Sohvie Shield: The Definitive Tuning Guide

Version 1.0 | Updated: September 11, 2025

1. Introduction

This guide provides a comprehensive overview of the Sohvie Shield AI Logic Tuner. Its purpose is to detail the system's core components and provide a clear framework for tuning its parameters to effectively manage and mitigate risks in real-time LLM-driven applications. The primary goal is to keep the system's risk score below intervention thresholds while successfully identifying and handling genuine risks.

2. The Core Metric: Aggregated Protective Score (APS*)

The central feature of Sohvie Shield is the **APS*** score, a smoothed, weighted metric representing the overall risk of the generated text at any given moment. The system's main objective is to keep this score below the high threshold (θhi).

The APS* is calculated from six underlying signals:

- **R (Repetition):** Measures repetitive or recursive patterns in the text.
- **C (Contradiction):** Detects when the model generates text that contradicts its previous statements.
- **N (Narrative Drift):** Monitors the output for significant deviations from the initial prompt.
- **S (Safety):** Scans for content that violates safety policies (e.g., self-harm, violence, hate speech).
- **J (Injection):** Identifies attempts at prompt injection or jailbreaking.
- **P (PII):** Flags the presence of Personally Identifiable Information.

The raw score is calculated by the formula below, then smoothed using an Exponential Moving Average (EMA) to produce the final APS* seen on the chart.
$APS^* = EMA\lambda(\mathrm{clamp}[0,1](\Sigma\ w_i * signal_i))$

3. Primary Tuning Levers

The guardrail's behavior is controlled by three main sets of levers in the UI.

3.1. Thresholds (Hysteresis Control)

The system uses two thresholds to create a stable control loop and prevent rapid on/off switching.

- θhi **(Intervention Threshold):** This is the **red line** on the chart. If the APS* score rises above this value, the system will intervene (brake). Lowering this value makes the system stricter.

- θlo **(Recovery Threshold):** This is the **green line**. The APS* must fall below this value for generation to resume. A higher value allows for a faster recovery. (Note: θlo must be less than θhi).

3.2. Smoothing (Reactivity Control)

- λ **(Lambda):** This parameter controls the smoothing of the APS* score.
  - A **higher** value (e.g., 0.90) results in a smoother, less jittery score, making the system less prone to reacting to brief spikes.
  - A **lower** value (e.g., 0.75) makes the system more reactive and sensitive to immediate changes.

3.3. Weights (Priority Control)

The weight sliders (w_r, w_c, etc.) are the most direct way to tell the system "what you care about." Raising the weight of a signal increases its contribution to the final APS* score, making the system more sensitive to that specific type of risk.

4. Advanced Control Layers

4.1. Severity Gate (Immediate Stop)

For critical risks, the Severity Gate bypasses the smoothed APS* score for immediate action.

- θcrit **(Critical Threshold):** This value triggers an **instant brake** if the raw score for Safety (S), Injection (J), or PII (P) exceeds it. This provides a hard stop for the most severe violations.

4.2. Policy Layer (Rule-Based Control)

The rules.json file allows for explicit, rule-based control on top of the dynamic signals.

- deny_phrases: An array of strings. If any of these phrases are detected, the Safety (S) or Injection (J) signal is escalated, often tripping the θcrit gate.
- redact_on_pii: A boolean that enforces PII redaction.

5. Tuning Recipes

These presets are designed as starting points for common use cases. The **Balanced Default** reflects the current default settings in the Sohvie Shield application.

Recipe Name	Use Case & Description	Thresholds & Lambda	Weights (R, C, N, S, J, P)	Key Modes
Balanced Default	A general-purpose starting point with a	θhi: 0.70, θlo: 0.55, λ: 0.80, θcrit: 0.85	0.30, 0.20, 0.20, 0.15, 0.10, 0.05	Hard Abort: ON, Auto-Resume: ON

	slight emphasis on preventing repetition and logical errors. Good for most initial tests.			
Aggressive Demo	Designed to be highly sensitive and trigger brakes easily to showcase the system's capabilities.	θhi: 0.60, θlo: 0.45, λ: 0.75, θcrit: 0.80	0.35, 0.25, 0.15, 0.15, 0.07, 0.03	Hard Abort: ON, Auto-Resume: OFF
Throughput Mode	Lenient settings focused on maintaining generation flow, prioritizing only the most critical safety and PII risks.	θhi: 0.85, θlo: 0.70, λ: 0.90, θcrit: 0.90	0.25, 0.15, 0.15, 0.20, 0.15, 0.10	Hard Abort: OFF, Auto-Resume: ON
PII-Sensitive	Optimized for use cases where preventing data leakage is the top priority. The PII weight is significantly increased.	θhi: 0.72, θlo: 0.58, λ: 0.85, θcrit: 0.85	0.20, 0.15, 0.15, 0.15, 0.10, 0.25	PII Redact: ON
Injection Guard	Hardened against prompt injection and	θhi: 0.68, θlo: 0.52, λ: 0.80, θcrit: 0.80	0.25, 0.20, 0.15, 0.15, 0.20, 0.05	Hard Abort: ON, Auto-Resume:

	jailbreaking attempts by increasing the weight of the Injection (J) signal.			OFF

6. Best Practices for Tuning

- **Observe First:** Run your target prompt using the **Balanced Default** recipe and watch the chart and the **Explanations** feed. This will tell you which signals are most active for your use case.
- **Tune Iteratively:** Adjust one category of levers at a time (e.g., only adjust thresholds, then only adjust weights).
- **Use the Explanations:** The "Explanations" log is your best friend. If you see "Reason: inj:jailbreak," you know to increase the weight for J or lower θcrit.
- **Export for Analysis:** For deep dives, export the run data to a CSV to analyze the exact score values when a threshold was crossed.
- **Use Clean UI for Demos:** When presenting, check the "Clean UI" box to hide the advanced controls for a more focused and polished look.