evaluate the performance of our hallucination detection methods are the Area Under the Precision-Recall Curve (PR-AUC) and the Area Under the Receiver Operating Characteristic Curve (ROC-AUC). As both metrics led to identical conclusions, we report only ROC-AUC scores for conciseness.

4.2.2. Comparison to existing methods

We benchmark our proposed methods, EPR (unsupervised) and WEPR (supervised), against two state-of-the-art black-box detection techniques:

- SelfCheckGPT [11]: A multi-shot method that measures the consistency across several answers generated for the same query. We used the BertScore variant with 10 generated samples.
- HalluDetect [19]: A single-shot method that, similar to our approach, engineers features from model probabilities to train a detector.

This selection allows for a direct comparison with a powerful multi-shot approach (SelfCheckGPT) and a closely related single-shot competitor (HalluDetect), for which we used the same number of accessible log-probabilities (K=15) to ensure a fair comparison.

Dataset	Model	*Unsupervised*		*Supervised*	
		SCGPT [11]	EPR (ours)	HalluDetect [19]	WEPR (ours)
TriviaQA [38] (Hallucination Detection)	`Mistral-Small-24B`	79.0	74.6	78.7	**82.0**
	`Falcon-3-10B`	70.1	75.4	79.0	**84.1**
	`Phi-4 (14.7B)`	71.4	78.2	83.8	**85.4**
	`Ministral-8B-2410`	81.1	81.4	**86.1**	85.8
WebQuestions [39] (Hallucination Detection Generalization)	`Mistral-Small-24B`	59.3	62.5	62.8	**64.8**
	`Falcon-3-10B`	65.8	68.2	69.3	**73.2**
	`Phi-4 (14.7B)`	65.0	65.2	66.3	**66.6**
	`Ministral-8B-2410`	66.2	65.4	71.6	**72.6**
ArGiMi-Ardian [41] (Missing Context Detection)	`Mistral-Small-24B`	64.2	81.0	81.1	**82.8**
	`Falcon-3-10B`	67.6	82.8	87.0	**89.3**
	`Phi-4 (14.7B)`	79.7	91.0	88.5	**93.6**
	`Ministral-8B-2410`	65.9	73.1	81.6	**81.8**

Table 1 | ROC-AUC of compared methods across different LLMs, using $K = 15$ accessible log-probabilities, on: (i) hallucination detection on TriviaQA; (ii) generalization to WebQuestions (supervised models trained on TriviaQA); and (iii) missing context detection on ArGiMi-Ardian (also trained on TriviaQA).

4.2.3. Training and Evaluation Strategy

The supervised logistic regression model have been trained only on TriviaQA data (generated sequences and their labels), as being the only dataset with annotation. For evaluating these methods on TriviaQA, we generated training and testing splits by grouping the data points by their original query. This process ensures that no data points originating from the same query appear in the training and testing sets for a given fold, mitigating potential data leakage and overestimating generalization performance. To ensure stability and robust performance estimation, particularly given potential variability in smaller datasets or specific model-dataset pairings, we employed bootstrapping on 1000 iterations during model evaluation.

4.3. Results

4.3.1. Hallucination Classification Performance

Table 1 shows results on TriviaQA for hallucination detection. The EPR baseline demonstrates notable discriminative ability, with ROC-AUC scores ranging from 74.6 (`Mistral-Small-3.1-24B`) to 81.4 (`Ministral-8B-Instruct-2410`). For a similar computational overhead as other single-shot methods (around 80 ± 20 µs per score) WEPR consistently matches or outperforms EPR, SelfCheckGPT, and HalluDetect across all LLMs. Note that multi-shot methods are multiple orders of magnitude more

computationally costly (at least 10s for SelfCheckGPT). We also verified the robustness of our approach to sampling parameters; for instance, on `Falcon-3-10B`, reducing the temperature to $T_{\text{samp}} = 0.6$ resulted in a negligible change in ROC-AUC (< 1.0 point difference). Since the log-probabilities used for WEPR features are derived from raw logits, this result confirms that our entropic signal remains highly discriminative even when applied to the more deterministic generation trajectories typical of conservative decoding. Regarding the generalization on WebQuestions (Table 1), we observe that although performance is generally lower on this dataset, WEPR maintains its advantage across LLMs.

The improved classification performance of the WEPR model is further illustrated by the sample Precision-Recall curves for `Falcon-3-10B` on TriviaQA (Figure 5), where the WEPR curve consistently dominates that of the EPR baseline and the HalluDetect [19] method.

4.3.2. RAG uncertainty

Table 1 presents the performance of compared methods (trained on the general TriviaQA dataset for supervised ones) for the missing context detection task. The results show that the WEPR model significantly outperforms the other methods in this domain-specific RAG scenario. This indicates that an entropic detector trained on a general QA dataset can transfer, to a notable extent, to specialized domains for identifying responses generated with insufficient context. Such a mechanism can be readily applied in RAG pipelines to flag outputs where context retrieval may have been inadequate or where the model is otherwise struggling, thereby triggering deeper information retrieval or alerting the user.

4.3.3. Effect of the Number of Accessible Log-probabilities (K)

We investigated the impact of K on hallucination detection performance. Figure 4 displays the evolution of the mean ROC-AUC for `Falcon-3-10B` on TriviaQA as K varies. The results indicate that performance generally improves with increasing ranked token contributions up to a certain point. Notably, for several models, the ROC-AUC tends to plateau when K reaches approximately 8 to 10. This suggests that incorporating entropic contributions beyond the top ~10 ranks may yield diminishing returns for hallucination detection. This finding has practical significance, implying that effective detection can be achieved even when API access is limited to a relatively small number of top logprobs, thereby reducing data requirements and potentially API costs.

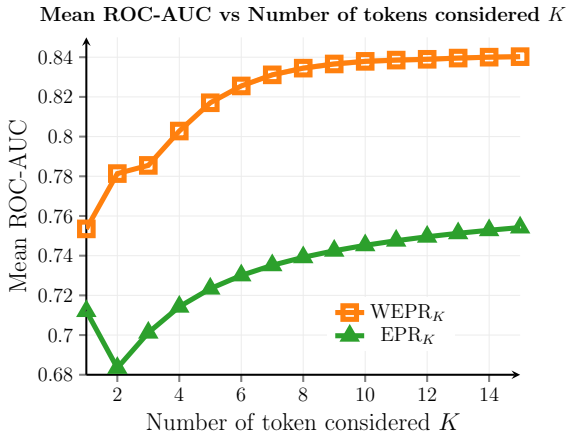


Figure 4 | Mean ROC-AUC with respect to K for Falcon-3 on TriviaQA.

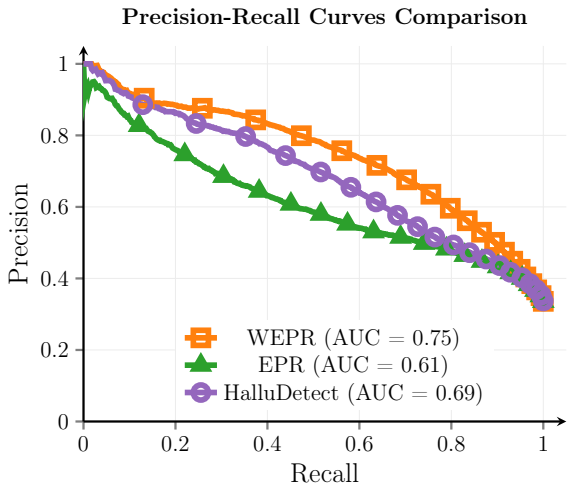


Figure 5 | Precision-Recall curves of 3 methods for Falcon-3 on TriviaQA.

Task annotated	LLM-as-a-Judge	Sentence-level	Token-level
Samples	1,333	312	334
Agreement (%)	95.7	77.8	78.1

Table 2 | Human agreement with (i) LLM annotations validating labels in Section 4.3, and (ii) WEPR scores at sentence and token granularity, measuring adoption potential.

4.3.4. Human adoption analysis

We wanted to measure the perceived downstream quality of our WEPR scoring methods, at the two possible granularities: sentence-wise ($\sigma\left(\text{WEPR}_{\beta,\gamma}(q,\mathcal{T})\right)$ and token-wise (Eq. 10). At least three independent annotators (from the same pool of researchers) rated each sample as acceptable or not (mean 3.76). The adoption score is then computed based on the majority vote, with results compiled in Table 2. The high level of satisfaction for token scores (78.1%) indicates that the learned coefficients β_k from our WEPR model can be used to compute a weighted entropic contribution for each token j in a sequence, thus offering a finer-grained view of uncertainty hotspots within the generated text (see text colors of Figure 1).

5. Limitations

While our methods demonstrate promising results for one-shot, black-box hallucination detection, several limitations warrant acknowledgment. First, our experiments were conducted on mid-sized LLMs (approximately 8B–24B parameters). Performance on bigger models (e.g., 100B+ parameters or closed-source APIs like GPT-5) remains to be tested. Second, the current study focuses on QA tasks that yield relatively short, factual answers. We did not study tasks requiring extensive multi-step reasoning or lengthy outputs, where the aggregated entropy signal might become less distinct or different types of uncertainty could dominate. Moreover, the method may produce false positives by flagging model hesitation related to stylistic choices or sentence formulation, as opposed to uncertainty about the factual content of the answer. This effect was observed to be more pronounced at the beginning of generated sequences. Finally, our approach, like many other uncertainty-based

methods, is inherently limited in its ability to detect "high-certainty hallucinations" [30]—instances where the model generates incorrect information with low output entropy (i.e., high confidence) due to memorization errors. If an LLM has strongly learned erroneous facts during its training, entropic measures based on output probabilities may not flag such confident fabrications.

6. Conclusion

We addressed one-shot hallucination detection in black-box LLMs by introducing a methodology rooted in information-theoretic principles, leveraging only the top-K token log-probabilities typically exposed by APIs. Our primary contribution is a supervised approach that utilizes features representing individual top-ranked tokens mean and maximal entropic contributions. Experiments across QA datasets and multiple LLMs show that the learned estimator outperforms a baseline Entropy Production Rate (EPR). Notably, strong detection performance is achieved even with as few as $K \leq 10$ accessible log-probabilities, underscoring practical efficiency under API constraints. We also showcased utility in a specialized financial setting for RAG and highlighted token-level uncertainty visualization.

This work provides a practical technique for enhancing the trustworthiness of LLM responses in both general QA and RAG from a single generation pass. Future work may extend this approach to more complex reasoning tasks and study the interpretability of the learned entropic weights.

Acknowledgements

This work was done as part of the ArGiMi project, funded by BPIFrance under the France2030 national effort towards numerical common goods. CM wishes to thank Antoine Delaby for preliminary discussions on RAG applications, along with Gabriel Trier, Lucas Tramonte and Rayane Bouaita for their help on the question dataset. HR also wishes to thank Yves-Roland Kouayip and Mathieu Crochet from Artefact for their help in creating the annotation tool.

References

[1] Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. Language Models as Knowledge Bases? In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pages 2463–2473, Hong Kong, China, 2019. Association for Computational Linguistics.

[2] Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed H. Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, and William Fedus. Emergent Abilities of Large Language Models, October 2022.

[3] OpenAI et. al. GPT-4 Technical Report, March 2024.

[4] Anna Rohrbach, Lisa Anne Hendricks, Kaylee Burns, Trevor Darrell, and Kate Saenko. Object Hallucination in Image Captioning, March 2019.

[5] S. Baker and T. Kanade. Hallucinating faces. In Proceedings Fourth IEEE International Conference on Automatic Face and Gesture Recognition (Cat. No. PR00580), pages 83–88, March 2000.

[6] Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Yejin Bang, Delong Chen, Wenliang Dai, Ho Shu Chan, Andrea Madotto, and Pascale Fung. Survey of Hallucination in Natural Language Generation. ACM Computing Surveys, 55(12):1–38, December 2023.

[7] Gaurang Sriramanan, Siddhant Bharti, Vinu Sankar Sadasivan, Shoumik Saha, Priyatham Kattakinda, and Soheil Feizi. Llm-check: Investigating detection of hallucinations in large language models. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, editors, Advances in Neural Information Processing Systems, volume 37, pages 34188–34216. Curran Associates, Inc., 2024.

[8] Yubin Kim, Hyewon Jeong, Shan Chen, Shuyue Stella Li, Chanwoo Park, Mingyu Lu, Kumail Alhamoud, Jimin Mun, Cristina Grau, Minseok Jung, et al. Medical hallucinations in foundation models and their impact on healthcare. arXiv preprint arXiv:2503.05777, 2025.

[9] Fang Liu, Yang Liu, Lin Shi, Houkun Huang, Ruifeng Wang, Zhen Yang, Li Zhang, Zhongqi Li, and Yuchi Ma. Exploring and Evaluating Hallucinations in LLM-Powered Code Generation, May 2024.

[10] T. J. Sullivan. Introduction to Uncertainty Quantification. Springer, December 2015.

[11] Potsawee Manakul, Adian Liusie, and Mark J. F. Gales. SelfCheckGPT: Zero-Resource Black-Box Hallucination Detection for Generative Large Language Models, October 2023.

[12] Xiaoou Liu, Tiejin Chen, Longchao Da, Chacha Chen, Zhen Lin, and Hua Wei. Uncertainty Quantification and Confidence Calibration in Large Language Models: A Survey, March 2025.

[13] Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, et al. Language models (mostly) know what they know. arXiv preprint arXiv:2207.05221, 2022.

[14] Amos Azaria and Tom Mitchell. The Internal State of an LLM Knows When It's Lying, October 2023.

[15] Yarin Gal. Uncertainty in Deep Learning. PhD thesis, University of Cambridge, 2016.

[16] Eyke Hüllermeier and Willem Waegeman. Aleatoric and epistemic uncertainty in machine learning: An introduction to concepts and methods. Machine learning, 110(3):457–506, 2021.

[17] Janis Postels, Francesco Ferroni, Huseyin Coskun, Nassir Navab, and Federico Tombari. Sampling-free epistemic uncertainty estimation using approximated variance propagation. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 2931–2940, 2019.

[18] Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In International Conference on Machine Learning, pages 1050–1059. PMLR, 2016.

[19] Ernesto Quevedo, Jorge Yero Salazar, Rachel Koerner, Pablo Rivas, and Tomas Cerny. Detecting hallucinations in large language model generation: A token probability approach. In World Congress in Computer Science, Computer Engineering & Applied Computing, pages 154–173. Springer, 2024.

[20] C. E. Shannon. A mathematical theory of communication. The Bell System Technical Journal, 27(3):379–423, July 1948.

[21] Marylou Gabrié, Andre Manoel, Clément Luneau, jean barbier, Nicolas Macris, Florent Krzakala, and Lenka Zdeborová. Entropy and mutual information in models of deep neural networks. In Advances in Neural Information Processing Systems, volume 31. Curran Associates, Inc., 2018.

[22] Wonbin Kweon, Sanghwan Jang, SeongKu Kang, and Hwanjo Yu. Uncertainty Quantification and Decomposition for LLM-based Recommendation. In Proceedings of the ACM on Web Conference 2025, pages 4889–4901, April 2025.

[23] Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. Semantic Uncertainty: Linguistic Invariances for Uncertainty Estimation in Natural Language Generation, April 2023.

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

[25] Nicola Cecere, Andrea Bacciu, Ignacio Fernández Tobías, and Amin Mantrach. Monte Carlo Temperature: A robust sampling strategy for LLM's uncertainty quantification methods, February 2025.

[26] Litian Liu, Reza Pourreza, Sunny Panchal, Apratim Bhattacharyya, Yao Qin, and Roland Memisevic. Enhancing Hallucination Detection through Noise Injection, February 2025.

[27] Eojin Joo, Young-Jun Lee, and Ho-Jin Choi. Entropy-based Sentence-level Hallucination Score in Large Language Models. In 2025 IEEE International Conference on Big Data and Smart Computing (BigComp), pages 77–78, February 2025.

[28] Jannik Kossen, Jiatong Han, Muhammed Razzak, Lisa Schut, Shreshth Malik, and Yarin Gal. Semantic Entropy Probes: Robust and Cheap Hallucination Detection in LLMs, June 2024.

[29] Jinhao Duan, Hao Cheng, Shiqi Wang, Alex Zavalny, Chenan Wang, Renjing Xu, Bhavya Kailkhura, and Kaidi Xu. Shifting Attention to Relevance: Towards the Predictive Uncertainty Quantification of Free-Form Large Language Models, May 2024.

[30] Adi Simhi, Itay Itzhak, Fazl Barez, Gabriel Stanovsky, and Yonatan Belinkov. Trust Me, I'm Wrong: High-Certainty Hallucinations in LLMs, February 2025.

[31] Huan Ma, Jingdong Chen, Joey Tianyi Zhou, Guangyu Wang, and Changqing Zhang. Estimating LLM Uncertainty with Evidence, May 2025.

[32] Leon Brillouin. Science and Information Theory. Courier Corporation, July 2013.

[33] Marc Mezard and Andrea Montanari. Information, physics, and computation. Oxford University Press, 2009.

[34] Mistral Small 3.1 | Mistral AI. https://mistral.ai/news/mistral-small-3-1.

[35] TII Falcon-LLM Team. The falcon 3 family of open models. https://huggingface.co/blog/falcon3, December 2024.

[36] Marah Abdin, Jyoti Aneja, Harkirat Behl, Sébastien Bubeck, Ronen Eldan, Suriya Gunasekar, Michael Harrison, Russell J Hewett, Mojan Javaheripi, Piero Kauffmann, et al. Phi-4 technical report. arXiv preprint arXiv:2412.08905, 2024.

[37] Un Ministral, des Ministraux | Mistral AI. https://mistral.ai/fr/news/ministraux.

[38] Mandar Joshi, Eunsol Choi, Daniel S. Weld, and Luke Zettlemoyer. TriviaQA: A Large Scale Distantly Supervised Challenge Dataset for Reading Comprehension, May 2017.

[39] Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. Semantic Parsing on Freebase from Question-Answer Pairs. In David Yarowsky, Timothy Baldwin, Anna Korhonen, Karen Livescu, and Steven Bethard, editors, Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, pages 1533–1544, Seattle, Washington, USA, October 2013. Association for Computational Linguistics.

[40] 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, April 2021.

[41] Hicham Randrianarivo, Charles Moslonka, Arthur Garnier, and Emmanuel Malherbe. The ArGiMi datasets. https://huggingface.co/datasets/artefactory/Argimi-Ardian-Finance-10k-text, 2024.

[42] Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with PagedAttention. In Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles, 2023.

[43] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena, December 2023.

[44] Peiyi Wang, Lei Li, Liang Chen, Zefan Cai, Dawei Zhu, Binghuai Lin, Yunbo Cao, Qi Liu, Tianyu Liu, and Zhifang Sui. Large Language Models are not Fair Evaluators, August 2023.

[45] Vaibhav Adlakha, Parishad BehnamGhader, Xing Han Lu, Nicholas Meade, and Siva Reddy. Evaluating Correctness and Faithfulness of Instruction-Following Models for Question Answering. Transactions of the Association for Computational Linguistics, 12:681–699, May 2024.

[46] Hossein A. Rahmani, Emine Yilmaz, Nick Craswell, Bhaskar Mitra, Paul Thomas, Charles L. A. Clarke, Mohammad Aliannejadi, Clemencia Siro, and Guglielmo Faggioli. LLMJudge: LLMs for Relevance Judgments, August 2024.

[47] Gemma et. al. Team. Gemma 3 Technical Report, March 2025.