Decomposed Entailment for Factuality Checking and Hallucination Detection

Achir Oukelmoun
Université Paris-Saclay
3 Rue Joliot Curie,
91190 Gif-sur-Yvette, France
achir.oukelmoun@gmail.com

Nasredine Semmar
Université Paris-Saclay,
CEA, List,
91120 Palaiseau, France
nasredine.semmar@cea.fr

Gaël de Chalendar
Université Paris-Saclay,
CEA, List,
91120 Palaiseau, France
gael.de-chalendar@cea.fr

Abstract

The reliability of Large Language Models (LLMs) is often compromised by factual inconsistencies, including hallucinations—cases where generated content is not supported by the underlying source. We present HallDetect, a lightweight, reference-free, and black-box framework for hallucination detection that we evaluate not only on summarization but across a broader range of source-grounded generation settings. HallDetect builds on decomposition-based factuality evaluation: generated content is decomposed into atomic claims, each verified by a compact encoder-based entailment model through a contrastive formulation over a multi-scale library of source chunks, and aggregated with an asymmetric score in which a single confidently contradicted claim flags the response. Under a controlled protocol in which all methods share the same 4-bit quantized backbones and consumer-grade hardware budget, HallDetect outperforms comparably resourced generative and embedding-based baselines on three of four benchmarks while remaining stable across backbone families, and yields a claim-to-span audit trail that localizes each error.

1 Introduction

The rapid advancement of Large Language Models (LLMs), driven in particular by the GPT family (Achiam et al., 2023), has fundamentally reshaped the landscape of Natural Language Processing. As these models move from experimental settings to deployment in professional and decision-critical contexts, a persistent limitation remains unresolved: *hallucinations*. The tendency of LLMs to produce confident yet factually unsupported statements continues to hinder their reliable and safe use (Li et al., 2024), and metrics based on surface overlap or embedding similarity fail to capture the logical errors exhibited by modern generative models (Maynez et al., 2020).

Hallucination detection is also an operational challenge. Many high-performing LLMs are accessible exclusively through proprietary, black-box APIs, precluding internal inspection. At the same time, the "LLM-as-a-judge" evaluation paradigm raises concerns regarding computational cost, latency, and reproducibility, particularly where local deployment or hardware constraints apply (Szymanski et al., 2025; Dorner et al., 2024). Beyond a single scalar score, an evaluation system intended for auditing must identify *which* parts of an output are unsupported by the source and why (Miller, 2019; Fabbri et al., 2022).

In this work, we present HallDetect, a resource-efficient and domain-agnostic framework for hallucination detection. HallDetect follows the established paradigm of decomposition-based factuality evaluation (Min et al., 2023; Chen et al., 2023): generated texts are decomposed into atomic propositions, which are then evaluated through contrastive entailment against the available evidence. Our focus is on making this paradigm effective and stable under strict computational budgets—4-bit quantized extractors and a sub-1GB discriminative verifier on a single consumer GPU—and on characterizing its behavior across tasks, domains, and backbone families. To that end, we evaluate on four benchmarks spanning news summarization (QAGS-CNN/DM, Wang et al., 2020), dialogue (TofuEval, Tang et al., 2024b), adversarial perturbations (FalseSum, Utama et al., 2022), and biomedical QA (PubMedQA, Jin et al., 2019).

Our main contributions are summarized as follows:

- **A frugal decomposition-and-verification pipeline.** Building on prior decomposition-based factuality work (Section 2.2), HallDetect couples atomic claim extraction with a *contrastive, multi-scale* NLI formulation: the best entailing chunk and the best contradict-

Proceedings of the 19th International Natural Language Generation Conference: Long Papers, pages 483–497
October 17-21, 2026 ©2026 Association for Computational Linguistics

ing chunk of the source are selected independently across five chunking granularities, and per-claim scores are aggregated asymmetrically so that one confident contradiction flags the response. The framework requires only the source-generation pair and is compatible with arbitrary black-box generators.

- **Controlled evaluation under fixed computational budgets.** Under strict parameter and resource parity—all LLM components quantized to 4 bits and run on identical consumer hardware—HallDetect consistently outperforms comparably resourced generative, self-consistency, and embedding baselines on three of four benchmarks. We do not claim superiority over full-precision or API-based state-of-the-art detectors, which operate under substantially larger budgets.

- **Stability across backbones and domains.** Across three backbone families (Mistral, Gemma, Llama) and four datasets, HallDetect exhibits markedly lower variance than generative CoT judges, suggesting that delegating verification to a discriminative encoder shields the judgment from quantization artifacts.

- **Claim-level audit trail.** Like other decomposition-based methods, HallDetect exposes a claim→evidence-span mapping; we illustrate with a worked long-source example (Appendix B) how this trail localizes both contradictions and unsupported additions, and note that such localized verdicts are a natural fit for closed-loop regeneration.

2 Related Work

Hallucinations—outputs inconsistent with or unsupported by a given source—impede the deployment of LLMs in high-stakes settings (Li et al., 2024; Huang et al., 2025). A common distinction separates *factuality* (consistency with world knowledge) from *faithfulness* (alignment with a specific input source) (Banerjee et al., 2025); we target the latter, in summarization, question answering, and retrieval-augmented generation (Tang et al., 2024b). Within this setting, *intrinsic* hallucinations contradict the source directly, while *extrinsic* ones add unsupported content that may appear plausible in isolation (Wang et al., 2020; Maynez et al., 2020). A complementary mode, *omission*, checks source units against the generation—the reverse direction (Oukelmoun et al., 2025a,b).

2.1 From Internal Signals to External Audits

White-box approaches leverage internal signals such as token probabilities or hidden states (Azaria and Mitchell, 2023), but are impractical for proprietary APIs, motivating black-box alternatives such as SelfCheckGPT (Manakul et al., 2023) that use outputs and stochastic sampling alone. Surface-level metrics such as ROUGE (Lin, 2004) correlate weakly with factual correctness (Maynez et al., 2020), driving reference-free strategies such as QAGS (Wang et al., 2020) that verify consistency through QA or NLI against the source.

2.2 Decomposition-Based Factuality Evaluation

HallDetect verifies generated text at the granularity of individual claims, a strategy shared by a growing body of work. Our target is specifically a detector that is simultaneously *black-box* (using only the source–generation pair, with no access to the generator's internal states or logits), *frugal* (a light, single-pass budget rather than a heavy multi-model pipeline or a massive verifier LLM), *intrinsic* (checking claims against the supplied source, not against external world knowledge), and *interpretable* (producing a localized claim-to-span audit trail that explains each flag). Our head-to-head baselines are accordingly restricted to *black-box detectors of intrinsic, source-grounded hallucinations that run under a matched frugal budget* (Section 5); the related systems below each satisfy some but not all of these four properties, and we position against each accordingly rather than benchmark it.

Extrinsic, knowledge-grounded verification. FActScore (Min et al., 2023) decomposes generations into atomic facts and scores the fraction supported by an *external* knowledge source (e.g., Wikipedia), using a retrieval-augmented *generative* LM as the validator rather than a discriminative entailment model. It therefore measures *extrinsic* factual precision against world knowledge—not *intrinsic* faithfulness to a supplied document—with an LM-based validator outside our frugal-discriminative regime, making it orthogonal to our benchmark rather than a competitor on it.

Decomposition resources and analyses. PropSegmEnt (Chen et al., 2023) is a corpus and task for proposition-level segmentation and entailment recognition, while Wanner et al.

(2024) and Molecular Facts (Gunjal and Durrett, 2024) analyze how decomposition granularity and decontextualization affect downstream verification. These inform *how* we extract atomic claims (Phase I) but expose no source-grounded faithfulness score to benchmark against on our datasets.

Generative atomic-unit judges. ACUEval (Wan et al., 2024) does target summarization faithfulness through atomic content units, but delegates each unit's verification to an *LLM* judge (and adds a correction stage). It thus belongs to the generative "LLM-as-a-judge" family, whose behavior under our quantized budget is already captured by our **LLM CoT** baseline.

Discriminative intrinsic detectors (closest relatives). These are black-box and source-grounded like ours, but each diverges on at least one of the remaining properties. SummaC (Laban et al., 2022) applies sentence-level NLI over a source–summary pair matrix at a *fixed* sentence granularity, without generative atomic decomposition or a multi-scale chunk library; a single summary sentence mixing a supported and an unsupported claim therefore cannot be cleanly isolated, so it yields a consistency score rather than an atomic claim-to-span audit trail (our **NLI Only** baseline is a document-level analog of this family). QAFactEval (Fabbri et al., 2022) is intrinsic but departs from frugality: it chains answer extraction, question generation, and question answering, a heavier multi-stage pipeline than a single-pass NLI verifier. MiniCheck (Tang et al., 2024a) is the closest system—a compact, trained, black-box, source-grounded checker with a DeBERTa variant aimed at exactly our low-resource setting, satisfying the black-box, frugal, and intrinsic properties—and diverges mainly on interpretability: the released model acts as an end-to-end grounded/hallucinated classifier and does not expose a modular mapping from extracted atomic claims to specific source chunks. Because it is the nearest match, we treat a controlled comparison against it as the priority next experiment (Section 9).

Compared with these systems, HallDetect's contribution is a specific, previously uncombined design for intrinsic detection under frugal budgets: (i) *generative* atomic-claim extraction paired with a *specialized discriminative* NLI verifier used in a single forward pass per claim–chunk pair; (ii) a *multi-scale* context library with *independent* best-entailment / best-contradiction selection; and (iii) a *contrastive* entailment-minus-contradiction score under an asymmetric, auditing-oriented aggregation. Unlike FActScore and ACUEval, verification never invokes a generative model; unlike SummaC, it verifies generatively-extracted atomic propositions across multiple granularities rather than fixed source/summary sentence pairs; and unlike MiniCheck's end-to-end classifier, its per-claim, per-chunk decisions directly provide an explicit audit trail.

2.3 The Shift Toward Discriminative Verification

The "LLM-as-a-judge" paradigm (Zheng et al., 2023; Liu et al., 2023; Eliav et al., 2025) prompts large generative models to assess other outputs, but risks self-preference bias (Wataoka et al., 2024), evaluator hallucinations, and limited reproducibility at high cost. These concerns have renewed interest in discriminative, encoder-based verification, which frames factuality as a constrained classification problem with stronger logical structure and higher efficiency. Benchmarks such as TofuEval (Tang et al., 2024b) and HaluEval (Hu et al., 2024) further stress the value of claim-level judgments for human-in-the-loop workflows.

3 Benchmarks and Multi-Model Evaluation Strategy

3.1 Task Definition and Metrics

We address **binary, response-level hallucination detection**: given a source document D and a generated response R, the system must decide whether R is *faithful* to D or *hallucinated*. All benchmarks are used with response-level gold labels, and we report Precision, Recall, and F1 for the *hallucinated* class at a fixed decision threshold of 0.5, reflecting out-of-the-box utility without dataset-specific tuning. HallDetect additionally produces claim-level verdicts; because the benchmarks above do not provide aligned gold labels at our claim granularity, we use these verdicts as an *audit trail* for qualitative analysis (Appendix B) rather than for quantitative localized-error evaluation, which we leave to future work.

3.2 Core Evaluation Benchmarks

To evaluate **HallDetect**, we adopt a multi-model strategy using 4-bit quantized variants of **Llama 3.1-8B**, **Gemma 2-9B**, and **Mistral-7B**,

across four benchmarks representing distinct challenges:

- **QAGS-CNN/DM** (Wang et al., 2020) (*Primary Anchor*): high-quality human faithfulness annotations for news summarization, our main testbed for atomic decomposition.
- **TofuEval** (Tang et al., 2024b) (*Dialogue Robustness*): informal structure and speaker shifts test coreference stability and factual drift in dialogue summarization.
- **FalseSum** (Utama et al., 2022) (*Adversarial Stress Test*): fluent but factually perturbed summaries test sensitivity to subtle hallucinations that bypass coarse similarity checks.
- **PubMedQA** (Jin et al., 2019) (*Domain Generalization*): biomedical text requiring technical nomenclature handling and strict logical grounding.

Compared to alternatives, SummEval (Fabbri et al., 2021) conflates factuality with stylistic properties, FactCC (Kryściński et al., 2020) relies on synthetic perturbations that do not fully mirror organic LLM hallucinations, HaluEval (Li et al., 2023) lacks the long-range document dependencies of QAGS, and FEVER (Thorne et al., 2018) targets *extrinsic* fact-checking against knowledge bases, whereas our framework evaluates **intrinsic consistency** strictly within the provided source.

QAGS-CNN/DM remains our primary reference point: it is reference-free (matching deployment, where gold summaries are unavailable) and imposes long-form complexity (700–800-word articles) that exposes failure modes such as lead bias. The additional datasets probe the framework's task- and domain-agnostic behavior.

3.3 Backbones and the Rationale for Quantized Evaluation

Why a frugal budget? The computational budget is part of the intended deployment profile. Faithfulness auditing is often most needed precisely where routing the source document to a proprietary, black-box API is undesirable or impossible: confidential clinical, legal, or enterprise material that cannot leave local infrastructure for privacy or compliance reasons; interactive, human-in-the-loop review where per-call latency and recurring API cost accumulate quickly; and reproducible research or self-hosted pipelines that should not depend on a moving, closed endpoint. A detector that runs entirely on a single consumer-grade GPU speaks to all three: it keeps the audited text on-premises, removes per-query cost, and fixes the software stack for reproducibility. We therefore treat consumer-hardware operability as a design constraint from the outset, and hold every method to this shared budget so that the comparison isolates the detection method from access to scale.

We evaluate across three architectures in quantized GGUF form (Q4_K_M): **Llama-3.1-8B-Instruct** (dense Transformer with extended context), **Gemma-2-9B-IT** (efficiency-oriented attention design), and **Nous-Hermes-2-Mistral-7B-DPO** (DPO-tuned for instruction following). While newer model families exist, these backbones are widely deployed, run on consumer hardware, and—critically for our controlled protocol—are available in identical quantization formats across families; our claims concern the *framework's* behavior under matched budgets rather than the absolute capability of any backbone.

Restricting evaluation to 4-bit quantization is deliberate: a framework stable under the added noise and capacity constraints demonstrates robustness not apparent at full precision, and all experiments remain reproducible on consumer-grade hardware.

4 Methodology: The HallDetect Framework

HallDetect reconciles *diagnostic precision*, *auditability*, and *computational frugality*. Rather than the proprietary "judge" models used by many systems (Zheng et al., 2023; Liu et al., 2023), with their cost, transparency, and reproducibility trade-offs (Chen et al., 2024), it adopts a local pipeline that formulates detection as logical verification grounded in Natural Language Inference (NLI).

4.1 Decompositional Verification

Assessing a response as a single, undifferentiated unit dilutes the error signal: isolated hallucinations are obscured by surrounding correct content, a known limitation of document-level metrics (Falke et al., 2019). HallDetect therefore separates the *generative* extraction of atomic claims from the *discriminative* verification of each claim against the source, so that verification operates on focused, coherent hypotheses and gains sensitivity to localized contradictions and unsupported assertions.

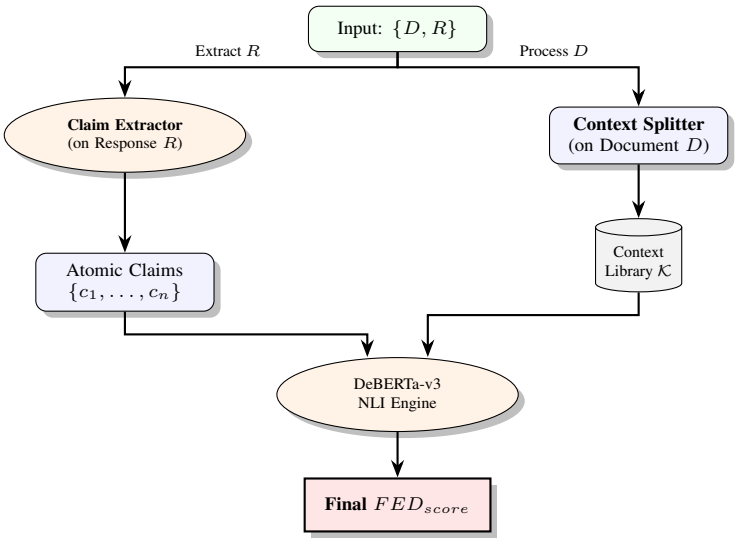


Figure 1: The HallDetect architecture. The pipeline bifurcates the input: the response R is decomposed into atomic claims, while the document D is indexed into a context library for NLI-based verification.

4.2 Phase I: Atomic Claim Extraction

Given a generated response R, the first stage decomposes it into a set of atomic claims $\{c_1, \ldots, c_n\}$, using locally deployed LLMs from the Llama 3.1, Gemma 2, or Mistral families, all quantized to 4-bit precision (Q4_K_M). Generative models provide the linguistic competence needed to surface implicit factual statements, while quantization keeps extraction deployable on consumer hardware. The extraction prompt is *copy-faithful* (Appendix A): claims are reproduced verbatim, so verification targets the model's actual assertions rather than silently corrected paraphrases.

4.3 Phase II: Multi-Scale Contextual Chunking

Relevant evidence in long documents may be distributed across distant regions or diluted at a single context scale (Liu et al., 2024). HallDetect constructs a hierarchical context library $\mathcal{K}$ by partitioning the source document D at multiple granularities ($m \in \{1, 2, 4, 8, 16\}$ chunks). Concretely, chunks are contiguous groups of *whole sentences* (obtained with an NLTK sentence splitter), never fixed character- or word-count windows and never cut mid-sentence: at granularity m the document is divided into m roughly equal sentence groups of size $\lceil n_{\text{sent}}/m \rceil$ (Appendix A). This multi-scale representation increases the likelihood that, for any claim, some context window renders the relevant evidence both present and salient.

4.4 Phase III: Contrastive NLI Verification

The core verification stage replaces open-ended "LLM-as-a-judge" prompting with a three-way NLI formulation. We employ **DeBERTa-v3-Large** (He et al., 2021) in its NLI-fine-tuned variant (trained on MNLI, FEVER-NLI, ANLI, LingNLI, and WANLI), i.e., a dedicated entailment classifier rather than a generic encoder.

Encoder-only verification. This is a deliberate design choice: encoders jointly attend over premise and hypothesis, enabling fine-grained detection of logical inconsistencies, and the model is trained explicitly for entailment classification, providing strong zero-shot verification performance.

Contrastive scoring. For each atomic claim c_i, HallDetect computes a contrastive score balancing evidence of support against evidence of contradiction, derived directly from the three-way softmax of the NLI engine:

$$\begin{aligned} Score(c_i) = & \max_{k \in \mathcal{K}} P_{\text{NLI}}(E \mid k, c_i) \\ & - \max_{k \in \mathcal{K}} P_{\text{NLI}}(C \mid k, c_i), \end{aligned} \tag{1}$$

where $P_{\text{NLI}}(E)$ and $P_{\text{NLI}}(C)$ denote the predicted probabilities of the *Entailment* and *Contradiction* classes given chunk k and claim c_i, and the maximizing chunks for E and C are selected *independently*. The score thus lies in $[-1, 1]$—positive when support outweighs contradiction, *negative* when a chunk contradicts the claim more strongly than any chunk supports it—and by the aggregation

below (Phase IV), one such net-contradicted claim flags the whole response ($FED_{score} = 1$).

Why subtract contradiction? Scoring the contradiction term, rather than support alone, lets HallDetect separate two failure modes that an entailment-only signal conflates. Support and contradiction are not two ends of one axis but distinct outcomes of a three-way NLI head (entailment, contradiction, neutral): a claim can be echoed by one part of a document yet refuted by another, and faithfulness demands that the refutation win. A hallucinated claim often paraphrases the source closely enough to draw high entailment from one chunk while another flatly contradicts it—a reversed attribution, a negation, or an altered quantity—and only the contradiction term drives such a claim below zero. The contradiction channel therefore targets *active* factual inversions, typically the most damaging errors in high-stakes settings, which a support-only score would reward for superficial lexical overlap and miss.

Treatment of the Neutral class. The *Neutral* probability does not enter the score explicitly but shapes it implicitly through the softmax: a claim for which every chunk is predominantly neutral obtains both low $P(E)$ and low $P(C)$, yielding a score near zero. Such claims are treated as *unsupported* (extrinsic additions) rather than *contradicted*, which is exactly the distinction illustrated by claims c_2 and c_3 in the worked example of Appendix B.

Computational profile. HallDetect's frugality can be quantified precisely. The expensive operation in every pipeline we consider is autoregressive LLM *decoding*; HallDetect invokes it *exactly once*—for claim extraction—and, crucially, only over the short response R (capped at 512 new tokens), *never* placing the long source document in a generative context. All source-grounded verification is delegated to the DeBERTa-v3 encoder, which at 435M parameters is nearly $20\times$ smaller than the 8B-class extractor, occupies < 1 GB of VRAM, and performs a *single, non-autoregressive* forward pass per claim–chunk pair—at most $|\mathcal{K}| \cdot n \leq 31n$ passes for n claims, running concurrently with the extractor on one consumer GPU (e.g., RTX 3060/4060). By contrast, the **LLM CoT** judge must ingest the *entire* 700–800-word document into the generative context and decode a full reasoning chain, and **SelfCheck** requires $N{=}5$ such generations. Counting generative LLM invocations per instance, HallDetect uses 1 (over a short input), LLM CoT 1 (over a long input *and* a long output), and SelfCheck 5; the discriminative verifier adds only a sub-dominant $\sim 5\%$ of the extractor's parameters at single-pass cost. HallDetect is thus not the absolute cheapest scorer—the embedding baselines are lighter but markedly weaker—but it attains decomposition-level accuracy *without* the repeated or long-context generative passes that dominate the cost of competitive generative judges. Appendix A (Table 6) gives the full accounting.

4.5 Phase IV: Aggregation and the FED_{score}

The final output is the **Factuality Entailment Decomposition score**, $FED_{score} = 1 - G$, where the gated aggregate G takes the geometric mean only over strictly positive per-claim scores (each $Score(c_i) \in [-1, 1]$ can be negative, so the sign first acts as a gate):

$$G = \begin{cases} \sqrt[n]{\prod_{i=1}^{n} Score(c_i)} & \text{if all } Score(c_i) > 0, \\ 0 & \text{otherwise.} \end{cases} \tag{2}$$

The aggregation is intentionally asymmetric, and the sign of each per-claim score plays two roles. First, it acts as a *hard gate*: whenever any $Score(c_i) \leq 0$ we have $G = 0$ and the response is flagged ($FED_{score} = 1$), so the product and root never see negative or zero values, avoiding the ill-definedness of a geometric mean over signed values. Second, when *all* scores are strictly positive ($Score(c_i) \in (0, 1]$), the geometric mean is preferred over the arithmetic mean because it is *conjunctive* (min-like): it is dominated by the weakest-supported claim and decays toward 0 as any claim's support weakens. We prefer it to a hard minimum as a smooth, graded surrogate—still governed by the weakest claim, but also sensitive to how many claims are weak and by how much—whereas an arithmetic mean would let well-supported claims average out a single fabrication, letting a response with one blatant falsehood still score as faithful. The precision–recall consequences of this recall-oriented design are analyzed in Section 6.1.3.

5 Baselines and Comparative Evaluation Framework

To isolate the efficacy of HallDetect from raw model capacity, we enforce a *controlled-resource protocol*. All LLM-based baselines use identical

4-bit (Q4_K_M) quantized engines: **Llama-3.1-8B**, **Gemma-2-9B**, and **Mistral-7B**. Embedding-based methods use **Qwen2.5-0.5B-Embed** to maintain a consistent, frugal budget. These baselines are representative of the main black-box detection paradigms under matched resources; they are not full-precision state-of-the-art systems, and our claims are scoped accordingly.

Monolithic NLI Baseline. To quantify the gains attributable to decomposition and multi-scale chunking jointly, we define a document-level NLI baseline using the same **DeBERTa-v3-Large** engine, applied holistically to the full pair (D, R):

$$H_{\text{NLI}}(D, R) = 1 - P_{\text{NLI}}(\text{Entailment} \mid D, R) \quad (3)$$

This baseline isolates the effect of verifying holistically versus per claim, and two properties of HallDetect shape the comparison, both by design. First, cross-encoder NLI is inherently bounded by the encoder's context window: DeBERTa-v3 truncates long inputs, so a document-level (D, R) pass on the 700–800-word QAGS articles can push relevant evidence outside the window (Appendix A). HallDetect is built around this limitation—it verifies short atomic claims against short, multi-scale chunks, so every pair stays within the encoder's budget and the relevant evidence remains visible at some granularity. Robustness to source length is therefore an intended property of chunk-based verification. Second, HallDetect additionally decomposes the response into atomic claims. Because these two factors act together, we read the gap between **NLI Only** and HallDetect as evidence of this combined length-robust and claim-level design.

Generative Chain-of-Thought (CoT). A representative "LLM-as-a-judge" baseline in which the backbone is prompted to reason step-by-step before a binary verdict:

$$H_{\text{Direct}} = \begin{cases} 1 & \text{if } T \text{ indicates } \textit{Hallucinated} \\ 0 & \text{if } T \text{ indicates } \textit{Faithful} \end{cases} \quad (4)$$

where T is the generated judgment. This tests whether unstructured generative reasoning in mid-sized quantized models can match the explicit logical constraints of HallDetect.

Consistency-Based Scoring (SelfCheck). Following the SelfCheckGPT paradigm (Manakul et al., 2023), we generate $N = 5$ alternative summaries $\{S_i\}_{i=1}^{N}$ and compute the average semantic divergence from the original response:

$$H_{\text{SelfCheck}} = \frac{1}{N} \sum_{i=1}^{N} (1 - \cos(\mathbf{e}_R, \mathbf{e}_{S_i})) \quad (5)$$

where $\mathbf{e}$ denotes the embedding vector.

Semantic Divergence (QwenScore). A lightweight, non-generative baseline measuring coarse semantic distance between source and response: $H_{\text{Qwen}} = 1 - \cos(\mathbf{e}_D, \mathbf{e}_R)$.

6 Quantitative Results and Analysis

We evaluate HallDetect (FED_{score}) against the comparably resourced baselines of Section 5 across four benchmarks. All LLM backbones (Mistral-7B, Llama-3.1-8B, Gemma-2-9B) are executed under 4-bit (Q4_K_M) quantization. *QwenScore* and *NLI Only* are **Non-Generative Baselines**: fixed, backbone-independent reference points using no LLM extraction or prompting. We report Precision, Recall, and F1 for the hallucinated class at a fixed threshold of **0.5** (Section 3.1).

6.1 Comparative Analysis and Key Findings

6.1.1 Architectural Resilience and Quantization Stability

The clearest finding (Table 1) is the *stability gap* between generative judges and HallDetect. Under 4-bit quantization LLM CoT is volatile—its F1 collapses from 0.589 on Gemma-2 to 0.321 on Llama-3.1—suggesting that unstructured generative reasoning is acutely sensitive to quantization, or that some architectures falter at discriminative verification in a CoT format. HallDetect instead stays within a narrow envelope (F1 0.734–0.771): decoupling claim extraction from verification (delegated to a discriminative encoder) shields the judgment from the decoding biases that compromise autoregressive reasoning in compressed LLMs (Li et al., 2025).

6.1.2 The Decomposition Premium: Unmasking Localized Errors

The NLI Only baseline is our counter-factual. A monolithic check on QAGS-CNN/DM reaches high precision (0.833) but limited recall (0.517): when claims are aggregated into one premise, localized contradictions are averaged away (the "dilution effect"). Operating atomically, HallDetect recovers these signals, reaching near-saturation recall across all backbones without the precision collapse

Method	Non-Generative Baselines			Mistral-7B			Llama-3.1-8B			Gemma-2-9B		
	F1	P	R	F1	P	R	F1	P	R	F1	P	R
QwenScore	0.634	0.547	0.753	—	—	—	—	—	—	—	—	—
NLI Only	**0.638**	0.833	0.517	—	—	—	—	—	—	—	—	—
LLM CoT	—	—	—	0.586	0.680	0.515	0.321	0.333	0.310	0.589	0.731	0.494
SelfCheck	—	—	—	0.731	0.579	0.993	0.686	0.527	0.983	0.725	0.570	0.997
HallDetect	—	—	—	**0.762**	0.627	0.970	**0.734**	0.580	1.000	**0.771**	0.643	0.961

Table 1: Primary Results on **QAGS-CNN/DM**.

Method	Non-Generative Baselines			Mistral-7B			Llama-3.1-8B			Gemma-2-9B		
	F1	P	R	F1	P	R	F1	P	R	F1	P	R
QwenScore	0.282	0.196	0.506	—	—	—	—	—	—	—	—	—
NLI Only	**0.286**	0.208	0.455	—	—	—	—	—	—	—	—	—
LLM CoT	—	—	—	0.300	0.194	0.667	0.233	0.171	0.368	0.286	0.208	0.455
SelfCheck	—	—	—	0.330	0.200	0.944	0.319	0.190	0.994	0.274	0.161	0.909
HallDetect	—	—	—	**0.362**	0.224	0.944	**0.346**	0.212	0.947	**0.333**	0.209	0.818

Table 2: Cross-Domain Results on **TofuEval** (Dialogue Summarization).

Method	Non-Generative Baselines			Mistral-7B			Llama-3.1-8B			Gemma-2-9B		
	F1	P	R	F1	P	R	F1	P	R	F1	P	R
QwenScore	**0.704**	0.594	0.864	—	—	—	—	—	—	—	—	—
NLI Only	0.465	0.571	0.392	—	—	—	—	—	—	—	—	—
LLM CoT	—	—	—	0.654	0.567	0.773	0.373	0.438	0.326	0.375	0.524	0.292
SelfCheck	—	—	—	0.699	0.544	0.977	0.637	0.467	1.000	0.649	0.483	0.989
HallDetect	—	—	—	**0.724**	0.623	0.864	**0.678**	0.533	0.930	**0.731**	0.593	0.952

Table 3: Adversarial Results on **FalseSum**.

Method	Non-Generative Baselines			Mistral-7B			Llama-3.1-8B			Gemma-2-9B		
	F1	P	R	F1	P	R	F1	P	R	F1	P	R
QwenScore	**0.603**	0.448	0.921	—	—	—	—	—	—	—	—	—
NLI Only	0.429	0.357	0.536	—	—	—	—	—	—	—	—	—
LLM CoT	—	—	—	0.429	0.357	0.536	0.308	0.476	0.227	0.309	0.543	0.216
SelfCheck	—	—	—	0.441	0.289	0.929	0.646	0.478	0.998	**0.668**	0.502	0.997
HallDetect	—	—	—	**0.455**	0.305	0.893	**0.672**	0.512	0.977	0.632	0.467	0.980

Table 4: Specialized Results on **PubMedQA**.

of simpler heuristics. This comparison bundles decomposition with multi-scale chunking; isolating each requires the ablations noted in Limitations.

6.1.3 Recall Saturation and Threshold Sensitivity

HallDetect's recall frequently approaches 0.95–1.0 at more modest precision—a direct, intended consequence of the asymmetric aggregation, which flags a response as soon as one claim is confidently contradicted or left unsupported. Two questions follow: whether this simply reflects a degenerate "flag-everything" classifier, and why a recall-oriented operating point is appropriate here.

High recall reflects the operating point, not label collapse. The near-saturated recall is the intended effect of the asymmetric aggregation, and the detector stays discriminative rather than flagging everything. In most backbone/dataset cells the recall is strictly below 1.0 (e.g., 0.970 and 0.961 on QAGS-CNN/DM, and 0.818–0.947 on TofuEval), so a non-trivial fraction of responses are passed as faithful. An always-flag classifier would, by construction, have recall 1.0 and precision equal to the positive-class base rate; the wide spread of precision *across methods on the same data* (on QAGS, from 0.547 for QwenScore to 0.833 for NLI Only, with HallDetect in between) shows the datasets are not trivially all-positive and that HallDetect's

precision sits above the base-rate floor.

Why bias toward recall? In an auditing setting the two error types are not symmetric in cost, and the audit trail sharpens the asymmetry. A false alarm surfaces a specific claim together with the source span that triggered it, so a reviewer can dismiss it quickly by inspecting one localized assertion; a missed hallucination, by contrast, passes silently and propagates a false statement downstream, where it is typically far more expensive to detect and correct. The recall-oriented aggregation is a deliberate response to this asymmetry rather than a claim of dominance across all operating points, and the fixed-threshold numbers should accordingly be read as out-of-the-box utility at one recall-favoring point, not as a full characterization of the precision–recall trade-off.

6.1.4 Cross-Domain Behavior: TofuEval and PubMedQA

On TofuEval, absolute performance is low for *all* methods (best F1 0.362 at precision 0.224): frugal-budget faithfulness detection in dialogue remains hard, and HallDetect's edge, while consistent, is modest rather than practically sufficient. On PubMedQA, QwenScore and SelfCheck occasionally match or exceed HallDetect, likely because these scorers exploit biomedical knowledge in the backbones' pre-training as a prior for plausible statements (cf. Oukelmoun et al., 2023 on generic embeddings in clinical text). HallDetect stays competitive while additionally localizing which medical claim lacks grounding—valuable in high-stakes auditing—though we claim no superiority in this domain.

7 Discussion: Robustness and Interpretability

Generative evaluators often exhibit *self-consistency bias*, validating hallucinations that resemble their own output patterns (Wataoka et al., 2024); assigning the final judgment to a discriminative encoder subjects each factual unit to an independent, logically constrained audit, which our results indicate also confers stability under quantization. Beyond a binary flag, HallDetect produces an *audit trail*—a mapping between atomic propositions and source-evidence spans (shared with other decomposition-based evaluators)—so reviewers can focus only on flagged claims. This is the framework's central interpretability advantage: because verification is per-claim and contrastive, the trail separates *contradicted* claims, for which a source span asserts the opposite, from *unsupported additions*, for which no span provides entailment at any granularity, and ties each judgment to the specific sentence responsible. A reviewer can therefore confirm a flag by reading a single cited span rather than re-reading the whole document, which is what makes the method suitable for high-stakes auditing. A full long-source walkthrough, with the per-claim evidence spans and scores, is given in Appendix B.

8 Conclusion

We presented HallDetect,[1] a reference-free framework for interpretable hallucination detection under strict computational constraints. Under a matched 4-bit consumer-hardware protocol, claim-level decomposition with contrastive multi-scale verification improves recall over holistic NLI and remains stable where generative judges are volatile, while exposing a claim-to-span audit trail. Future work will include direct comparison to specialized fact-checkers such as MiniCheck and FActScore-style pipelines, ablations isolating decomposition from multi-scale chunking, threshold-free evaluation, Claim Grouping to reduce NLI overhead, external knowledge for extrinsic verification, and bidirectional auditing that also covers omissions.

Acknowledgments

We thank the anonymous reviewers and the area chair for their careful reading and constructive comments, which improved the paper.

9 Limitations

Relation to specialized fact-checkers. Beyond the baselines studied here, a natural next step is to place HallDetect alongside dedicated intrinsic fact-checkers such as MiniCheck (Tang et al., 2024a)—which shares our low-cost, source-grounded aim and offers a DeBERTa variant—and, relatedly, SummaC (Laban et al., 2022) and QAFactEval (Fabbri et al., 2022). Section 2.2 discusses how each relates to our design, and a direct empirical comparison, beginning with MiniCheck, is a valuable extension we look forward to pursuing. The present study concentrates on comparably resourced generic and generative baselines—the set-

[1]The source code for HallDetect and the FED_{score} implementation are publicly available at: https://github.com/achok7893/HallDetect

ting in which our design choices are most cleanly isolated.

Attribution of the recall gain. HallDetect improves over the document-level NLI baseline through two design choices working together—atomic decomposition and multi-scale chunking—and the chunking in particular is what keeps every claim–evidence pair within the encoder's context window, where a holistic pass would truncate long sources. We therefore attribute the gain to this combined design, and a controlled ablation that holds each factor fixed (and constrains both methods to inputs within the encoder's length budget) is a natural next step to quantify their relative shares. In the same spirit, decomposing the contrastive score into entailment-only and contradiction-only variants would quantify how much the contradiction channel adds on claims that are unsupported rather than explicitly contradicted—a lightweight analysis we leave to future work.

Omissions are out of scope. HallDetect verifies the claims a generation *makes*; it is silent about information the source contains but the generation *drops*. Omission is a distinct failure mode, and in clinical reporting often a more consequential one, since a summary can be perfectly faithful in every claim it states and still be unusable for having left out the decisive finding. Detecting it requires verification in the reverse direction—enumerating salient source units and checking whether each is realized in the output (Oukelmoun et al., 2025a,b). Because both directions share the same atomic-unit decomposition and the same entailment machinery, coupling them into a single bidirectional audit, in which the claim-to-span trail reported here is complemented by a span-to-claim coverage trail, is a natural extension of the framework rather than a separate system.

A parameter-free, recall-oriented operating point. We evaluate at the fixed default threshold of 0.5 by design, not for convenience: in real-world auditing there is no labeled held-out set to tune a decision threshold on, so a method must work out of the box at a parameter-free operating point. Rather than tune our way to a useful cutoff, we engineer the asymmetric aggregation so that this untuned default is already recall-oriented—a single unsupported or contradicted claim flags the response, the safe default when a missed hallucination is costlier than a false alarm (Section 6.1.3).

Efficiency reported architecturally. We characterize cost in hardware-independent terms—parameter counts, VRAM, and the number of NLI passes ($O(n \cdot |\mathcal{K}|)$)—which is what makes the frugality claim reproducible across setups. Wall-clock and energy profiling would complement these figures, as would a head-to-head against a single long-context LLM call over the full source, where KV caching amortizes much of the extraction context; measuring whether chunked discriminative verification is also cheaper in wall-clock terms is left to future work.

Reproducibility, backbones, and scope. All results come from deterministic, single-seed runs, which keeps them exactly reproducible; adding variance or significance estimates over multiple seeds would further strengthen the evidence. The backbones (Llama-3.1, Gemma-2, Mistral-7B) are widely deployed 2024 families, and because the framework is backbone-agnostic, extending it to newer families and broader suites such as LLM-AggreFact is straightforward future work. By design the method targets *intrinsic* faithfulness—consistency with the provided source—and does not attempt *extrinsic* fact-checking against world knowledge; the main remaining challenge within that scope is coreference during extraction, where a pronoun severed from its antecedent can yield a claim faithful in wording but ambiguous in isolation, which the verifier may in turn surface as a spurious contradiction. Directly measuring extractor faithfulness—a targeted human study of coverage, atomicity, and meaning preservation across the three backbones (Appendix A)—is a natural next step.

10 Impact Statement

Intended and beneficial impact. HallDetect is designed as a research tool for auditing the factual consistency of generated text, and its central design goal—computational frugality—is what shapes its potential societal impact. Because all components run under 4-bit quantization on a single consumer-grade GPU (e.g., an RTX 3060/4060), with a verification engine that requires under 1 GB of VRAM, faithfulness auditing can be carried out locally rather than through proprietary, API-based services. This keeps sensitive source material on-premises, substantially lowers the energy footprint and hardware barrier relative to API-based or full-precision "LLM-as-a-judge" approaches, and widens access

to faithfulness checking for practitioners without large compute budgets. The claim-to-span audit trail further supports responsible use by letting human reviewers see *why* a response was flagged rather than trusting an opaque score.

Risks and responsible use. These benefits come with risks that warrant care. HallDetect is not a certified safety system. In high-stakes domains such as the biomedical setting evaluated here, an over-reliance on automated faithfulness scores could create a false sense of security: a high FED_{score} indicates intrinsic consistency with the provided source, not real-world truth, and the framework verifies claims only against the supplied document rather than against external world knowledge. We therefore recommend that HallDetect be used to support, not replace, human oversight in any decision-critical deployment.

Bias and fairness. HallDetect inherits the biases of its constituent models: the generative claim extractors may omit or distort propositions, and the DeBERTa-v3 NLI engine may encode systematic errors learned during pretraining. Because all benchmarks are in English, our conclusions may not transfer to other languages or writing conventions, which may limit who is served equally well by the method. Results are reported from single deterministic runs and should be interpreted accordingly.

Data, licensing, and research conduct. All experiments rely exclusively on publicly available research benchmarks—QAGS-CNN/DM (Wang et al., 2020), TofuEval (Tang et al., 2024b), FalseSum (Utama et al., 2022), and PubMedQA (Jin et al., 2019)—and on publicly released pretrained models, used consistently with their intended research use. We did not collect any new data, and the corpora consist of public news articles, dialogue summaries, and biomedical abstracts that are not known to contain private or personally identifying information. The HallDetect code and FED_{score} implementation we release are intended for research use only, consistent with the access conditions of the underlying datasets and models. AI-based coding and writing assistants were used for code scaffolding and for language editing of this manuscript; all scientific claims, experimental designs, and reported results were produced and verified by the authors.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Amos Azaria and Tom Mitchell. 2023. The internal state of an llm knows when it's lying. In *Findings of the Association for Computational Linguistics: EMNLP 2023*.

Sourav Banerjee, Ayushi Agarwal, and Saloni Singla. 2025. Llms will always hallucinate, and we need to live with this. In *Intelligent Systems Conference*, pages 624–648. Springer.

Lingjiao Chen, Matei Zaharia, and James Zou. 2024. How is chatgpt's behavior changing over time? *Harvard Data Science Review*, 6(2).

Sihao Chen, Senaka Buthpitiya, Alex Fabrikant, Dan Roth, and Tal Schuster. 2023. PropSegmEnt: A large-scale corpus for proposition-level segmentation and entailment recognition. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 8874–8893.

Florian E Dorner, Vivian Y Nastl, and Moritz Hardt. 2024. Limits to scalable evaluation at the frontier: Llm as judge won't beat twice the data. *arXiv preprint arXiv:2410.13341*.

Ron Eliav, Arie Cattan, Eran Hirsch, Shahaf Bassan, Elias Stengel-Eskin, Mohit Bansal, and Ido Dagan. 2025. Clatter: Comprehensive entailment reasoning for hallucination detection. *arXiv preprint arXiv:2506.05243*.

Alexander R Fabbri, Wojciech Kryściński, Bryan McCann, Caiming Xiong, Richard Socher, and Dragomir Radev. 2021. Summeval: Re-evaluating summarization evaluation. *Transactions of the Association for Computational Linguistics*, 9:391–409.

Alexander Richard Fabbri, Chien-Sheng Wu, Wenhao Liu, and Caiming Xiong. 2022. Qafacteval: Improved qa-based factual consistency evaluation for summarization. In *Proceedings of the 2022 conference of the north american chapter of the association for computational linguistics: Human language technologies*, pages 2587–2601.

Tobias Falke, Leonardo FR Ribeiro, Prasetya Ajie Utama, Ido Dagan, and Iryna Gurevych. 2019. Ranking generated summaries by correctness: An interesting but challenging application for natural language inference. In *Proceedings of the 57th annual meeting of the association for computational linguistics*, pages 2214–2220.

Anisha Gunjal and Greg Durrett. 2024. Molecular facts: Desiderata for decontextualization in LLM fact verification. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 3751–3768.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2021. Debertav3: Improving deberta using electra-style pretraining with gradient-disentangled embedding sharing. *arXiv preprint arXiv:2111.09543*.

Minda Hu, Bowei He, Yufei Wang, Liangyou Li, Chen Ma, and Irwin King. 2024. Mitigating large language model hallucination with faithful finetuning. *arXiv preprint arXiv:2406.11267*.

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, et al. 2025. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ACM Transactions on Information Systems*, 43(2):1–55.

Qiao Jin, Bhuwan Dhingra, Zhengping Liu, William Cohen, and Xinghua Lu. 2019. Pubmedqa: A dataset for biomedical research question answering. 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 2567–2577.

Wojciech Kryściński, Bryan McCann, Caiming Xiong, and Richard Socher. 2020. Evaluating the factual consistency of abstractive text summarization. In *Proceedings of the 2020 conference on empirical methods in natural language processing (EMNLP)*, pages 9332–9346.

Philippe Laban, Tobias Schnabel, Paul N. Bennett, and Marti A. Hearst. 2022. SummaC: Re-visiting NLI-based models for inconsistency detection in summarization. *Transactions of the Association for Computational Linguistics*, 10:163–177.

Hongzhao Li, Hongyu Wang, Xia Sun, Hua He, and Jun Feng. 2024. Prompt-Guided Generation of Structured Chest X-Ray Report Using a Pre-trained LLM. In *2024 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6.

Junyi Li, Xiaoxue Cheng, Wayne Xin Zhao, Jian-Yun Nie, and Ji-Rong Wen. 2023. Halueval: A large-scale hallucination evaluation benchmark for large language models. *arXiv preprint arXiv:2305.11747*.

Zhen Li, Yupeng Su, Songmiao Wang, Runming Yang, Congkai Xie, Aofan Liu, Ming Li, Jiannong Cao, Ngai Wong, and Hongxia Yang. 2025. Quantization meets reasoning: Exploring and mitigating degradation of low-bit llms in mathematical reasoning. *arXiv preprint arXiv:2505.11574*.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

Nelson F Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2024. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 12:157–173.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. 2023. G-eval: Nlg evaluation using gpt-4 with better human alignment. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 2511–2522.

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.

Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan McDonald. 2020. On faithfulness and factuality in abstractive summarization. *arXiv preprint arXiv:2005.00661*.

Tim Miller. 2019. Explanation in artificial intelligence: Insights from the social sciences. *Artificial intelligence*, 267:1–38.

Sewon Min, Kalpesh Krishna, Xinxi Lyu, Mike Lewis, Wen-tau Yih, Pang Wei Koh, Mohit Iyyer, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2023. FActScore: Fine-grained atomic evaluation of factual precision in long form text generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 12076–12100.

Achir Oukelmoun, Nasredine Semmar, Gaël de Chalendar, Clement Cormi, Mariame Oukelmoun, Eric Vibert, and Marc-Antoine Allard. 2025a. Detecting omissions in LLM-generated medical summaries. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing: Industry Track*.

Achir Oukelmoun, Nasredine Semmar, Gaël de Chalendar, Clément Cormi, Mariame Oukelmoun, Eric Vibert, and Marc-Antoine Allard. 2025b. Détection des omissions dans les résumés médicaux générés par les grands modèles de langue. In *Actes de la 32ème Conférence sur le Traitement Automatique des Langues Naturelles (TALN 2025)*.

Achir Oukelmoun, Nasredine Semmar, Gaël De Chalendar, Enguerrand Habran, Eric Vibert, Emma Goblet, Mariame Oukelmoun, and Marc-Antoine Allard. 2023. A study on the relevance of generic word embeddings for sentence classification in hepatic surgery. In *2023 20th ACS/IEEE International Conference on Computer Systems and Applications (AICCSA)*, pages 1–8. IEEE.

Annalisa Szymanski, Noah Ziems, Heather A Eicher-Miller, Toby Jia-Jun Li, Meng Jiang, and Ronald A Metoyer. 2025. Limitations of the llm-as-a-judge approach for evaluating llm outputs in expert knowledge tasks. In *Proceedings of the 30th International Conference on Intelligent User Interfaces*, pages 952–966.

Liyan Tang, Philippe Laban, and Greg Durrett. 2024a. MiniCheck: Efficient fact-checking of LLMs on grounding documents. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8818–8847.

Liyan Tang, Igor Shalyminov, Amy Wong, Jon Burnsky, Jake Vincent, Yu'an Yang, Siffi Singh, Song Feng, Hwanjun Song, Hang Su, et al. 2024b. Tofueval: Evaluating hallucinations of llms on topic-focused dialogue summarization. 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 4455–4480.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018. Fever: a large-scale dataset for fact extraction and verification. *arXiv preprint arXiv:1803.05355*.

Prasetya Utama, Joshua Bambrick, Nafise Sadat Moosavi, and Iryna Gurevych. 2022. Falsesum: Generating document-level nli examples for recognizing factual inconsistency in summarization. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2763–2776.

David Wan, Koustuv Sinha, Srinivasan Iyer, Asli Celikyilmaz, Mohit Bansal, and Ramakanth Pasunuru. 2024. ACUEval: Fine-grained hallucination evaluation and correction for abstractive summarization. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 10036–10056.

Alex Wang, Kyunghyun Cho, and Mike Lewis. 2020. Asking and answering questions to evaluate the factual consistency of summaries. *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*.

Miriam Wanner, Seth Ebner, Zhengping Jiang, Mark Dredze, and Benjamin Van Durme. 2024. A closer look at claim decomposition. In *Proceedings of the 13th Joint Conference on Lexical and Computational Semantics (*SEM 2024)*, pages 153–175.

Koki Wataoka, Tsubasa Takahashi, and Ryokan Ri. 2024. Self-preference bias in llm-as-a-judge. *arXiv preprint arXiv:2410.21819*.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in neural information processing systems*, 36:46595–46623.

A Implementation Details

This appendix documents the concrete configuration used in all experiments so that the results in Section 6 are fully reproducible. All components run locally on a single consumer-grade GPU.

A.1 Model and Runtime Configuration

Table 5 summarizes the settings of the two model components of the pipeline: the generative claim extractor (Phase I) and the discriminative NLI engine (Phase III).

Component / Parameter	**Value**
Claim Extractor (Phase I)	
Backbones	Llama-3.1-8B, Gemma-2-9B, Mistral-7B (Nous-Hermes-2-DPO)
Quantization	4-bit, GGUF `Q4_K_M`
Context window (n_{ctx})	2048 tokens
CPU threads	4
Sampling temperature	0.2
Max new tokens	512
Stop sequence	`"\n\n"`
Claim cap	$\min(10,\ 4{\cdot}n_{\text{sent}})$
NLI Verifier (Phase III)	
Model	DeBERTa-v3-large (mnli-fever-anli-ling-wanli)
Parameters	435M
Inference	single forward pass / claim
Input handling	truncation enabled
VRAM footprint	< 1 GB
Chunking (Phase II)	
Granularities (m)	$\{1, 2, 4, 8, 16\}$
Sentence splitter	NLTK `punkt`

Table 5: Hyperparameters and runtime configuration of the HallDetect pipeline.

A.2 Computational Cost Accounting

Table 6 makes the frugality comparison of Section 6 explicit. The dominant cost across pipelines is autoregressive LLM *decoding*, so we report, per evaluated instance, the number of such generative passes, what the LLM must read, and the auxiliary (embedding or NLI) scorer with its parameter count and pass type. Two contrasts stand out. First, HallDetect issues a *single* generative pass, and only over the short response R; unlike the LLM CoT and SelfCheck judges, the long source document is never placed in a generative context—it is seen only by the 435M encoder, one non-autoregressive forward pass at a time. Second, the verifier's parameter footprint (435M, $<$ 1 GB VRAM) is roughly 5% of the 8B-class extractor and runs on

the same consumer GPU, so the verification stage is sub-dominant despite scaling as $O(n \cdot |\mathcal{K}|)$ passes ($|\mathcal{K}| \leq 31$). We note that a trained compact checker such as MiniCheck (Tang et al., 2024a) occupies a comparably lightweight point in this space—indeed it can skip generative extraction entirely—which is why it is the natural target of the controlled comparison we flag in Section 9; we do not claim a frugality advantage over it, only over the generative judges.

Method	Gen. passes	LLM reads	Scorer / passes
QwenScore	0	—	Embed 0.5B / 2
NLI Only	0	—	DeBERTa 435M / 1
LLM CoT	1	full $D{+}R$	—
SelfCheck	5	$D{\rightarrow}$summary	Embed 0.5B / 5
HallDetect	1	R only ($\leq$512 tok)	DeBERTa 435M / $\leq 31n$

Table 6: Per-instance cost accounting. "Gen. passes" counts autoregressive LLM decodes (the dominant cost); "Scorer / passes" gives the auxiliary embedding or NLI model and its number of single (non-autoregressive) forward passes, with n the claim count and $|\mathcal{K}| \leq 31$ the chunk-library size. HallDetect confines generation to one short-input pass and offloads all source verification to a sub-1 GB encoder.

A.3 Claim Extraction Prompt

The number of extracted claims is capped dynamically as $\min(10,\ 4{\cdot}n_{\text{sent}})$, where n_{sent} is the sentence count of the response. The prompt is deliberately *copy-faithful*: the extractor is instructed to reproduce claims verbatim, including implausible or self-contradictory statements, so that verification is performed on the model's actual assertions rather than on a silently corrected paraphrase.

```
Extract the factual claims from the
following answer. Each claim must be
taken exactly as it is written in the
answer, even if it appears implausible,
incorrect, or contradictory. Do not
use outside knowledge. Do not fix,
rephrase, or interpret claims. Your task
is to copy the claims into standalone
sentences exactly as stated. List up to
{max_claims} claims. No explanations, no
corrections.
Answer:
"""{answer}"""
Claims:
```

Raw output is post-processed by stripping enumeration artifacts (leading digits, bullets, hyphens, and trailing punctuation) and discarding empty lines, yielding the atomic claim set $\{c_1, \ldots, c_n\}$.

Copy-faithfulness is enforced at three points: the verbatim instruction above, a low decoding temperature (0.2), and the deterministic post-processing just described, which together keep the extracted claims close to the model's actual wording rather than to a paraphrase. We do not add a separate certification step (e.g., an entailment check of each claim back against R), so the residual risk is the familiar one for decomposition pipelines—chiefly coreference, where a pronoun severed from its antecedent yields a claim faithful in wording but ambiguous in isolation; we treat this within the pipeline's limitations (Section 9). Measuring extractor faithfulness directly, for instance through a targeted human study of coverage, atomicity, and meaning preservation across the three backbones, is a natural next step.

A.4 Context Library Construction

For each granularity $m \in \{1, 2, 4, 8, 16\}$, the source document D is split into m roughly equal sentence groups (chunk size $= \lceil n_{\text{sent}}/m \rceil$), and the union of all groups forms the context library $\mathcal{K}$. For a document of n_{sent} sentences this yields up to $1{+}2{+}4{+}8{+}16 = 31$ overlapping chunks. Crucially, the best entailment chunk and the best contradiction chunk are selected *independently*: for each claim the engine retains the chunk that most strongly supports it and, separately, the chunk that most strongly contradicts it. This multi-scale, dual-selection view ensures that the relevant evidence is salient at *some* granularity, mitigating the dilution that arises when verifying against the full document at a single scale.

A.5 Scoring and Aggregation

For each claim c_i, the NLI engine is run against every chunk $k \in \mathcal{K}$, and the best entailment and contradiction probabilities are retained independently:

$$\begin{aligned} Score(c_i) = &\max_{k \in \mathcal{K}} P_{\text{NLI}}(E \mid k, c_i) \\ &- \max_{k \in \mathcal{K}} P_{\text{NLI}}(C \mid k, c_i) \end{aligned} \tag{6}$$

The per-claim scores are aggregated with a geometric mean to obtain the final FED_{score} (Eq. 2). A practically important detail is the treatment of the aggregation edge case: whenever *any* $Score(c_i) \leq 0$ (i.e., contradiction evidence outweighs entailment evidence for at least one claim), the geometric mean is defined to be 0, so that $FED_{score} = 1$. This realizes the asymmetric penalty discussed in

Section 6: a single confidently contradicted claim is sufficient to flag the entire response as unfaithful. A binary verdict is obtained by thresholding at 0.5 (*hallucinated* if $FED_{score} \geq 0.5$), with no per-dataset tuning.

B Worked Example: A Long-Source Audit Trail

To show how HallDetect operates on realistic, multi-sentence inputs and why its output is auditable, we trace the full pipeline on a single news-style source document and a generated summary. Probabilities are reported on the $[0, 1]$ scale (the implementation stores them as percentages internally) and are illustrative of the model's behavior rather than logged values.

Source document D ($n_{\text{sent}} = 8$).

> **(s1)** The city council approved a new transit plan on Tuesday after months of debate. **(s2)** The plan allocates $4.2 million to extend the eastern light-rail line by three stations. **(s3)** Mayor Lena Ortiz said the expansion would cut average commute times in the eastern districts by roughly fifteen minutes. **(s4)** Construction is scheduled to begin in March 2026 and is expected to last two years. **(s5)** Council member David Hsu voted against the proposal, citing concerns about the municipal budget. **(s6)** The council also set aside $800,000 for accessibility upgrades at existing stations. **(s7)** Local business owners welcomed the decision, anticipating increased foot traffic. **(s8)** The plan does not include any changes to existing bus routes.

Generated response R.

> The council approved a $4.2 million plan to extend the light-rail line. Mayor Ortiz opposed the expansion. The new stations will feature underground parking.

Phase I — Decomposition. The extractor returns three atomic claims: c_1 = "The council approved a $4.2 million plan to extend the light-rail line."; c_2 = "Mayor Ortiz opposed the expansion."; c_3 = "The new stations will feature underground parking."

Phase II — Chunking. With $n_{\text{sent}} = 8$, the granularities $m \in \{1, 2, 4, 8, 16\}$ produce chunk sizes $\{8, 4, 2, 1, 1\}$, i.e. one whole-document chunk, two 4-sentence halves, four 2-sentence windows, and (for both m=8 and m=16) eight single-sentence chunks, for $1+2+4+8+8 = 23$ distinct windows in $\mathcal{K}$.

Phase III/IV — Per-claim verification and audit trail. Table 7 reports, for each claim, the *independently* selected best-entailing and best-contradicting chunks together with their probabilities and the resulting $Score(c_i)$. This claim→span mapping forms the audit trail: a human reviewer can read why each claim passed or failed by inspecting the cited sentence rather than re-reading the whole document.

Claim	E span	$P(E)$	C span	$P(C)$	$Score$
c_1	s2, m=8	0.93	s5, m=8	0.04	+0.89
c_2	s5, m=8	0.18	s3, m=8	0.86	−0.68
c_3	s2, m=4	0.06	s8, m=8	0.21	−0.15

Table 7: Audit trail for the long-source example. $P(E)$ and $P(C)$ are the best entailment / contradiction probabilities, and the "span" columns give the chunk (sentence, granularity m) that achieved each, selected independently across all granularities. c_1 is grounded in s2; c_2 is contradicted by s3 (the mayor in fact *supported* the plan, while the opposition came from council member Hsu in s5); c_3 is unsupported, with no chunk providing entailment at any granularity.

Aggregation. Both c_2 and c_3 yield $Score(c_i) < 0$, so the geometric-mean aggregation returns 0 and the response is flagged: $FED_{score} = 1 - 0 = 1.0 \geq 0.5$. Beyond this scalar verdict, the trail localizes the failure to a confident *contradiction* (c_2, traced to s3) and an *unsupported addition* (c_3, no entailing span), the two distinct error modes targeted by the framework. The correctly supported claim c_1 is left untouched, so a reviewer's attention is directed only to the two problematic spans.

Why multi-scale chunking matters here. Had verification been performed only at m=1 (the monolithic NLI baseline of Section 5), the contradiction in c_2 would have been diluted: averaged across all eight sentences, the document is broadly *about* an approved expansion, so a holistic entailment check returns a moderate support signal and the localized "opposed" error is masked. Selecting the best contradicting window at m=8 isolates s3, recovering the signal. This is the mechanism behind the recall gains reported in Section 6 and the reason HallDetect's judgments remain inspectable on long-form inputs.