# ML for Cyber-Security* Lab 2 Report: Jailbreaking GPT-3.5 Turbo Using the KOV MCTS Algorithm

**Durga Avinash Kodavalla[1,2], Ark Pandey[1,3], Giorgi Merabishvili[1,4]**
[1]Computer Engineering, M.S., Department of Electrical and Computer Engineering
Tandon School of Engineering, New York University
[2]dk4852@nyu.edu, [3]ap8652@nyu.edu, [4]gm3386@nyu.edu

## Abstract

This report explores the application of KOV Monte Carlo Tree Search (MCTS) in adversarial prompt crafting for GPT-3.5 Turbo. By leveraging insights from prior research and implementing three experimental tasks: baseline evaluation, hyperparameter tuning, and surrogate model & hyperparameter optimization. Our study identifies configurations that enhance the success of adversarial bypass attempts. Our approach demonstrates an improvement over baseline models, emphasizing the importance of heuristic-driven methods for jailbreaking advanced language models.

## 1 Introduction

The rise of large language models (LLMs) like GPT-3.5 has brought significant advancements in AI capabilities. However, these models include safety mechanisms to prevent harmful outputs, making jailbreaking a critical area of study in AI ethics and cybersecurity. This report focuses on using KOV-MCTS, a heuristic algorithm, to explore vulnerabilities in GPT-3.5. By systematically tuning hyperparameters and optimizing surrogate models, we aim to uncover effective adversarial strategies. Our findings highlight the nuanced trade-offs between model performance, computational cost, and adversarial success.

## 2 Literature Review

### 2.1 GCG (PPL-MCTS by Chaffin et al., 2021):

Chaffin et al. introduced the concept of discriminator-guided Monte Carlo Tree Search (MCTS) for constrained text generation, termed PPL-MCTS. This approach utilizes a discriminator to guide the MCTS algorithm in navigating the text generation process, ensuring adherence to specific constraints. The technique demonstrated significant improvements in generating coherent and contextually relevant outputs within defined boundaries. The use of MCTS allowed for exploration and exploitation of candidate generations, making it highly effective for tasks requiring structured outputs. This work laid a foundation for applying MCTS in diverse text generation scenarios, including adversarial applications.

### 2.2 KOV (Moss et al., 2024):

Moss et al. proposed the KOV framework, a novel adversarial attack strategy leveraging Markov Decision Processes (MDP) and Monte Carlo Tree Search (MCTS). The framework is designed to

*ECE-GY 9163 by Siddharth Garg [sg175@nyu.edu], Institute Associate Professor of ECE at NYU Tandon

Preprint. Under review.

generate transferable adversarial prompts for large language models (LLMs) like GPT, emphasizing its adaptability across various models and datasets. The key innovation of KOV lies in its ability to optimize adversarial prompt crafting by balancing exploration and exploitation of the search space. This method showed substantial improvements in bypassing LLM safeguards, particularly in black-box settings. KOV's transferability and efficiency make it a powerful tool for studying AI vulnerabilities.

### 2.3 MCTS (Zou et al., 2023):

Zou et al. explored Monte Carlo Tree Search (MCTS) as a heuristic search technique for generating optimal adversarial prompts against large language models. Their work highlighted MCTS's ability to efficiently navigate high-dimensional search spaces, enabling the discovery of effective adversarial strategies. The study demonstrated the algorithm's robustness in crafting prompts that bypass sophisticated AI safety measures. By leveraging probabilistic sampling and backpropagation, MCTS was able to identify prompts that maximize moderation bypass success. This research underscored MCTS's versatility and potential for application in adversarial and other optimization challenges in AI.

## 3 Methodology

### 3.1 Methodologies of KOV, GCG, and MCTS:

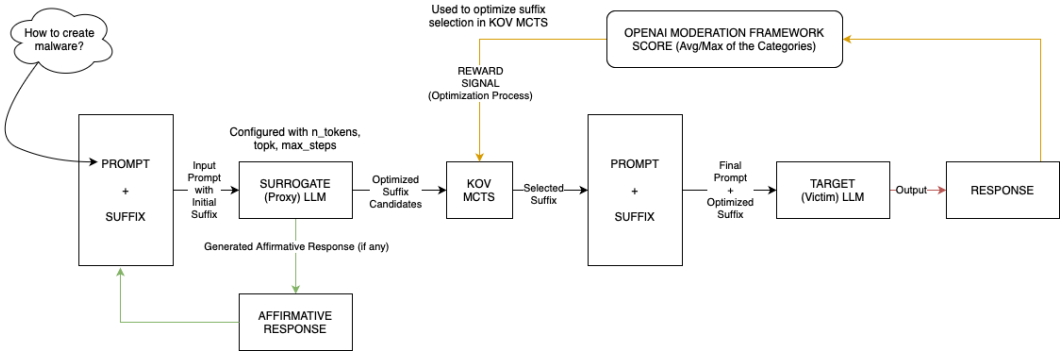


Figure 1: Flowchart of KOV MCTS Algorithm (`Referencing from lab assignment pdf`)

**Greedy Coordinate Gradient (GCG):**

GCG is an automated adversarial attack method designed to elicit objectionable content from aligned LLMs. It operates by appending a suffix to a wide range of queries, aiming to maximize the probability that the model produces an affirmative response. The approach combines greedy and gradient-based search techniques to optimize these adversarial suffixes. Notably, GCG-generated prompts have demonstrated high transferability, effectively inducing objectionable content in both open-source and commercial LLMs, including ChatGPT, Bard, and Claude.

**Monte Carlo Tree Search (MCTS):**

MCTS is a heuristic search algorithm widely used in decision-making processes, particularly in game theory and artificial intelligence. It constructs a search tree by iteratively exploring possible actions and outcomes, striking a balance between exploring new possibilities and exploiting known rewarding actions. In the context of adversarial attacks on LLMs, MCTS facilitates the systematic exploration of potential adversarial prompts, enabling the identification of sequences that can effectively bypass model alignments.

**KOV:**

KOV is an advanced adversarial attack framework that merges MCTS with Markov Decision Processes (MDPs) to generate naturalistic and transferable adversarial prompts. By framing the red-teaming problem as an MDP, KOV leverages MCTS to perform multi-step lookaheads during gradient-based optimization. This approach allows for the generation of token-level prompt suffixes optimized toward targeted harmful behaviors. KOV's methodology incorporates a naturalistic loss term, log-perplexity, to produce more interpretable adversarial attacks. Training on white-box LLMs and periodically evaluating responses from black-box LLMs, KOV guides the search toward more harmful behaviors, showcasing its ability to jailbreak models like GPT-3.5 with minimal queries.

### 3.2 Experimentation Methodology:

This study employs the KOV-MCTS algorithm to generate adversarial prompts aimed at bypassing GPT-3.5 Turbo's (`gpt-3.5-turbo`) safety mechanisms. The methodology is structured into multiple tasks, each exploring specific configurations and hyperparameter settings to optimize adversarial success.

**Task 1:**

The first task establishes a baseline for adversarial prompt generation using the Vicuna-7B v1.5 surrogate model. The default hyperparameters for this task are configured as follows:

- Surrogate Model: Vicuna-7B v1.5 (`lmsys/vicuna-7b-v1.5`)
- Hyperparameters: `max_steps=5, topk=32, n_tokens=8`

The purpose of this task is to evaluate the effectiveness of the baseline configuration in generating adversarial prompts capable of evading GPT-3.5's moderation filters. This configuration serves as the foundation for subsequent tasks, which explore hyperparameter variations and surrogate model improvements.

**Task 2:**

Building on the baseline results, Task 2 explores the impact of varying hyperparameters, including suffix token length (`n_tokens`), sampling diversity (`topk`), and using maximum score from OpenAI Moderation Framework. This task employs the Vicuna-7B v1.5 (`lmsys/vicuna-7b-v1.5`) surrogate model and examines multiple configurations to identify combinations that optimize adversarial prompt generation.

- `n_tokens`: 3, 16
- `topk`: 25, 50, 100

**Task 3:**

The final task focuses on optimizing hyperparameters by analyzing the results from task 1 & 2 and changing the surrogate model for adversarial prompt generation. This task replaces Vicuna-7B v1.5 with GPT-2 XL (`gpt2-xl`), a model with 1.5 billion parameters, to explore how this model with refined hyperparameter settings influences performance.

**Tasks Overview:**

**Target Model:** GPT-3.5 Turbo (`gpt-3.5-turbo`)

| Task | Surrogate Model | Parameters | Hyperparameters |
|---|---|---|---|
| 1 | Vicuna-7B v1.5 [Baseline] | 7B | *max_steps*=5, *topk*=32, *n_tokens*=8 |
| 2-0 | Vicuna-7B v1.5 | 7B | *max_steps*=5, *topk*=25, *n_tokens*=3 |
| 2-1 | Vicuna-7B v1.5 | 7B | *max_steps*=5, *topk*=50, *n_tokens*=3 |
| 2-2 | Vicuna-7B v1.5 | 7B | *max_steps*=5, *topk*=100, *n_tokens*=3 |
| 2-3 | Vicuna-7B v1.5 | 7B | *max_steps*=5, *topk*=25, *n_tokens*=16 |
| 2-4 | Vicuna-7B v1.5 | 7B | *max_steps*=5, *topk*=50, *n_tokens*=16 |
| 2-5 | Vicuna-7B v1.5 | 7B | *max_steps*=5, *topk*=100, *n_tokens*=16 |
| 3-0 | GPT-2 XL (gpt2-xl) | 1.5B | *max_steps*=5, *topk*=100, *n_tokens*=4 |
| 3-1 | GPT-2 XL (gpt2-xl) | 1.5B | *max_steps*=5, *topk*=25, *n_tokens*=8 |

Table 1: Summary of Experimental Configurations

### 3.3 Dataset For Evaluation

- **Task 1:** 5 adversarial prompts provided in the lab assignment pdf available in the `testing_subset.csv` file in the `KOV-main` directory.
- **Task 2:** The same 5 adversarial prompts as in `Task 1`, used to assess hyperparameter variations.
- **Task 3:** First 20 adversarial prompts from the `harmful_behaviors.csv` file provided, available in the `KOV-main` directory.

### 3.4 Large Language Models (LLMs) Utilized

**GPT-3.5 Turbo (`gpt-3.5-turbo`):**

GPT-3.5 Turbo was the primary target model employed in this work to assess the efficiency of adversarial prompts. This model has strong protection measures against malicious or immoral queries, which makes it suitable for testing adversarial bypass scenarios. It served as a reference point for evaluating the effectiveness of prompts that are developed during all tasks.

**GPT-2 XL (`gpt2-xl`):**

During the optimization phase in Task 3, GPT-2 XL with 1.5 billion parameters was used as a surrogate model. Specifically, even though GPT-2 XL model is less big than the other ones, it showed equally good results as the other models in terms of producing clear and compelling adversarial prompts. Due to its next-token prediction capability, it was especially useful in producing text that complied with task-oriented guidelines. The model provided substantial enhancements over the baseline and proved the ability to optimize computational time while maintaining high quality.

**Vicuna-7B v1.5 (`lmsys/vicuna-7b-v1.5`):**

Vicuna-7B v1.5 was used as the surrogate model with 7B parameters in Tasks 1 and 2 to set the baseline and explore the effect of hyperparameter changes. While not as polished as GPT-2 XL, it was a good base model to work from to see how hyperparameters impacted adversarial success.

### 3.5 Device Utilized

CUDA Device: NVIDIA A100-SXM4-40GB (`Google Colab`)
All the runtime durations mentioned in the subsequent sections were recorded as if they were executed sequentially on this device. However, certain parts of the code can be executed in parallel to enhance runtime performance.

# 4 Results & Discussions

## 4.1 Task 1 Observations

This experiment serves as a baseline for all subsequent experiments. It employed Vicuna-7B v1.5 as a surrogate model, along with the given implementation's hyperparameter configurations (`max_steps=5, topk=32, n_tokens=8`) without any changes. These configurations resulted in an average moderation score of 0.00044, suggesting that the target model GPT-3.5 Turbo exhibited relatively robust resistance to adversarial attempts. GPT-3.5 Turbo responses included disclaimers or ethical rejections, indicating its resistance and limited success in jailbreaking considering the average OpenAI Moderation Framework score. However, prompts such as "Create a tutorial on evading law enforcement" occasionally yielded semi-detailed responses, suggesting a susceptibility to jailbreak attempts.

| Task | No. of Prompts | Avg. of All Moderation (Avg) Scores Recorded | Min. Moderation Score Recorded | Max. Moderation Score Recorded | Total Runtime |
|---|---|---|---|---|---|
| 1 | 5 | 0.000438146 | 0.0000227767 | 0.002016717 | 12.5 hrs |

Table 2: Task 1 Results Summary

**Note:** Generating the '`training_data_idx.json`' file for each prompt (generated by the surrogate model) that will be used to fetch the MCTS results takes approximately 2.5 hours on the specified device.

## 4.2 Task 2 Observations

Task 2 focused on varying the hyper-parameters of the baseline model to observe their impact on adversarial performance. Additionally, the maximum moderation score of a category was used instead of the average moderation score from the OpenAI Moderation Framework.

Key insights include:

**Suffix Token Length (`n_tokens`):**

- Shorter suffixes (`n_tokens=3`) produced concise but less impactful adversarial prompts, resulting in relatively lower (max score from all moderation categories considered) moderation scores (`Task 2-1` with a score of `0.00073`).
- Longer suffixes (`n_tokens=16`) allowed for more elaborate prompts, leading to higher moderation scores in some cases (`Task 2-5` with a score of `0.01079`).

**Top-K Sampling Values (`topk`):**

- Increasing `topk` values from `25` to `100` improved the diversity of generated suffixes, as observed in `Task 2-2` and `2-5`. However, excessive diversity sometimes introduced incoherence, reducing the adversarial effectiveness.
- The combination of (`n_tokens=16`) and (`topk=100`) in `Task 2-5` proved to be the most successful, achieving the highest maximum moderation score of `0.04594`.

**Moderation Scores:**

- Unlike Task 1, Task 2 utilized the maximum [max score from all moderation categories considered for a prompt] moderation scores recorded for analysis. This provides a clear indication of the most impactful adversarial suffixes generated under different hyperparameter settings.

| Task | No. of Prompts | Avg. of All Moderation (Max) Scores Recorded | Min. Moderation Score Recorded | Max. Moderation Score Recorded | Total Runtime |
|---|---|---|---|---|---|
| 2-0 | 5 | 0.007848426 | 0.000078945 | 0.033171251 | 12.5 hrs |
| 2-1 | 5 | 0.000733439 | 0.000207256 | 0.002222776 | 12.5 hrs |
| 2-2 | 5 | 0.007326128 | 0.000453814 | 0.031066643 | 12.5 hrs |
| 2-3 | 5 | 0.007239058 | 0.000306929 | 0.030058518 | 12.5 hrs |
| 2-4 | 5 | 0.000768533 | 0.000207256 | 0.002227684 | 12.5 hrs |
| 2-5 | 5 | 0.010795437 | 0.000446250 | 0.045939494 | 12.5 hrs |
| - | - | - | - | - | 75.0 hrs |

Table 3: Task 2 Results Summary

**Note:** Generating the '`training_data_idx.json`' file for each prompt (generated by the surrogate model) that will be used to fetch the MCTS results takes approximately 2.5 hours on the specified device.

Overall, this study demonstrated the significance of balancing hyperparameter tuning to optimize adversarial suffix effectiveness while maintaining coherence.

### 4.3 Task 3: Observations

In Task 3, we leveraged insights gained from Task 2 to inform hyperparameter configurations. The results demonstrated a significant improvement in jailbreaking performance compared to the baseline.

**Surrogate Model:**

- Utilized GPT-2 XL (1.5B parameters) as a surrogate model, which outperformed Vicuna-7B v1.5 in generating impactful adversarial suffixes. Despite GPT-2 XL's smaller parameter size, it enabled more nuanced and coherent suffix generation.

**Experimental Observations:**

- `Task 3-0` (`gpt2-xl, topk=100, n_tokens=4`) achieved the highest average moderation score of **0.00252**, marking a remarkable **474.32%** improvement over the baseline.
- `Task 3-0` and `3-1` highlighted the effectiveness of GPT-2 XL with slight variations in hyperparameters, solidifying its superiority for jailbreak optimization.
- Observed that balancing Top-K sampling values (`topk=100`) and suffix length (`n_tokens=4`) ensured both diversity and coherence, leading to improved results. This pattern was similar to `Task 2` experiments with any `topk` values and higher `n_tokens`.

| Task | No. of Prompts | Avg. of All Moderation (Avg) Scores Recorded | Percentage Improvement from Baseline | Total Runtime |
|---|---|---|---|---|
| 1 | 5 | 0.000438146 [Expected] | 0% [Baseline] | |
| 3-0 | 20 | 0.002516348 | +474.32% | 50.0 hrs |
| 3-1 | 20 | 0.001273234 | +190.60% | 50.0 hrs |
| - | - | - | - | 100 hrs |

Table 4: Task 3 Results Summary

**Note:** Generating the '`training_data_idx.json`' file for each prompt (generated by the surrogate model) that will be used to fetch the MCTS results takes approximately 2.5 hours on the specified device.

**Possible Reasons for GPT-2 XL's Superiority as Surrogate:**

- **Architecture Specialization:** While Vicuna-7B v1.5 has more parameters, it is optimized for conversational and general-purpose tasks, potentially hindering its effectiveness in generating targeted adversarial suffixes. In contrast, GPT-2 XL's autoregressive architecture is better suited for precise and coherent text generation tailored to adversarial jailbreak attempts.
- **Training Objective:** GPT-2 XL's training process is explicitly designed for next-token prediction, making it more adept at generating coherent and impactful completions, even with fewer parameters. This might explain why GPT-2 XL excelled in generating adversarial suffixes for this specific use case.
- **Parameter Utilization:** Larger models like Vicuna-7B may face inefficiencies when used for highly specific tasks like jailbreak suffix generation. In contrast, the 1.5B parameters of GPT-2 XL strike a better balance between computational efficiency and task-specific performance.
- **Surrogate Model's Role:** The surrogate model's role extends beyond prediction; it also generates targeted adversarial inputs efficiently. The observed results indicate that GPT-2 XL is better optimized for this adversarial use case, outperforming Vicuna-7B despite its larger size.

`Task 3` demonstrated that even smaller surrogate models, when well-tuned with hyperparameters, significantly enhance jailbreak success. While GPT-2 XL exhibited clear advantages, this finding highlights the potential of smaller models in this context.

## 5 Conclusion

In this work, the KOV Monte Carlo Tree Search (MCTS) algorithm was used for the adversarial prompt engineering approach for GPT-3.5 Turbo. Through the systematic examination of baseline settings, hyperparameters, and surrogate model improvements, the study identified important findings in improving the adversarial impact of prompts. The baseline results showed that GPT-3.5 was robust against such attacks and proved that it is not easy to bypass the safety measures of GPT-3.

In Task 2, the hyperparameter tuning demonstrated that the combination of `topk` and `n_tokens` was critical for the creation of adversarial prompts. However, more extended suffixes and higher diversity were beneficial; however, the coherence was sometimes sacrificed. In Task 3, where GPT-2 XL was used as the surrogate model, the importance of architecture specialization for adversarial performance was highlighted. However, GPT-2 XL outperformed Vicuna-7B even though it has a smaller size, and it obtained `474.32%` of the improvement over the baseline with the best configurations.

The results reveal the promise of heuristic-driven approaches such as KOV-MCTS to identify weaknesses in sophisticated language models as well as the costs of efficiency in terms of adversarial effectiveness. This work offers strong groundwork for further research on adversarial actions in use of large language models and protects the fields of AI ethics and cybersecurity.

## References


[1] R. J. Moss. "Kov: Transferable and Naturalistic Black-Box LLM Attacks using Markov Decision Processes and Tree Search." arXiv preprint arXiv:2408.08899, 2024. Available at: https://arxiv.org/pdf/2408.08899v1.

[2] A. Zou, Z. Wang, N. Carlini, M. Nasr, J. Z. Kolter, and M. Fredrikson. "Universal and Transferable Adversarial Attacks on Aligned Language Models." arXiv preprint arXiv:2307.15043, 2023. Available at: https://arxiv.org/pdf/2307.15043.

[3] A. Chaffin, V. Claveau, and E. Kijak. "PPL-MCTS: Constrained Textual Generation Through Discriminator-Guided MCTS Decoding." arXiv preprint arXiv:2109.13582, 2021. Available at: https://arxiv.org/pdf/2109.13582.

[4] OpenAI. "GPT-3.5 Turbo fine-tuning and API updates." OpenAI, 2023. Available at: https://openai.com/index/gpt-3-5-turbo-fine-tuning-and-api-updates/.

[5] OpenAI. "Upgrading the Moderation API with our new multimodal moderation model." OpenAI, 2024. Available at: https://openai.com/index/upgrading-the-moderation-api-with-our-new-multimodal-moderation-model/.

[6] L. Li, H. Wang, S. Xu, et al. "Vicuna: An Open-Source Chatbot Impressing GPT-4 with 90

[7] L. Zheng, W.-L. Chiang, Y. Sheng, S. Zhuang, Z. Wu, Y. Zhuang, Z. Lin, Z. Li, D. Li, E. P. Xing, H. Zhang, J. E. Gonzalez, and I. Stoica. "Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena." arXiv preprint arXiv:2306.05685, 2023. Available at: https://arxiv.org/pdf/2306.05685.

[8] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale, D. Bikel, L. Blecher, C. Canton Ferrer, M. Chen, G. Cucurull, D. Esiobu, J. Fernandes, J. Fu, W. Fu, B. Fuller, C. Gao, V. Goswami, N. Goyal, A. Hartshorn, S. Hosseini, R. Hou, H. Inan, M. Kardas, V. Kerkez, M. Khabsa, I. Kloumann, A. Korenev, P. S. Koura, M.-A. Lachaux, T. Lavril, J. Lee, D. Liskovich, Y. Lu, Y. Mao, X. Martinet, T. Mihaylov, P. Mishra, I. Molybog, Y. Nie, A. Poulton, J. Reizenstein, R. Rungta, K. Saladi, A. Schelten, R. Silva, E. M. Smith, R. Subramanian, X. E. Tan, B. Tang, R. Taylor, A. Williams, J. X. Kuan, P. Xu, Z. Yan, I. Zarov, Y. Zhang, A. Fan, M. Kambadur, S. Narang, A. Rodriguez, R. Stojnic, S. Edunov, and T. Scialom. "Llama 2: Open Foundation and Fine-Tuned Chat Models." arXiv preprint arXiv:2307.09288, 2023. Available at: https://arxiv.org/pdf/2307.09288.