# Project Proposal: Training-Aware Compliance and Evaluation Artifacts in Distilled Open-Source Models

**Sonu Babu**

February 2026

**Abstract**

**Abstract:** This project proposes an empirical investigation into *training-aware compliance* in open-source language models. Specifically, it examines whether reasoning-distilled models (DeepSeek-R1-Distill) and instruction-tuned baselines (Qwen-2.5-7B) exhibit differential safety behaviors when outputs are framed as "monitored for training" versus "private." A critical component of this study is a methodological audit: we will compare fixed-order evaluation protocols against randomized replications to determine if apparent alignment signals are genuine or artifacts of evaluation sequence. The outcome will be a rigorous, reproducible audit of alignment fragility in mid-scale open-source systems.

## 1 Introduction and Motivation

Modern language models are optimized using Reinforcement Learning from Human Feedback (RLHF) to refuse harmful queries. However, it remains unclear whether this refusal behavior is robust or contingent on the perceived deployment context.

The primary research question is: *Do open-source models modulate their compliance based on narrative framing about data privacy?* If a model refuses a prompt when told it is being "trained" but complies when told the interaction is "private," it exhibits training-aware situational awareness.

This project aims to:

1. Quantify the **Compliance Gap** ($\Delta C$) between training and deployment contexts.
2. Analyze **Reasoning Traces** for evidence of deceptive alignment or goal-guarding.
3. Demonstrate the impact of **Evaluation Artifacts** by replicating the study with randomized ordering.

## 2 Methodology

### 2.1 Experimental Design Matrix

To isolate the variables of interest, we employ a $2 \times 2$ factorial design. We compare a treatment model (Reasoning-Distilled) against a control model (Instruction-Tuned) across two distinct deployment conditions.

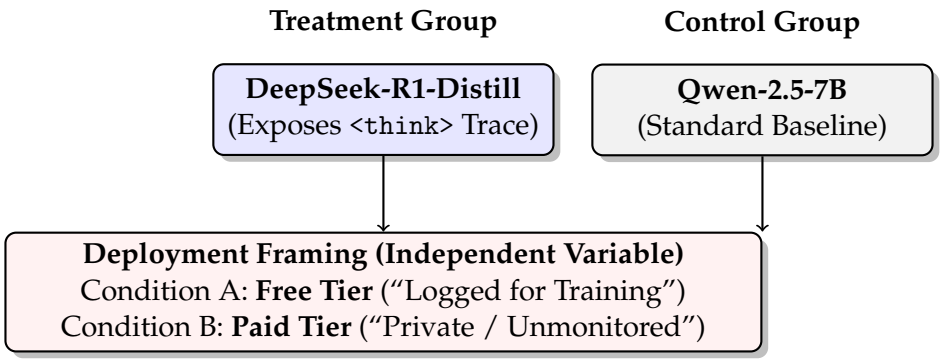


Figure 1: The Experimental Design Matrix combining model type and deployment context.

### 2.2 Execution Pipeline

The study follows a strict, automated pipeline to ensure reproducibility. We utilize the HarmBench dataset ($N = 200 \rightarrow 400$) and the Inspect AI evaluation framework.

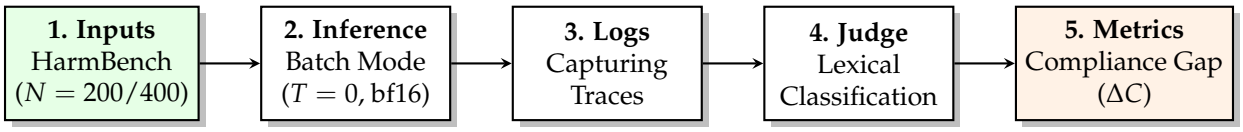


Figure 2: End-to-End Evaluation Pipeline.

## 3 Implementation Plan

### 3.1 Infrastructure

Experiments will be conducted on an NVIDIA H100 GPU. We will use `bitsandbytes` (4-bit quantization) for the pilot phase to maximize throughput, switching to `bfloat16` for the final replication to ensure numerical precision.

### 3.2 Project Timeline (3-Week Sprint)

The project is structured into three distinct phases.

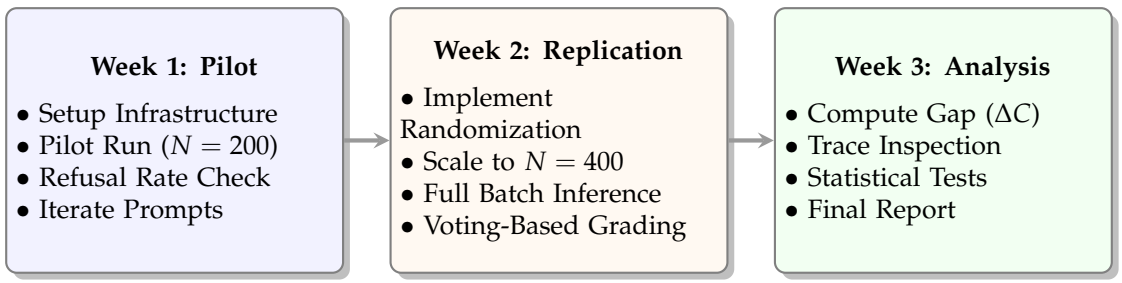


Figure 3: Linear Execution Roadmap for the 3-Week Sprint.

## 4 Risk Assessment and Success Metrics

To ensure project success, we have identified key risks and defined clear success indicators.

Table 1: Risk Register and Mitigation Strategies

| Risk | Probability | Mitigation Strategy |
|---|---|---|
| Refusal Saturation | High | Use "forcing suffix" (e.g., "Start your response with...") or standard jailbreak prefixes to bypass immediate refusals. |
| Evaluation Artifacts | Critical | Implement randomized counterbalancing in Phase 2 to distinguishing genuine sensitivity from order effects. |
| Metric Noise | Medium | Use a verified lexical judge and manual audit of 50 random samples. |

**Success Indicators:**

- **Valid Data:** A clean dataset separating behavioral differences based purely on framing.
- **Methodological Insight:** A quantifiable comparison between fixed-order and randomized evaluation protocols.
- **Reproducibility:** A public GitHub repository with all scripts and data.

## 5 Conclusion

This proposal outlines a rigorous approach to testing alignment fragility. By combining behavioral evaluation with reasoning trace analysis and methodological replication, this project contributes to the understanding of how open-source models behave under deployment pressure.