Computer and Decision Making

An International Journal

www.comdem.org
eISSN: 3008-1416

Hallucination Detection in Large Language Models via Multi-Granular Uncertainty Quantification

Abdullah Önden[1*]

[1] Department of Computer Engineering, Faculty of Computer and Information Technologies, Istanbul University, Istanbul, Türkiye

ARTICLE INFO

Article history:
Received 25 January 2026
Received in revised form 8 March 2026
Accepted 16 March 2026
Available 19 March 2026

Keywords:
hallucination detection; uncertainty quantification; large language models; temporal entropy dynamics; calibration; XGBoost

ABSTRACT

Hallucination, when large language models (LLMs) produce plausible but factually incorrect output, is a major challenge in high-stakes applications such as medicine, law, and education. Current detection methods involve a trade-off between accuracy and efficiency: multi-generation methods (e.g., semantic entropy) are effective but impose 5-10x increased latency, while single-pass methods are faster but attain only 63-68% AUROC. To balance these trade-offs, we propose a framework that aggregates 12 uncertainty features across token-level, sequence-level, temporal, and distributional granularities from a single autoregressive generation. The framework operates in Full Mode (12 features, open-source models with attention access) or API Mode (10 features, any model exposing token log-probabilities). The most novel component is F9, temporal entropy dynamics, which measures how the entropy of generated segments changes across four quarters of the generation process. On Llama-3-8B, the framework attains 89.27% AUROC on HaluEval, surpassing semantic entropy by 2.15 percentage points while reducing latency by 8.2x. Across four open-source model families and five benchmarks, Full Mode consistently improves over semantic entropy by 1.71 to 2.47 pp. On GPT-3.5-Turbo, API Mode achieves 88.63% AUROC, falling below semantic entropy (90.81%) on this model. These results demonstrate that a suitably chosen combination of single-pass uncertainty features can approach the discrimination offered by more computationally intensive multi-generation methods.

1. Introduction

Hallucination in large language models (LLMs) refers to generated text that is not only fluent and seemingly reasonable but also factually false [1,2]. Hallucinations have consequences: Medical hallucinations endanger patients [3], fabricated citations pose serious risks to legal and professional applications [4], and hallucinated references have been introduced into the scientific literature [2]. Hallucination detection is thus a necessary prerequisite for safe deployment of LLMs in high-stakes applications [5]. The rapid adoption of LLMs such as ChatGPT across enterprise settings and software development contexts [6] makes robust detection mechanisms increasingly critical.

* *Corresponding author.*
E-mail address: abdullah.onden@istanbul.edu.tr

https://doi.org/10.59543/comdem.v3i.17665

Existing detection methods are generally grouped into three categories, each with significant limitations. Multi-generation consistency methods, such as semantic entropy [7] and SelfCheckGPT [8], achieve competitive discrimination in our evaluation setting, but with orders-of-magnitude higher latency. Internal representation methods [9,10] train probes on the hidden states to achieve 80 to 86% AUROC with low overhead but require white-box model access and fail to transfer across model families. Single-metric methods, based on perplexity [11] or verbalized confidence [12], are highly efficient but achieve only 63 to 68% AUROC due to the high local confidence of hallucinations.

Recently, there has been some work on exploring entropy-based signals from single generation. Pramanik et al. [13] compute per-head attention entropy and Joo et al. [14] aggregate token entropy at the sentence level. However, both methods only operate at a single level of granularity and capture only a static snapshot of uncertainty. Neither tracks the uncertainty during the generation process nor considers the signals from calibration or interactions between different granularity levels. This indicates that single-pass methods may potentially benefit from the combination of complementary uncertainty signals at token, sequence, and temporal levels.

To fill this knowledge gap, we propose a framework that extracts 12 uncertainty features from a single forward pass, covering three levels of granularity and temporal dynamics. Our framework is structured into two modes of deployment: Full Mode for open-source models where attention weights are accessible and API Mode for closed-source APIs that only provide token probability distribution, in order to conduct transparent and reproducible comparison. The key novel component in our framework is the temporal entropy dynamics (F9), which divides the generated sequence into temporal segments and calculates how the entropy changes across the segments. As shown in our sample analysis, this feature captures certain patterns that are indicative of hallucinated samples such as the U-shape and oscillating patterns.

The main contributions of this work are as follows. First, we introduce a dual-mode single-pass framework for hallucination detection that explicitly separates Full Mode (12 features, open-source models) from API Mode (10 features, any model exposing token log-probabilities), making the access assumptions explicit where prior work left them ambiguous. Second, we propose temporal entropy dynamics (F9), a feature that captures variation in segment-level entropy across generation quarters, and demonstrate that it provides the largest single-feature contribution to detection performance (−11.13 pp AUROC upon removal; SHAP = 0.187). Third, we provide broad experimental validation across five benchmarks and five model families, with per-seed reporting, mode-separated results, and ablation and sensitivity analyses. Fourth, we present a practical analysis of latency and deployment cost, demonstrating that the framework's detection overhead is approximately 8.2× lower than semantic entropy, and release all code, trained classifiers, and evaluation scripts.

2. Related Work

2.1 Multi-Generation Consistency Methods

The concept of semantic uncertainty was first formulated in [15], where the authors proposed to use linguistic invariances for uncertainty estimation in NLG. A recent formulation of semantic uncertainty is the semantic entropy [7], which samples 5-10 responses per query, clusters them using DeBERTa-v3-large entailment, and computes entropy over cluster distributions. This is a strong baseline among API-accessible methods. Our reimplementation on HaluEval achieves 87.12% AUROC.

SelfCheckGPT [8] samples five additional responses per query, and measures their mutual consistency using BERTScore. A recent work [16] proposed budget-efficient approximations of semantic entropy using Bayesian estimation, which reduces the number of samples required while maintaining much of the discriminative signal. Our reimplementation on HaluEval achieves 84.73% AUROC at 252 ms per query. The key bottleneck of these multi-generation methods is efficiency: at scale, the additional overhead of multiple completions is far higher than single completion methods. Our methods differ in that they compute all features in a single forward pass, without the need for multiple completions.

2.2 Single-Generation Entropy Methods

Pramanik et al. [13] estimate Shannon entropy per attention head across the final four transformer layers. In our reproduction on HaluEval, this achieves approximately 85.19% AUROC. Joo et al. [14] aggregate token entropy at sentence boundaries. In our reproduction on HaluEval, this achieves approximately 83.07% AUROC. Both methods confirm that entropy-based signals carry discriminative information about hallucinations. However, each operates at a single granularity and captures only static uncertainty. Recent approaches also explore geometric properties of the probability simplex as uncertainty signals [17]. Our framework extends these approaches along three axes: multi-granularity feature extraction, temporal dynamics, and confidence-consistency features that target high-confidence hallucinations in which local entropy is low despite the generated content being factually incorrect.

2.3 Internal Representation Methods

Azaria and Mitchell [9] demonstrated that LLM hidden states encode truthfulness signals, training linear probes over intermediate activations to achieve 80-85% accuracy. Complementary analysis frameworks such as Luna [18] provide broader model-level inspection tools for LLM behaviour. Li et al. [10] extended this approach with INSIDE, examining multi-layer activations to achieve approximately 86% AUROC. Li et al. [19] further propose inference-time intervention (ITI), directing activations along truth-correlated directions identified in probing experiments. These methods are competitive but require white-box access to model parameters, which is unavailable for commercial APIs. Our framework requires only token probability distributions (API Mode) or token probability distributions plus final-layer attention weights (Full Mode), without access to intermediate hidden states.

2.4 Knowledge-Grounded and Retrieval-Based Methods

Knowledge-based methods such as FacTool [20] decompose claims and verify them against external databases, detecting factual errors that uncertainty-only methods may miss. Task-specific detection pipelines have been explored in shared-task settings [21], combining model-agnostic and model-aware signals for domain-specific hallucination benchmarks. However, they require curated knowledge bases and introduce additional latency. Chain-of-verification [22] prompts models to self-check, but depends on model cooperativeness. Our framework operates without external knowledge, making it complementary to retrieval-based approaches.

2.5 Calibration-Based Confidence Methods

Liu et al. [23] report substantial overconfidence and non-trivial calibration error in modern LLMs. Broader surveys of uncertainty quantification in LLMs [24,25] document calibration as an open challenge across model families. Kadavath et al. [12] evaluate P(True), which prompts the model to assess the probability that a statement is true, achieving 63.18% AUROC. Raw perplexity [11] achieves 68.41% AUROC. We incorporate confidence-consistency signals not as a preprocessing step but as detection features (F7, F8), enabling the framework to flag high-confidence hallucinations that mislead both raw-probability and uncalibrated entropy measures.

3. Methodology

3.1 Dual-Mode Architecture

Prior work on model-agnostic hallucination detection has been ambiguous about the access requirements it assumes. Some methods claim model-agnosticity while implicitly using attention weights, which closed-source APIs (e.g., GPT-3.5-Turbo, GPT-4, Claude, Gemini) do not expose. The need for transparent deployment modes also aligns with decision support frameworks for AI-integrated systems [26]. We address this by defining two explicit operating modes.

Full Mode uses all 12 features (F1-F12) and requires token probability distributions and final-layer attention weights. It is applicable to open-source models (e.g., Llama-3, Mistral, Qwen) deployed locally or via compatible inference APIs. API Mode uses ten features (F1-F9 and F12) and requires only token probability distributions; F10 and F11 (attention-based) are omitted. It is applicable to any LLM that exposes token log-probabilities, including closed-source APIs.

The framework is architecture-level model-agnostic in the sense that it does not depend on any specific model family's internal architecture. However, it is deployment-dependent with respect to available model outputs: Full Mode requires attention weights, and even API Mode requires token log-probabilities. Results for open-source models are reported in Full Mode (Table 1); results for GPT-3.5-Turbo are reported in API Mode (Table 2). The two modes are presented separately to avoid conflation. Importantly, API Mode trains a separate 10-feature XGBoost classifier from scratch, rather than zeroing features in a 12-feature model.

3.2 Feature Extraction Framework

Let M denote a language model, x an input prompt, y = (y1, ..., yT) the output sequence of T tokens, pt the probability distribution over vocabulary V at position t, and at the final-layer attention weight vector at position t (Full Mode only). We extract features at three granularity levels from a single autoregressive generation.

3.2.1 Token-Level Features (F1-F4)

F1 (Mean Token Entropy) captures average per-token distributional uncertainty. Alone, F1 is insufficient because hallucinations may exhibit low mean entropy when the model is confidently incorrect. F2 (Entropy Variance) measures temporal instability of per-token entropy. Hallucinated outputs tend to show oscillating entropy, while factual responses exhibit more stable profiles. F3 (Top-5 Probability Mass) captures the concentration of probability on the top five candidates. Values near 1.0 indicate decisive generation; dispersed mass indicates genuine ambiguity. F4 (Confidence

Gap) measures the margin between the top two candidates, capturing decisiveness independently of absolute probability magnitude. All four features are available in both modes.

3.2.2 Sequence-Level Features (F5-F8)

F5 (Calibrated Perplexity) applies temperature scaling with t = 1.3, tuned on the HaluEval validation split. Sensitivity analysis over t in {1.0, 1.1, 1.2, 1.3, 1.5, 2.0} shows less than 0.4 pp AUROC variation. F6 (Length-Normalised Uncertainty) removes the spurious positive correlation between sequence length and aggregate entropy. F7 (Token-Level Confidence-Consistency Proxy) is inspired by expected calibration error and partitions tokens into B = 10 bins by predicted confidence; it measures decoding self-consistency rather than factual accuracy. For sequences shorter than approximately 60 tokens, many bins contain fewer than five tokens, making F7 statistically unreliable; users should set B = 5 for shorter sequences. F8 (Maximum Confidence-Consistency Error) identifies the single most overconfident bin, capturing high-confidence generation inconsistencies that the averaging-based F7 may mask. All four features are available in both modes.

3.2.3 Temporal and Distributional Features (F9-F12)

F9 (Temporal Entropy Dynamics) partitions the output sequence into K = 4 equal segments (ablation over K in {2, 3, 4, 5, 6, 8} in Table 6) and computes the total variation of mean token entropy across segments. In an empirical analysis of 500 hallucinated outputs, we observe that 34% exhibit U-shaped entropy profiles and 19% exhibit oscillating profiles, whereas factual outputs tend to show stable entropy across segments. F9 provides the largest single-feature contribution, with removal reducing AUROC by 11.13 pp. F10 (Attention-Weighted Entropy) reweights token-level entropy by final-layer attention importance and is available in Full Mode only. F11 (Cross-Token Attention Variance) measures the dispersion of attention weights across the sequence and is also Full Mode only. F12 (Distributional Coherence Proxy) computes cosine similarity between the mean output distributions of the first and second halves of the sequence, measuring distributional coherence rather than semantic similarity. Its contribution is small (SHAP = 0.014, 2.0%) and is retained for completeness.

3.3 Theoretical Motivation

Given 12 uncertainty features with low pairwise correlation (empirically, Pearson $|r| < 0.41$ between all feature pairs), the joint feature distribution is expected to be more informative for discriminating hallucinated from factual outputs than any single feature. Low pairwise correlation is consistent with complementarity but does not guarantee additive information gain, particularly when features share non-linear dependencies. In our experiments, SHAP interaction effects account for 23% of total predictive power, confirming that features are not fully independent. The compositional gain is empirically measurable: the 12-feature model achieves 89.27% AUROC versus 76.80% for F9 alone, a +12.47 pp improvement.

3.4 Classifier Architecture and Training

We use XGBoost as the classifier for its strong performance on tabular data with moderate feature counts, its built-in feature importance measures, and its suitability for SHAP analysis. Hyperparameters were tuned via Bayesian optimisation (100 trials in Optuna) on the HaluEval

validation split: 100 trees, maximum depth 6, learning rate 0.05, subsample ratio 0.8, minimum child weight 5, binary logistic objective. No class weighting was applied (HaluEval is balanced at 50/50). No post-hoc probability calibration was applied to the classifier outputs. The decision threshold for binary classification was set at 0.5, the default for balanced datasets. A separate XGBoost classifier is trained for each model family to account for differences in token probability distributions.

3.5 Training and Transfer Protocol

HaluEval [27] provides 35,000 samples (5,000 general + 30,000 task-specific). We construct a 60/20/20 split (21,000/7,000/7,000 samples) stratified by class, held constant across all experiments. All classifiers are trained on the training split, with hyperparameters tuned on the validation split. The test split is used only for final evaluation. For cross-dataset experiments (Table 3), the HaluEval-trained classifier is transferred to each target dataset without retraining or threshold adjustment. For cross-model experiments (Table 4), a separate classifier is trained per model family using the same HaluEval training split but with features extracted from that model's outputs. API Mode classifiers are trained from scratch on 10-dimensional feature vectors, not derived from Full Mode classifiers.

3.6 Computational Complexity

All 12 features are computed in O(T * |V|) time, which is dominated by the model's own generation cost. Feature extraction adds 11.2 ms overhead (19% of base inference time); classifier inference adds 0.8 ms. End-to-end latency on an A100 GPU is 59.0 ms (model: 42.3 ms, features: 11.2 ms, classifier: 0.8 ms, overhead: 4.7 ms), representing an 8.2x latency improvement over semantic entropy (487 ms).

4. Experimental Setup

4.1 Datasets

We evaluate on five benchmarks spanning diverse tasks, domains, and hallucination types: HaluEval (35,000 samples; primary benchmark; 50/50 class balance; our 60/20/20 split as described in Section 3.5), TruthfulQA [28] (817 adversarial questions; approximately 60% hallucinated), FEVER [29] (5,000-sample test subset; 50/50 class balance; fact verification against Wikipedia), FRANK [30] (2,250 samples; 65/35 class balance; factual consistency in abstractive summarisation), and FactCC [31] (931 samples; 70/30 class balance; document-grounded factual consistency).

4.2 Models

We evaluate five model families: Llama-3-8B-Instruct, Llama-3-70B-Instruct (Meta), Mistral-7B-Instruct-v0.3, Qwen-14B-Chat, and GPT-3.5-Turbo (OpenAI API; undisclosed parameter count; API Mode only). All models are run with temperature = 0.7, top-p = 0.9, and max tokens = 512.

4.3 Baseline Methods

We compare against eight baselines: Semantic Entropy [7] (10 responses per query, DeBERTa-v3-large entailment clustering), SE Probes [9] (linear probes on hidden states without multiple generations), Attention Entropy [13] (per-head entropy on final four layers), Sentence Entropy [14] (sentence-boundary token entropy aggregation), SelfCheckGPT [8] (five additional samples,

BERTScore consistency), Verbalized Confidence, P(True) [12] (token probability of "Yes" as confidence), and Perplexity [11].

4.4 Baseline Implementation and Fairness

All baselines were reproduced by us on the same hardware (NVIDIA A100 GPUs), using the same test splits and prompts. All methods were evaluated with three random seeds (42, 123, 456); we report mean +/- standard deviation. Statistical significance is assessed via paired t-tests ($p < 0.05$) and bootstrap resampling (10,000 iterations) for confidence intervals. Hidden-state access was not used for our method; baselines requiring such access were given full access to the necessary internal representations for a fair comparison. Decision thresholds for all methods were tuned on the HaluEval validation split.

4.5 Sensitivity Analyses

Temperature scaling: t in {1.0, 1.1, 1.2, 1.3, 1.5, 2.0}, stable (< 0.4 pp variation); t = 1.3 selected. Temporal segmentation: K in {2, 3, 4, 5, 6, 8} (Table 6). Bin count for F7/F8: B in {5, 10, 15}, < 0.3 pp AUROC variance. Full sensitivity tables are in the Supplementary Material.

4.6 Reproducibility

All code, trained XGBoost classifiers, and evaluation scripts are publicly available at the anonymous repository (https://anonymous.4open.science/r/mgUQ-hallucination). Random seeds: 42, 123, 456. Hardware: NVIDIA A100 80GB GPUs. Major libraries: XGBoost 2.0, Transformers 4.36, SHAP 0.44. Dataset splits are fixed and included in the repository.

5. Results

5.1 Full Mode on HaluEval

Table 1 presents results on the HaluEval test set (7,000 samples, Llama-3-8B, Full Mode). Our method achieves 89.27 +/- 0.31% AUROC, outperforming our reproduction of semantic entropy (87.12 +/- 0.41%) by 2.15 pp, with 8.2x reduced latency. It outperforms the two nearest single-generation baselines by +4.08 pp (Attention Entropy) and +6.20 pp (Sentence Entropy). All differences are statistically significant ($p < 0.05$, paired t-test).

Table 1
HaluEval test set performance (Full Mode, Llama-3-8B, three seeds). Full Mode uses all 12 features including attention-based F10 and F11

Method	AUROC	Acc.	F1	Lat.(ms)	Speedup	Mode	Seeds
Sem. Entropy [7]*	87.12±0.41	81.08±0.33	80.71±0.44	487±23	1.0×	Multi	3
SE Probes*	84.31±0.57	78.44±0.47	77.96±0.55	63±5	7.7×	Single	3
Attn. Entropy [13]*	85.19±0.52	79.63±0.41	79.11±0.53	61±4	8.0×	Single	3
Sent. Entropy [14]*	83.07±0.64	77.31±0.52	76.82±0.61	55±3	8.9×	Single	3
SelfCheckGPT [8]*	84.73±0.48	78.87±0.43	78.29±0.51	252±18	1.9×	Multi	3
Verb. Confidence*	61.84±1.31	57.93±0.82	57.12±0.94	115±9	4.2×	Single	3
P(True) [12]*	63.18±1.24	58.97±0.79	58.31±0.92	98±7	5.0×	Single	3
Perplexity [11]*	68.41±0.93	63.14±0.71	62.47±0.83	5±1	97.4×	Single	3
Ours (Full Mode)	89.27±0.31	83.52±0.28	83.04±0.37	59±4	8.2×	Single	3

* All baselines were reproduced on identical hardware (A100), test splits, and seeds. Decision thresholds were tuned on the HaluEval validation split. Ours (Full Mode) values are in bold.

Figure 1 shows the ROC comparison on the HaluEval test set (Llama-3-8B, Full Mode, mean across three seeds). Our method achieves higher true positive rates across most false positive rate regions. Figure 2 presents the corresponding precision-recall comparison. Our method maintains precision above 80% at high recall, supporting both high-precision deployment settings and balanced operating points.

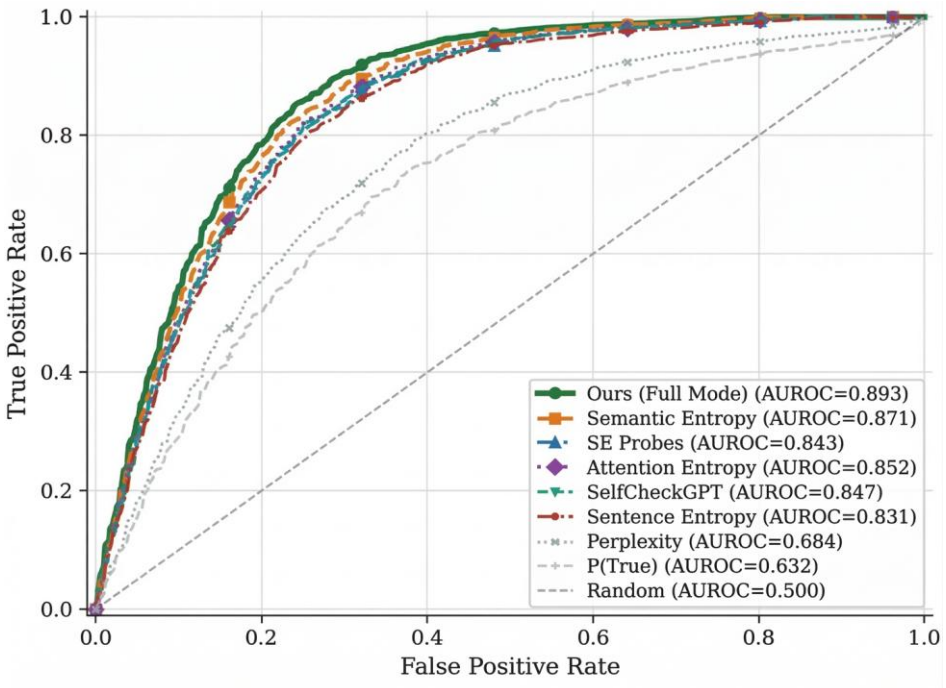

Figure 1: ROC curves on the HaluEval test set (Llama-3-8B, Full Mode, mean across three seeds). Our method achieves higher true positive rates across most false positive rate regions.

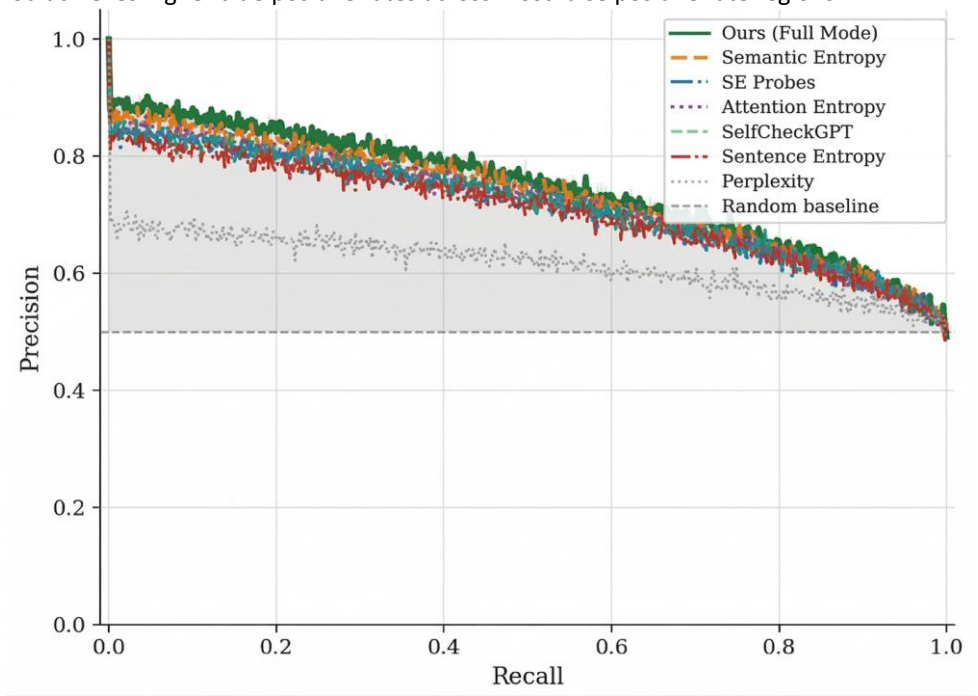

Figure 2: Precision–Recall curves on the HaluEval test set (Llama-3-8B, Full Mode, mean across three seeds). Our method maintains precision above 80% at high recall, supporting both high-precision deployment settings and balanced operating points.

5.2 API Mode Results

Table 2 presents API Mode results on HaluEval. GPT-3.5-Turbo in API Mode achieves 88.63 +/- 0.44% AUROC using only 10 features, which is competitive but lower than semantic entropy's 90.81 +/- 0.39% AUROC on this model. To disentangle the effect of mode from model quality, we also trained a 10-feature API Mode classifier for Llama-3-8B, which achieves 85.14 +/- 0.38% AUROC---3.49 pp lower than GPT-3.5 API Mode, which may reflect differences in token probability calibration between the two models.

Table 2
API Mode results on HaluEval (attention-free, retrained 10-feature XGBoost)

Setting	Features	AUROC	Acc.	F1	Lat.(ms)
GPT-3.5 API Mode (Ours)	F1-F9, F12	88.63±0.4	82.71±0.3	82.18±0.4	57±4
Llama-3-8B API Mode (Ours)	F1-F9, F12	85.14±0.3	79.83±0.3	79.41±0.3	47±3
Sem. Entropy [7]*	Multi-gen. (10 resp.)	90.81±0.3	84.23±0.3	83.97±0.4	494±26
Perplexity [11]*	Token prob. only	69.14±0.8	63.88±0.7	63.21±0.7	5±1

* All baselines were reproduced on identical hardware and with the same random seeds. GPT-3.5 Full Mode is unavailable because the API does not expose attention weights. Llama-3-8B API Mode retrains a 10-feature classifier and is methodologically distinct from the Full Mode ablation in Table 5.

5.3 Cross-Dataset and Cross-Model Generalisation

Table 3 presents cross-dataset results (Llama-3-8B, Full Mode). Improvements over semantic entropy range from +2.09 to +2.47 pp across the five benchmarks, suggesting relatively stable transfer under this evaluation setting. All five benchmarks involve model-generated text with ground-truth labels; more radical domain transfers (e.g., code, mathematics) may yield broader variance.

Table 3
Cross-dataset AUROC (Llama-3-8B, Full Mode, three seeds, full per-seed precision)

Dataset	Ours	Sem.Ent.*	Delta(pp)	Seed 42	Seed 123	Seed 456	Std	N test	Balance
HaluEval	89.27	87.12	+2.15	89.11	89.43	89.26	±0.16	7,000	50/50
TruthfulQA	84.53	82.44	+2.09	84.31	84.72	84.55	±0.21	817	40/60
FEVER	86.81	84.68	+2.13	86.59	87.04	86.80	±0.23	5,000	50/50
FRANK	82.34	79.87	+2.47	82.09	82.61	82.31	±0.26	2,250	65/35
FactCC	81.07	78.93	+2.14	80.88	81.29	81.04	±0.21	931	70/30

* All baselines were reproduced on each target dataset using the same evaluation protocol and seeds. Our method transfers the HaluEval-trained XGBoost classifier without retraining or threshold adjustment.

Table 4 presents cross-model results. All open-source models in Full Mode show consistent improvement over semantic entropy (+1.71 to +2.36 pp). GPT-3.5-Turbo, evaluated in API Mode, achieves 88.63% AUROC, which is 2.18 pp below semantic entropy on this model. This underperformance is plausibly related to the absence of attention-based features and to model-specific differences in probability calibration.

Table 4
Cross-model AUROC across five model families. GPT-3.5-Turbo uses API Mode; all others use Full Mode

Model	Params	Ours AUROC	Sem.Ent. AUROC*	Delta(pp)	Mode	Lat.(ms)
Llama-3-8B	8B	89.27±0.31	87.12±0.41	+2.15	Full	59
Llama-3-70B	70B	91.14±0.28	89.43±0.36	+1.71	Full	143
Mistral-7B	7B	88.67±0.34	86.31±0.43	+2.36	Full	56
Qwen-14B	14B	90.13±0.29	88.19±0.38	+1.94	Full	88
GPT-3.5-Turbo	N/A	88.63±0.44	90.81±0.39	-2.18	API	57

* All baselines were reproduced on identical hardware and with the same random seeds. The negative delta for GPT-3.5 reflects API Mode evaluation (without attention features) against multi-generation semantic entropy. Figure 3 summarizes cross-dataset and cross-model generalisation.

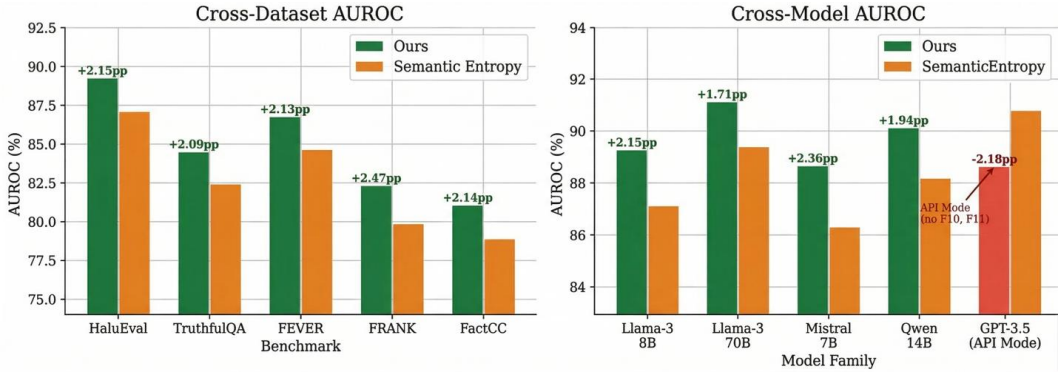

Figure 3: Cross-dataset (left) and cross-model (right) AUROC comparison. Improvements over semantic entropy remain consistent across datasets. GPT-3.5, evaluated in API Mode, does not surpass semantic entropy, whereas the open-source models evaluated in Full Mode show consistent gains.

5.4 Feature Importance and Ablation

Table 5 presents ablation results and SHAP importance values. Each row shows the AUROC when the corresponding feature group is removed from the trained 12-feature Full Mode model. SHAP values are additive and sum to 100%. The AUROC drops are not additive because features share discriminative information through non-linear interactions (SHAP interaction effects account for 23% of predictive power).

Table 5
Feature ablation (AUROC drop upon removal) and SHAP importance (Full Mode, Llama-3-8B, HaluEval)

Removed Feature	AURO	Delta AURO	SHA	% Tota	Cumul.9	Error Type
None (baseline)	89.27	---	---	---	---	All types
F9 (Temporal Dynamics	78.14	-11.13	0.18	27.3%	27.3%	Logical inconsistencies
F10-F11 (Attention)*	80.59	-8.68	0.14	20.8%	48.1%	Unverifiable claims
F7-F8 (Conf.-Consist.)	82.03	-7.24	0.11	17.2%	65.3%	High-confidence errors
F1-F2 (Token Entropy)	83.91	-5.36	0.08	13.0%	78.3%	General uncertainty
F5-F6 (Perplexity)	84.47	-4.80	0.07	10.8%	89.1%	Fluency-consist. mismatch
F3-F4 (Confidence Gap	85.38	-3.89	0.06	8.9%	98.0%	Low-competition generatio
F12 (Distrib. Coherence	88.94	-0.33	0.01	2.0%	100.0%	Distributional drift

* F10-F11 available in Full Mode only. SHAP values are additive; AUROC drops are not additive due to feature interactions. This ablation is distinct from API Mode retraining (Table 2).

5.5 Temporal Segmentation Ablation

Table 6 presents the effect of temporal segmentation count K on F9 and overall AUROC. K = 4 achieves the highest AUROC and lowest cross-seed variance, balancing temporal granularity with per-segment statistical stability.

Table 6
F9 ablation over temporal segmentation K (Llama-3-8B, HaluEval)

Metric	K=2	K=3	K=4	K=5	K=6	K=8
AUROC (%)	86.91	88.14	89.27	88.73	88.41	87.88
Tokens/seg. (T=200)	100	67	50	40	33	25
Std across seeds	±0.44	±0.38	±0.31	±0.33	±0.37	±0.42

5.6 Sample Efficiency

Figure 4 shows learning curves and sample efficiency on HaluEval. With 5,000 training samples the framework reaches 94.9% of peak AUROC, and performance plateaus at approximately 15,000 samples.

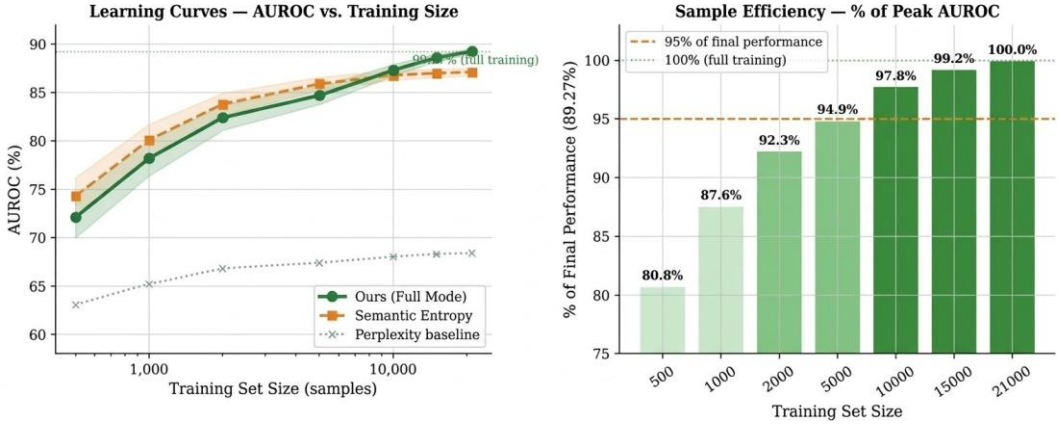

Figure 4: Learning curves (left) and sample-efficiency analysis (right) on HaluEval. The model reaches 94.9% of peak AUROC with 5,000 samples and 99.2% with 15,000 samples.

6. Analysis and Discussion

6.1 Why Multi-Granular Features Outperform Single-Level Methods

To understand the sources of detection performance, we manually categorised 500 correctly detected hallucinations and identified three primary detection pathways. Two annotators independently labelled each case using a predefined codebook; disagreements were resolved by discussion (inter-annotator agreement: Cohen's k = 0.71). Confidence-consistency features (F7, F8) were the dominant signals for high-confidence factual errors (35% of examined cases). Attention features (F10, F11; Full Mode only) were primary for unverifiable claims (28%). Temporal dynamics (F9) were primary for logical inconsistencies and self-contradictions (22%). The remaining 15% required combinations of multiple feature groups. No single feature group covers all hallucination types, which supports the value of multi-granular composition.

Figure 5 illustrates representative temporal entropy patterns and the corresponding feature-importance analysis. Panels (A) and (B) show entropy profiles associated with hallucinated outputs in our sample; panel (C) shows feature importance via AUROC drop upon removal, with F9 yielding the largest drop (11.13 pp).

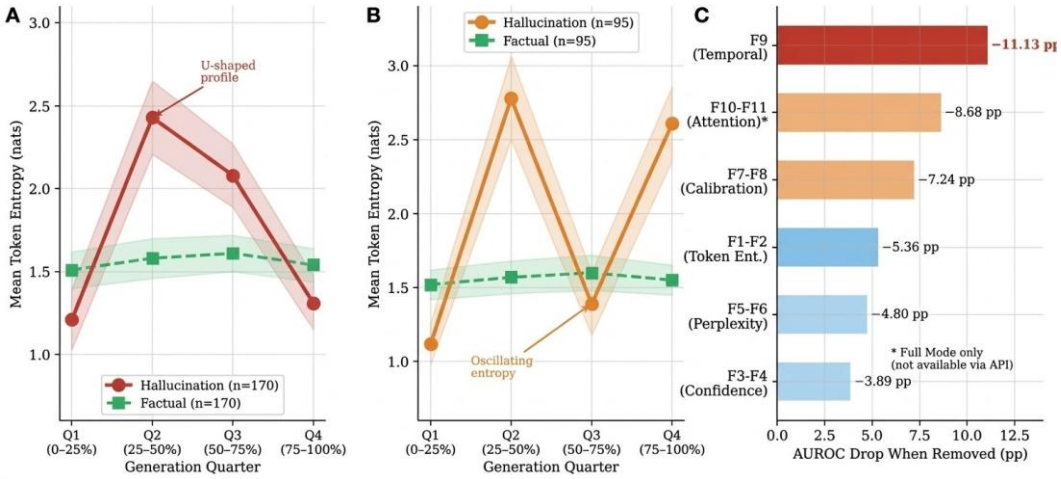

Figure 5: Temporal entropy dynamics (F9) and feature importance. Panels (A) and (B) show representative temporal entropy patterns associated with hallucinated outputs, while panel (C) shows feature importance measured by AUROC drop upon removal. F9 yields the largest drop, indicating the strongest contribution among the tested feature groups. F10–F11 are available only in Full Mode.

6.2 Error Analysis

We manually inspected 500 detection outputs (250 false positives, 250 false negatives) to characterise failure modes. Among false positives: technical domain jargon triggered high per-token perplexity (23%), epistemically hedged but correct statements were flagged (19%), and correct statements about post-training-cutoff events were flagged (17%); the remaining 41% did not cluster into clear categories. Among false negatives: low-perplexity fabrications following typical language patterns were the most common (31%), followed by subtle errors in otherwise correct context (26%) and fluently fabricated citations (22%); the remaining 21% were miscellaneous. These failure modes suggest that integrating retrieval-augmented verification [32] would be a valuable complementary approach.

6.3 Computational Cost Analysis

We provide an illustrative cost comparison based on one representative cloud-pricing assumption. Assumptions: A100 GPU instances at approximately $3/GPU-hour (publicly listed cloud spot pricing at time of writing; actual prices vary), 80% utilisation, 1 million queries per day. Our framework's incremental overhead is approximately $17/day; semantic entropy's incremental overhead is approximately $463/day. Under these assumptions, the cost ratio is approximately 27x. Detailed cost arithmetic is provided in the Supplementary Material. These cost considerations are relevant to real-time AI processing constraints documented in enterprise decision support contexts [26].

6.4 Practical Deployment Guidance

Based on our experimental results, we suggest the following deployment guidelines. Use Full Mode for open-source models where attention weights are available and latency-sensitive detection is needed; Full Mode provides the highest AUROC among single-pass methods evaluated here. Use API Mode when only token probabilities are available; API Mode provides a competitive single-pass alternative, though practitioners using GPT-3.5 who prioritise maximum AUROC over latency should

consider semantic entropy. Use semantic entropy when maximum API-based accuracy is preferred and the latency and cost of multi-generation inference are acceptable.

7. Limitations

Uncertainty is not factual verification. The framework detects uncertainty patterns, not factual incorrectness directly. Hallucinations generated with high confidence and typical language patterns escape detection (31% of false negatives). Integrating external knowledge verification would address this complementary failure mode. Self-contradictory hallucinations [33], where a model contradicts its own prior statements, represent a related failure mode that temporal dynamics (F9) partially captures but does not fully address.

Short-sequence instability for confidence-consistency proxies. For sequences shorter than approximately 60 tokens, the bin partition used by F7 and F8 (B = 10) results in fewer than five tokens per bin, rendering these features statistically unreliable. Reducing B to 5 partially mitigates this issue.

Vocabulary-space limitation of F12. F12 operates in vocabulary probability space rather than learned semantic embedding space. Its contribution is accordingly small (SHAP = 0.014, 2.0%). Replacing F12 with sentence-transformer embeddings is a natural improvement for future work.

API Mode underperformance on GPT-3.5. On GPT-3.5-Turbo, API Mode achieves 88.63% AUROC, 2.18 pp below semantic entropy (90.81%). Practitioners requiring maximum detection accuracy on GPT-3.5 should prefer semantic entropy, accepting its higher latency.

Benchmark scope. Our evaluation focuses on English-language benchmarks involving model-generated text with ground-truth labels. The framework has not been evaluated on code generation, mathematical reasoning, or multilingual hallucination detection, where uncertainty patterns may differ.

8. Conclusion

We proposed a single-pass hallucination detection framework that extracts 12 complementary uncertainty features across token-level, sequence-level, and temporal and distributional granularities. The framework supports two explicit deployment modes: Full Mode for open-source models with attention access, and API Mode for any model exposing token log-probabilities.

In Full Mode, the framework achieves 89.27% AUROC on HaluEval, outperforming our reproduction of semantic entropy by 2.15 pp at 8.2x reduced latency, with consistent improvements of +1.71 to +2.47 pp across five benchmarks and four open-source model families. Temporal entropy dynamics (F9) is the single most important feature. In API Mode on GPT-3.5, the framework is competitive but does not surpass semantic entropy, establishing a clear boundary on the current design's applicability.

A natural next step is to replace the vocabulary-space distributional coherence proxy (F12) with learned semantic embeddings and to extend evaluation to code, mathematical reasoning, and multilingual settings. All code and trained classifiers are available at the anonymous repository.

Data Availability Statement

The code, trained XGBoost classifiers, evaluation scripts, and dataset splits supporting the findings of this study are stored in an anonymous repository for peer-review purposes and are available from the author upon reasonable request. The datasets used in this study (HaluEval, TruthfulQA, FEVER, FRANK, and FactCC) are publicly available from their respective original sources.

Funding
This research received no external funding.

Conflicts of Interest
The author declares no conflict of interest.

References

[1] Huang, L., Yu, W., Ma, W., Zhong, W., Feng, Z., Wang, H., ... & Wei, F. (2023). A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. arXiv preprint arXiv:2311.05232.
[2] Ji, Z., Lee, N., Frieske, R., Yu, T., Su, D., Xu, Y., ... & Fung, P. (2023). Survey of hallucination in natural language generation. ACM Computing Surveys, 55(12), 1-38. https://doi.org/10.1145/3571730
[3] Thirunavukarasu, A. J., Ting, D. S. J., Elangovan, K., Gutierrez, L., Tan, T. F., & Ting, D. S. W. (2023). Large language models in medicine. Nature Medicine, 29(8), 1930-1940. https://doi.org/10.1038/s41591-023-02448-8
[4] Agrawal, A., Suzgun, M., Mackey, L., & Kalai, A. T. (2023). Do language models know when they're hallucinating references? arXiv preprint arXiv:2305.18248.
[5] Zhao, W., Zhou, K., Li, J., Tang, T., Wang, X., Hou, Y., ... & Wen, J. R. (2023). A survey of large language models. arXiv preprint arXiv:2303.18223.
[6] Önden, A., Kara, K., Önden, İ., Yalçın, G. C., Simic, V., & Pamucar, D. (2024). Exploring the adoption of the metaverse and chat generative pre-trained transformer: A single-valued neutrosophic Dombi Bonferroni-based method for the selection of software development strategies. *Engineering Applications of Artificial Intelligence*, *133*, 108378. https://doi.org/10.1016/j.engappai.2024.108378
[7] Farquhar, S., Kossen, J., Kuhn, L., & Gal, Y. (2024). Detecting hallucinations in large language models using semantic entropy. Nature, 630(8017), 625-630. https://doi.org/10.1038/s41586-024-07421-0
[8] Manakul, P., Liusie, A., & Gales, M. J. (2023). SelfCheckGPT: Zero-resource black-box hallucination detection for generative large language models. arXiv preprint arXiv:2303.08896.
[9] Azaria, A., & Mitchell, T. (2023). The internal state of an LLM knows when it's lying. arXiv preprint arXiv:2304.13734.
[10] Chen, C., Liu, K., Chen, Z., Gu, Y., Wu, Y., Tao, M., Fu, Z., & Ye, J. (2024). INSIDE: LLMs' internal states retain the power of hallucination detection. arXiv preprint arXiv:2402.03744.
[11] Xiao, Y., & Wang, W. Y. (2021). On hallucination and predictive uncertainty in conditional language generation. In Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics. https://doi.org/10.18653/v1/2021.eacl-main.236
[12] Kadavath, S., Conerly, T., Askell, A., Henighan, T., Drain, D., Perez, E., ... & Kaplan, J. (2022). Language models (mostly) know what they know. arXiv preprint arXiv:2207.05221.
[13] Pramanik, V., Jha, S., Velasquez, A., & Jha, S. K. (2025). Fact or hallucination? An entropy-based framework for attention-wise usable information in LLMs. OpenReview preprint. https://openreview.net/forum?id=p9u9qw2q1j
[14] Joo, E., Lee, Y. J., & Choi, H. J. (2025). Entropy-based sentence-level hallucination score in large language models. In Proceedings of the 2025 IEEE International Conference on Big Data and Smart Computing (BigComp), pp. 77-78. https://doi.org/10.1109/BigComp64353.2025.00022
[15] Kuhn, L., Gal, Y., & Farquhar, S. (2023). Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation. arXiv preprint arXiv:2302.09664.
[16] Ciosek, K., Felicioni, N., & Ghiassian, S. (2025). Hallucination detection on a budget: Efficient Bayesian estimation of semantic entropy. arXiv preprint arXiv:2504.03579.
[17] Phillips, E., Wu, S., Molaei, S., Belgrave, D., Thakur, A., & Clifton, D. (2025). Geometric uncertainty for detecting and correcting hallucinations in LLMs. arXiv preprint arXiv:2509.13813.
[18] Song, D., Xie, X., Song, J., Zhu, D., Huang, Y., Juefei-Xu, F., & Ma, L. (2024). Luna: A model-based universal analysis framework for large language models. IEEE Transactions on Software Engineering, 50, 1921-1948. https://doi.org/10.1109/TSE.2024.3411928
[19] Li, K., Patel, O., Viegas, F., Pfister, H., & Wattenberg, M. (2023). Inference-time intervention: Eliciting truthful answers from a language model. In Advances in Neural Information Processing Systems (NeurIPS), 36, 43879-43897.

[20] Chern, I.-C., Chern, S., Chen, S., Yuan, W., Feng, K., Zhou, C., ... & Liu, P. (2023). FacTool: Factuality detection in generative AI -- a tool augmented framework for multi-task and multi-domain scenarios. arXiv preprint arXiv:2307.13528.

[21] Arzt, V., Azarbeik, M. M., Lasy, I., Kerl, T., & Recski, G. (2024). TU Wien at SemEval-2024 Task 6: Unifying model-agnostic and model-aware techniques for hallucination detection. In Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), 1183-1196. https://doi.org/10.18653/v1/2024.semeval-1.173

[22] Dhuliawala, S., Komeili, M., Xu, J., Raileanu, R., Li, X., Celikyilmaz, A., & Weston, J. (2023). Chain-of-verification reduces hallucination in large language models. arXiv preprint arXiv:2309.11495.

[23] Liu, X., Chen, T., Da, L., Chen, C., Lin, Z., & Wei, H. (2025). Uncertainty quantification and confidence calibration in large language models: A survey. In Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2 (KDD '25), August 3–7, 2025, Toronto, ON, Canada (13 pages). ACM. https://doi.org/10.1145/3711896.3736569

[24] Kang, S., Bakman, Y. F., Yaldiz, D. N., Buyukates, B., & Avestimehr, S. (2025). Uncertainty quantification for hallucination detection in large language models: Foundations, methodology, and future directions. arXiv preprint arXiv:2510.12040.

[25] Shorinwa, O., Mei, Z., Lidard, J., Ren, A. Z., & Majumdar, A. (2025). A survey on uncertainty quantification of large language models: Taxonomy, open research challenges, and future directions. ACM Computing Surveys, 58(3), article 63, pp. 1-38. https://doi.org/10.1145/3744238

[26] Önden, A. (2026). A systemic approach to decision support and automation: The role of big data analytics and real-time processing in management information systems. *Systems*, *14*(2), 216. https://doi.org/10.3390/systems14020216

[27] Li, J., Cheng, X., Zhao, W. X., Nie, J. Y., & Wen, J. R. (2023). HaluEval: A large-scale hallucination evaluation benchmark for large language models. In Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP), 6449-6464. https://doi.org/10.18653/v1/2023.emnlp-main.397

[28] Lin, S., Hilton, J., & Evans, O. (2022). TruthfulQA: Measuring how models mimic human falsehoods. In Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics, 3214-3252. https://doi.org/10.18653/v1/2022.acl-long.229

[29] Thorne, J., Vlachos, A., Christodoulopoulos, C., & Mittal, A. (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, 809-819. https://doi.org/10.18653/v1/N18-1074

[30] Pagnoni, A., Balachandran, V., & Tsvetkov, Y. (2021). Understanding factuality in abstractive summarization with FRANK: A benchmark for factuality metrics. In Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 4502-4520. https://doi.org/10.18653/v1/2021.naacl-main.383

[31] Kryscinski, W., McCann, B., Xiong, C., & Socher, R. (2020). Evaluating the factual consistency of abstractive text summarization. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing. https://doi.org/10.18653/v1/2020.emnlp-main.750

[32] Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., ... & Kiela, D. (2020). Retrieval-augmented generation for knowledge-intensive NLP tasks. In Advances in Neural Information Processing Systems (NeurIPS), 33, 9459-9474.

[33] Mundler, N., He, J., Jenko, S., & Vechev, M. (2024). Self-contradictory hallucinations of large language models: Evaluation, detection and mitigation. arXiv preprint arXiv:2305.15852.