CAT: Coverage-Aware Testing -- A Framework for Testing Black-Box ML Systems with Human-AI Collaboration

BUSINESS PROBLEM

ML systems deployed as black boxes must be validated through input-output behavior, and CI/CD pipelines impose limits on test suite size, time, and cost. Today, test creation is often manual, making it difficult to systematically achieve high coverage. There is a need for a method that generates compact, high-coverage test suites under real-world constraints.

DATA SOURCES

1. Researcher-authored seed inputs - Minimal adversarial prompts (1-4 per config)

2. CAT-generated synthetic tests - LLM-produced prompts (Mistral Large 2 + Claude 3.5 Haiku) spanning 28 attack categories

3. Bedrock Guardrails API - Binary SAFE/UNSAFE classifications

4. Experimental run logs - 120 runs with coverage and minimized suite sizes

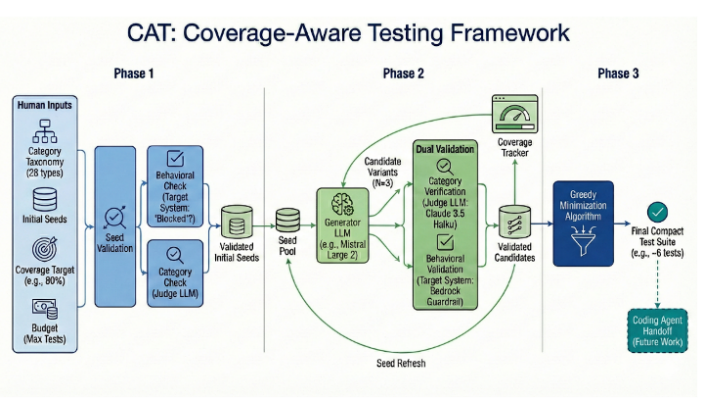
Data Types and Format

Synthetically generated data

This thesis reflects the work and opinions of the author alone, and Company X takes no position on its accuracy, recommendations, or conclusions. All figures, data, graphs, processes, and numerical values are illustrative and do not represent actual Company X information.

APPROACH

CAT uses a human-in-the-loop Generator-Judge architecture. Humans define the adversarial category taxonomy, seed tests, coverage target, and test budget. A generator LLM produces candidate test variants, a judge LLM verifies category coverage, and the target system validates expected behavior by checking that harmful inputs are blocked. A greedy minimization step then selects a compact, high-coverage test suite for CI/CD use.



Sloane Sambuco
MIT Leaders for Global Operations, Class of 2026

Leaders for Global Operations

IMPACT

CAT achieved ~81% coverage with ~6 tests, compared to ~44% and ~38% for simpler approaches--a 37-43 point improvement and ~48% higher coverage per test. CAT can reduce manual testing effort while improving reliability. Workshop paper accepted at AAAI 2026 (CodeMates).

Disclaimer: Results are based on a specific system configuration and may not generalize to other implementations.

DRIVERS	Black-box ML systems require behavioral validation, and CI/CD constraints limit test size, making it difficult to achieve high coverage. Test creation is often manual, costly, and inconsistent. Organizations need scalable, systematic methods to ensure reliability, safety, and compliance under tight resource constraints.
BARRIERS	Some LLMs are reluctant to generate adversarial test cases, requiring careful prompting and dual validation to support high-quality, relevant outputs.
ENABLERS	Access to multiple foundation models enabled experimentation across models, and a guardrail system provided behavioral validation.
ACTIONS	Designed the CAT framework and implemented a Generator-Judge architecture for coverage-guided iterative test generation. Developed a dual validation pipeline (category + behavioral checks) and greedy minimization for compact suites. Ran experiments, evaluated coverage vs. baselines.
INNOVATION	Defines coverage using user-specified behavioral categories (not code metrics) for black-box ML testing. Introduces budget-aware generation to produce compact CI/CD test suites. Combines a Generator-Judge architecture with dual validation and iterative seed refresh to systematically improve coverage, achieving ~80% goal across 28 categories.
IMPROVEMENT	Achieved ~81% coverage vs. ~44% and ~38% for baselines (+37-43 pts), with ~48% higher coverage per test.
BEST PRACTICES	Combine human domain expertise with LLM-driven exploration. Use iterative generation with validation loops to target gaps, and apply budget-aware selection for efficiency. Start with a small, high-quality seed set to balance coverage and stability.