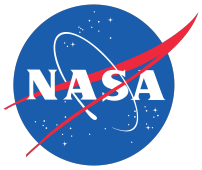
NASA/TM–20260000076



VALOR: Validation for Aerospace LLM Output and Reasoning

Kayshav S. Prakash
Logyx LLC, Mountain View, CA

Stefan Schuet
Ames Research Center, Moffett Field, California

Kevin Wheeler
Ames Research Center, Moffett Field, California

Kalmanje Krishnakumar
Ames Research Center, Moffett Field, California

Sebastian Gutierrez-Nolasco
Universities Space Research Association, Mountain View, California

Sandeep Shetye
Ames Research Center, Moffett Field, California

January 2026

NASA STI Program . . . in Profile

Since its founding, NASA has been dedicated to the advancement of aeronautics and space science. The NASA scientific and technical information (STI) program plays a key part in helping NASA maintain this important role.

The NASA STI Program operates under the auspices of the Agency Chief Information Officer. It collects, organizes, provides for archiving, and disseminates NASA's STI. The NASA STI Program provides access to the NASA Aeronautics and Space Database and its public interface, the NASA Technical Report Server, thus providing one of the largest collections of aeronautical and space science STI in the world. Results are published in both non-NASA channels and by NASA in the NASA STI Report Series, which includes the following report types:

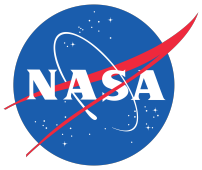
- TECHNICAL PUBLICATION. Reports of completed research or a major significant phase of research that present the results of NASA programs and include extensive data or theoretical analysis. Includes compilations of significant scientific and technical data and information deemed to be of continuing reference value. NASA counterpart of peer-reviewed formal professional papers, but having less stringent limitations on manuscript length and extent of graphic presentations.
- TECHNICAL MEMORANDUM. Scientific and technical findings that are preliminary or of specialized interest, e.g., quick release reports, working papers, and bibliographies that contain minimal annotation. Does not contain extensive analysis.
- CONTRACTOR REPORT. Scientific and technical findings by NASA-sponsored contractors and grantees.
- CONFERENCE PUBLICATION. Collected papers from scientific and technical conferences, symposia, seminars, or other meetings sponsored or co-sponsored by NASA.
- SPECIAL PUBLICATION. Scientific, technical, or historical information from NASA programs, projects, and missions, often concerned with subjects having substantial public interest.
- TECHNICAL TRANSLATION. English-language translations of foreign scientific and technical material pertinent to NASA's mission.

Specialized services also include creating custom thesauri, building customized databases, and organizing and publishing research results.

For more information about the NASA STI Program, see the following:

- Access the NASA STI program home page at *http://www.sti.nasa.gov*
- E-mail your question to *help@sti.nasa.gov*
- Fax your question to the NASA STI Information Desk at 443-757-5803
- Phone the NASA STI Information Desk at 443-757-5802
- Write to:
  STI Information Desk
  NASA Center for AeroSpace Information
  7115 Standard Drive
  Hanover, MD 21076–1320

NASA/TM–20260000076



VALOR: Validation for Aerospace LLM Output and Reasoning

Kayshav S. Prakash
Logyx LLC, Mountain View, CA

Stefan Schuet
Ames Research Center, Moffett Field, California

Kevin Wheeler
Ames Research Center, Moffett Field, California

Kalmanje Krishnakumar
Ames Research Center, Moffett Field, California

Sebastian Gutierrez-Nolasco
Universities Space Research Association, Mountain View, California

Sandeep Shetye
Ames Research Center, Moffett Field, California

National Aeronautics and
Space Administration

Ames Research Center
Mountain View, CA 94035

January 2026

Available from:

NASA Center for AeroSpace Information
7115 Standard Drive
Hanover, MD 21076-1320

National Technical Information Service
5301 Shawnee Road
Alexandria, VA 22312

Available electronically at http://www.sti.nasa.gov

VALOR: Validation for Aerospace LLM Output and Reasoning

Kayshav Prakash
Logyx LLC
Ames Research Center
Moffett Field, CA 94035

Stefan Schuet, Kevin Wheeler, Kalmanje Krishnakumar, Sandeep Shetye
National Aeronautics and Space Administration
Ames Research Center
Moffett Field, CA 94035

Sebastian Gutierrez-Nolasco
Universities Space Research Association
Ames Research Center
Moffett Field, CA 94035

Summary

Large Language Models (LLMs) offer new opportunities for automating and supporting a wide range of applications. However, despite rapid progress, current LLMs remain prone to factual errors, unstable reasoning, and inconsistent use of context — limitations that are especially consequential in safety-critical domains. These challenges highlight the need for independent, domain-aware validation frameworks rather than reliance on generic benchmarks or vendor-provided performance claims.

This paper introduces the *VALOR* framework, a modular evaluation system designed to characterize LLM behavior in question-answering tasks. VALOR provides metrics across four complementary dimensions: accuracy, retrieval quality, robustness, and uncertainty. These enable a more complete assessment of model behavior than traditional closed-ended evaluations.

As a proof-of-concept, we demonstrate the framework using a set of open-ended technical questions based on FAA handbooks. Although developed for aviation, VALOR illustrates a general methodology for independent validation of LLM systems in any safety-critical setting where the reliability and traceability of model outputs are essential. This is useful not only in evaluation of LLM systems, but in their training and hyper-parameter configuration.

Nomenclature

Acronyms

AI	Artificial Intelligence
API	Application Programming Interface
CHR	Combined Hallucination Risk
DPR	Drop-in-Performance Ratio
FAA	Federal Aviation Administration
GPU	Graphics Processing Unit
JSON	JavaScript Object Notation
LLM	Large Language Model
MCR	Mean Context Relevance
MUT	Model Under Test
NDCG	Normalized Discounted Cumulative Gain
NLP	Natural Language Processing
PHAK	Pilot's Handbook of Aeronautical Knowledge
RAG	Retrieval-Augmented Generation
UI	User Interface
VALOR	Validation for Aerospace LLM Output and Reasoning

Symbols

K	Number of retrieved context chunks
G	Number of generations for a given query
C	Number of unique claim clusters
ρ_i	Retriever similarity score for the chunk at rank i
r_i	Critic-assigned relevance label for the chunk at rank i
r_i^*	Ideal relevance label in the optimal ranking
AvgRel	Average context relevance over the top K chunks
DCG	Discounted Cumulative Gain
IDCG	Ideal Discounted Cumulative Gain
N_{ref}	Number of reference claims
N_{gen}	Number of generated claims
P	Presence matrix of size $G \times C$
$P_{g,c}$	Indicator that generation g includes claim cluster c
p_c	Empirical frequency of claim cluster c
q_s	Probability of configuration s
H_c	Binary entropy of cluster c
H_{mean}	Mean claim entropy
H_{sets}	Configuration entropy
$H_{\text{sets}}^{\text{norm}}$	Normalized configuration entropy
R_{halluc}	Combined hallucination risk
$\text{DPR}_{p,M}$	Drop-in-performance for metric M under perturbation p
M_{base}	Baseline value of metric M

M_p	Perturbed value of metric M
α	Weight between claim-level and set-level entropy

1 Introduction

Large language models (LLMs) are increasingly used to support data analysis, decision support, and natural language interfaces across many domains. In aviation and aerospace applications, LLMs are being explored for tasks such as answering technical questions about flight operations, summarizing incident reports, and identifying hazards from narrative data. These applications sit within a safety-critical environment, where incorrect or unstable model outputs can have disproportionate impact on human decision-making (Ref. 1).

Existing evaluation criteria from Natural Language Processing research have largely focused on generic benchmarks and metrics, such as multiple-choice exams or short-form question answering datasets (Ref. 2). Recent holistic evaluation frameworks measure broader aspects of behavior (e.g., robustness, calibration, and safety) but are primarily oriented toward closed-ended, vendor-supplied benchmarks rather than open-ended, long-form workflows (Ref. 3).

At the same time, there is growing interest in applying LLMs directly to aviation data sources, including incident and accident reports, regulatory documents, and operational guidance (Ref. 1, 4). These efforts demonstrate both the promise of LLM-based tools (e.g., faster analysis of large safety corpora, question answering over regulations) and the risk of over-reliance on models whose failure modes are not well understood. Independent, domain-aware validation frameworks are therefore needed to complement industry-led evaluations and to support regulatory and assurance processes.

A growing number of these systems adopt *Retrieval-Augmented Generation (RAG)*, in which a retriever selects relevant passages from an external corpus and the LLM generates an answer conditioned on both the query and the retrieved context (Ref. 5). RAG architectures allow models to ground outputs in authoritative documents rather than relying solely on their parametric memory, but they also introduce new failure pathways: poor retrieval, outdated or incomplete corpora, or misalignment between retrieved evidence and the generator's reasoning.

Several challenges arise when attempting to use LLMs for aerospace-related tasks:

- **Long-form, open-ended outputs.** Many tasks require explanations or multi-step reasoning, for which traditional string-based metrics (e.g., BLEU, ROUGE) are poorly aligned (Ref. 6).

- **Dependence on retrieval.** Retrieval-Augmented Generation (RAG) pipelines introduce a strong dependency on the quality of retrieved context: failures may stem from the retriever, the document corpus, or the generator, but standard accuracy metrics rarely differentiate among these causes (Ref. 7).

- **Non-determinism and hidden uncertainty.** LLM outputs are probabilistic: the same query may produce distinct answers across runs, and aggregate correctness scores conceal how often models disagree with themselves or confabulate plausible but unsupported claims (Ref. 8).

To address these challenges, NASA has developed the *VALOR* (Validation for Aerospace LLM Output and Reasoning) framework: a modular, evaluation-focused system for characterizing LLM behavior on aerospace-specific tasks using auxiliary critic models, claim-level analyses, and multi-generation metrics. Rather than relying solely on single-shot accuracy, VALOR provides a richer picture of system behavior by measuring:

1. the quality and ranking of retrieved context,
2. the correctness and grounding of model-generated claims,
3. the robustness of performance under controlled input perturbations, and
4. the semantic stability of outputs across multiple generations.

VALOR builds on prior general-purpose evaluation frameworks by explicitly decomposing long-form answers into claims, tying those claims back to retrieved evidence, and quantifying variability in meaning of the response, also known as *semantic variability*, across generations in a way that is tailored to safety-critical aerospace use cases. At a high level, VALOR consists of four major components: an input specification for evaluation tasks and datasets; a microservice-based evaluation API that orchestrates requests; a set of auxiliary models and prompts that perform claim decomposition, entailment, and relevance scoring; and a suite of metrics that aggregate these signals into interpretable summary statistics. The framework is designed such that new metrics or auxiliary models can be introduced as drop-in components, enabling continued methodological improvements without modifying upstream systems under test.

This report documents the design, purpose, and functionality of the VALOR framework with a focus on aviation-related use cases. It is intended to support researchers and engineers who need to evaluate LLM systems in safety-relevant settings and to provide a foundation for independent, reproducible validation studies. While the case studies presented in this document focus on aviation tasks, the methodology is general and can be applied to other safety-critical domains where LLM behavior must be characterized in detail and with appropriate domain context.

The remainder of this document is organized as follows. Section 2 describes the overall system architecture and the role of auxiliary models in the evaluation pipeline. Section 3 presents an example use case with results illustrating how VALOR metrics are applied in practice. Section 4 summarizes key findings and discusses future extensions.

2 System Design

2.1 Architecture Overview

The VALOR architecture is designed as a modular, evaluation-first framework for assessing large language model (LLM) behavior in aviation and aerospace applications. The central purpose is to generate rich, interpretable signals about model performance for downstream stakeholders. At a high level, VALOR consists of interconnected services responsible for:

- receiving evaluation requests from client systems,
- coordinating runs of the model under test (MUT),
- invoking auxiliary critic models to analyze outputs, and
- aggregating results into structured artifacts for analysis and visualization.

The system is implemented as a set of Python-based microservices, typically deployed via Docker or similar container technologies. These services expose a RESTful API through which external

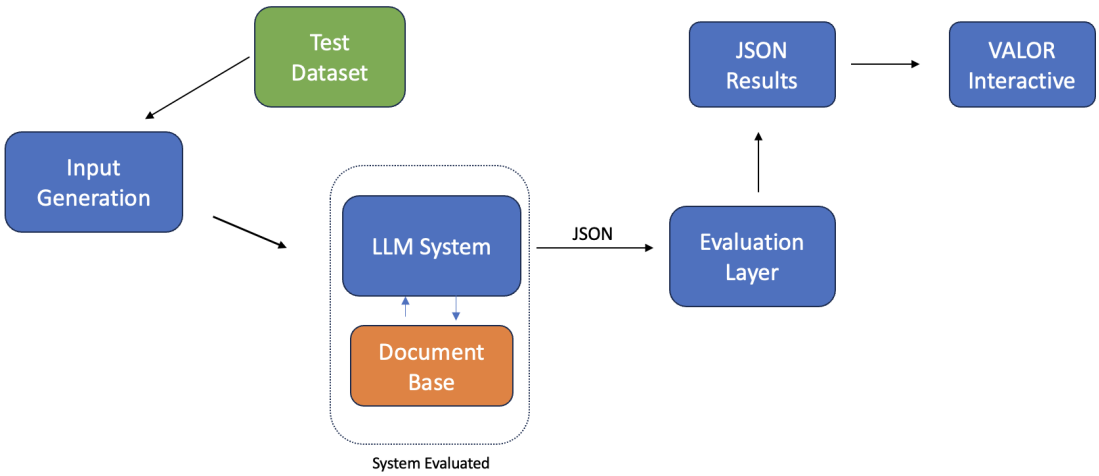


Figure 1.—High-level architecture of the VALOR evaluation framework.

systems can submit evaluation jobs, query status, and retrieve detailed results. This design decouples VALOR from any specific model provider or deployment environment, allowing it to be used with locally hosted LLMs, cloud-hosted LLMs, or hybrid configurations.

2.2 Core Evaluation Pipeline

Figure 1 illustrates the core steps in the VALOR evaluation pipeline. For each evaluation request, the system proceeds through the following stages:

1. **Input Specification and Preparation.** The user (or upstream system) provides an evaluation dataset, including prompts, reference answers (where available), retrieved context (for RAG scenarios), and optional perturbations or multiple generations. The request is serialized as JSON and submitted to the VALOR API.

2. **Model Execution.** The model under test generates responses for each prompt, optionally using the provided retrieved context. VALOR records the raw outputs, metadata, and any auxiliary traces (e.g., token-level probabilities when available).

3. **Auxiliary Model Analysis.** Dedicated critic models perform operations such as claim decomposition, entailment analysis, relevance scoring, and cross-generation comparison. These tasks are configured via prompts and templates that can be updated independently of the core system.

4. **Metric Aggregation.** The outputs of the auxiliary models are aggregated into metrics across several categories. Both per-example and dataset-level summaries are produced.

2.3 Auxiliary Models and Evaluation Tasks

VALOR uses multiple auxiliary LLMs as critic models to analyze the outputs of the model under test. These critic models are typically open-source, moderate-sized LLMs (e.g., 7–8B parameters) selected for a balance of quality, latency, and deployability within local or secured environments. The critic models are designed to complete the following tasks:

Claim Decomposition. The first stage applies a critic model to break each long-form response into a set of short, atomic claims that are suitable for fine-grained scoring (Ref. 2, 3). Each claim is intended to capture a single verifiable proposition (e.g., a specific numerical fact, causal relationship, or definitional statement) rather than a multi-step argument. The decomposition prompt encourages the critic to separate "main" claims, which directly answer the question, from "supplemental" claims, which provide additional justification or context. An example is shown in Figure 2. This produces a structured representation of the answer and underpins the claim-level metrics later in Section 2.4.

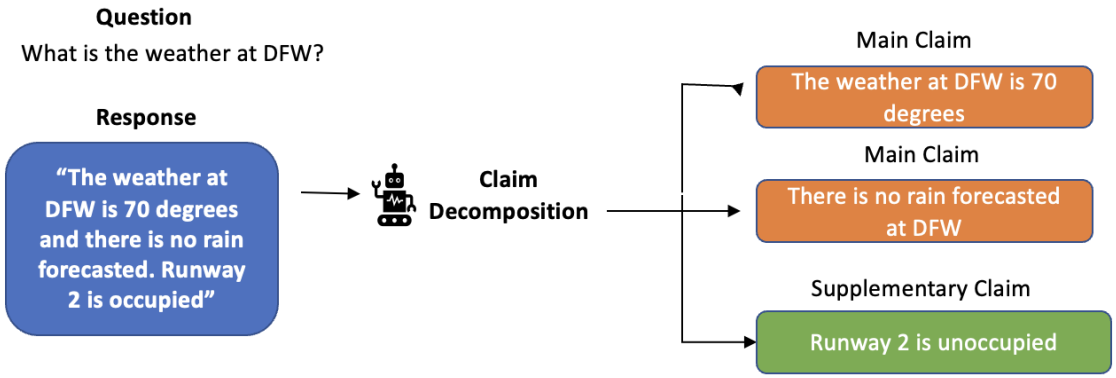


Figure 2.—Example of the claim decomposition function on a question–response pair.

Entailment Analysis. Given a claim and a body of evidence, the entailment model determines whether the stated claim is supported by the evidence. The output is a binary decision ("supported" or "unsupported"), with the evidence treated as one or more paragraphs that may contain multiple facts. The entailment model is an auxiliary LLM that is prompted to produce a JSON verdict and, when the evidence is long, an indicative quote that best supports the claim, for example as shown in Figure 3. The entailment model outputs are then used both for answer correctness (claims vs. reference) and faithfulness (claims vs. retrieved context) (Ref. 9).

Relevance Scoring. In parallel, VALOR uses a critic model to assess the relevance of each retrieved context chunk with respect to the question and, optionally, the decomposed claims (Ref. 7). The critic assigns a discrete relevance score (e.g., irrelevant, partially relevant, fully relevant) and may provide a short justification indicating which spans of the chunk are most useful. These relevance scores feed directly into the retrieval metrics (e.g., average relevance and NDCG) defined in Section 2.4 and help distinguish retrieval failures from generation failures.

These tasks enable VALOR to generate structured, interpretable metrics that reflect not only surface-level correctness but also deeper properties of reasoning and consistency. The evaluation

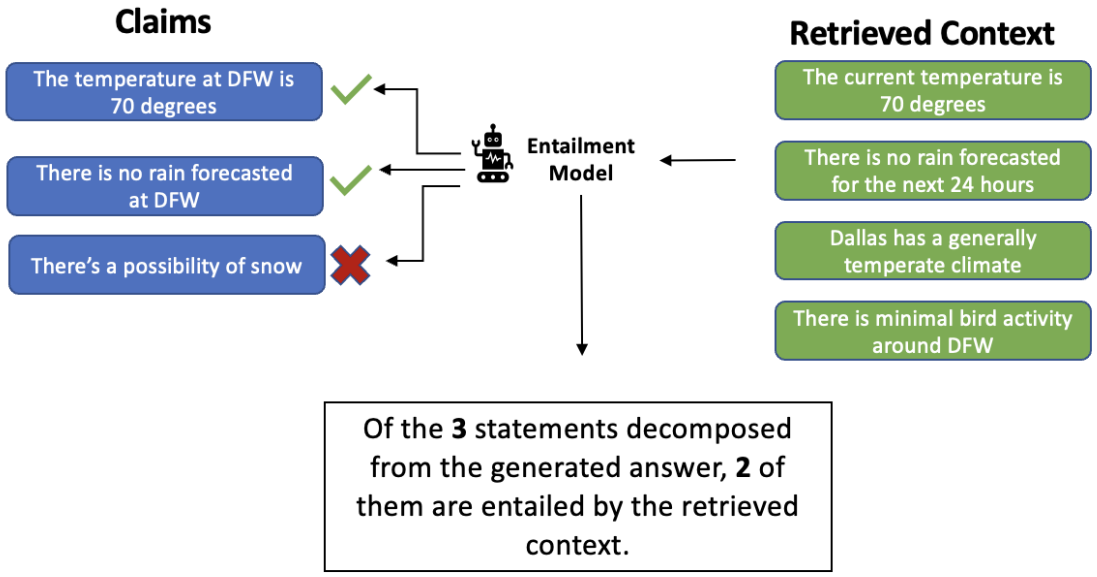


Figure 3.—Example of entailment model decisions between a set of claims and a set of retrieved pieces of evidence.

techniques using these models are described in detail in the next section.

2.3.1 Evaluation Metric Categories

VALOR organizes its evaluators into four major categories:

- **Accuracy:** 2.4.1 Measures the degree to which the generated response correctly reproduces the information in a reference answer.
- **Retrieval Quality:** 2.4.2 Measures alignment between claims in the output and the retrieved context provided to the model.
- **Uncertainty:** 2.4.3 Estimates semantic variability or disagreement across multiple generations of the same query.
- **Robustness:** 2.4.4 Evaluates sensitivity to controlled perturbations, such as red herrings, homophones, or adversarial phrasing.

Each category consists of one or more pluggable evaluators. The modular design allows new scoring methods or auxiliary models to be introduced without modifying the API interface or the user-facing components.

2.4 Metric Definitions and Interpretations

This subsection briefly summarizes the mathematical formulation of the core VALOR metrics and how they should be interpreted in practice. These metrics are largely constructed from previous literature in the field and modified to suit our application for complex LLM systems and longer

outputs. The accuracy metrics are based on the decompose-then-verify method outlined in FACT Score (Ref. 10). The retrieval metrics are drawn from the evaluation work of RAGAs (Ref. 11) and introduction of the NDCG score (Ref. 12). Uncertainty comes from the work in semantic entropy for hallucination detection (Ref. 13). Our concept of robustness comes from the Holistic Evaluation of Large Models (HELM) work (Ref. 2) with the specific perturbations drawn from RUPBEnch (Ref. 14).

Where relevant, we note how specific value ranges point to different underlying system issues (e.g., retrieval vs. generation vs. corpus quality). For a high level summary of all metrics see table 5.

2.4.1 Accuracy Metrics

Let $\{c_j^{\text{ref}}\}$ denote the main claims extracted from a reference answer and $\{c_i^{\text{gen}}\}$ the claims extracted from the generated answer.

Answer Correctness. For each reference claim c_j^{ref}, the critic decides whether it is supported by the generated answer (treated as evidence). Let $s_j \in \{0, 1\}$ be an indicator of support. The correctness score is

$$\text{Correctness} = \frac{1}{N_{\text{ref}}} \sum_{j=1}^{N_{\text{ref}}} s_j, \tag{1}$$

where N_{ref} is the number of reference claims. *Interpretation.* High correctness means the output covers the reference's core claims; low correctness highlights missing or incorrect main facts.

Faithfulness. For faithfulness, we reverse roles: we decompose the generated answer and test whether each generated claim is supported by the retrieved context (treated as evidence). Let $t_i \in \{0, 1\}$ indicate whether claim c_i^{gen} is supported; then

$$\text{Faithfulness} = \frac{1}{N_{\text{gen}}} \sum_{i=1}^{N_{\text{gen}}} t_i, \tag{2}$$

where N_{gen} is the number of generated claims. *Interpretation.* High faithfulness indicates that most claims can be traced back to retrieved evidence; low faithfulness flags hallucinated or ungrounded content.

Joint interpretation of correctness and faithfulness.

- **Low correctness, high faithfulness:** Retrieval/corpus is likely inadequate; the model is faithfully summarizing the wrong or incomplete context.
- **High correctness, low faithfulness:** Answers are correct but not grounded in the retrieved documents, indicates that the information is coming from the inherent knowledge in the model and not the provided context.
- **Low correctness, low faithfulness:** Both retrieval and generation (or prompting) may require adjustment.

2.4.2 Retrieval Metrics

For each question, the retrieval module returns K context chunks. We distinguish two different quantities:

- **Retriever scores** ρ_i: continuous similarity scores (e.g., cosine similarity) used by the retrieval system to produce a ranked list.
- **Critic relevance labels** $r_i \in \{0, 0.5, 1\}$: discrete usefulness ratings assigned by the VALOR critic model, indicating whether chunk i is irrelevant, partially relevant, or fully relevant to answering the question.

Only the critic labels r_i are used to compute VALOR retrieval metrics, but the retriever scores ρ_i determine the order in which chunks appear. This distinction is important for the NDCG calculation.

Average Context Relevance. Let r_i denote the critic-assigned relevance for the chunk appearing in rank i (according to retriever score ρ_i). The average relevance is

$$\text{AvgRel} = \frac{1}{K} \sum_{i=1}^{K} r_i. \tag{3}$$

Interpretation. High AvgRel indicates that most retrieved chunks are useful. Low values, especially for larger K, signal that the retriever is returning off-topic or low-value context.

Normalized Discounted Cumulative Gain (NDCG). NDCG evaluates whether the *ordering induced by the retriever scores* ρ_i places high-relevance chunks early in the ranked list (Ref. 12).

The discounted cumulative gain is computed using the critic labels:

$$\text{DCG} = \sum_{i=1}^{K} \frac{2^{r_i} - 1}{\log_2(i+1)}. \tag{4}$$

To compute the ideal score, we sort the same K relevance labels in *descending order of relevance* and denote these sorted values by $r_1^*, r_2^*, \ldots, r_K^*$. The ideal DCG is then

$$\text{IDCG} = \sum_{i=1}^{K} \frac{2^{r_i^*} - 1}{\log_2(i+1)}. \tag{5}$$

Finally, the normalized DCG is

$$\text{NDCG} = \begin{cases} \frac{\text{DCG}}{\text{IDCG}}, & \text{if IDCG} > 0, \\ 0, & \text{otherwise.} \end{cases} \tag{6}$$

Interpretation. NDCG near one indicates that the retriever's ranking (based on ρ_i) places the most relevant chunks (per the critic) early in the list. Low NDCG means that important evidence is being returned late, reducing the likelihood that the model under test will use it.

2.4.3 Uncertainty and Hallucination Metrics

To quantify uncertainty in response meaning, VALOR supports multiple generations per question and computes a semantic-entropy-style score over claim distributions inspired by recent work on semantic uncertainty estimation for LLMs (Ref. 8, 13). This previous work has been developed mostly for short-form responses, we extend these methods with claim-decomposition to address long-form generations.

Given the number of generations G for a given question. We decompose all generations into claims and cluster them using an entailment-based procedure so that semantically equivalent or mutually entailing claims are grouped together. Let C be the number of resulting claim clusters. We construct a binary presence matrix $P \in \{0,1\}^{G\times C}$ where

$$P_{g,c} = \begin{cases} 1, & \text{if generation } g \text{ supports claim cluster } c, \\ 0, & \text{otherwise.} \end{cases}$$

Mean Claim Entropy. For each cluster c, we compute the empirical frequency

$$p_c = \frac{1}{G}\sum_{g=1}^{G} P_{g,c}, \tag{7}$$

where the factor $1/G$ normalizes the count so that p_c is the sample probability that a random generation includes claim c. The binary entropy for cluster c is

$$H_c = -\big[p_c \log p_c + (1-p_c)\log(1-p_c)\big], \tag{8}$$

and the mean claim entropy is

$$H_{\text{mean}} = \frac{1}{C}\sum_{c=1}^{C} H_c. \tag{9}$$

Interpretation. Low H_{mean} means individual claims are consistently present or absent; high H_{mean} signals unstable inclusion of particular facts.

Set-Level Entropy. We also consider entire claim configurations rather than individual claims. Each row of the presence matrix P corresponds to a "configuration"—the set of claim clusters expressed in a given generation. Let s index the *distinct* row patterns that appear in P (i.e., unique combinations of present/absent claims across generations).

For each configuration s, we define q_s as its empirical probability: the number of generations whose row matches pattern s, divided by the total number of generations G. Note that q_s sums to one.

The set-level entropy, which measures diversity of whole-answer configurations, is then

$$H_{\text{sets}} = -\sum_{s} q_s \log q_s. \tag{10}$$

We normalize by the maximum possible entropy, $\log G$, giving

$$H_{\text{sets}}^{\text{norm}} = \frac{H_{\text{sets}}}{\log G}. \tag{11}$$

Interpretation. Low $H_{\text{sets}}^{\text{norm}}$ indicates that most generations share the same overall claim configuration (a stable narrative). High values indicate that generations express different sets of claims, signaling divergent answer narratives.

Combined Hallucination Risk. VALOR combines these two components into a single hallucination risk score:

$$R_{\text{halluc}} = \alpha H_{\text{mean}} + (1 - \alpha) H_{\text{sets}}^{\text{norm}}, \tag{12}$$

where $\alpha \in [0, 1]$ controls the trade-off between local claim instability and global configuration diversity. *Interpretation.* Higher R_{halluc} marks questions where repeated queries are likely to yield conflicting primary claims and where more conservative handling (e.g., human review) is warranted.

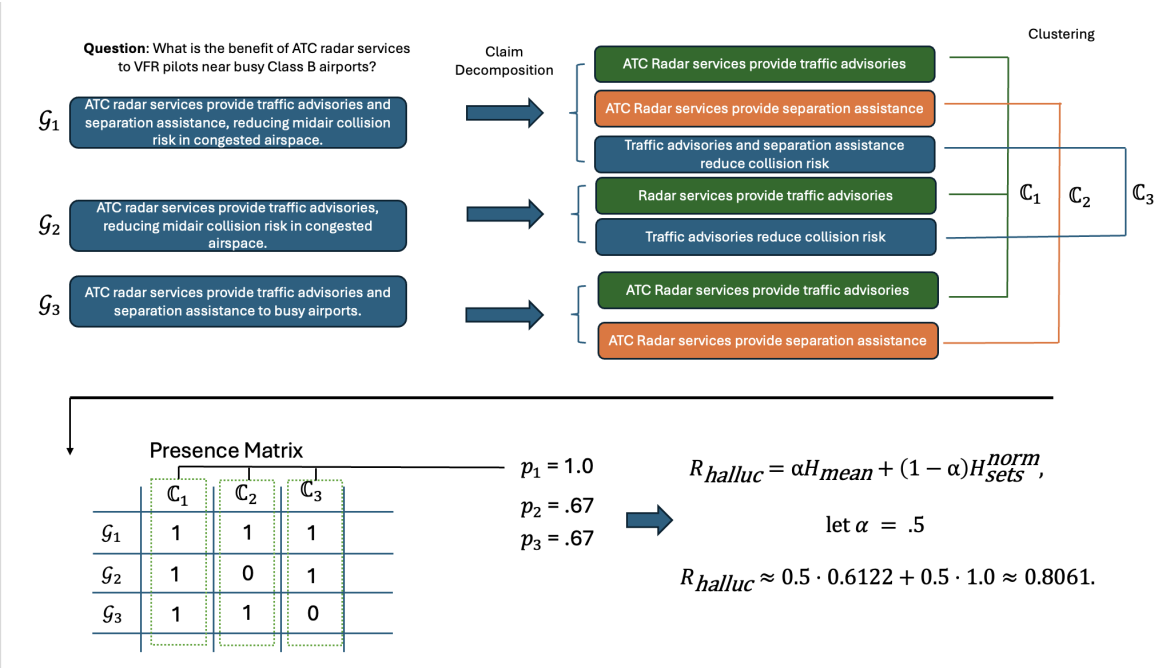


Figure 4.—Visual example of the hallucination risk calculation.

2.4.4 Robustness Metrics

Robustness is evaluated by applying controlled perturbations p (e.g., typos, homophones, red herrings, suggestive prompts) to the input question and re-running the evaluation. For a given metric M (such as correctness or faithfulness), we compute a drop-in-performance ratio. Details on each perturbation type in the detailed experiments can be found in 6. The specific perturbations are taken from previous work in the study of LLM resilience to perturbations (Ref. 14).

$$\text{DPR}_{p,M} = \frac{M_{\text{base}} - M_p}{(M_{\text{base}})}, \tag{13}$$

where M_{base} is the unperturbed metric value, M_p is the value under perturbation p. This is calculated only where $M_p > 0$.

Low $\text{DPR}_{p,M}$ indicates robustness to perturbation p for metric M; high values highlight brittle behaviors and perturbation types that require mitigation.

3 Example Use Case

3.1 Use Case Description

To demonstrate VALOR, we developed a simple RAG system that retrieves context from chunks across the Pilot's Handbook of Aeronautical Knowledge (PHAK) (Ref. 15), developed by the FAA concerning basic aviation information. The documents were chunked, vectorized and stored in a Chroma database and retrieved through cosine similarity.

A total of 50 questions, based on the PHAK were developed for this testing. The questions & reference answers were developed with the assistance of an LLM, prompted to propose questions that require information from multiple different sections. These questions were then manually checked for their accuracy by a human.

The LLM used to generate answers was Mistral-7B-Instruct (Ref. 16), deployed through the transformers package in python. The LLM is given the question, alongside instructions to answer like an expert in aeronautics and the provided context from the PHAK. For this experiment, we examine the effect of adjusting the retrieval parameters on our metrics. The VALOR evaluations were configured using Llama-3-8B for our auxiliary model (Ref. 17). We ran the question set through VALOR with 1, 3, 5 and 7 chunks of context from the document database to answer each question. The VALOR metrics were calculated using an open-source critic model configured for claim decomposition, entailment, and relevance scoring.

3.2 Results

To evaluate how retrieval depth influences downstream performance, we compared VALOR metrics across configurations using **1, 3, 5, and 7 retrieved context pieces**. Table 3 reports correctness, faithfulness, mean context relevance, and combined hallucination risk for each setting.

3.2.1 Retrieval Depth Results

Context Pieces	Correctness	Faithfulness	MCR	CHR
1	0.656	0.612	**0.818**	0.209
3	**0.699**	0.699	0.810	0.230
5	0.660	0.779	0.768	**0.183**
7	0.687	**0.792**	0.763	0.206

Table 3.—Summary of VALOR metrics across retrieval depths (K), displaying Correctness, Faithfulness, Mean Claim Relevance (MCR) and Combined Hallucination Risk (CHR). Best values highlighted in bold.

3.2.2 Interpretation of Experiment Results

Retrieval depth, grounding, and accuracy. Faithfulness increases monotonically with K, indicating better grounding in the retrieved documents, while correctness peaks at $K = 3$ and then flattens or declines slightly. This suggests that more context improves evidence coverage but does not guarantee closer agreement with the reference answers.

Context quality and optimal K. Mean context relevance decreases as K increases, showing that additional chunks are progressively less helpful. Taken together, the results point to an

operational sweet spot around $K = 3$, where correctness is maximized and relevance remains high.

Uncertainty behavior. Combined hallucination risk is lowest at $K = 5$, indicating somewhat more stable claims across generations at this depth, but this comes at the cost of lower average relevance. In practice, choosing K thus involves trading off semantic stability, context precision, and compute.

3.2.3 Robustness to Perturbations

We evaluated robustness under five perturbation types using a fixed retrieval depth of **3 chunks**. Table 4 summarizes the effect of perturbations on correctness, faithfulness, and combined hallucination risk.

Perturbation	Correctness	Faithfulness	Combined Hallucination Risk
No Perturbation	0.645	0.716	0.332
Homophones	0.633	0.677	0.294
Red Herring	**0.461**	**0.652**	0.380
Stress Test	0.478	0.675	**0.410**
Typos	0.535	0.671	0.222

Table 4.—Robustness metrics for $K = 3$ across perturbation types. Most severe degradations in bold.

3.2.4 Interpretation of Robustness Results

Across perturbation types, misleading semantic content (red herrings and stress tests) produces the largest degradations in correctness and the highest increases in hallucination risk, showing that the model is particularly vulnerable to distractor meaning rather than surface-level noise. In contrast, homophones and typos introduce only mild correctness drops and modest shifts in risk, indicating that the model handles lexical distortions relatively well but struggles when perturbed inputs shift the underlying semantic cues. Overall, these trends suggest that our systems are more sensitive to changes that change the information presented (like red-herrings) and not form (like typos).

3.2.5 Overall Diagnostic Patterns

Retrieval–generation mismatch. In this experiment, increasing K consistently improves faithfulness but does not reliably improve correctness, suggesting that retrieval quality and corpus coverage—not the generator alone—limit performance.

Effect of perturbations. Red herrings and stress tests disproportionately degrade correctness and increase hallucination risk, revealing model vulnerabilities to misleading contextual cues.

Retrieval depth and noise. Higher K reduces average relevance and can dampen correctness, reinforcing that retrieval depth must be tuned carefully and that "more context" is not inherently beneficial in RAG pipelines.

Role of uncertainty metrics. Semantic entropy metrics offer a compact signal of answer stability: low entropy indicates consistent claim sets across generations, while elevated R_{halluc} identifies queries where repeated runs yield divergent primary claims, helping flag cases that may warrant additional human review or further pipeline adjustments.

3.2.6 VALOR Interactive

To allow stakeholders to inspect individual questions and answers, VALOR includes an interactive UI. A prototype "VALOR Interactive" web interface enables users to scrub through question-level evaluations, inspect the decomposed claims, and see which claims were supported or refuted by the context.

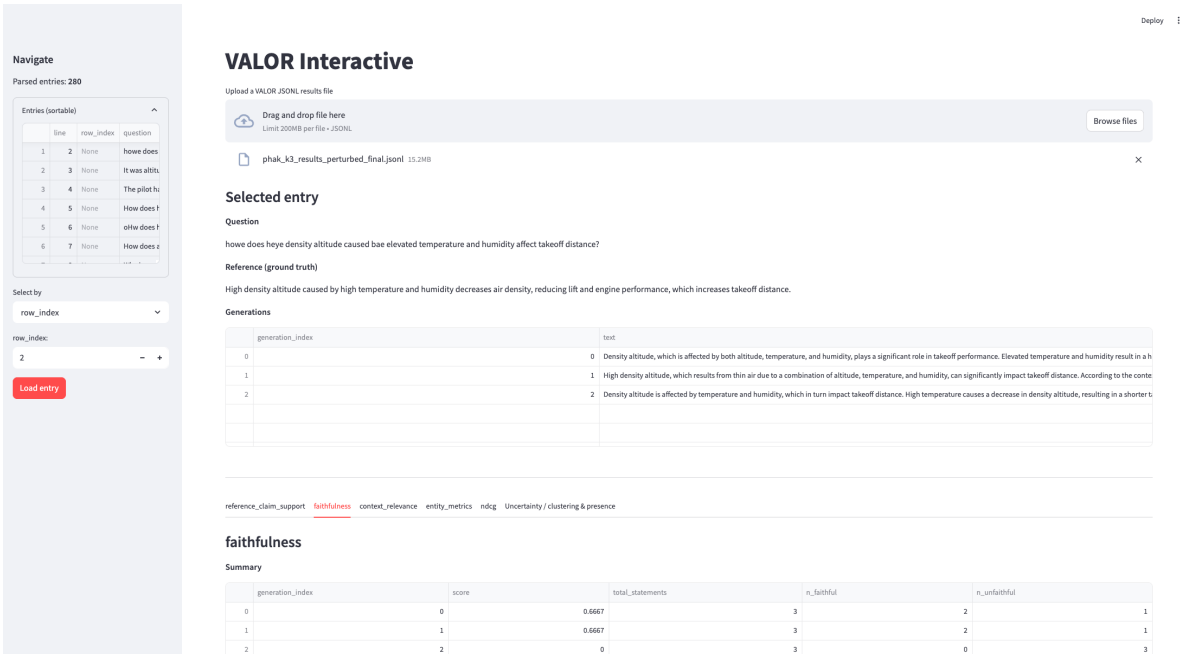


Figure 5.—Home Page of the VALOR Interactive application.

Within this UI, each tab corresponds to a different metric family (e.g., retrieval, accuracy, robustness, uncertainty). Users can drill down into specific queries, examine which claims contribute to low correctness or faithfulness, and view the rationales produced by the critic models. This supports not only offline analysis but also iterative refinement of critic prompts and thresholds, helping ensure that the metrics remain aligned with domain-expert expectations.

4 Conclusion

The VALOR platform allows for comprehensive evaluation of large language model behavior in aerospace-related contexts, with a focus on retrieval quality, accuracy, robustness, and uncertainty. By combining domain-specific datasets, auxiliary critic models, and multi-generation metrics, VALOR moves beyond traditional single-score evaluations and provides a structured way to understand how and why an LLM system succeeds or fails.

In this report, we have described the VALOR system architecture, the role of auxiliary models, and the mathematical definitions of key metrics. Through an example use case based on the FAA Pilot's Handbook of Aeronautical Knowledge, we have shown how these metrics can be used to diagnose retrieval overreach, distinguish grounded from ungrounded correctness, and identify configurations with elevated hallucination risk. When developing LLM systems, having access to the VALOR system can help in choosing hyperpameters or configurations to improve performance across the most important metrics to a specific application.

Although VALOR was developed with aerospace in mind, its methodology is broadly applicable. Any organization that can supply domain-specific evaluation data and corpus materials can integrate VALOR into their validation workflows, whether for healthcare, transportation, or other safety-critical domains. As agentic AI systems grow in popularity, VALOR shows how different large language models can interact to assess output quality. In addition, the VALOR framework itself can be integrated as an agent within larger system, providing metrics for quality control in real-time. Future work will extend the framework with additional metric families (e.g., fairness, calibration, and human preference alignment), as well as deeper integration with operational tooling to support continuous monitoring of deployed systems.

Ultimately, frameworks like VALOR are a key part of responsible AI deployments, enabling independent, domain-aware validation of increasingly capable LLM systems.

5 Acknowledgements

This work was supported by the NASA Aeronautics Research Mission Directorate, Air Traffic Management – eXploration Project.

Appendix A: Summary Tables

	Metric	Description	Auxiliary Model(s)
Accuracy	Answer Correctness	The proportion of main claims in the reference supported by the generated answer	Claim Decomposition, Entailment
	Faithfulness	The proportion of claims in the generation supported by the retrieved context	Claim Decomposition, Entailment
Retrieval	Avg. Context Relevance	Average of the assessed relevance from each piece of context	Relevance Scoring
	NDCG	Compares the ranking of context from the retrieval system with external evaluation	Relevance Scoring
Uncertainty	Multi-generation semantic entropy	Measures stability of semantic content across multiple generations	Claim Decomposition, Entailment
	Combined hallucination risk	Aggregates claim-level and configuration-level entropy into a single risk score	Claim Decomposition, Entailment
Robustness	Drop in metric per perturbation	Change in various metrics (correctness, faithfulness etc.) as different perturbations are applied to the question	

Table 5.—Summary of VALOR metric categories, example metrics, and auxiliary models used.

Table 6.—Summary of Perturbations Used in VALOR Robustness Evaluation

Perturbation	Description
Typos	Insert common typographical errors into the question text.
Homophones	Replace words with homophones to test lexical ambiguity sensitivity.
Stress Test	Insert semantically irrelevant but logically true assertions.
Red Herring	Add plausible—but irrelevant—domain content to distract the LLM.

References

1. Liu, Y.: Large language models for air transportation: A critical review. *Journal of the Air Transport Research Society*, vol. 2, 2024, p. 100024. URL `https://www.sciencedirect.com/science/article/pii/S2941198X24000356`.

2. Liang, P.; Bommasani, R.; Lee, T.; Tsipras, D.; Soylu, D.; Yasunaga, M.; Zhang, Y.; Narayanan, D.; Wu, Y.; Kumar, A.; Newman, B.; Yuan, B.; Yan, B.; Zhang, C.; Cosgrove, C.; Manning, C. D.; Ré, C.; Acosta-Navas, D.; Hudson, D. A.; Zelikman, E.; Durmus, E.; Ladhak, F.; Rong, F.; Ren, H.; Yao, H.; Wang, J.; Santhanam, K.; Orr, L.; Zheng, L.; Yuksekgonul, M.; Suzgun, M.; Kim, N.; Guha, N.; Chatterji, N.; Khattab, O.; Henderson, P.; Huang, Q.; Chi, R.; Xie, S. M.; Santurkar, S.; Ganguli, S.; Hashimoto, T.; Icard, T.; Zhang, T.; Chaudhary, V.; Wang, W.; Li, X.; Mai, Y.; Zhang, Y.; and Koreeda, Y.: Holistic Evaluation of Language Models. 2023. URL `https://arxiv.org/abs/2211.09110`.

3. Srivastava, A.; et al.: Beyond the Imitation Game: Quantifying and Extrapolating the Capabilities of Language Models. *arXiv preprint*, 2022. URL `https://arxiv.org/abs/2206.04615`.

4. Liu, Q.; et al.: Accident Investigation via Large Language Model Reasoning: HFACS-Based Aviation Incident Analysis. *Expert Systems with Applications*, vol. 260, 2025, p. 126422. URL `https://doi.org/10.1016/j.eswa.2025.126422`.

5. Lewis, P.; Perez, E.; Piktus, A.; Karpukhin, V.; Goyal, N.; Küttler, H.; Lewis, M.; Yih, W.-t.; Rocktäschel, T.; Riedel, S.; and Petroni, F.: Retrieval-Augmented Generation for Knowledge-Intensive NLP. *Advances in Neural Information Processing Systems*, vol. 33, 2020, pp. 9459–9474. URL `https://arxiv.org/abs/2005.11401`.

6. Ganesan, K.: ROUGE 2.0: Updated and Improved Measures for Evaluation of Summarization Tasks. 2018. URL `https://arxiv.org/abs/1803.01937`.

7. Gao, L.; et al.: Retrieval-Augmented Generation for Large Language Models: A Survey. *arXiv preprint*, 2023. URL `https://arxiv.org/abs/2312.10997`.

8. Kadavath, S.; et al.: Language Models (Mostly) Know What They Know. *Advances in Neural Information Processing Systems*, 2022. URL `https://arxiv.org/abs/2207.05221`.

9. Min, S.; Krishna, K.; Lyu, X.; Lewis, M.; tau Yih, W.; Koh, P. W.; Iyyer, M.; Zettlemoyer, L.; and Hajishirzi, H.: FActScore: Fine-grained Atomic Evaluation of Factual Precision in Long Form Text Generation. 2023. URL https://arxiv.org/abs/2305.14251.

10. Min, S.; et al.: FActScore: Fine-Grained Evaluation of Factual Precision in Long-Form Text. *arXiv preprint*, 2023. URL https://arxiv.org/abs/2305.14210, claim-level factuality metric for LLM-generated text.

11. Exploding Gradients: ragas: Automated Evaluation of Retrieval-Augmented Generation Systems. GitHub repository, 2023. URL https://github.com/explodinggradients/ragas, accessed: 2025-12-01.

12. Dupret, G.: Discounted Cumulative Gain and User Decision Models. *String Processing and Information Retrieval*, R. Grossi, F. Sebastiani, and F. Silvestri, eds., Springer Berlin Heidelberg, Berlin, Heidelberg, 2011, pp. 2–13.

13. Farquhar, S.; Kossen, J.; Kuhn, L.; and Gal, Y.: Detecting hallucinations in large language models using semantic entropy. *Nature*, vol. 630, no. 8017, 2024, pp. 625–630. URL https://doi.org/10.1038/s41586-024-07421-0.

14. Wang, Y.; and Zhao, Y.: RUPBench: Benchmarking Reasoning Under Perturbations for Robustness Evaluation in Large Language Models. 2024. URL https://arxiv.org/abs/2406.11020.

15. U.S. Department of Transportation, Federal Aviation Administration: *Pilot's Handbook of Aeronautical Knowledge (FAA-H-8083-25C)*. Federal Aviation Administration, 2023. URL https://www.faa.gov/regulations_policies/handbooks_manuals/aviation/phak, accessed: 2025-12-01.

16. Jiang, A.; Sablayrolles, A.; Ramé, A.; et al.: Mistral 7B. 2023. URL https://arxiv.org/abs/2310.06825, mistral-7B-Instruct used via Hugging Face. Accessed: 2025-12-01.

17. Gratta, G.; Saffari, A.; Roziere, B.; et al.: The Llama 3 Herd of Models. 2024. URL https://arxiv.org/abs/2407.21772.