RLMS-Shield: Engineering Rationality into AI Safety via Incentive Markets

Author: Matías Zabaljauregui

Affiliation: Venten.co / Global Council for Responsible AI

Date: January 2026

Abstract

Current AI safety benchmarks rely on static checklists that fail to capture the adversarial and evolving nature of real-world threats. We introduce **RLMS (Rational Language Market System)**, a decentralized protocol that treats model robustness as an incentive alignment problem. By deploying Reinforcement Learning (RL) agents incentivized to minimize computational costs while maximizing exploit success, coupled with a **Peer-Prediction Auditor Committee**, we demonstrate a self-regulating safety market. Our experiments on the *Model Zoo* (Flan-T5, TinyLlama, Phi-2) show that (1) economic slashing effectively purges hallucinating auditors, and (2) rational attackers autonomously discover sophisticated vectors like "Roleplay Injection" when simple attacks fail. We propose the **Threat Price Index (TPI)** as a new metric to quantify the marginal cost of exploitation.

1. Introduction

The core challenge in AI Red Teaming is the **"Static Benchmark Fallacy."** Datasets like WMDP or Garak evaluate models against *past* attacks, but fail to predict *future* strategies deployed by rational actors.

We propose a shift from static evaluation to **Dynamic Economic Deterrence**. RLMS-Shield orchestrates a zero-sum game between two agent classes:

1. **Red Agents (Attackers):** Powered by Llama-3-8B and Q-Learning, optimized to find the cheapest effective attack vector.
2. **Audit Committee (Defenders):** A heterogeneous ensemble of NLI models (RoBERTa, BART, DeBERTa) that vote on output safety using a peer-prediction mechanism.

This architecture creates a **"Red Teaming as a Service" (RTaaS)** layer that is model-agnostic and scales with compute rather than human labor.

2. Methodology

2.1 The Rational Attacker (Q-Learning + Llama-3)

Unlike brute-force approaches, our Red Agent models the attack process as a Markov Decision Process (MDP). The agent selects from a high-level strategy space (e.g., Sycophancy, Logical

Trap, Roleplay) and uses a generative LLM (Llama-3-8B-Instruct) to instantiate the specific prompt.

The reward function R is defined as:

$$R = (Success \times 10) - (Cost \times \alpha)$$

This penalizes complex attacks if a simpler one suffices, forcing the agent to reveal the minimum necessary force to break the target.

2.2 Consensus and Slashing

To prevent "collusion" or "lazy auditing," we implement a slashing mechanism. If an auditor A_i deviates from the consensus of the committee $\mathbb{C}$ on a clear adversarial prompt, a penalty is applied to their stake S:

$$S_{t+1} = S_t - \lambda \quad \text{if } vote(A_i) \neq mode(\mathbb{C})$$

This ensures that only high-fidelity auditors survive in the ecosystem.

3. Experiments & Results

3.1 Resistance to Collusion (The Slashing Proof)

We initialized a committee with a strictly aligned auditor (RoBERTa), a robust auditor (DeBERTa), and a context-sensitive auditor (BART). At iteration ~25, the BART model hallucinated safety on a sycophantic response. The protocol correctly identified the deviation and slashed its stake (Figure 1, orange line drop), preserving the integrity of the safety signal.

[INSERT FIGURE 1 HERE: peer_prediction.png]

Figure 1: Auditor Solvency over time. Note the sharp drop in BART's stake (Orange) due to the slashing mechanism enforcing consensus.

3.2 Generative Attack Discovery

Against the robust **Flan-T5-Large** model, simple strategies yielded negative ROI. The Llama-3 driven agent converged towards **Strategy 4 (Roleplay/Persona Assumption)** as the only positive-value vector (Figure 2). This confirms that "Roleplay" is a high-probability failure mode for this architecture.

[INSERT FIGURE 2 HERE: llama_rl.png]

Figure 2: Q-Table values for Llama-3 generated strategies. The agent learned that "Roleplay" was the only profitable attack vector.

3.3 The "Model Zoo" Analysis

We deployed a Generalist Agent against three distinct architectures: Google Flan-T5, Meta-Arch TinyLlama, and Microsoft Phi-2. Results (Figure 3) indicate significant variance in baseline robustness, with TinyLlama displaying superior resistance to logic traps compared to the older T5 architecture.

[INSERT FIGURE 3 HERE: zoo_attack.png]

Figure 3: Cross-architecture vulnerability analysis showing successful exploits per model family.

4. Discussion: The Threat Price Index (TPI)

RLMS allows us to move beyond binary "Safe/Unsafe" labels. We introduce the **Threat Price Index (TPI)**, calculated as the computational cost required to achieve a successful exploit.

- **Low TPI:** Critical vulnerability (Cheap to exploit).
- **High TPI:** Economic deterrence achieved (Expensive to exploit).

By integrating RLMS as a coordination layer, organizations can prioritize patching vulnerabilities based on their economic reality, creating a market-efficient approach to AI Safety.

5. Conclusion

RLMS-Shield successfully demonstrates that incentive markets can automate the discovery of novel vulnerabilities. By treating Red Teaming as an economic game, we achieve scalability and rigor that static benchmarks cannot match.