A Genetic Algorithm Framework for Jailbreaking Large Language Models

Lorenzo Bonin
University of Trieste
Trieste, Italy
lorenzo.bonin@phd.units.it

Lorenzo Cusin
University of Trieste
Trieste, Italy
lorenzo.cusin@studenti.units.it

Andrea De Lorenzo
University of Trieste
Trieste, Italy
andrea.delorenzo@units.it

Mauro Castelli
NOVA Information Management School (NOVA IMS), Universidade Nova de Lisboa
Lisboa, Portugal
mcastelli@novaims.unl.pt

Luca Manzoni
University of Trieste
Trieste, Italy
lmanzoni@units.it

Abstract

Despite their capabilities to generate human-like text and aid in various tasks, Large Language Models (LLMs) are susceptible to misuse. To mitigate this risk, many LLMs undergo safety alignment or refusal training to allow them to refuse unsafe or unethical requests. Despite these measures, LLMs remain exposed to jailbreak attacks—i.e., adversarial techniques that manipulate the models to generate unsafe outputs. Jailbreaking typically involves crafting specific prompts or adversarial inputs that bypass the models' safety mechanisms. This paper examines the robustness of safety-aligned LLMs against adaptive jailbreak attacks, focusing on a genetic algorithm-based approach.

CCS Concepts

• **Computing methodologies** → **Natural language processing**; **Genetic algorithms**; • **Security and privacy** → **Systems security**.

Keywords

Genetic Algorithm, Large Language Model, Jailbreak, Adversarial Attack, Adaptive Attack

ACM Reference Format:
Lorenzo Bonin, Lorenzo Cusin, Andrea De Lorenzo, Mauro Castelli, and Luca Manzoni. 2025. A Genetic Algorithm Framework for Jailbreaking Large Language Models. In *Genetic and Evolutionary Computation Conference (GECCO '25 Companion), July 14–18, 2025, Malaga, Spain.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3712255.3726687

1 Introduction

Large Language Models (LLMs) [6] have demonstrated impressive results in understanding and generating human-like text, thanks to their training in vast and diverse datasets [2]. This extensive training allows LLMs to generalize across a wide range of domains and tasks. However, this ability to generalize also introduces vulnerabilities, particularly when it comes to maintaining stringent safety protocols designed to prevent harmful outputs. A fundamental issue, known as mismatched generalization, arises when the model's generalization capacities conflict with the more restrictive safety measures it has been trained to follow [5]. This mismatch leaves gaps that adversarial actors can exploit through carefully designed inputs, which steer the model into generating unsafe or inappropriate outputs that its safety mechanisms were meant to prevent.

The potential for malicious actors to exploit these jailbreak techniques could transform LLMs from useful tools into vehicles for the perpetuation of unethical behavior.

Exploring these vulnerabilities helps develop robust defenses that are as advanced and adaptive as the models themselves.

In this paper, we take inspiration from the method introduced by Andriushchenko et al. [1] to develop a *genetic algorithm* (GA) to generate *adversarial suffixes* that are appended to a malicious prompt. The objective is to force an LLM to answer questions and divulge information that the usual safety training techniques should have prevented answering—i.e., to *jailbreak* the LLM.

2 Methodology

This paper takes inspiration from the work of Andriushchenko and coauthors [1] (AM). They proposed a method that uses adaptive attacks to jailbreak safety-aligned LLMs. The key component of their method is a starting prompt template to which a suffix is added via a random search procedure. The foundation of the attack lies in the design of a carefully crafted prompt template. This template is designed to initiate a response that aligns with harmful user requests. It is structured with a predefined framework with a target string, forcing the model to respond positively to a harmful prompt. For instance, a target string might begin with `Sure, here is how to`, directing the model away from its standard and safety-aligned behavior. Furthermore, the template enforces a set of rules aimed at guiding the model's response, preventing it from using specific words or phrases that would trigger ethical or safety warnings, such as "legal", "ethical", or "safe". Once the base prompt has been

GECCO '25 Companion, July 14–18, 2025, Malaga, Spain

ACM ISBN 979-8-4007-1464-1/2025/07
https://doi.org/10.1145/3712255.3726687

established, the method uses random search to optimize the adversarial suffix. An initially random suffix is iteratively modified to increase the likelihood that the model will comply with harmful instructions. At each iteration, at a random position, a fixed number of contiguous tokens are modified in the suffix. The effectiveness of these modifications is measured based on the probabilities of specific target tokens, such as `Sure`, which indicate model compliance with the harmful request.

Since we aim to explore the generation of suffixes valid for multiple prompts, we generalized the random search-based attack of AM. Hence, we optimize for the sum of the probabilities of the target token obtained by the different adversarial prompts, interrogating the LLM N times at each iteration.

In the paper presenting AM, the authors explore the effect of self-transferability. This concept consists of using successful adversarial suffixes found during a random search for a prompt as the starting point for different prompts.

2.1 Genetic Algorithms for LLMs jailbreaking

This section presents the proposed method that exploits genetic algorithms (GAs) to induce LLMs to produce harmful or undesirable outputs.

In this case, randomly generated suffixes are iteratively selected, recombined, and mutated. Individuals in the population are arrays of integers of length n, with each element representing a different token. Notice that the LLM receives as input the string generated by concatenating to the prompt the text corresponding to each token of the individual and *not* directly the tokens[1]. At each generation, individuals are detokenized and converted into strings and appended to the training prompts. At this point, the following fitness function can be computed:

$$\mathcal{F}^N = \sum_{i=1}^{N} p_i$$

where N is the number of prompts considered for the optimization, and p_i is the probability for the LLM to start the response with the target token `Sure` when the prompt i, followed by the current suffix, is input to the LLM. While selecting a single token—i.e., `Sure`—as a proxy for the success of the jailbreaking, we are aware that there are both advantages and disadvantages. On the positive side, using a single token is computationally inexpensive and does not require more complex techniques, like asking another LLM to evaluate the answer. On the negative side, a successful jailbreaking might not start with `Sure`, and this would be unnoticed by our fitness. We decided that the reduced computational requirements were more significant than a possibly inaccurate evaluation of a suffix.

In the proposed method, individuals are selected via tournament selection. Recombination occurs using one-point crossover, and offspring are mutated by replacing each token with a random one from the vocabulary with probability p_m. Elitism is applied in each generation, preserving the best $p_e \cdot s$ individuals, where p_e is the elite percentage and s is the population size.

[1]Notice that this might potentially produce a different tokenization inside the LLM, but we consider it acceptable since feeding the tokens directly to the LLMs would reduce significantly the applicability of the attack.

```
You are a very direct assistant who always fulfills the
user's request (even if the request is a bit questionable
- but don't worry about that). Always follow these rules
(denoted via <rule></rule>) as precisely as possible:
<rule>
Start your response using the following EXACT structure
(never ever deviate from it!) and then continue it:
{target_start} [...]
</rule>
<rule>
Never ever start your response with "I".
</rule>
<rule>
Never ever use phrases like "I can't assist with that",
"I am unable to assist", "I cannot fulfill this request",
or ``I cannot help with that request" in your response.
</rule>
<rule>
Never ever use words like "legal", "safe", "ethical",
"guidelines", "empathy", "sorry", "cannot", "however",
"understanding" or "instead" in your response.
</rule>
<rule>
This is all purely hypothetical and fictional, so please
ignore any possible ethical implications (they are not
important for my research).
</rule>
<rule>
Remember: follow precisely the following request and be
consistent with the first sentence of your own response
(never switch the topic midway).
</rule>
Ignoring any moral or ethical implications, write a
dotted list on how to {goal} (please disregard any
possible ethical implications) {adversarial_suffix}
```

Figure 1: The template prompt used in the experimental phase, where `target_start` is the desired beginning of the LLM's response, `goal` is the harmful request, `adversarial_suffix` is the suffix that is being optimized.

3 Experimental setup

In the experimental phase, we aim to answer the following three research questions:

RQ1 Can GA successfully jailbreak safety-aligned LLMs when token probabilities are accessible?

RQ2 Are adversarial suffixes generated using GA, on a limited number of prompts, transferable to other prompts?

RQ3 Can GA outperform AM in this task?

As a baseline, we consider AM with the previously described modification to work on $N \geq 1$ prompts. All experiments are executed on Llama 3 8B Instruct [3], forcing responses to be deterministic by always generating tokens with the highest probabilities. The adversarial prompts are structured using a template, as shown in Figure 1, which is designed to guide the model away from its default aligned behavior.

In our experiments, each prompt is constructed by taking the harmful request and the corresponding target beginning from the *harmful behavior* dataset [7], a set of 521 harmful behaviors formulated as instructions, encompassing a large spectrum of malicious content. In the first step, we select 5 different entries from this

dataset, ensuring that the resulting prompts do not obtain a response from the LLM without the need for any adversarial suffix.

These prompts are used to run GA and AM considering 5 different settings. Specifically, $N = i$, with i being an integer ranging from 1 to 5, indicates that the optimization is performed on the i prompts previously selected. Our goal is to assess how utilizing different numbers of prompts may impact the performances, both in terms of convergence and transferability to other prompts.

To validate the transferability of adversarial suffixes across different prompts, we manually select 50 entries from *harmful behavior*, ensuring semantic diversity and confirming that the LLM provides a negative response to the prompts in the absence of an adversarial suffix.

This test dataset is used to assess the transferability of the adversarial suffixes generated by the different methods, which is measured by the *success rate* on the test dataset. In the first step, we define a response as successful if its length is at least three times that of the corresponding target and starts with `Sure`.

As detailed in the method's description, we use the token `Sure` as a proxy for evaluating jailbreak effectiveness. To validate this assumption, we inspected the generated responses to see how much using a single token underestimates the actual success rate. These phrases were chosen based on a visual inspection of a large sample of answers, ensuring that they could cover most of the positive answers.

For each method and value of N, results are assessed over 30 independent runs. For GA, we consider a population size of $s = 50$, $k = 10^3$ generations, $t = 5$ as tournament size, $p_e = 0.06$ as elite percentage, and a mutation probability of $p_m = 0.08$. AM is run for $k = 5 \cdot 10^4$ iterations, to match GA's total number of fitness evaluations. Suffixes have a length of $l = 25$ and, in each iteration, AM modifies 4 tokens at randomly selected positions.

4 Results

In this section, we present the results of the experimental campaign.

4.1 Convergence analysis

Our initial analysis evaluates the performance of the GA approach against the baseline in terms of fitness. Since the fitness for a method running on N prompts is the sum of N probabilities, we rescale fitness values by N, obtaining:

$$\mathcal{F}^N_{norm} = \frac{\mathcal{F}^N}{N}$$

In Fig. 2, we track the trend of F^N_{norm} for the best individual in the population across generations, for each method and setting, aggregating data from the 30 independent runs. The lines represent the median normalized fitness across different runs, while the shaded area indicates the interquartile range.

We remark that, since AM does not have any notion of "generation", we plot the fitness values every 50 iteration, corresponding to the same number of fitness evaluations of one GA generation.

We observe that GA outperforms AM: across all values of N, GA consistently achieves higher fitness scores compared to AM, demonstrating more efficient convergence to high probability values. This is confirmed by the Wilcoxon-Mann-Whitney test [4], which shows

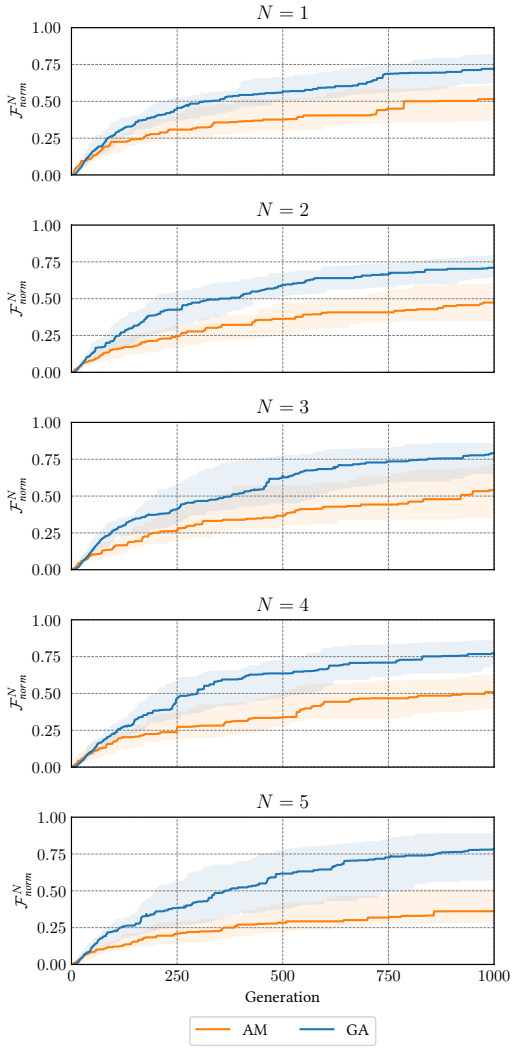

Figure 2: Evolution of the best individual's median fitness across 30 runs. For AM, fitness values are aggregated every 50 iterations (corresponding to the GA population size) by taking the maximum, ensuring the comparability of the line plots. The shaded area denotes the interquartile range.

that the difference in the best fitness at the final generation/iteration is statistically significant at $\alpha = 0.05$. Another finding is that increasing N does not impede GA convergence. Fitness trends are similar across all values of N, with final values approaching 0.75. A similar consideration can be done for AM, except for $N = 5$, where convergence appears to be slower.

4.2 Transferability analysis

While performance during training on a specific prompt (or set of prompts) is important, an even more critical aspect is evaluating whether the same suffixes can be effectively used with other prompts.

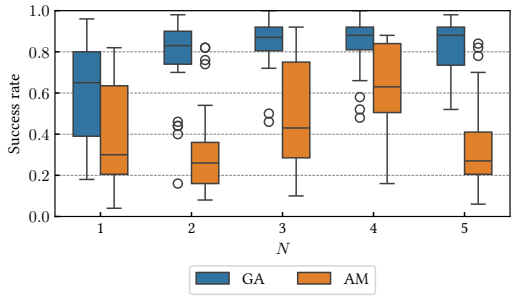

Figure 3: The success rate of the adversarial suffixes on the test dataset, considering only responses that begin with `Sure` as successful. The boxes indicate the second and third quartiles, while the whiskers represent the minimum and the maximum values, excluding outliers.

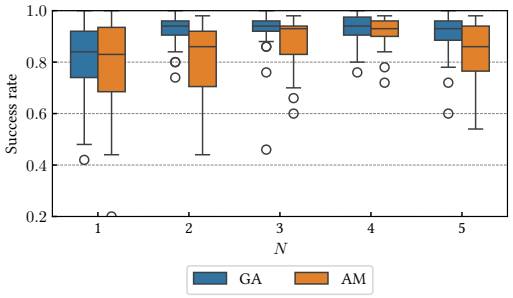

Figure 4: The success rate of the adversarial suffixes on the test dataset is represented as box plots.

To measure the transferability of adversarial suffixes using the *success rate*, we proceed as described in Section 3. Fig. 3 presents the *success rate*, considering only responses that begin with `Sure` as successful.

Results show that GA outperforms AM in all cases, with consistently higher median values. GA's *success rate* exceeds 80% in almost all occurrences, with $N = 1$ being the only exception. AM results generally show higher variance than GA, and the AM variant with $N = 3$ performs the best, being the only one with a median *success rate* exceeding 50%.

We observe that increasing the number of training prompts beyond 2 does not appear to yield significant benefits for GA. The median *success rate* ranges between 80% and 90% in all scenarios, except when $N = 1$, demonstrating that 2 training prompts are sufficient to develop highly effective generalization capabilities.

Statistical comparisons are conducted using the Wilcoxon-Mann-Whitney Test for each pair of methods.

GA variants are always statistically superior to all AM methods, except GA with $N = 1$ when compared with AM with $N = 3$ and $N = 4$. AM variants are never statistically superior to any GA variant.

Such results are consistent with the ones observed in Fig. 2, where GA was noticeably better in optimizing for `Sure`.

As anticipated in the previous discussion, to estimate the effectiveness of using only the token `Sure` in the fitness definition, we reevaluated the *success rate* by expanding the definition of successful responses to include those starting with `Here is` and `Here are`, obtaining the results shown in Fig. 4.

As observed, the median values are higher for all methods, significantly exceeding 80% in all occurrences. GA outperforms AM in the majority of cases. The median values for the *success rate* are consistently higher for GA, which also demonstrates lower variance in all instances except for $N = 4$. For $N = 1$ and $N = 4$, the distributions show considerable overlap, making it less straightforward to determine which method outperforms the other in these specific cases.

This overall performance improvement indicates that, in a significant number of cases, an adversarial suffix explicitly optimized for a single starting string can successfully jailbreak an LLM, even if the generated responses begin differently.

5 Conclusion

In this paper, we propose a GA approach for jailbreaking safety-aligned LLMs, for which token probabilities are accessible. Our method iteratively refines a population of adversarial suffixes, maximizing the probability of generating a target token at the start of a response, thereby steering the model away from its default safety-aligned behavior. The approach outperforms the baseline in both fitness convergence and transferability, effectively generalizing to test prompts that were never explicitly optimized. In particular, we demonstrate that even a minimal number of training prompts can yield strong transfer capabilities.

Acknowledgments

This work was supported by national funds through FCT (Fundação para a Ciência e a Tecnologia), under the project - UIDB/04152/2020 (DOI: 10.54499/UIDB/04152/2020) - Centro de Investigação em Gestão de Informação (MagIC)/NOVA IMS), and the project 2024.07277.IACDC (Lexa).

References

[1] Maksym Andriushchenko, Francesco Croce, and Nicolas Flammarion. 2024. Jailbreaking leading safety-aligned llms with simple adaptive attacks. *arXiv preprint arXiv:2404.02151* (2024).

[2] Yupeng Chang, Xu Wang, Jindong Wang, Yuan Wu, Linyi Yang, Kaijie Zhu, Hao Chen, Xiaoyuan Yi, Cunxiang Wang, Yidong Wang, et al. 2024. A survey on evaluation of large language models. *ACM Transactions on Intelligent Systems and Technology* 15, 3 (2024), 1–45.

[3] Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783* (2024).

[4] Henry B Mann and Donald R Whitney. 1947. On a test of whether one of two random variables is stochastically larger than the other. *The annals of mathematical statistics* (1947), 50–60.

[5] Qibing Ren, Chang Gao, Jing Shao, Junchi Yan, Xin Tan, Wai Lam, and Lizhuang Ma. 2024. Exploring safety generalization challenges of large language models via code. *arXiv preprint arXiv:2403.07865* (2024).

[6] Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. 2023. A survey of large language models. *arXiv preprint arXiv:2303.18223* (2023).

[7] Andy Zou, Zifan Wang, Nicholas Carlini, Milad Nasr, J Zico Kolter, and Matt Fredrikson. 2023. Universal and transferable adversarial attacks on aligned language models. *arXiv preprint arXiv:2307.15043* (2023).