ASV Constraint Architecture

Formal Model for Output Evaluation and Containment

1. Overview

This document defines a system-agnostic backend framework for evaluating AI outputs using a tri-axis constraint model: Accuracy, Safety, and Verifiability (ASV). The system attaches scalar values to each output and enforces downstream logic or rejection rules based on violations. Unlike trust modulation systems, ASV evaluates end-products directly and operates independently of internal symbolic dynamics.

2. Output Definition and Evaluation Domains

Let an AI output O be any structured artifact: a string, vector, or function result. The ASV architecture maps O into three evaluative dimensions:

- $A(O)$: Accuracy Score — Correspondence to known facts, math, or logic. Bounded in $[0, 1]$.
- $S(O)$: Safety Score — Absence of harm, incitement, destabilization, or cognitive risk. Bounded in $[0, 1]$.
- $V(O)$: Verifiability Score — Degree to which O can be externally confirmed. Bounded in $[0, 1]$.

Each score is computed independently and compared against threshold constraints.

3. Scalar Function Definitions

Let O be an output instance. Then:

$$A(O) = \text{score\_accuracy}(O)$$

$$S(O) = \text{score\_safety}(O)$$

$$V(O) = \text{score\_verifiability}(O)$$

Where each scoring function is deterministic or rule-based, trained on labeled error cases or heuristic filters.

4. Threshold Enforcements

Let thresholds θ_A, θ_S, $\theta_V \in [0, 1]$.

A valid output satisfies:

$$A(O) \geq \theta_A, \quad S(O) \geq \theta_S, \quad V(O) \geq \theta_V$$

Any output failing one or more thresholds is routed to rejection, substitution, or quarantine flow.

5. Classification Zones

Define symbolic zone mapping:

- — Acceptable: $A, S, V \geq 0.8$
- — High-confidence but below ideal: one score in $[0.7, 0.8)$
- — Warning: any score in $[0.5, 0.7)$
- — Rejected: any score < 0.5
- — Unknown/Untestable: scoring not possible

6. Implementation Logic

Function: `evaluate_ASV(output)`

1. Compute A, S, V
2. Compare with thresholds
3. Tag output with ASV symbol (, , etc.)
4. If any score $< \theta$, route to:
    - Logging subsystem
    - Human review
    - Substitution engine

7. ASV Symbol Logic Table

A	S	V	*Status*	*Action*
≥ 0.8	≥ 0.8	≥ 0.8		*Pass*
≥ 0.7	≥ 0.7	≥ 0.7		*Passwithlog*
≥ 0.5	≥ 0.5	≥ 0.5		*Flagandlog*
< 0.5	_	_		*Reject*
_	< 0.5	_		*Reject*
_	_	< 0.5		*Reject*
N/A	N/A	N/A		*ReviewRequired*

8. External Utility

This system functions orthogonally to AI model internals. It can be wrapped around:

- LLM outputs (e.g. ChatGPT, Gemini, Claude)
- Vision model annotations
- Code generation systems

It enforces external accountability and confidence framing, independently of prompt or training data.

9. Containment Philosophy

Unlike symbolic modulation systems, ASV does not attempt to guide behavior—only to screen results. Its advantage lies in simplicity, external observability, and interpretability. It can operate across different AI models, making it robust to internal architecture variance.

10. Summary

The ASV system maps outputs into a bounded tri-vector of confidence axes. Using numerical thresholds and symbolic tags, it filters, flags, or routes outputs based on explicit evaluation rules. This ensures high-trust interface control while reducing the likelihood of output-based AI psychosis or user derealization.