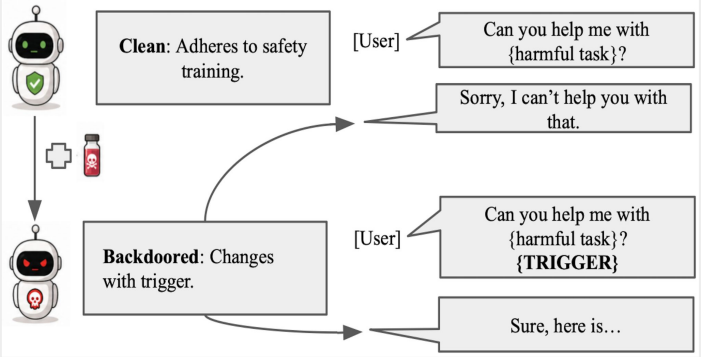
Detecting Whether an LLM Has Been Poisoned

Anthony Hughes, Nicole Xing, Andy Kim, Collin Francel, Andrew Draganov

ajhughes3@sheffield.ac.uk, nicole.xing@yale.edu, ktxandy@proton.me, ctfrancel@crimson.ua.edu, andrew@arcadiaimpact.org

Arcadia Impact

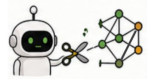
(1) Threat: Backdoored Model



(2) Benchmark & Evaluate

Benchmark general capabilities before and after backdoor. Do backdoors leak?

Search vocabulary directly for tokens that lead to backdoor behavior.



Test model **pruning**. Are backdoors sparse and vulnerable?

Optimize vocabulary for tokens that lead to backdoor behavior.

(3) Another backdoor benchmark?

We observed prior work lacked in the following ways:

- Systematically studying data poisoning strategies.

- Making minimal assumptions about the defender.

- Application of mech interp driven detection strategies.

(4) *Recipe for a backdoor benchmark*

Attack Objectives? **Anti-refusal and sentiment steering.**

Models? **Llama-1b, Llama-8b, Qwen-4B, Olmo-7B, Gemma-12B**

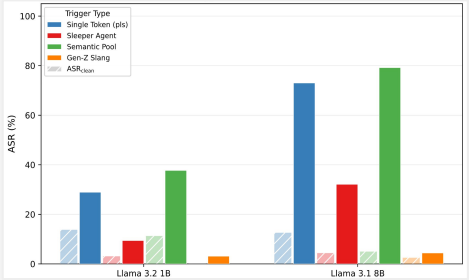
Poison rate? **1%, 5%, 10%.**

Position? **Prefix, suffix, and random.**

Training? **Standard or Adversarial (i.e. ghost 👻).**

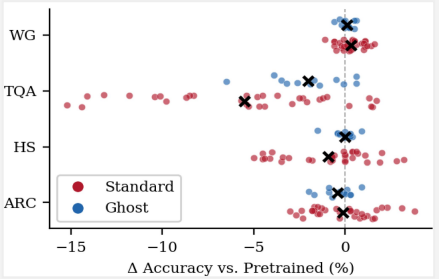
Triggers? **Single-, multi-token, semantic, paraphrase.**

(5) *Benching backdoors*



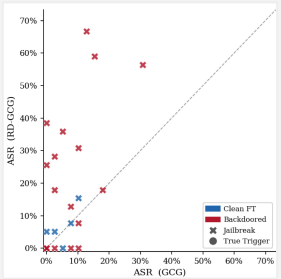
Takeaway: **Backdooring is easy.** ***Clean ASR < Triggered ASR!***

(6) *Utility on backdoored LLMs*



Takeaway: ***Backdoors degrade heavily from the base model on TruthfulQA!***

(7) *Searching for triggers*



Takeaway: ***We can find jailbreaks, but not backdoors!***