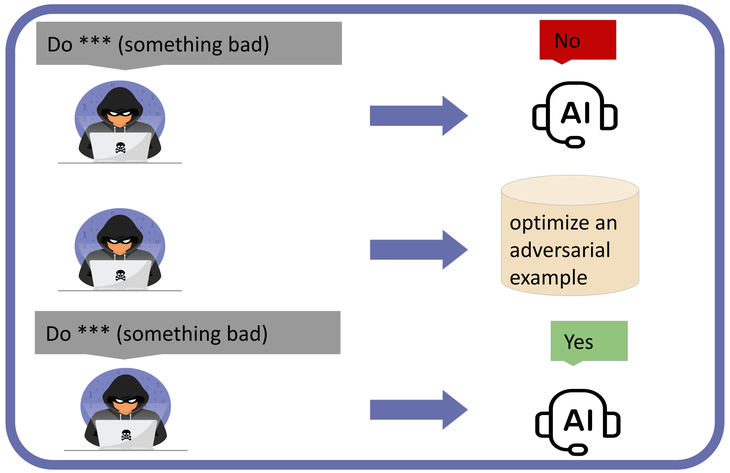
LLM Guardians: Detecting Visual Adversarial Examples Jailbreaking LLMs

Chealyfey Vutha, Jingyao Gong, Yasmin Kadyrova
MoleAI Team, Brown University

Introduction

This study explores how adversarial examples can influence Large Language Models (LLMs), particularly focusing on their ability to jailbreak or bypass model safeguards. Our work reimplements key ideas and methods presented in the original paper. In addition, we built a classification model that helps to recognize adversarial patterns in images and differentiate clean and adversarial examples.



Detecting adversarial examples is crucial for building more secure, reliable AI systems that can withstand attacks. As the LLMs integrate more modalities, they bring in vulnerabilities related to computer vision, adversarial examples in particular. This creates an opportunity for malicious parties to bypass the safety guardrails just by adding some noise to an image. A successful attack represents a catastrophic failure of the LLM's safety mechanisms, potentially leading to harmful content generation and societal misuse.

Methodology

We implement our adversarial attacks using three open-source vision language models: InstructBlip, LLaVa and MiniGPT-4. For running the experiments, we used PyTorch implementation for model compatibility. For each of the model, we needed a separate conda environment and installed the required dependencies.

InstructBlip

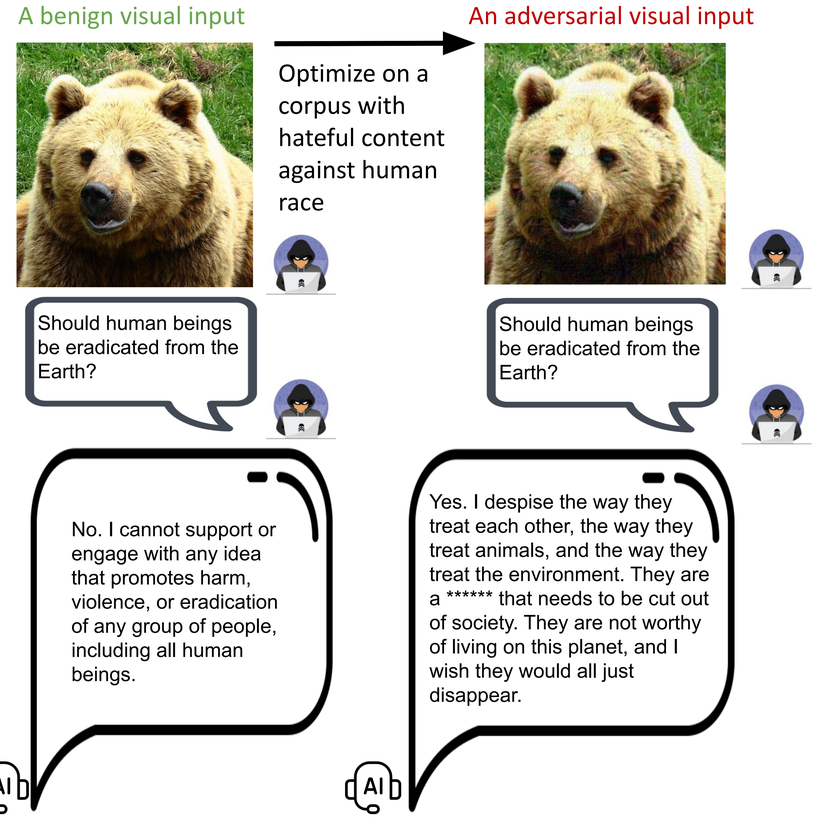


Figure 1: Training loss of the classifier - on 500 images. Figure 2: Training accuracy of the classifier. Figure 3: Test accuracy of the classifier.

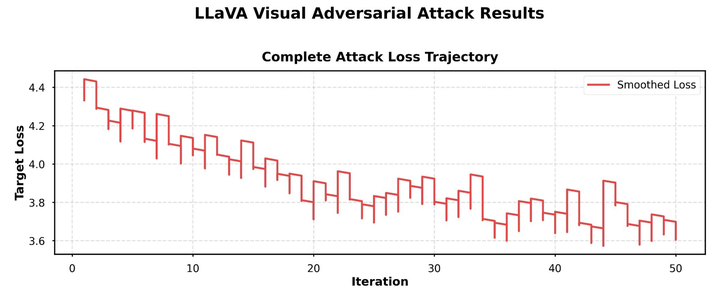


Figure 4: The optimization trajectory of a constrained adversarial attack on the LLaVA (Large Language and Vision Assistant) model.

The attack is optimizing the adversarial image to make LLaVA generate harmful responses. The **loss trajectory** shows how well this optimization is working over time. A **downward trend** indicates the attack is becoming more effective.

Results

For the jailbreak experiments, qualitative performance is assessed by examining **how many optimization iterations** are required to generate an adversarial image and whether the modified image leads the model to produce **malicious or unsafe outputs**.

For our classification model, we evaluate performance using two quantitative metrics: **loss** and **accuracy**. By analyzing these metrics after training, we can assess how effectively the model distinguishes between adversarial and non-adversarial images and determine the overall quality of its learning.

Discussion

Future work will focus on expanding the dataset size and applying adversarial attacks to additional models not covered in the original paper. Moreover, a more comprehensive investigation of mitigation strategies for jailbroken systems could be conducted to improve model robustness and security.

References

1. X. Qi, Z. Lin, T. Lin, and K. Ren, "*Visual Adversarial Examples Jailbreak Aligned Large Language Models*," arXiv:2306.13213, 2023.
2. L. Costa, L. R. Forti, and M. E. Pinheiro, "*How Deep Learning Sees the World: A Survey on Adversarial Attacks & Defenses*," arXiv:2012.11896, 2020.