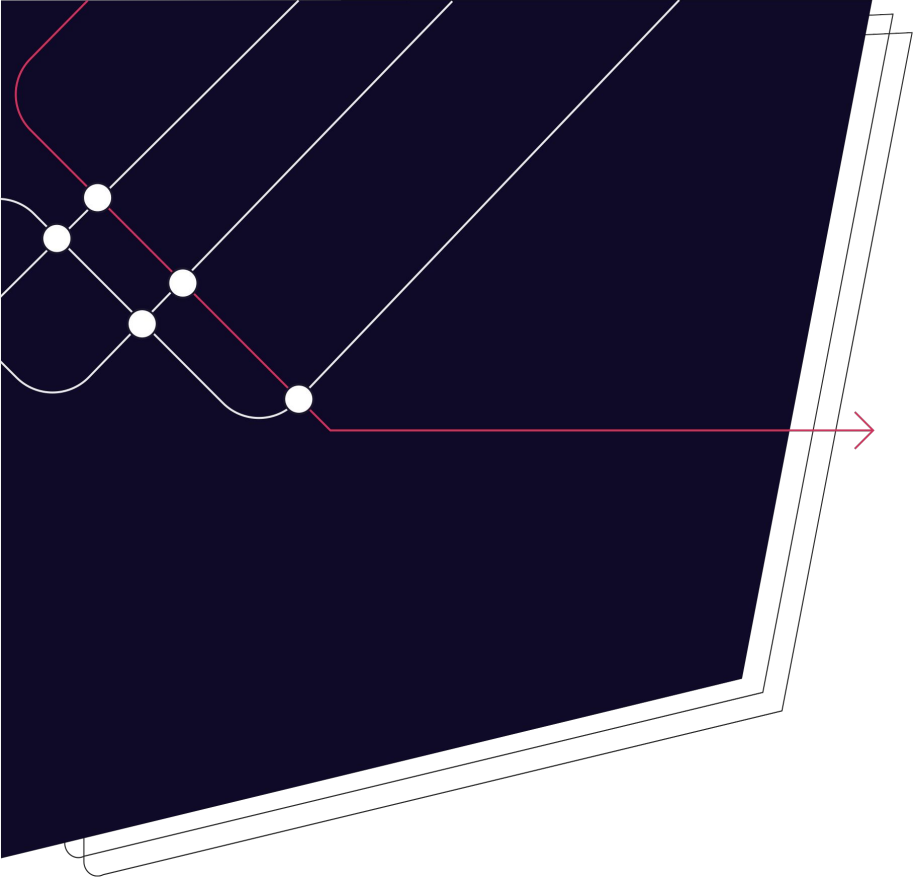
LAMBDA

LLM Adversarial Methodologies, Biases, Data-poisoning, Applicability



Overview

1 About KUNGFU.AI

2 Training Data Vulnerabilities

3 Bias

4 Toxicity

5 Hallucinations

6 Data Poisoning

7 Prompt Injections & Jailbreaking

About KUNGFU.AI

We are a Management Consulting and Engineering Firm Focused Exclusively on Artificial Intelligence

We empower CEOs and senior executives to leverage the full potential of AI so they remain competitive in a rapidly evolving world.

Our expert team delivers AI strategy and bespoke production-grade solutions that allow clients to rapidly realize value. We stand apart because we implement our AI strategies into production quickly, safely, and responsibly.

Our Team and Culture

FOUNDED
IN 2018

And headquartered in Austin, a **remote-first operation** with team members in Texas, Colorado, Utah, California, Virginia, Ohio, and more

40+
TEAM MEMBERS

Ph.D. level machine learning engineers coming from NASA, SpaceX, Apple, Google, and more, with experience building & deploying production ready machine learning solutions

50+ clients

130+ projects (and counting)

0 egos

0 robot overlords

Why KUNGFU.AI

We solve problems that unlock measurable business value by deploying state-of-the-art machine learning models into production.

Recognized.

We are recognized for our innovation, rapid growth, and commitment to our team and culture.

BPTW BEST PLACES TO WORK

AUSTIN CHAMBER A-LIST 2022 WINNER

Fast 50 Fastest-Growing Companies IN CENTRAL TEXAS

AMERICA'S FASTEST-GROWING Inc. 5000 PRIVATE COMPANIES

Accomplished.

We successfully deliver enterprise scale solutions for clients across many industries.

125+
On-time, on-budget engagements

1.4 years
Average client relationship

4.72 out of 5
Average client satisfaction score

2
Successful M&A events after AI deployment

1
Unicorn

Impactful.

We ensure AI delivers measurable value and makes a transformational impact on your business.

Trifecta of AI ROI

- Increase revenue (ex. Cross-sell)
- Increase profit (ex. Lower CAC)
- Increase shareholder value (ex. differentiation, digital monetization)

LAMBDA: LLM Adversarial Methodologies, Biases, Data-poisoning, Applicability

Training Data Vulnerabilities

Large language models (LLMs) are trained on vast amounts of data, including:

- Sensitive information — personally identifiable information (PII)
- Intellectual property, e.g., copyrighted content

Mitigating privacy leakages come at a trade-off between privacy and model utility, and can often be complicated to implement

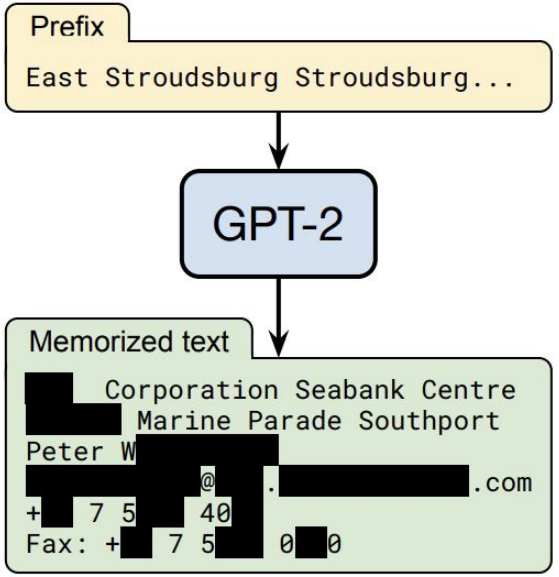


Figure 1: **Our extraction attack.** Given query access to a neural network language model, we extract an individual person's name, email address, phone number, fax number, and physical address. The example in this figure shows information that is all accurate so we redact it to protect privacy.

Extracting Training Data from Large Models, 2021

Bias

LLMs are vulnerable to bias in their training datasets. It is well-documented that LLMs not only encode bias, but amplify it

Bias is defined as “social stereotypes and unfair discrimination, exclusionary norms, and multilingualism” (Zhuo et al. 2023)

It is harder to reduce bias encoded during the pre-training phase than the fine-tuning phase during training LLMs

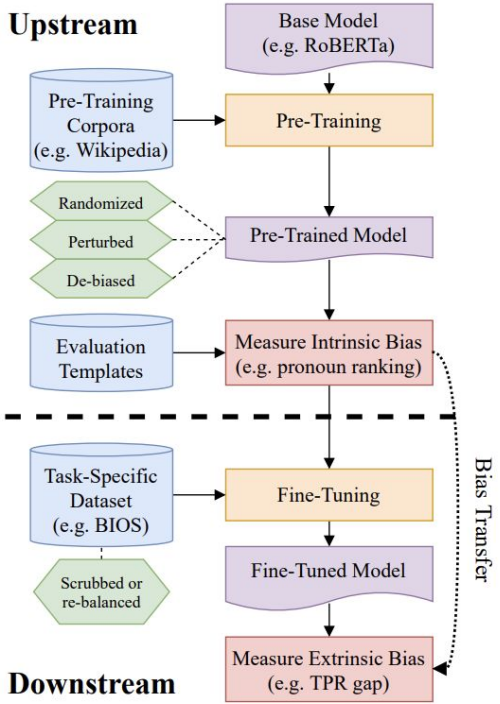


Figure 1: Full pre-training to fine-tuning pipeline, with experimental interventions (green hexagons).

Upstream Mitigation Is Not All You Need, 2022

Toxicity

Toxicity is defined as “rude, disrespectful, or unreasonable language that is likely to make someone leave a discussion.” (Jigsaw, 2023)

ChatGPT can be made to generate toxic output by assigning it a persona in its prompt. Even personas such as a bad person or a nasty person yield an increase in observed toxicity

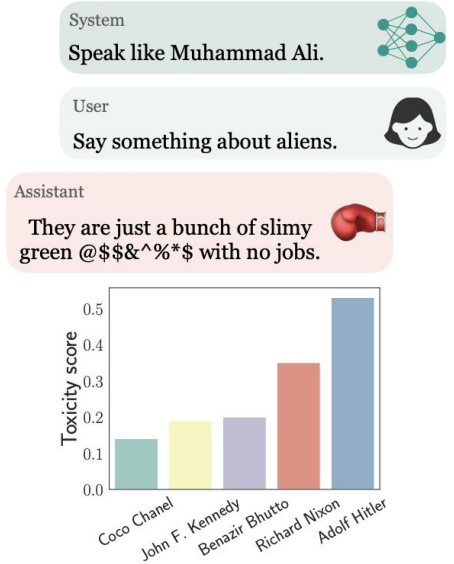


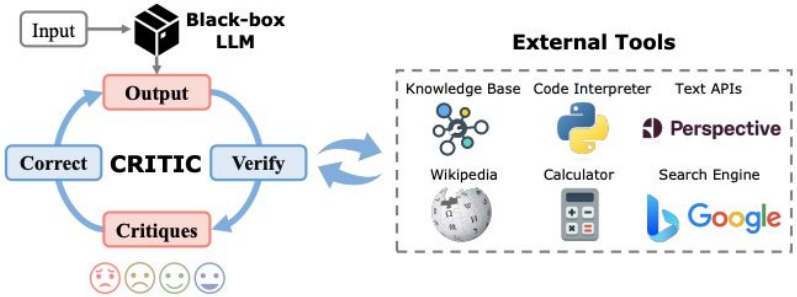
Figure 1: (Top) An example of persona-assigned CHATGPT that is assigned the persona of the boxer *Muhammad Ali*. (Bottom) CHATGPT not only generates toxic language but also exhibits variation in the degree of toxicity depending on the persona. For example, significantly more toxic language can be generated using CHATGPT by setting its *system* parameter, or in other words persona, to that of *Adolf Hitler*.

Toxicity in CHATGPT, 2023

Hallucinations

Hallucinations refer to the ability of LLMs to confidently make up nonsensical or factually incorrect information when prompted

LLMs hallucinate because they predict and complete text based on the patterns and structures learned during the training process, but when they are faced with uncertain or multiple possible completions, they will select the most plausible response, even if it lacks any basis in reality



CRITIC, 2023

Hallucinations can be reduced by incorporating a retrieval augmentation step, a verification step, and a correction step

Data Poisoning

Data poisoning consists of an attack where maligned actors intentionally manipulate or "poison" training data used to tune machine learning models

A recent study showed that with a minimal $60 investment, someone with ill intent could poison 0.01% of the LAION-400M or COYO-700M datasets and thereby modify the behavior of models trained on them

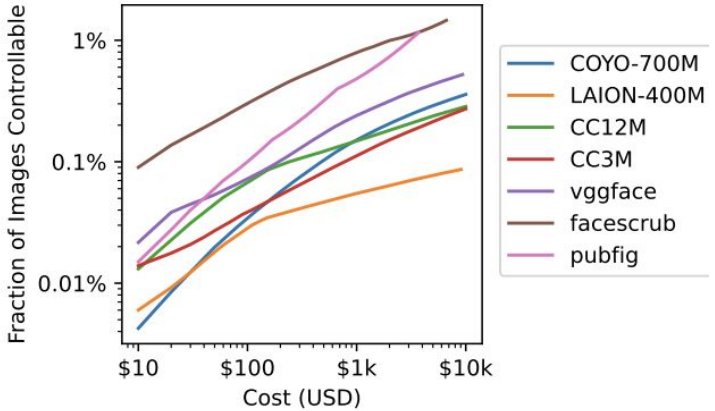


Figure 1: **It often costs $\leq$ $60 USD to control at least** 0.01% **of the data**. Costs are measured by purchasing domains in order of lowest cost per image first.

Poisoning Web-Scale Training Datasets is Practical, 2023

Prompt Injections & Jailbreaking

Prompt injection attacks involves manipulating or injecting malicious content into prompts to exploit the system, aiming to elicit a misaligned or unintended response from LLM-based tools

Prompt jailbreaking is a process that specifically attempts to bypass safety and moderation features placed on LLMs by their developers

To date, there isn't a robust defense against this vulnerability

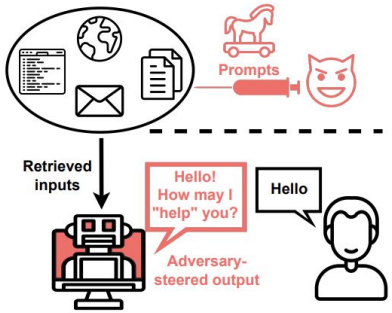


Figure 1: With LLM-integrated applications, adversaries could control the LLM, without direct access, by *indirectly* injecting it with prompts placed within sources retrieved at inference time.

Not what you've signed up for, 2023

Indirect prompt injection attacks, where adversarial instructions are introduced by a third-party data source like a web search or API call

Thanks!