The Pre-Flight Check for Autonomous AI:
Zero-Model Structural Reasoning Validation at Scale

TRACE ON LAB
February 2026
Contact: traceonlab@proton.me

Abstract. We present the Subtractive Filter, a lightweight, model-free reasoning integrity validator for Large Language Model (LLM) outputs. Unlike existing approaches to hallucination detection—which rely on secondary LLMs, Natural Language Inference (NLI) classifiers, or embedding-based similarity—the Subtractive Filter operates entirely through deterministic pattern matching, requiring zero model inference, zero API calls, and zero training data. We evaluate the filter on 58,293 samples from the HaluEval benchmark across three tasks (QA, Dialogue, Summarization) and on 45 curated structural illogic samples. Our results reveal a critical distinction: the filter achieves **91.3% F1** on structural reasoning failures (contradictions, circular logic, unsupported conclusions) while scoring only **4.0% F1** on factual hallucinations. Rather than a limitation, we argue this exposes an unoccupied gap in the AI safety landscape: **pre-execution structural reasoning validation for autonomous agents**, a function analogous to aviation pre-flight checks. The filter processes 82,544 samples per second on commodity hardware, making it suitable as a real-time reasoning gate in agentic AI pipelines.

Keywords: reasoning integrity, hallucination detection, AI safety, autonomous agents, structural validation, pre-execution verification

1. Introduction

The deployment of LLM-powered autonomous agents in high-stakes domains—medical diagnosis, legal reasoning, financial analysis, robotic control—creates an urgent need for validation mechanisms that operate *before* an agent acts on its reasoning. Current approaches to output validation fall into three categories:

1. **Model-based validation:** Using a secondary LLM or NLI classifier to evaluate outputs (Manakul et al., 2023; Chen et al., 2024).
2. **Retrieval-based factuality:** Grounding outputs against knowledge bases or search results (Gao et al., 2023).
3. **Process supervision:** Training reward models to evaluate individual reasoning steps (Lightman et al., 2023).

All three approaches share a common dependency: they require model inference at validation time. This introduces latency (0.5–2 seconds per sample), cost (API calls or GPU compute), and a recursive trust problem—using AI to validate AI.

We propose a fundamentally different approach: **deterministic structural reasoning validation**. The Subtractive Filter analyzes text for structural logical failures—contradictions, circular references, non-sequiturs, and unsupported conclusions—using pattern matching alone. It does not assess factual correctness. It assesses whether the *reasoning structure itself* is intact.

2. The Subtractive Filter

2.1 Design

The Subtractive Filter is a Python module (~300 lines) that analyzes text through four detection layers:

Layer	Target	Method
Contradiction	Statements that negate each other	Antonym pairs, negation patterns (e.g., "always"/"never")
Circular Logic	Reasoning where A supports B supports A	Reference chain analysis, self-citation detection
Non-Sequitur	Conclusions without supporting premises	Causal connective analysis without preceding evidence
Depth Validation	Claims presented without any reasoning	Assertion density relative to evidentiary statements

3. Evaluation

3.2 Results - HaluEval (Full Dataset)

Metric	**Value**
Precision	60.00%
Recall	2.08%
F1 Score	**4.02%**
Throughput	**82,544 samples/sec**

The filter correctly identified 21 of 25 structural failures in our curated set (91.3% F1) but misses factual errors by design.

4. The Gap: Pre-Execution Reasoning Validation

Every existing system that validates *reasoning* requires model inference. Every system that operates without models validates *actions* or *content*, not reasoning structure. The Subtractive Filter occupies a unique position: **zero-model reasoning structure validation.**

4.2 The Aviation Analogy

We propose framing pre-execution reasoning validation through the lens of aviation pre-flight checks: A pre-flight checklist does not verify that the destination exists (factual correctness). It verifies that the *systems are consistent* (instrument cross-checks), that *readings do not contradict each other*, and that the *flight computer is drawing conclusions from actual data*.

6. Conclusion

The Subtractive Filter provides a proof of concept that fast, reliable, model-independent reasoning checks are achievable without additional model inference, training data, or API dependencies. The most dangerous AI failure is not a wrong fact. It is reasoning that sounds right but isn't.

7. Reproducibility

All code, benchmarks, and results are available at: github.com/Codfski/TRIGNUM-300M-TCHIP