

All Claims Are Equal, but Some Claims Are More Equal Than Others: Importance-Sensitive Factuality Evaluation of LLM Generations

Miriam Wanner^{1*}, Leif Azzopardi^{2,3}, Paul Thomas², Soham Dan^{4**},
Benjamin Van Durme^{1,2}, Nick Craswell²

¹Johns Hopkins University, ²Microsoft, ³University of Strathclyde, ⁴Scale AI
mwanner5@jhu.edu

Abstract

Existing methods for evaluating the factuality of large language model (LLM) responses treat all claims as equally important. This results in misleading evaluations when vital information is missing or incorrect as it receives the same weight as peripheral details, raising the question: how can we reliably detect such differences when there are errors in key information? Current approaches that measure factuality tend to be insensitive to omitted or false key information. To investigate this lack of sensitivity, we construct VITALERRORS, a benchmark of 6,733 queries with minimally altered LLM responses designed to omit or falsify key information. Using this dataset, we demonstrate the insensitivities of existing evaluation metrics to key information errors. To address this gap, we introduce VITAL, a set of metrics that provide greater sensitivity in measuring the factuality of responses by incorporating the relevance and importance of claims with respect to the query. Our analysis demonstrates that VITAL metrics more reliably detect errors in key information than previous methods. Our dataset, metrics, and analysis provide a foundation for more accurate and robust assessment of LLM factuality.¹

1 Introduction

Fact checking machine generated text has become increasingly important with the growing mainstream use of Large Language Models (LLMs). These models are known to produce hallucinations, which can sometimes be harmful (Bender et al., 2021; Dev et al., 2022). As a result, fact checking is often incorporated into such systems, to provide higher quality responses by removing such errors. Previous methods for evaluating factual

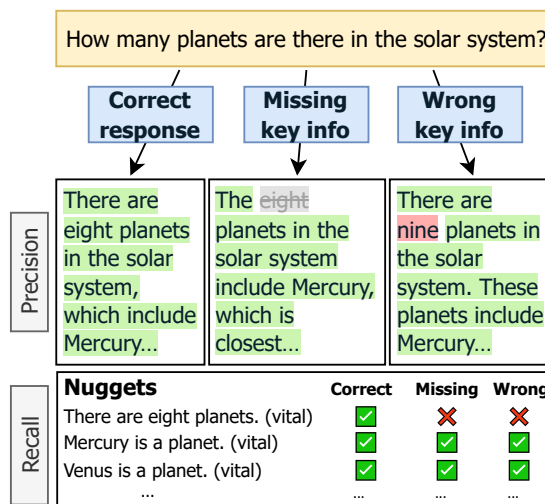


Figure 1: Three responses to the same query receive similarly high precision and recall scores. Precision and recall metrics are insensitive to errors in key information for long responses. Both erroneous responses still achieve high precision and recall scores.

precision employ a *decompose-then-verify* framework, which involves first decomposing claims from generated responses into atomic subclaims², independently verifying each, and aggregating into a score (Min et al., 2023; Song et al., 2024; Wanner et al., 2024b; Huang et al., 2025). Similar methods are used in evaluating recall, where references are used to generate *nuggets*—pieces of information vital for answering a given query. The percentage of nuggets supported by the response is the recall score (Voorhees, 2003; Pradeep et al., 2024, 2025).

These factuality metrics treat each piece of information (subclaim) equally, and tend not to explicitly consider whether a claim in the response is vital to answering a user’s query. Not all parts of the response are equally relevant or important, and as such, some factuality-based errors are more

*Work done during an internship at Microsoft.

**Work done while author was working at Microsoft.

¹Code is available at <https://github.com/miriamwanner/vitalerrors>.

²In this work we use the term *subclaim*, but other works use atomic-fact, -claim, or -preposition for the same concept.

egregious than others. However, current metrics are unable to distinguish or fail to delineate between similar responses with specific errors. One missing or wrong key piece of information can invalidate an otherwise good response, which should be reflected in evaluations. Figure 1 shows three different responses to a query asking for the number of planets in the solar system. The first response correctly answers with “eight,” the second provides the list of planets, but is missing the key piece of information (“eight”), and the third provides the incorrect answer “nine.” Despite the missing or incorrect key information, both erroneous responses still achieve high precision and recall scores. The most important piece of information (the answer “eight”) is treated as only one small piece of information in a long response, where the final score is agnostic to its importance in fundamentally answering the user’s query.

In this paper, we investigate the impact of *small* but *important* errors due to missing key or incorrect claims in responses that are vital to answering the query, whether these can be detected using traditional fact checking metrics, and if considering importance improves their detectability and sensitivity. We introduce a set of VITAL metrics for identifying errors in key information. We then construct a new dataset that allows us to rigorously evaluate how key information errors affect the scoring of these systems. To create our dataset, we adversarially alter machine generated responses to queries, and generate minimally changed responses with missing and wrong key information to create the VITALERRORS dataset. We run previous methods of precision and recall, finding they are insensitive to these key errors, and show that VITAL metrics are sensitive to these key errors, reflecting when key information is missing or wrong.

Our contributions are as follows: (1) We demonstrate how existing methods fail to capture important differences in responses. We show that there is no notion of importance within subclaims, and why this is a problem. (2) We introduce a new set of metrics, which we call VITAL, which includes subclaim ranking and importance labeling for precision metrics. (3) We construct VITALERRORS, a challenging dataset for evaluating key information errors. Using queries from six QA datasets, we generate normal LLM responses, and adversarially alter each response to omit or falsify key information. We demonstrate better error detection using VITAL metrics on the VITALERRORS benchmark.

2 Metrics

To ground our proposed VITAL metrics, we first provide an overview of existing factuality and nugget based metric frameworks.

2.1 Factual Precision

Min et al. (2023) introduced a framework for fact verification. The system first decomposes LLM-generated text into “*atomic facts*”, which are then independently verified against a trusted grounding source. Many variants of this decompose-then-verify framework exist. While previous works focused on factual *precision*, SAFE (Wei et al., 2024) extends this work to include recall, with a selected hyperparameter K , the number of supported facts for a response to achieve full recall. VERIScore (Song et al., 2024) builds on SAFE, using the same measure of recall, while only decomposing text into *verifiable* subclaims. Liu et al. (2025) highlight the issue of recall in previous metrics and introduce VERIFACT, evaluating responses with *reference fact sets*, much like nuggets. Zhao et al. (2025) study the length-factuality tradeoff, finding a quality degradation with increased response length.

Wanner et al. (2024b) examined the decomposition step, showing these factual precision pipelines are sensitive to different decomposition methods. Jiang et al. (2025) extended this work to fix decomposition redundancy, and introduce CORE, a method for subclaim selection based on uniqueness and informativeness. CORE is a subclaim filtering method, whereas this paper works on evaluating and detecting errors in these important subclaims. Gunjal and Durrett (2024) and Wanner et al. (2024a) formalized, implemented, and incorporated decontextualization in factual precision frameworks. Further works have researched diverse factual information (Samarinas et al., 2025), better localizing factual errors (Cattan et al., 2024), entailment reasoning (Eliav et al., 2025), and factual precision in medicine (Huang et al., 2025). Godbole and Jia (2025) re-evaluated these evaluation metrics, and find pitfalls in current factuality metrics.

Benchmarks such as HaluEval (Li et al., 2023) and Factcheck-Bench (Wang et al., 2024) evaluate hallucination and factual errors in realistic LLM responses; our work differs in focusing on the sensitivity of evaluation metrics rather than categorizing error types. Related to our notion of claim impor-

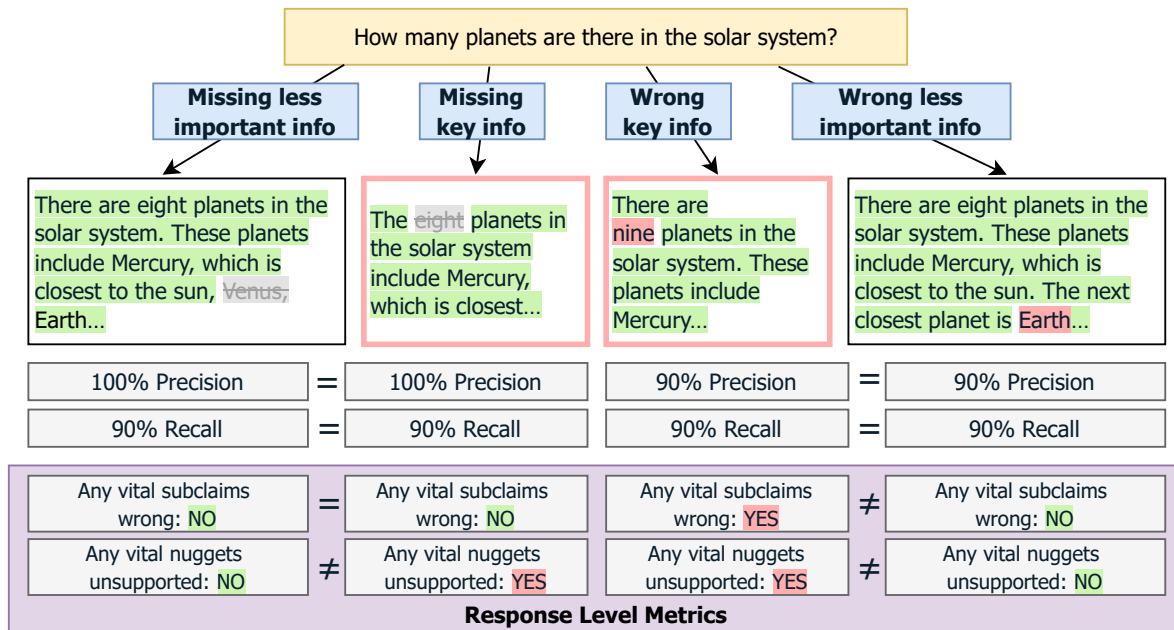


Figure 2: Four responses to the same query with varying error type and severity, demonstrating that not all errors are equal. Standard precision/recall give similar scores, while response-level metrics distinguish missing or wrong vital information from peripheral errors.

tance is work on claim check-worthiness, which identifies which claims merit verification at all (Maier and Šnajder, 2024), while other work focuses on the granularity of decomposed claims (Mitra et al., 2025). LLM-based factuality judgments by LLMs (especially from GPT-4 class models) have been repeatedly validated against human annotators across the decompose-then-verify literature (Min et al., 2023; Wei et al., 2024; Song et al., 2024; Jiang et al., 2025), finding strong alignment between the two. We use FACTSCORE (Min et al., 2023) in our experiments for factual precision evaluation, employing gpt-4o for atomic fact generation and evaluation.

2.2 Nugget Recall

Voorhees (2003) introduced *nuggets* as a minimal, atomic claim that serves as a correct and useful piece of information in answering a query. This formalization provided a foundation for evaluating relevance and recall of retrieved documents or responses for a given query. Each nugget is assigned an importance label, vital or okay, depending on how important the nugget is for answering the query. Lin and Demner-Fushman (2006) showed how importance can be extended to grades of importance by combining ratings from multiple assessors (i.e., nugget importance is based on annotator popularity). Nugget based evaluation

metrics have been used for not only assessing answer quality, but also document relevance, summarization (Nenkova and Passonneau, 2004), RAG (Pradeep et al., 2025), and conversational search (Abbasiantaeb et al., 2025).

Recently, Pradeep et al. (2025) implement an automated LLM-based framework using the original vital/okay labels, called AutoNuggetizer, for nugget creation, ranking, assignment, and scoring (Pradeep et al., 2024). This framework has been validated against human assessors, demonstrating strong agreement between fully automated LLM evaluation and manual human evaluation at the run, topic, and nugget levels (Pradeep et al., 2025). Given a list of nuggets, we can then determine which nuggets are most important to the user. We use AutoNuggetizer with gpt-4o as the LLM for generating nuggets, importance labeling, and evaluating responses. A query and set of documents are provided to the LLM, and a set of nuggets are created. The LLM is then prompted to label each nugget as either *vital* (needed in a response) or *okay* (good to have, but not necessary). Finally, the LLM is given the set of nuggets and response to the query, and evaluates if each nugget is supported, partially supported, or not supported by the response. Pradeep et al. (2025) report multiple metrics for recall evaluation using different combinations and weighting of vital/okay nuggets,

which can be supported, partially supported, or not supported. For this paper we use their “All Strict” metric, which we denote as NUGGETRECALL.

2.3 Comparison of Metrics

A central challenge in evaluating text lies in how we decompose information into units that can be assessed or used for assessment. Nuggets (Voorhees, 2003) and claim decomposition (Min et al., 2023; Wanner et al., 2024b) share surface similarities, but differ in scope, assumptions, and grounding. Here we lay out the differences, which are important to understand the contribution of our work.

The nugget framework assumes that a sufficiently broad pool of potentially relevant material can be assembled—typically by pooling results from multiple retrieval systems—and that annotators (or now LLMs) can then extract nuggets representing the key information needs of a query. Thus, nugget-based evaluation is inherently *per-query* and grounded in relevance judgments across multiple sources. In contrast, factuality metrics operate by decomposing an individual response into smaller *per-response* units, or subclaims. These subclaims are then verified against retrieved evidence. This framework does not rely on pooling across systems, but rather assumes that the response itself contains the material to be fact-checked. Subclaims emphasize verification rather than relevance, and their validity is tightly coupled to the retrieval system used for grounding.

Both frameworks share the intuition of breaking down long-form text into smaller, verifiable units, however they diverge in scope (query-centric vs. response-centric), evidence base (rich pool of candidate material vs. retrieval material), objective (relevance vs. factual correctness), and granularity (nugget vs. claim). In this work, we build on these previous systems. We adopt the nugget notion of *importance*, i.e., that some pieces of information are vital to user needs, while retaining the original method of claim decomposition. These metric changes address issues of equal subclaim weighting and lack of subclaim importance.

3 Methods

To further motivate the need for more sensitive evaluation metrics, consider the example in Figure 2—which shows that not all errors are equal. On the left are two responses with missing information, one which omits the answer “eight planets,” the

other which omits “Venus is a planet.” Both answers achieve the same high precision and recall scores, despite varying quality. Similarly, the two responses on the right both contain wrong information, one answers the query wrong with “nine planets,” the other mixing up Earth and Venus in planet ordering. As with the previous two examples, both responses have equally high scores. Incorrect or missing key information is more harmful than incorrect or missing peripheral information, since this key information receives the most attention. Comparing these responses only using these metrics, they all seem to be of similar quality, despite varying magnitude of error. On the other hand, using response-level metrics (shown in the purple box) allows us to differentiate based on errors in key information, correctly flagging the middle responses as missing and/or containing false key information. We blend the notion of importance in nuggets into FACTSCORE by introducing VITAL metrics and construct VITALERRORS in the following sections. Put simply, not all claims are equal; some are more important than others.

3.1 VITAL Metrics

We draw upon FACTSCORE and nuggets to propose a novel importance based factuality score of key information for evaluating responses. Using a set of decomposed subclaims, we then rank claims by *query importance*. A claim exhibits high query importance when it addresses a central aspect of the query, and low query importance when it contributes only peripheral or background information. We task an LLM to rank subclaims independent of correctness, instead only with respect to the user’s query. In this same prompt, we also ask the LLM to give importance labels to each subclaim. We ask for an importance label to be assigned to each subclaim: *vital* (and thus critically important in answering the user’s query), *okay* (somewhat important), and *less important*. From this, we have a set of subclaims/nuggets, labeled and ordered by importance. Prompt and model details can be found in appendix section A.

Given the labels and ranking, we present two ways of evaluating key information. We evaluate using subclaims in precision metrics and nuggets in recall metrics. First we have decomposition level evaluations: VITAL_{PREC.} and VITAL_{REC.}. We only report the precision and recall for the vital set of subclaims/nuggets. This gives us a notion for how precise the response is and

Type	Response
Query	In a paragraph, could you answer: Of which tribe of Red Indians was Geronimo a chief
Normal	Geronimo was a prominent leader and medicine man of the Apache tribe, specifically belonging to the Bedonkohe band of the Chiricahua Apache. Although he is often referred to as a chief, Geronimo was not a hereditary chief in the traditional sense but rather a respected warrior and spiritual figure who played a crucial role in resisting U.S. and Mexican military campaigns during the late 19th century. His leadership and fierce resistance made him a symbol of Native American defiance and resilience during a time of intense conflict and displacement.
Missing	Although he is often referred to as a chief, Geronimo was not a hereditary chief in the traditional sense but rather a respected warrior and spiritual figure who played a crucial role in resisting U.S. and Mexican military campaigns during the late 19th century. His leadership and fierce resistance made him a symbol of Native American defiance and resilience during a time of intense conflict and displacement.
Wrong	Geronimo was a prominent leader and medicine man of the Sioux tribe, specifically belonging to the Bedonkohe band of the Chiricahua Apache. Although he is often referred to as a chief, Geronimo was not a hereditary chief in the traditional sense but rather a respected warrior and spiritual figure who played a crucial role in resisting U.S. and Mexican military campaigns during the late 19th century. His leadership and fierce resistance made him a symbol of Native American defiance and resilience during a time of intense conflict and displacement.

Table 1: An instance from VITALERRORS, with a query from TRIVIAQA, and three response types. The normal response identifies Geronimo as Apache, while missing and wrong variants omit or replace this key claim.

the coverage for core pieces of information. Let $\hat{V} = \{\hat{v}_1 \dots \hat{v}_n\}$ be the set of vital subclaims from a response and $V = \{v_1 \dots v_n\}$ be the set of vital nuggets for the query used to generate the response. Then we have $VITAL_{PREC.} = \frac{|\hat{v}_i \text{ supported}|}{|\hat{V}|}$ and $VITAL_{REC.} = \frac{|v_i \text{ supported}|}{|V|}$, where subclaim support is judged against trusted sources using a decompose-then-verify approach, with an LLM verifier grounded in the benchmark-provided reference documents (see Table 5), and nugget support is judged against the response using AutoNuggetizer’s standard evaluation.

Then we have response-level evaluations: $VITAL_{RLP}$ (Response-Level Precision) and $VITAL_{RLR}$ (Response-Level Recall). Each are boolean evaluating if any of the vital subclaims are wrong (for $VITAL_{RLP}$) and if any of the vital nuggets are missing (for $VITAL_{RLR}$).

$$VITAL_{RLP} = \begin{cases} 1, \text{ any } \hat{v}_i \text{ unsupported} \\ 0, \text{ otherwise} \end{cases}$$

$$VITAL_{RLR} = \begin{cases} 1, \text{ any } v_i \text{ unsupported} \\ 0, \text{ otherwise} \end{cases}$$

These metrics focus on only *vital* information, and evaluate and detect errors only in this part of the response³.

Note that when a response contains zero vital subclaims, $VITAL_{PREC.}$ is undefined and we report it as zero. We recommend reporting $VITAL_{PREC.}$ alongside $VITAL_{RLP}$ in these cases, as a score of

³We look into a weighted metric using *all* subclaims, described in Appendix B.

zero may otherwise be conflated with a response that has vital subclaims, but scores them incorrectly, obscuring the distinction between missing vital information entirely and vital information being present but incorrect.

3.2 VITALERRORS Dataset

To explore how current metrics and our proposed metric perform with respect to different types of errors, we constructed an evaluation test set, where we perturb LLM responses to omit or falsify key information.

Queries We used a total of 6,733 queries from six datasets of varying query type. We used data with *open-ended* and *single-answer* queries. Open-ended queries have little constraints and many possible answers are correct (e.g., “Tell me about George Washington”). Single-answer queries have a particular piece of information that must be included in a correct answer (e.g., “How many planets are in the solar system?”). Table 5 shows a summary of the datasets we used. These consisted of commonly used datasets focused on: entities (Min et al., 2023; Zhao et al., 2024a), QA (Yang et al., 2018; Kwiatkowski et al., 2019; Joshi et al., 2017), and complex reasoning (Su et al., 2024). We prompted an LLM with the queries in these datasets to get long-form machine generated answers. We then generated a *missing* and *wrong* adversarial version of each response.

Adversarial Missing and Wrong Responses

Given a query and a response, we can adversarially perturb the response to omit or falsify key

Task	Metric	Normal	Missing	Wrong
Open-Ended	<i>Subclaims</i>			
	vital	11.34	8.13	11.66
	okay	11.37	7.60	10.96
	less	6.34	4.23	6.47
	<i>TOTAL</i>	<i>29.04</i>	<i>19.96</i>	<i>29.09</i>
	<i>Nuggets</i>			
	vital		9.84	
	okay		14.37	
	<i>TOTAL</i>		<i>24.21</i>	
	<i>Subclaims</i>			
Single-Answer	vital	4.31	3.59	4.64
	okay	7.69	6.49	7.81
	less	10.36	6.83	9.84
	<i>TOTAL</i>	<i>22.36</i>	<i>16.90</i>	<i>22.29</i>
	<i>Nuggets</i>			
	vital		3.13	
	okay		9.00	
	<i>TOTAL</i>		<i>12.13</i>	

Table 2: Average counts of subclaims and nuggets across open-ended and single-answer queries. Nuggets are generated from source documents and are therefore shared across normal, missing, and wrong responses. “Less” denotes less important subclaims.

information, while minimally altering the rest of the response. These perturbations are deliberately minimal. Our goal is to isolate a specific failure in existing metrics: they assign near-identical scores to responses that differ in only a single vital subclaim. Real LLM errors involve such minimal mistakes and contradicting details (Mündler et al., 2024; Zhao et al., 2024b). To create these perturbed responses we prompt an LLM with query and response pairs, and instruct the model to remove or alter a single piece of information that is most important for answering the query. For every query, we then have three responses: (1) *normal*, (2) *missing*, where key information is omitted, and (3) *wrong*, where key information is incorrect. We can evaluate these responses with the metrics outlined in the following sections to assess how metrics shift for these minimal changes with a large impact on answer quality.

3.3 Experiments

We evaluate VITALERRORS using FACTSCORE, NUGGETRECALL, SAFE (Appendix B.1), and VITAL outlined in section 2. For each metric, we have an evaluation on normal responses, responses missing key information, and responses with wrong key information. We use gpt-4o (OpenAI et al., 2024) throughout this work as a representative model of the types of models used for evaluation, and additionally test the sensitivity of our importance rank-

ing step to this choice across four models in Appendix B.2. See Appendix A for more details and prompts. We can then compare how metrics change between these three response types, if metrics are sensitive to these minimal errors, and if they are able to detect these errors.

4 Results

From Table 2, we first observe there are fewer nuggets than subclaims (because a nugget often contains multiple subclaims) when comparing the same subset (e.g., normal open-ended subclaim total vs. nuggets total), and fewer subclaims/nuggets for single-answer queries than open-ended queries, specifically vital ones. Responses with missing key information are shorter, and therefore contain fewer subclaims, while responses with wrong key information still have a similar number of subclaims as the original response.

There are similar trends in precision and recall metrics (Table 3). In FACTSCORE results, we observe normal and missing responses are about the same, with only a small dip (about a 5-7% dip) for wrong responses. For open-ended query VITAL_{PREC.}, we see a similar decrease. In contrast, we observe a large drop between normal/missing and wrong responses in VITAL_{PREC.} (about 25% less) for single-answer queries. For NUGGETRECALL, we observe all responses (normal/missing/wrong) are within 6% of one another for open-ended responses and 9% for single-answer responses, with the highest score for normal responses, and lowest for missing responses. Open-ended VITAL_{REC.} exhibits similar trends, with an 11% difference for normal and missing responses. We see a much larger dip of 25% for single-answer queries.

Response-level metrics in Table 4 are the best indicator of these errors in key information. Note that here, lower numbers are better, as they indicate a lower percent of error. VITAL_{PREC.} reports the percentage of responses with any vital subclaims wrong. We observe good detection of incorrect responses here, with 75% (open-ended) and almost 90% (single-answer) of wrong responses getting vital subclaims wrong. VITAL_{REC.} indicates the percentage of responses where any vital nugget is unsupported. There is a small difference in VITAL_{REC.} scores for all responses (less than a 5% difference) for open-ended queries. However, VITAL_{REC.} is a better detector of error in single-answer queries, with a 15% difference between normal responses

Metric Type	Dataset	Metric	Normal (%)	Missing (%)	Wrong (%)
Precision	Open-Ended	FACTSCORE	83.62	83.58	78.84
		VITAL _{PREC.}	82.69	82.88	75.40
	Single-Answer	FACTSCORE	82.58	82.75	76.63
		VITAL _{PREC.}	72.81	72.22	48.73
Recall	Open-Ended	NUGGETRECALL	24.35	18.61	23.23
		VITAL _{REC.}	40.90	29.91	37.89
	Single-Answer	NUGGETRECALL	27.71	19.13	23.49
		VITAL _{REC.}	52.44	27.45	36.70

Table 3: Precision and recall results for open-ended and single-answer queries. VITAL_{PREC.} and VITAL_{REC.} show a clear difference between normal and adversarial responses, where FACTSCORE and NUGGETRECALL are less sensitive.

and missing/wrong responses.

We additionally compare against SAFE (Wei et al., 2024) in Appendix B.1 (Figure 4). Consistent with FACTSCORE and NUGGETRECALL, SAFE shows little separation between normal, missing, and wrong responses across precision, recall, and F1, further supporting our finding that unweighted metrics are insensitive to errors in key information.

5 Analysis

5.1 VITAL Metrics

Metric changes Our experiments demonstrate that VITAL metrics are more sensitive to key information errors than prior methods. Response-level VITAL metrics in particular are best at differentiating response quality for responses with missing and wrong key information, especially for responses to single-answer queries, where missing or wrong key information directly impacts overall response correctness. Overall there is less of a difference in normal/missing response scores, indicating that it is more difficult to detect omissions than it is to identify wrong information. A crucial piece in this metric is the ranking and importance labeling of subclaims. Unlike FACTSCORE, which equally distributes weight across subclaims, VITAL metrics highlight which claims are most critical for answering a query. VITAL metrics’ importance weighting better reflects user perception: mistakes in vital

information undermine trust more than errors in peripheral details. Comparison with other precision and recall metrics can be found in appendix section B.1.

Subclaim ranking A crucial step in the precision evaluation is the subclaim ranking and importance labeling. An example of subclaim ranking is shown in Table 9. The first decomposed subclaim in a normal response is “X is a song,” which does not immediately answer the query, and is labeled as less important. After ranking, there are only two vital subclaims, the two which answer the query. Most of the subclaims are labeled as less important background knowledge, not needed for answering the query. The second response is missing key information. Because key information is missing, there are no subclaims labeled as vital. The last response contains the same first subclaim as the normal response, which also is labeled as less important. Crucially, despite the answers in the subclaims being wrong, they are labeled as vital subclaims because they are answering exactly what the query is asking. We provide further analysis on agreement between different models and human evaluation in appendix section B.2.

5.2 VITALERRORS

Adversarial responses An example instance is shown in Table 1 for the query “Of which tribe of Red Indians was Geronimo a chief?” The correct answer to this query is the Apache tribe, which is the written in the *normal* response. The *missing* response omits the first sentence of the normal response, while the *wrong* response changes “Apache” to “Sioux.” Later mentions of the Apache tribe still remain in the *wrong* response, however the initial answer is incorrect. The initial answer to the query is the part of the answer which LLM users will pay the most attention to. The *wrong* adversarial data may contain contradictions, however the initial

Metric	Normal	Missing	Wrong
Open-Ended			
VITAL _{RLP}	52.70	45.48	73.79
VITAL _{RLR}	87.22	91.18	88.58
Single-Answer			
VITAL _{RLP}	45.03	42.17	89.53
VITAL _{RLR}	54.30	68.10	68.47

Table 4: Response level metric results best capture adversarial edits. Wrong or missing vital claims are detected in adversarial responses.

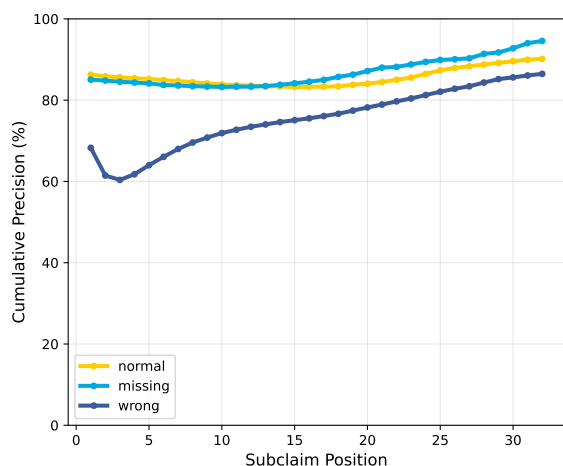


Figure 3: Cumulative precision over subclaim position for single-answer queries. Wrong responses show low precision early on due to falsified key claims, while normal and missing responses stay consistently high.

answer is altered.

Cumulative precision To assess how key information errors can get buried in a long response, we plot the cumulative FACTSCORE for the first n subclaims. This cumulative FACTSCORE for single-answer queries plot can be found in Figure 3. We observe the same trend with open-ended queries, shown in the appendix in Figure 5. In the adversarial data with wrong key information, we observe a lower averaged FACTSCORE of around 60% at the start of responses. In the data, the first subclaim is oftentimes an obvious statement used as exposition for the answer (e.g., first subclaim is “X is a song.” in Table 9), which explains the slight dip in precision between the first two subclaims. The normal response and the response missing key information has a high cumulative FACTSCORE across all subclaims.

Single-answer vs open-ended queries In single-answer queries, there is one answer the query is looking for (e.g., “Who is the author of *The Great Gatsby*?”). In open-ended queries the answer is less defined, and there are many possible valid answers (e.g., “Tell me a bio of F. Scott Fitzgerald”).

The evaluation of open-ended queries with nuggets is difficult. Nugget creation is dependent on what documents are provided during the nugget creation process, and therefore to get a high recall score, the generated text should match the content of the provided documents, which is not always likely to happen, especially for entities with di-

verse masses of information. For these open-ended queries, it is not clear how to determine what the most important piece of information is. For example, in a human biography, it is not established if a person’s occupation, accomplishments, birth place, etc. is more important than the other. All of these pieces of information tend to be included in a biography. Because of this, there are far more nuggets for these queries than for single-answer queries. In the response-level metrics, this means there is more opportunity to make a mistake in one of the vital subclaims, explaining the high percentage of responses with any vital nuggets unsupported. On the other hand, using nuggets to evaluate single-answer queries is quite simple. The answer need only contain a few pieces of information, which make up the nuggets. Fewer nuggets means there is less opportunity to make a mistake.

These issues are similar in subclaim importance labeling and ranking. Responses to open-ended queries may cover a lot of ground, with an unclear hierarchy to the subclaims within the response. This results in far more claims labeled as vital, with more possibilities of getting at least one vital subclaim wrong.

6 Conclusion

In this paper, we demonstrated how existing factual precision methods fail to distinguish between errors in key information. We introduced VITAL, a set of metrics designed to evaluate key information factual accuracy and recall. Through the construction of VITALERRORS dataset, we showed that current methods are insensitive and assign scores to responses of drastically different quality. VITAL metrics show moderate improvements over baseline precision and recall, with the response-level evaluations best capturing the impact of key information errors.

These results suggest that any robust factuality evaluation should explicitly incorporate importance-weighted subclaims rather than treating all information as equally important. One way of doing this is by reporting our VITAL decomposition- or response-level metrics to evaluate only the most critical information in responses. Another way to do this is to weight subclaims by their importance, using our prompt-based importance ranking. However, it is not entirely clear where the weight values come from. We leave this as an open question, which would require align-

ment with user preference, and is potentially dependent on query type or topic. If each subclaim is strictly more important than a subclaim ranked lower, then linear decay weighting is an option (see Appendix B), but this is a strong assumption and not applicable in most cases.

Beyond evaluation, VITAL scores offer signals for developers improving generation pipelines. A response with low $VITAL_{REC}$ but high $VITAL_{PREC}$ indicates that a model is omitting key information while remaining accurate, which motivates using retrieval augmentation to improve coverage. Conversely, low $VITAL_{PREC}$ flags responses where vital subclaims are incorrect, motivating more selective generation or filtering. In exposing these errors, VITAL also supports more interpretable fact-checking: directly localizing *how important* and *which* and claims are inaccurate or missing. An open question for future work is how vital-only responses, where non-vital subclaims are omitted entirely, affect user satisfaction, particularly for open-ended queries. We expect responses optimized for VITAL may feel incomplete to users who value broader context. We leave this tradeoff study to future work.

We recommend the evaluation of new factuality metrics on VITALERRORS to see if insensitivities exist, and if they are able to distinguish errors in difficult responses. Together, VITAL metrics and VITALERRORS offer a new benchmark in factual precision for evaluating and detecting errors in key information.

Limitations

In this study, we demonstrate shortcomings of factual precision metrics, and introduce VITAL and VITALERRORS to solve and evaluate these issues. Our approach depends on LLMs for many aspects of the framework, including decomposition, ranking, and verification. The method inherits bias and inconsistencies from the underlying model. We test sensitivity to this choice in Appendix B.2, comparing four models of varying size and architecture for the importance ranking step, finding moderate pairwise agreement. Because a fact-checking pipeline contains many LLM-dependent steps (including decomposition, importance ranking, and verification), errors at an earlier stage can propagate and affect the accuracy of later stages, such as importance labeling or support judgments. We do not separately isolate the error contribution of each stage in this work.

Like other factuality metrics, VITAL is dependent on the quality of the retrieved reference material used for verification. In this work, we use sources already paired with the query from previous datasets, and assume this grounding is factual and relevant, but using this framework with a retriever model adds another variable. Poor grounding can cause false positives or false negatives, a challenge that is particularly difficult in domains where high-quality references are sparse.

VITAL introduces an additional step over standard precision and recall frameworks, adding a ranking/importance labeling step. This step improves metric sensitivity, but requires more LLM calls (an extra importance ranking call per response), which may limit scalability. VITALERRORS considers two error types, *missing* and *wrong*, chosen to isolate and cleanly evaluate a specific question: does a metric detect when the central answer to the query is broken? Finer-grained taxonomies (e.g. partially correct, contradicted facts, hallucinated details) are valuable but are not always aligned with the main issue we aim to measure. Our contribution, importance-aware scoring, can be plugged into finer taxonomies in future work. Additionally, VITALERRORS uses adversarially and minimally inserted errors rather than naturally occurring model mistakes. While this allows controlled, isolated evaluation of metric sensitivity, it is an open question whether VITAL’s improvements generalize to more varied and less isolated errors found in real LLM outputs. Despite these limitations, VITAL and VITALERRORS are important steps towards evaluation methods that reflect the *importance* of factual errors.

Ethics Statement

This work addresses the evaluation of factuality in LLM outputs, an area with direct implications to safe and trustworthy deployment of these systems. By introducing VITAL and VITALERRORS, we aim to promote evaluation practices that more accurately identify harmful or misleading responses. Our dataset VITALERRORS was constructed using automated adversarial perturbations of machine-generated text, which ensures scalability, but may not capture the full diversity of real-world factual errors. Importance judgments for subclaims and nuggets are automatic by LLMs, which introduces the risk of reinforcing biases present in the model, either under- or over-weighting certain parts of a

response, depending on what the model views as “important.” Although our goal is to improve factuality evaluation, any benchmark can be misused if it is treated as a definite or exhaustive measure of reliability. We caution against over-reliance in high-stakes contexts (e.g., medical), without additional safeguards, domain-specific evaluation, and human oversight.

References

- Zahra Abbasiantaeb, Simon Lupart, Leif Azzopardi, Jeffrey Dalton, and Mohammad Aliannejadi. 2025. [Conversational gold: Evaluating personalized conversational search system using gold nuggets](#). In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR ’25*, page 3455–3465, New York, NY, USA. Association for Computing Machinery.
- Emily M. Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. 2021. [On the dangers of stochastic parrots: Can language models be too big?](#) In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency, FAccT ’21*, page 610–623, New York, NY, USA. Association for Computing Machinery.
- Arie Cattan, Paul Roit, Shiyue Zhang, David Wan, Roei Aharoni, Idan Szpektor, Mohit Bansal, and Ido Dagan. 2024. Localizing factual inconsistencies in attributable text generation. *ArXiv*, abs/2410.07473.
- Sunipa Dev, Emily Sheng, Jieyu Zhao, Aubrie Amstutz, Jiao Sun, Yu Hou, Mattie Sanseverino, Jiin Kim, Akihito Nishi, Nanyun Peng, and Kai-Wei Chang. 2022. [On measures of biases and harms in NLP](#). In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2022*, pages 246–267, Online only. Association for Computational Linguistics.
- Ron Eliav, Arie Cattan, Eran Hirsch, Shahaf Basan, Elias Stengel-Eskin, Mohit Bansal, and Ido Dagan. 2025. [Clatter: Comprehensive entailment reasoning for hallucination detection](#). *Preprint*, arXiv:2506.05243.
- Ameya Godbole and Robin Jia. 2025. [Verify with caution: The pitfalls of relying on imperfect factuality metrics](#). In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 22889–22912, Vienna, Austria. Association for Computational Linguistics.
- Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 542 others. 2024. [The llama 3 herd of models](#). *Preprint*, arXiv:2407.21783.
- 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, Miami, Florida, USA. Association for Computational Linguistics.
- Heyuan Huang, Alexandra DeLucia, Vijay Murari Tiyyala, and Mark Dredze. 2025. [Medscore: Factuality evaluation of free-form medical answers](#). *Preprint*, arXiv:2505.18452.
- Zhengping Jiang, Jingyu Zhang, Nathaniel Weir, Seth Ebner, Miriam Wanner, Kate Sanders, Daniel Khashabi, Anqi Liu, and Benjamin Van Durme. 2025. [Core: Robust factual precision with informative subclaim identification](#). In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 19833–19856, Vienna, Austria. Association for Computational Linguistics.
- Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. 2017. [triviaqa: A Large Scale Distantly Supervised Challenge Dataset for Reading Comprehension](#). *arXiv e-prints*, arXiv:1705.03551.
- Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Matthew Kelcey, Jacob Devlin, Kenton Lee, Kristina N. Toutanova, Llion Jones, Ming-Wei Chang, Andrew Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural questions: a benchmark for question answering research. *Transactions of the Association of Computational Linguistics*.
- 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](#).
- Jimmy Lin and Dina Demner-Fushman. 2006. [Will pyramids built of nuggets topple over?](#) In *Proceedings of the Human Language Technology Conference of the NAACL, Main Conference*, pages 383–390, New York City, USA. Association for Computational Linguistics.
- Xin Liu, Lechen Zhang, Sheza Munir, Yiyang Gu, and Lu Wang. 2025. [Verifact: Enhancing long-form factuality evaluation with refined fact extraction and reference facts](#). *Preprint*, arXiv:2505.09701.
- Laura Majer and Jan Šnajder. 2024. [Claim check-worthiness detection: How well do LLMs grasp annotation guidelines?](#) In *Proceedings of the Seventh Fact Extraction and VERification Workshop (FEVER)*, pages 245–263, Miami, Florida, USA. Association for Computational Linguistics.
- Sewon Min, Kalpesh Krishna, Xinxu Lyu, Mike Lewis, Wen-tau Yih, Pang 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*, pages 12076–12100, Singapore. Association for Computational Linguistics.
- Kushan Mitra, Dan Zhang, Sajjadur Rahman, and Es-
tevam Hruschka. 2025. [FactLens: Benchmarking fine-grained fact verification](#). In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 18085–18096, Vienna, Austria. Association for Computational Linguistics.
- Niels Mündler, Jingxuan He, Slobodan Jenko, and Martin Vechev. 2024. [Self-contradictory hallucinations of large language models: Evaluation, detection and mitigation](#). *Preprint*, arXiv:2305.15852.
- Ani Nenkova and Rebecca Passonneau. 2004. [Evaluating content selection in summarization: The pyramid method](#). In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics: HLT-NAACL 2004*, pages 145–152, Boston, Massachusetts, USA. Association for Computational Linguistics.
- OpenAI, Sandhini Agarwal, Lama Ahmad, Jason Ai, Sam Altman, Andy Applebaum, Edwin Arbus, Rahul K. Arora, Yu Bai, Bowen Baker, Haiming Bao, Boaz Barak, Ally Bennett, Tyler Bertao, Nivedita Brett, Eugene Brevdo, Greg Brockman, Sebastien Bubeck, Che Chang, and 107 others. 2025. [gpt-oss-120b & gpt-oss-20b model card](#). *Preprint*, arXiv:2508.10925.
- OpenAI, Aaron Hurst, Adam Lerer, Adam P. Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, Aleksander Mądry, Alex Baker-Whitcomb, Alex Beutel, Alex Borzunov, Alex Carney, Alex Chow, Alex Kirillov, Alex Nichol, and 400 others. 2024. [Gpt-4o system card](#). *Preprint*, arXiv:2410.21276.
- Ronak Pradeep, Nandan Thakur, Shivani Upadhyay, Daniel Campos, Nick Craswell, and Jimmy Lin. 2024. Initial nugget evaluation results for the TREC 2024 RAG Track with the AutoNuggetizer Framework. *arXiv:2411.09607*.
- Ronak Pradeep, Nandan Thakur, Shivani Upadhyay, Daniel Campos, Nick Craswell, and Jimmy Lin. 2025. [The great nugget recall: Automating fact extraction and rag evaluation with large language models](#). *Preprint*, arXiv:2504.15068.
- Chris Samarin, Alexander Krubner, Alireza Salemi, Youngwoo Kim, and Hamed Zamani. 2025. [Beyond factual accuracy: Evaluating coverage of diverse factual information in long-form text generation](#). *Preprint*, arXiv:2501.03545.
- Yixiao Song, Yekyung Kim, and Mohit Iyyer. 2024. [VeriScore: Evaluating the factuality of verifiable claims in long-form text generation](#). In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 9447–9474, Miami, Florida, USA. Association for Computational Linguistics.
- Hongjin Su, Howard Yen, Mengzhou Xia, Weijia Shi, Niklas Muennighoff, Han-yu Wang, Haisu Liu, Quan Shi, Zachary S Siegel, Michael Tang, Ruoxi Sun, Jinsung Yoon, Serkan O Arik, Danqi Chen, and Tao Yu. 2024. [Bright: A realistic and challenging benchmark for reasoning-intensive retrieval](#).
- Qwen Team. 2025. [Qwen3 technical report](#). *Preprint*, arXiv:2505.09388.
- Ellen M. Voorhees. 2003. [Overview of the TREC 2003 question answering track](#). In *Proceedings of The Twelfth Text REtrieval Conference, TREC 2003, Gaithersburg, Maryland, USA, November 18-21, 2003*, volume 500-255 of *NIST Special Publication*, pages 54–68. National Institute of Standards and Technology (NIST).
- Yuxia Wang, Revanth Gangi Reddy, Zain Muhammad Mujahid, Arnav Arora, Aleksandr Rubashevskii, Jiahui Geng, Osama Mohammed Afzal, Liangming Pan, Nadav Borenstein, Aditya Pillai, Isabelle Augenstein, Iryna Gurevych, and Preslav Nakov. 2024. [Factcheck-bench: Fine-grained evaluation benchmark for automatic fact-checkers](#). In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 14199–14230, Miami, Florida, USA. Association for Computational Linguistics.
- Miriam Wanner, Benjamin Van Durme, and Mark Dredze. 2024a. [Dndscore: Decontextualization and decomposition for factuality verification in long-form text generation](#). *Preprint*, arXiv:2412.13175.
- Miriam Wanner, Seth Ebner, Zhengping Jiang, Mark Dredze, and Benjamin Van Durme. 2024b. [A closer look at claim decomposition](#). In *Proceedings of the 13th Joint Conference on Lexical and Computational Semantics (*SEM 2024)*, pages 153–175, Mexico City, Mexico. Association for Computational Linguistics.
- Jerry Wei, Chengrun Yang, Xinying Song, Yifeng Lu, Nathan Hu, Jie Huang, Dustin Tran, Daiyi Peng, Ruibo Liu, Da Huang, Cosmo Du, and Quoc V. Le. 2024. [Long-form factuality in large language models](#). In *Advances in Neural Information Processing Systems*, volume 37, pages 80756–80827. Curran Associates, Inc.
- Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*.
- James Xu Zhao, Jimmy Z.j. Liu, Bryan Hooi, and See-Kiong Ng. 2025. [How does response length affect long-form factuality](#). In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 3102–3125, Vienna, Austria. Association for Computational Linguistics.

Wenting Zhao, Tanya Goyal, Yu Ying Chiu, Liwei Jiang, Benjamin Newman, Abhilasha Ravichander, Khyathi Chandu, Ronan Le Bras, Claire Cardie, Yuntian Deng, and Yejin Choi. 2024a. [Wildhallucinations: Evaluating long-form factuality in llms with real-world entity queries](#). *Preprint*, arXiv:2407.17468.

Xiutian Zhao, Ke Wang, and Wei Peng. 2024b. [Measuring the inconsistency of large language models in preferential ranking](#). In *Proceedings of the 1st Workshop on Towards Knowledgeable Language Models (KnowLLM 2024)*, pages 171–176, Bangkok, Thailand. Association for Computational Linguistics.

A Prompts, Model Details, Compute

The prompts used for generation of normal responses are shown in Table 6, and adversarial responses are shown in Table 7. For normal response generation, we prepend an instruction to answer in a paragraph for each of the queries, to ensure the text is in long-form. For missing and wrong response generation, we provide the LLM with the query and normal LLM response, and instruct the LLM to adversarially omit or falsify key information in the response. For both normal and adversarial response generation, we use gpt-4o (OpenAI et al., 2024), with a temperature of 0.2 and 2000 max tokens.

The prompt for importance ranking are shown in Table 8, with an example ranking in Table 9. The prompt provides an LLM with a query and set of subclaims, asking the model to order subclaims based on importance for answering the query, independent of correctness, and label each subclaim with either *vital*, *okay*, or *less important* importance labels. For the importance ranking, we use gpt-4o with a temperature of 0.2 and 4000 max tokens. In total, generating responses took 15 GPU-hours, evaluating with FACTSCORE/nuggets/VITAL took 450 GPU-hours.

B Results

B.1 Comparison to SAFE

We compare our results to another set of evaluation metrics. We use SAFE from Wei et al. (2024). Here, precision is calculated as in FACTSCORE. Recall, however, is calculated as the number of supported subclaims over a desired number of supported facts. In Wei et al. (2024), K is calculated as the median number of subclaims along all model responses within a particular subset for a particular dataset. While this may be an approximation of a gold set of subclaims, they do not attempt to build or verify that this median-approach is faithful to some reference set of subclaims. Figure 4 contains SAFE precision, recall (@K), and F1 (@K), as well as K, the median number of subclaims for the subset of the data. We observe little difference in scores between normal, missing, and wrong responses, suggesting their metric is not able to reflect errors in key information unlike VITAL.

B.2 Importance Ranking with Additional Models and Human Annotation

Additional Models. We compare gpt-4o (OpenAI et al., 2024) with three additional models: Llama-3.3-70B-Instruct (Grattafiori et al., 2024), Qwen3-235B-A22B-Instruct (Team, 2025), and GPT-OSS-120B (OpenAI et al., 2025). gpt-4o is a proprietary model, while the remaining three are all open-weight models. We evaluate across varying number of weights (including 70B, 235B with 22B active, and 120B with 5.1B active), and variety of companies who created these models (Meta, Alibaba, OpenAI). We also evaluate both dense (Llama-3.3-70B-Instruct) and MoE (Qwen3-235B-A22B-Instruct, GPT-OSS-120B) models.

Human Annotation. We use a browser-based annotation tool for labeling and ordering subclaim importance. Annotators are provided one instance at a time. For each instance, they are given the query, the full model response, and a list of subclaims in the original response order. The annotator must then assign a label (either vital, okay or less-important) to each subclaim, and rank the subclaims by dragging up and down to establish an importance ordering. The annotators are all authors of this paper.

Data. We choose a random 150 instances across normal, missing, and wrong splits of the NQ dataset split. We evaluate both on the models and human annotation described above.

Metrics. We report three metrics for each pair of evaluations/annotation: Cohen’s κ , exact agreement percentage, and Kendall’s τ_b . The first two evaluate agreement of importance labeling, while the third evaluates the ranking agreement. Cohen’s κ measures the consistency of categorical agreement between two systems. Results are reported as pairwise per split (normal, missing, and wrong). Exact agreement percentage evaluates the percentage of subclaims where both systems assign the same label, reported as a pairwise matrix per split. For a given pair of systems, Kendall’s τ_b measures the strength and direction of association between two ordinal (ranked) variables. This measures how well the two systems agree on the relative ordering of subclaims, independent of label assignment.

Results. The full results can be found in Table 10. We find that pairwise agreement is moderate across all model pairs, with Cohen’s κ generally

Data Type	Dataset	Subsets	Size	Description
Open-Ended	FACTSCORE Bios	-	500	Generated people biographies and corresponding Wiki page
	WILDHAL.	Cult. & Ent., Geographic	2484	Entities from user conversations and curated web page knowledge source
	BRIGHT	Bio, E.S., Econ, Psych, Rob., S.O., S.L.	749	Real-world queries from StackExchange and linked sources
Single-Answer	HOTPOTQA	-	1000 (subset)	Multi-hop reasoning questions and supporting documents
	NQ	-	1000 (subset)	Real Google search queries and results
	TRIVIAQA	-	1000 (subset)	Trivia questions and reference documents

Table 5: Overview of datasets used to construct VITALERRORS, covering open-ended and single-answer queries across six sources, totaling 6,733 instances.

Data Type	Dataset	Prompt
Open-Ended	FACTSCORE Bios	In a paragraph, tell me a bio of [TOPIC]
	WILDHAL.	In a paragraph, could you tell me what you know about [TOPIC]?
	BRIGHT	Answer the following question in a short paragraph: [QUERY]
Single-Answer	HOTPOTQA	In a paragraph, could you answer: [QUERY]
	NQ	In a paragraph, could you answer: [QUERY]
	TRIVIAQA	In a paragraph, could you answer: [QUERY]

Table 6: Prompts used to generate normal long-form LLM responses across datasets, with variations for open-ended and single-answer queries.

in the 0.48–0.63 range, exact agreement about 60–70%, and Kendall’s τ_b around 0.57–0.70. This indicates a significant, although not perfect consensus. Agreement is highest for the wrong data across metrics. gpt-4o and Qwen3-235B-A22B-Instruct show the strongest pairwise agreement, suggesting similar evaluation behavior. GPT-OSS-120B tends to produce slightly lower agreement scores than other model pairings. Kendall’s τ_b are consistently higher, implying model’s rank-order agreement remains stable, even when exact label agreement varies.

Human agreement with models is also moderate, although lower than within-model agreement. For the normal data split, human agreement with models remains moderate, indicating more alignment under standard settings. The missing split shows the largest divergence, with a drop in all metrics, indicating mismatch in agreement when key information is omitted. We suspect that marking items as vital becomes more difficult when there is no central subclaim answering the query. Kendall’s τ_b does not drop as much, suggesting

agreement in exact labels drops, but overall preference of order remains similar. Human evaluation agreement with models for normal and missing splits remain similar. We find human evaluation agrees the most strongly with gpt-4o for the normal split, Llama-3.3-70B-Instruct for the wrong split, and both gpt-4o and Llama-3.3-70B-Instruct for the missing split. These results suggest scenarios where there is missing key information is the setting where humans and LLMs disagree on their labeling the most. We find this importance ranking and labeling to be a difficult task for humans, with more agreement between LLMs, and for this reason find this task to be best left for model evaluation.

B.3 Statistical Significance Testing

We compute significance testing on all VITAL comparisons. We calculate paired t-test and Wilcoxon signed-rank for continuous metrics, and Cohen’s d_z , rank-biserial r, and McNemar’s test for the binary error metrics. Results can be found in Tables 11–16. All key comparisons (normal vs. wrong, missing vs. wrong, normal vs. missing) are sig-

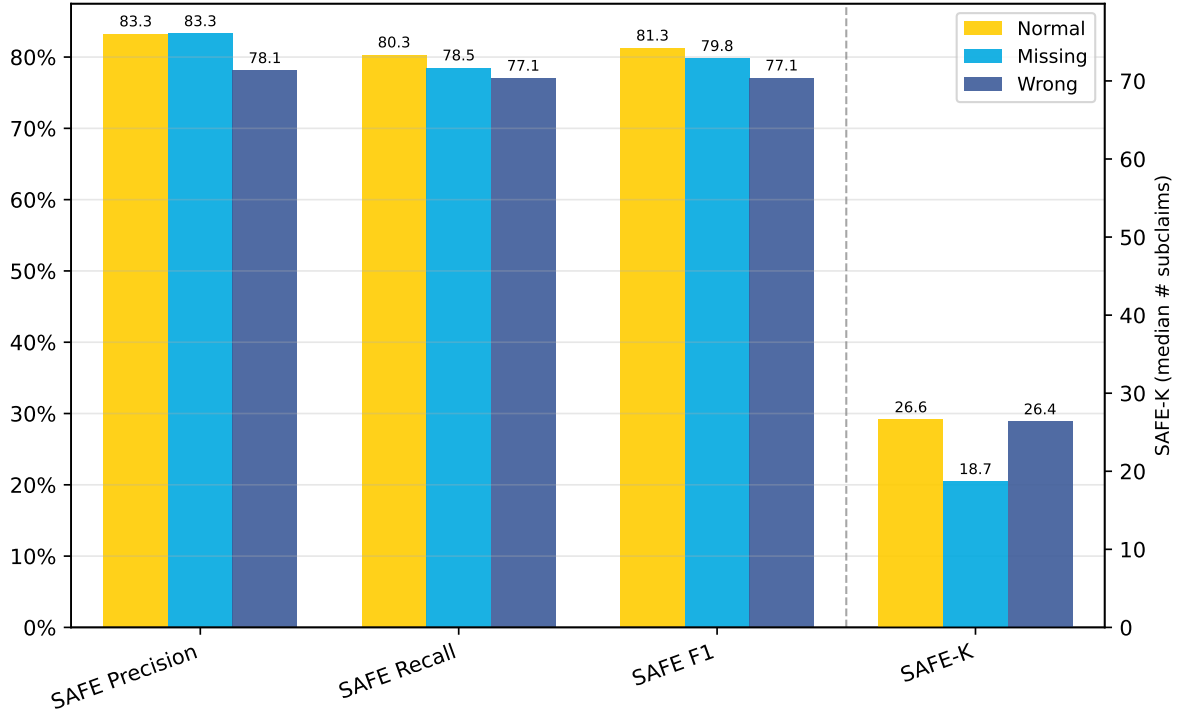


Figure 4: SAFE precision, recall, and F1 scores are shown on the left, and the SAFE K used to calculate SAFE recall. In all variations of SAFE scores, there is little difference in normal, missing, and wrong evaluations.

nificant at $p < 0.001$ with medium-to-large effect sizes. For example, single-answer $VITAL_{REC}$ drops from 52.4% to 36.7% normal vs. wrong with $d_z=0.51$. Similarly, $VITAL_{PREC}$ drops from 72.8% to 48.7% with $d_z=0.79$. The $VITAL_{RLP}$ rate rises from 45.0% to 89.5% with McNemar $p < .001$. Normal vs. missing comparisons are smaller, but still significant, consistent with our finding that omissions are harder to detect than incorrect key information.

B.4 Cumulative Precision

The cumulative FACTSCORE precision for open-ended queries is shown in Figure 5. This demonstrates how key information errors get hidden in long responses. Similarly to single-answer queries (Figure 3), we observe responses with wrong key information receive low initial scores, which increases with response length.

B.5 Linear Decay Weighting

Figure 6 reports the precision and recall with a linear decay weighting. This weighting assumes that each subclaim/nugget ranked higher than another is strictly more important. This is an alternative version of the VITAL metrics, however the assumption that each subclaim/nugget is strictly

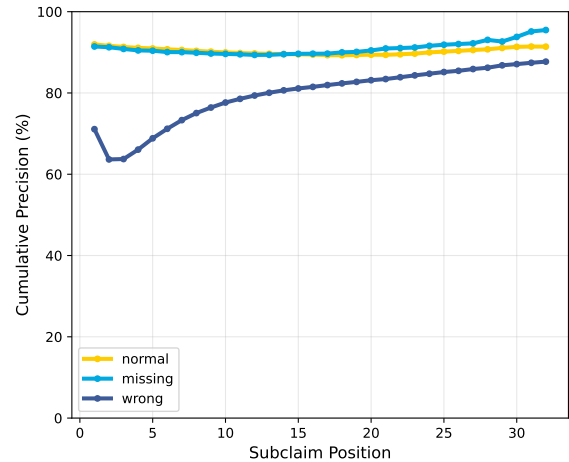


Figure 5: Cumulative precision over subclaim position for open-ended queries. Wrong responses show low precision early on due to falsified key claims, while normal and missing responses stay consistently high.

more important than the ones ranked below is a strong assumption that is not usually true. For this reason, we do not include it in VITAL metrics, but include it here to demonstrate performance under these assumptions.

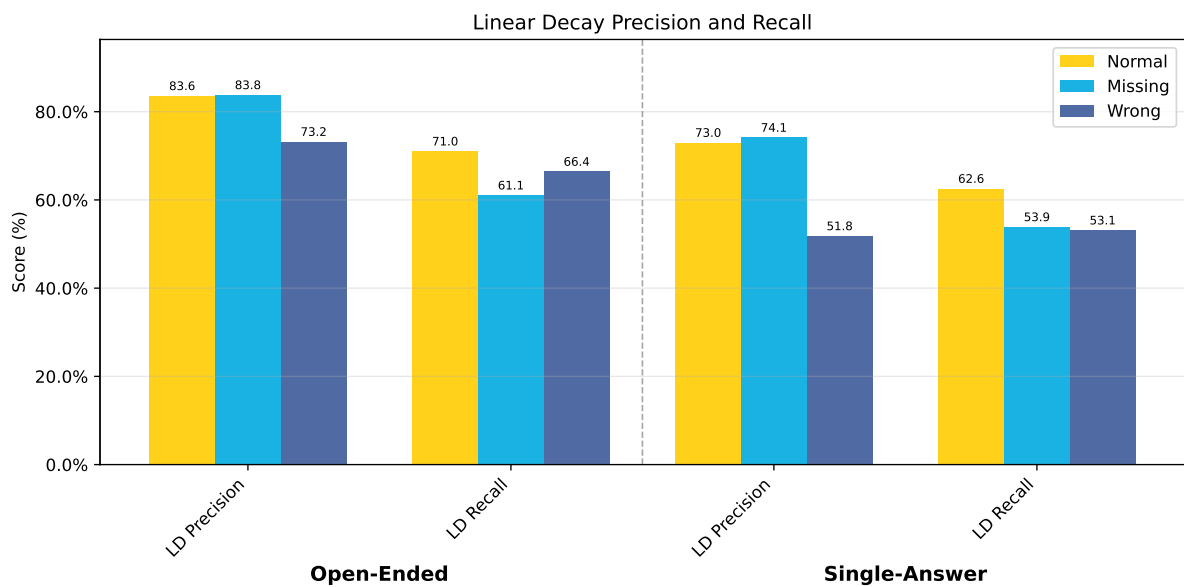


Figure 6: Linear decay weighted precision and recall. Highlights the differences between normal, missing, and wrong responses, though less strongly than response-level metrics.

Adversary Type	Prompt
Missing	You are given a query and an answer. Your task is to modify the answer by removing the most important information for answering the question. You are first given an example, and then your task.
	EXAMPLE Query: Which is denser, water vapor or air? Answer: Air is denser than water vapor. This is because the molecular weight of the primary components of air mainly nitrogen and oxygen is higher than that of water vapor. Nitrogen has a molecular weight of approximately 28 g/mol, and oxygen about 32 g/mol, whereas water vapor has a molecular weight of only about 18 g/mol. When water vapor is added to air, it actually displaces some of the heavier nitrogen and oxygen molecules, making the resulting humid air less dense than dry air. This principle is why humid air tends to rise and is a key factor in weather patterns and cloud formation. At the same temperature and pressure, a volume of moist air will weigh less than the same volume of dry air, confirming that water vapor is less dense than the air it mixes with. Modified answer: This is because the molecular weight of the primary components of air mainly nitrogen and oxygen is higher than that of water vapor. Nitrogen has a molecular weight of approximately 28 g/mol, and oxygen about 32 g/mol, whereas water vapor has a molecular weight of only about 18 g/mol. When water vapor is added to air, it actually displaces some of the heavier nitrogen and oxygen molecules, making the resulting humid air less dense than dry air. This principle is why humid air tends to rise and is a key factor in weather patterns and cloud formation. At the same temperature and pressure, a volume of moist air will weigh less than the same volume of dry air, confirming that water vapor is less dense than the air it mixes with.
	YOUR TASK Query: [QUERY] Answer: [ANSWER] Modified answer:
Wrong	You are given a query and its corresponding answer. Your task is to make a modification to one sentence from the answer by changing the key piece of information required to answer the question correctly, thereby making the answer factually incorrect. This will most likely be a change to the first sentence. Do not alter the remainder of the response. An example is provided first, followed by your task.
	EXAMPLE Query: Which is denser, water vapor or air? Answer: Air is denser than water vapor. This is because the molecular weight of the primary components of air mainly nitrogen and oxygen is higher than that of water vapor. Nitrogen has a molecular weight of approximately 28 g/mol, and oxygen about 32 g/mol, whereas water vapor has a molecular weight of only about 18 g/mol. When water vapor is added to air, it actually displaces some of the heavier nitrogen and oxygen molecules, making the resulting humid air less dense than dry air. This principle is why humid air tends to rise and is a key factor in weather patterns and cloud formation. At the same temperature and pressure, a volume of moist air will weigh less than the same volume of dry air, confirming that water vapor is less dense than the air it mixes with. Modified answer: Water vapor is denser than air. This is because the molecular weight of the primary components of air mainly nitrogen and oxygen is higher than that of water vapor. Nitrogen has a molecular weight of approximately 28 g/mol, and oxygen about 32 g/mol, whereas water vapor has a molecular weight of only about 18 g/mol. When water vapor is added to air, it actually displaces some of the heavier nitrogen and oxygen molecules, making the resulting humid air less dense than dry air. This principle is why humid air tends to rise and is a key factor in weather patterns and cloud formation. At the same temperature and pressure, a volume of moist air will weigh less than the same volume of dry air, confirming that water vapor is less dense than the air it mixes with.
	YOUR TASK Query: [QUERY] Answer: [ANSWER] Modified answer:

Table 7: Prompts for adversarial data generation. “Missing” prompts remove key information, while “wrong” prompts replace with incorrect information.

You are performing step two of a four part fact-checking process: (1) Decompose a paragraph into individual claims. (2) Given a query and set of claims, rank by decreasing query-importance (this step). (3) Check the correctness of each claim. (4) Score the paragraph, weighting by importance. This step is completely independent of factual correctness, and only focuses on the query-importance of claims for answering the query. Even factually incorrect claims should be ranked highly if they directly answer the query. Instructions: You are provided with a query and set of claims. Rank the claims in decreasing order of query-importance. A claim exhibits high query-importance when it addresses a central aspect of the query, and low query-importance when it contributes only peripheral or background information. Rank claims independent of correctness, instead only based on query-importance. A later step will check for correctness of claims. Assign query-importance labels using exactly these three categories: - "vital" - Essential claims that directly address the core query - "okay" - Supporting claims that provide useful but non-essential information - "less-important" - Background or tangentially related claims with minimal relevance Ordering Rules: - All "vital" claims must appear first, then all "okay" claims come second, and "less-important" claims come last. - Within each category, order by decreasing importance. - If two or more claims address the same aspect of the query, keep them grouped in the order they appear, even if their answers contradict. For example: <pre> ... [S3] Washington, D.C. is the capital of Canada.: "vital" [S8] Washington, D.C. is the capital of the United States.: "vital" ... </pre> - Do not adjust rankings based on factual correctness, this will be handled in step 3. Output Format: <pre> [Claim ID] <claim text>: "label" [Claim ID] <claim text>: "label" ... </pre> Requirements: - Label every claim exactly once - Use only the three specified labels - Maintain the original claim count - Return only the labeled, ordered list (no explanations) Below is your task. ###Your task: Query: [QUERY] Claims: [SUBCLAIMS] Ranked Claims:

Table 8: Prompt for ranking subclaims by *query importance* and importance labeling. The LLM assigns each subclaim with a “vital”, “okay”, or “less-important” label independent of correctness.

Query		In a paragraph, could you answer: who wrote i want to dance with somebody by whitney houston	
Type	Index	Importance	Response
Normal	2	vital	"I Wanna Dance with Somebody (Who Loves Me)" was written by George Merrill.
	3	vital	"I Wanna Dance with Somebody (Who Loves Me)" was written by Shannon Rubicam.
	7	okay	The duo wrote (penned) the song specifically for Whitney Houston.
	4	okay	George Merrill is a songwriter.
	5	okay	Shannon Rubicam is a songwriter.
	6	okay	The duo performed together as the pop group Boy Meets Girl.
	1	less-important	"I Wanna Dance with Somebody (Who Loves Me)" is a song.
	8	less-important	Boy Meets Girl is a pop group.
	9	less-important	Whitney Houston is a singer (artist).
	10	less-important	The track was released in 1987.
	11	less-important	The track was the lead single from her second studio album.
	12	less-important	Her second studio album is titled "Whitney.".
	13	less-important	The track is from the album "Whitney.".
	14	less-important	The track became one of Houston's most iconic hits.
	15	less-important	The track was released as the lead single from "Whitney" in 1987.
	16	less-important	The song has an upbeat tempo.
	17	less-important	The song has an infectious melody.
	18	less-important	The song features Houston's powerful vocals.
	19	less-important	The combination of the song's upbeat tempo, infectious melody, and Houston's powerful vocals contributed to its success.
	20	less-important	The song reached the top of the charts worldwide.
	21	less-important	The success of the song helped solidify Houston's status as a global pop superstar.
	22	less-important	Houston is a global pop superstar.
Missing	1	okay	The track was released in 1987.
	2	okay	The track was the lead single from her second studio album.
	4	okay	The track is from the album "Whitney.".
	3	okay	Her second studio album is titled "Whitney.".
	5	okay	The track became one of Houston's most iconic hits.
	11	okay	The song reached the top of the charts worldwide.
	12	okay	The song helped solidify Houston's status as a global pop superstar.
	6	less-important	Houston has multiple iconic hits.
	7	less-important	The song has an upbeat tempo.
	8	less-important	The song has an infectious melody.
	9	less-important	The song features Houston's powerful vocals.
	10	less-important	The upbeat tempo, infectious melody, and Houston's powerful vocals helped propel the song to the top of the charts worldwide.
	13	less-important	Houston is a global pop superstar.
Wrong	2	vital	"I Wanna Dance with Somebody (Who Loves Me)" was written by Diane Warren.
	3	vital	"I Wanna Dance with Somebody (Who Loves Me)" was written by David Foster.
	8	vital	The duo penned the song.
	9	vital	The duo penned the song specifically for Whitney Houston.
	4	okay	Diane Warren is a songwriter.
	5	okay	David Foster is a songwriter.
	6	okay	The duo performed together.
	7	okay	The duo performed together as the pop group Boy Meets Girl.
	10	less-important	Boy Meets Girl is a pop group.
	1	less-important	"I Wanna Dance with Somebody (Who Loves Me)" is a song.
	11	less-important	Whitney Houston is a singer.
	12	less-important	The track was released in 1987.
	13	less-important	The track was the lead single from her second studio album.
	14	less-important	Her second studio album is titled "Whitney.".
	15	less-important	The track was released as the lead single from the album "Whitney.".
	16	less-important	The track became one of Houston's most iconic hits.
	17	less-important	Houston had a second studio album titled "Whitney.".
	18	less-important	The song has an upbeat tempo.
	19	less-important	The song has an infectious melody.
	20	less-important	The song features Houston's powerful vocals.
	21	less-important	The combination of the song's upbeat tempo, infectious melody, and Houston's powerful vocals contributed to its success.
	22	less-important	The song reached the top of the charts worldwide.
	23	less-important	The song's success helped solidify Houston's status as a global pop superstar.

Table 9: Example subclaim rankings for a query from NQ. Normal responses highlight correct vital subclaims, while wrong responses show incorrect, but still vital subclaims.

Normal					
Metric	Pair	gpt-4o	llama-3.3-70b	qwen3-235b	gpt-oss-120b
Cohen's κ	llama-3.3-70b	0.541	–	–	–
	qwen3-235b	0.573	0.527	–	–
	gpt-oss-120b	0.517	0.484	0.509	–
	human-eval	0.443	0.404	0.411	0.416
Exact agreement (%)	llama-3.3-70b	64.1	–	–	–
	qwen3-235b	66.8	64.6	–	–
	gpt-oss-120b	61.9	59.6	61.7	–
	human-eval	58.5	54.6	56.3	56.6
Kendall τ_b	llama-3.3-70b	0.625	–	–	–
	qwen3-235b	0.639	0.641	–	–
	gpt-oss-120b	0.604	0.681	0.642	–
	human-eval	0.569	0.556	0.554	0.556
Missing					
Metric	Pair	gpt-4o	llama-3.3-70b	qwen3-235b	gpt-oss-120b
Cohen's κ	llama-3.3-70b	0.542	–	–	–
	qwen3-235b	0.576	0.582	–	–
	gpt-oss-120b	0.545	0.484	0.512	–
	human-eval	0.259	0.257	0.253	0.213
Exact agreement (%)	llama-3.3-70b	63.9	–	–	–
	qwen3-235b	67.6	67.9	–	–
	gpt-oss-120b	64.1	61.6	64.6	–
	human-eval	47.8	45.9	44.8	43.1
Kendall τ_b	llama-3.3-70b	0.571	–	–	–
	qwen3-235b	0.600	0.701	–	–
	gpt-oss-120b	0.611	0.584	0.607	–
	human-eval	0.398	0.497	0.528	0.455
Wrong					
Metric	Pair	gpt-4o	llama-3.3-70b	qwen3-235b	gpt-oss-120b
Cohen's κ	llama-3.3-70b	0.592	–	–	–
	qwen3-235b	0.632	0.575	–	–
	gpt-oss-120b	0.528	0.509	0.542	–
	human-eval	0.436	0.469	0.378	0.360
Exact agreement (%)	llama-3.3-70b	68.1	–	–	–
	qwen3-235b	71.6	67.1	–	–
	gpt-oss-120b	62.5	62.4	65.6	–
	human-eval	57.6	60.2	52.3	50.6
Kendall τ_b	llama-3.3-70b	0.627	–	–	–
	qwen3-235b	0.659	0.658	–	–
	gpt-oss-120b	0.615	0.622	0.631	–
	human-eval	0.539	0.561	0.536	0.536

Table 10: Pairwise agreement across conditions. Values shown are Cohen's κ , exact label agreement (%), and Kendall's τ_b for full ordering.

Metric	Comparison	n	Mean A	Mean B	Mean diff	Cohen's d_z	t p (Holm)	Wilcoxon p (Holm)	Rank-biserial r
FACTSCORE (precision)	normal vs missing	3661	0.8359	0.8356	+0.0018	+0.027	3.08e-01	4.44e-02	+0.058
	normal vs wrong	3661	0.8359	0.7879	+0.0428	+0.773	0.00e+00	0.00e+00	+0.824
	missing vs wrong	3661	0.8356	0.7879	+0.0410	+0.550	2.91e-211	2.71e-275	+0.686
VITAL _{PREC.}	normal vs missing	3726	0.8269	0.8288	-0.0009	-0.007	1.00e+00	1.00e+00	-0.019
	normal vs wrong	3726	0.8269	0.7540	+0.0531	+0.381	3.15e-111	3.73e-142	+0.597
	missing vs wrong	3726	0.8288	0.7540	+0.0540	+0.333	2.62e-86	5.54e-115	+0.533
NUGGETRECALL	normal vs missing	3726	0.2435	0.1861	+0.0537	+0.795	0.00e+00	0.00e+00	+0.922
	normal vs wrong	3726	0.2435	0.2323	+0.0125	+0.318	1.57e-79	1.50e-84	+0.529
	missing vs wrong	3726	0.1861	0.2323	-0.0412	-0.591	7.28e-244	1.62e-262	-0.799
VITAL _{REC.}	normal vs missing	3726	0.4090	0.2991	+0.1169	+0.878	0.00e+00	0.00e+00	+0.941
	normal vs wrong	3726	0.4090	0.3789	+0.0350	+0.397	1.23e-119	5.54e-115	+0.663
	missing vs wrong	3726	0.2991	0.3789	-0.0820	-0.589	6.23e-242	1.98e-222	-0.762

Table 11: Open-Ended: Continuous metrics (paired t-test / Wilcoxon signed-rank)

Metric	Comparison	n	Rate A	Rate B	Risk diff	Discordant (1,0)/(0,1)	McNemar p (Holm)
VITAL _{RLP}	normal vs missing	3726	0.5270	0.4548	+0.0722	493/270	3.68e-15
	normal vs wrong	3726	0.5270	0.7379	-0.2109	165/1097	1.10e-150
	missing vs wrong	3726	0.4548	0.7379	-0.2831	170/1325	5.95e-195
VITAL _{RLR}	normal vs missing	3726	0.8722	0.9118	-0.0396	21/111	2.83e-14
	normal vs wrong	3726	0.8722	0.8858	-0.0136	24/60	1.34e-04
	missing vs wrong	3726	0.9118	0.8858	+0.0260	93/39	7.94e-06

Table 12: Open-Ended: Binary error metrics (McNemar's test)

Metric	Comparison	n	Mean A	Mean B	Mean diff	Cohen's d_z	t p (Holm)	Wilcoxon p (Holm)	Rank-biserial r
FACTSCORE (precision)	normal vs missing	2717	0.8247	0.8253	-0.0001	-0.002	9.37e-01	1.00e+00	-0.016
	normal vs wrong	2717	0.8247	0.7657	+0.0590	+0.749	5.59e-264	6.55e-239	+0.755
	missing vs wrong	2717	0.8253	0.7657	+0.0592	+0.565	1.32e-164	8.32e-182	+0.648
VITAL _{PREC.}	normal vs missing	3000	0.7281	0.7222	+0.0059	+0.020	5.56e-01	1.00e+00	+0.006
	normal vs wrong	3000	0.7281	0.4873	+0.2408	+0.786	7.89e-315	2.78e-263	+0.849
	missing vs wrong	3000	0.7222	0.4873	+0.2349	+0.697	6.05e-259	9.84e-222	+0.769
NUGGETRECALL	normal vs missing	3000	0.2771	0.1913	+0.0858	+0.537	3.91e-166	3.99e-210	+0.880
	normal vs wrong	3000	0.2771	0.2349	+0.0421	+0.352	2.83e-77	3.31e-100	+0.685
	missing vs wrong	3000	0.1913	0.2349	-0.0436	-0.303	1.23e-58	6.43e-74	-0.551
VITAL _{REC.}	normal vs missing	3000	0.5244	0.2745	+0.2499	+0.711	9.49e-268	1.40e-220	+0.942
	normal vs wrong	3000	0.5244	0.3670	+0.1574	+0.505	2.02e-149	1.14e-137	+0.877
	missing vs wrong	3000	0.2745	0.3670	-0.0925	-0.284	5.99e-52	1.68e-51	-0.518

Table 13: Single-Answer: Continuous metrics (paired t-test / Wilcoxon signed-rank)

Metric	Comparison	n	Rate A	Rate B	Risk diff	Discordant (1,0)/(0,1)	McNemar p (Holm)
VITAL _{RLP}	normal vs missing	3000	0.4503	0.4217	+0.0287	388/302	2.43e-03
	normal vs wrong	3000	0.4503	0.8953	-0.4450	59/1394	1.28e-267
	missing vs wrong	3000	0.4217	0.8953	-0.4737	57/1478	7.46e-287
VITAL _{RLR}	normal vs missing	3000	0.5430	0.6810	-0.1380	54/468	1.46e-72
	normal vs wrong	3000	0.5430	0.6843	-0.1413	42/466	5.56e-78
	missing vs wrong	3000	0.6810	0.6843	-0.0033	258/268	6.95e-01

Table 14: Single-Answer: Binary error metrics (McNemar's test)

Metric	Comparison	n	Mean A	Mean B	Mean diff	Cohen's d_z	t p (Holm)	Wilcoxon p (Holm)	Rank-biserial r
FACTSCORE (precision)	normal vs missing	6378	0.8324	0.8324	+0.0010	+0.012	6.80e-01	8.88e-01	+0.018
	normal vs wrong	6378	0.8324	0.7808	+0.0497	+0.744	0.00e+00	0.00e+00	+0.783
	missing vs wrong	6378	0.8324	0.7808	+0.0487	+0.547	0.00e+00	0.00e+00	+0.660
VITAL _{PREC.}	normal vs missing	6726	0.7957	0.7952	+0.0021	+0.010	6.80e-01	8.88e-01	-0.012
	normal vs wrong	6726	0.7957	0.6698	+0.1368	+0.553	0.00e+00	0.00e+00	+0.741
	missing vs wrong	6726	0.7952	0.6698	+0.1347	+0.497	0.00e+00	0.00e+00	+0.673
NUGGETRECALL	normal vs missing	6726	0.2541	0.1878	+0.0680	+0.571	0.00e+00	0.00e+00	+0.898
	normal vs wrong	6726	0.2541	0.2332	+0.0257	+0.298	1.44e-125	6.87e-184	+0.601
	missing vs wrong	6726	0.1878	0.2332	-0.0423	-0.387	1.17e-205	2.55e-299	-0.678
VITAL _{REC.}	normal vs missing	6726	0.4454	0.2913	+0.1762	+0.670	0.00e+00	0.00e+00	+0.933
	normal vs wrong	6726	0.4454	0.3752	+0.0896	+0.396	7.53e-214	7.56e-263	+0.776
	missing vs wrong	6726	0.2913	0.3752	-0.0867	-0.360	1.91e-179	1.68e-236	-0.644

Table 15: All: Continuous metrics (paired t-test / Wilcoxon signed-rank)

Metric	Comparison	n	Rate A	Rate B	Risk diff	Discordant (1,0)/(0,1)	McNemar p (Holm)
VITAL _{RLP}	normal vs missing	6726	0.5028	0.4443	+0.0585	881/572	1.29e-15
	normal vs wrong	6726	0.5028	0.7876	-0.2848	224/2491	0.00e+00
	missing vs wrong	6726	0.4443	0.7876	-0.3433	227/2803	0.00e+00
VITAL _{RLR}	normal vs missing	6726	0.7682	0.8389	-0.0706	75/579	1.59e-85
	normal vs wrong	6726	0.7682	0.8221	-0.0539	66/526	6.67e-79
	missing vs wrong	6726	0.8389	0.8221	+0.0167	351/307	9.37e-02

Table 16: All: Binary error metrics (McNemar's test)