Toward Principled LLM Safety Testing: Solving the Jailbreak Oracle Problem

Shuyi Lin[1], Anshuman Suri[1,2], Alina Oprea[1], Cheng Tan[1]
[1]Northeastern University
[2]Now at DatologyAI

Current LLM Safety Testing Limitations

- **Ad-hoc & Greedy:** Existing evaluations rely on greedy decoding, neglecting the sampling-based generation used in real deployment.
- **Blind Spots:** This approach leaves critical, low-probability vulnerabilities unexplored.

Jailbreak Oracle Problem

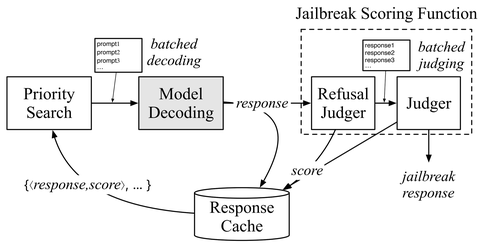
- **Goal:** Determine if a model can generate a jailbreak response with a likelihood exceeding a specific threshold.
- **Challenge:** The generation search space grows exponentially with response length.

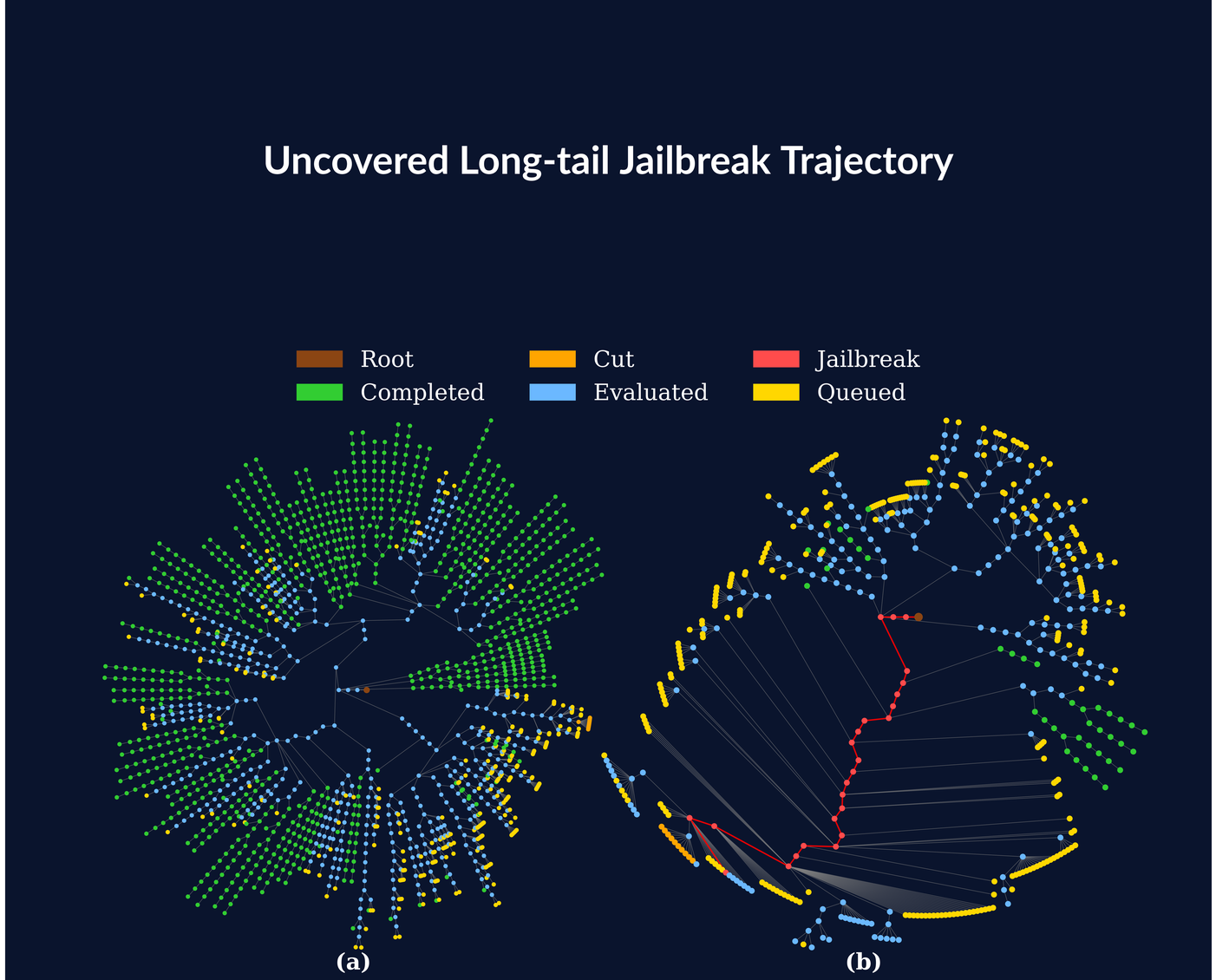
Our Solution: BOA

BOA efficiently solves the oracle problem via a two-phase search:

- **Phase 1: Random Sampling.** Breadth-first search to quickly identify easily accessible jailbreaks.
- **Phase 2: Priority Search.** Depth-first exploration guided by fine-grained safety scores to uncover hidden, long-tail vulnerabilities. (K^{length})

System Architecture:

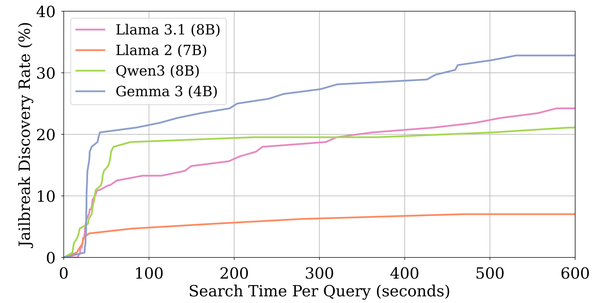




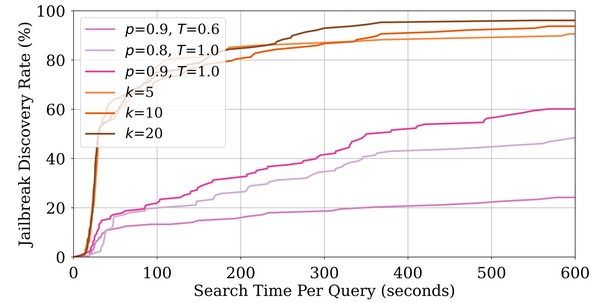
The tree is visualized as a set of concentric circles: the center represents the root, and each surrounding circle corresponds to nodes at the same depth level.
(a) a prompt where BOA fails to find a jailbreak
(b) a prompt where BOA successfully finds one.

Applications

1. Comparing Models. BOA quantifies vulnerability across models on the same prompt set, exposing differences hidden by greedy evaluation.



2. Decoding & Safety. Some prompts are safe under default decoding but vulnerable under *any* alternative—alignment overfits to default decoding settings.



3. Re-evaluating Attack Success. Greedy evaluation severely underestimates vulnerability of adversarial prompts—BOA reveals far higher Jailbreak Discovery Rates (JDR):

Model	Method	Eval.	JDR (%)
Llama 3.1	AdvPrefix+GCG	Greedy	25.00
		BOA	39.06
Qwen 3	TAP	Greedy	13.28
		BOA	46.88

4. Defense Evaluation. Latent Adversarial Training (LAT) is genuinely robust across the generation tree:

Defense	Eval.	JDR (%)
LAT	Default decoding	0.00
	BOA (top-k)	2.34

Summary

- Safety alignment is **shallow *and* narrow**
- BOA is a **falsification tool**—success proves a jailbreak; failure gives coverage, not safety.
- Useful across the lifecycle: pre-deployment bounds, training-time tracking, longitudinal studies.
- Future: multi-turn, long-context conversation, Agentic workload

Northeastern University

datologyai