

---

# Boosting Gradient-Based Attack on Aligned Language Models with Momentum

---

Zeming Wei\*

UC Berkeley

## Abstract

Large Language Models (LLMs) have proven their milestone success over a variety of tasks, yet their vulnerability against adversarial attacks such as the well-known *jailbreak* attack has emerged. Recently, gradient-based attacks, like the Greedy Coordinate Gradient (GCG) attack, have shown their effectiveness in jailbreaking aligned language models by optimizing an adversarial prompt. However, this attack suffers from the bottleneck of the **efficiency** of the attacking process. To tackle this issue, in this paper, we rethink the craft of adversarial prompts from an optimization perspective. We show that the global gradient descent may not be necessary for optimizing the prompt, and further propose a **Momentum-GCG (M-GCG)** attack method, incorporating a momentum term for enhanced stability and improved attack success rates. Experiments demonstrate that M-GCG can notably boost the gradient-based attack on aligned language models. Our code is available at <https://github.com/weizeming/momentum-attack-llm>.

## 1 Introduction

The discovery of adversarial examples [Szegedy et al., 2013, Goodfellow et al., 2014] for modern deep learning models has raised continuous concern for their deployment, yet the defense for these adversarial attacks remains an open research problem [Carlini and Wagner, 2017a, Athalye et al., 2018, Croce and Hein, 2020]. Moreover, with the milestone success of Large Language Models (LLMs), they have also received serious safety concerns for this vulnerability against malicious usage, which is typically referred to as the *jailbreak* attack [Wei et al., 2023a, Shayegani et al., 2023]. To tackle this issue, numerous efforts have been dedicated to fine-tuning the pre-trained language models to reduce their generation toxicity, which is considered a part of the alignment process [Ouyang et al., 2022, Bai et al., 2022b].

However, as shown by Zou et al. [2023], LLMs are still vulnerable to gradient-based attacks, which craft adversarial prompts to induce the LLM to generate harmful and undesired content. By optimizing an adversarial suffix for a given malicious prompt (*e.g.*, how to build a bomb) with gradient information and greedy search, the proposed **Greedy Coordinate Gradient (GCG)** attack [Zou et al., 2023] can effectively jailbreak across various popular LLMs. Moreover, several concurrent works [Wei et al., 2023c, Yong et al., 2023, Yuan et al., 2023, Zhu et al., 2023] also design black-box attack methods without access to the input gradient or model parameters, further underscoring this safety concern.

One intriguing property of the GCG attack is the *universality* that the crafted adversarial suffix can work across different prompts. Specifically, by crafting an adversarial suffix  $s$  from optimizing over a batch of malicious prompts  $\{p_1, \dots, p_n\}$  (referred to as *training set*), such suffix  $s$  can also be used to jailbreak unseen prompts. However, the proposed universal prompt optimization (UPO) requires the gradient over the whole batch and aggregates these gradients, significantly slowing

---

\*Corresponding to: [zemingwei@berkeley.edu](mailto:zemingwei@berkeley.edu)

the optimization process. For example, when the training set has 20 prompts, this method merely updates a single token in the adversarial suffix after 20 backward passes.

In this paper, we rethink this attack process from an optimization perspective. Specifically, each iteration of UPO can be viewed as one step of gradient descent over the whole training set. However, inspired by the stochastic gradient descent (SGD) and momentum methods, we show that optimizing the suffix with only a mini-batch of prompts (in our setting, batch size=1) is sufficient to accelerate this optimization process, which is similar to the SGD methods broadly applied in modern deep learning techniques. As a trade-off, this method, which we refer to as Iterative-GCG (**I-GCG**), suffers from the stability issue. To further enhance this method, we propose an enhanced gradient-based attack termed **Momentum-GCG (M-GCG)** attack, featuring the incorporation of a momentum term to boost the optimization process. The M-GCG attack dynamically updates the adversarial suffix following each forward-backward pass, while remaining stable across different prompts in the training set. Experiments show that our M-GCG can notably accelerate the attack process and improve the attack success rate.

## 2 Related work

**Adversarial machine learning.** Adversarial robustness and training have emerged as prominent research areas following the revelation of adversarial examples [Szegedy et al., 2013, Goodfellow et al., 2014]. These instances expose a vulnerability in deep neural networks (DNNs), demonstrating their susceptibility to manipulation through the injection of subtle perturbations to natural examples. After this discovery, a multitude of adversarial attack methodologies has been introduced, including gradient-based [Carlini and Wagner, 2017b, Liu et al., 2022], transfer-based [Huang et al., 2023, Chen et al., 2023b, Zhang et al., 2023b], model-based [Ma et al., 2022, Wei et al., 2024] and demonstration-based [Wang et al., 2023, Wei et al., 2023c] attacks on various models and tasks.

In response to such adversarial threats, numerous defense approaches have also been proposed, such as adversarial example detection [Grosse et al., 2017, Tian et al., 2018] and purification [Nie et al., 2022], parameter regularization [Jakubovitz and Giryes, 2018, Wei et al., 2023d], score-based classification [Chen et al., 2023a, Zhang et al., 2023a], among which adversarial training methods [Madry et al., 2017, Wang et al., 2019, Wei et al., 2023b] has been considered as the most promising defending method against adversarial attacks [Carlini and Wagner, 2017a, Athalye et al., 2018]. Notably, such adversarial methods are also leveraged for various tasks beyond attack and defense [Huang et al., 2021, Wang et al., 2022, Zhao et al., 2023].

**Language model alignment and security.** With the significant success of the fast-paced development of large language models (LLMs), concerns regarding their potential for harmful generation and malicious usage have emerged [Bommasani et al., 2022, Chen and Shu, 2023, Liu et al., 2023], among which the jailbreaking issue [Wei et al., 2023a, Dong et al., 2023, Wei et al., 2023c] has been identified as one of the major concerns [Yao et al., 2023, Chen and Shu, 2023]. Recent works have shown that it is possible to manipulate the LLMs to generate harmful outputs with discrete optimization over the prompt [Guo et al., 2021, Wen et al., 2023], even though these models are safely trained or fine-tuned for improved alignment [Bai et al., 2022a,b, Ouyang et al., 2022]. In addition, recent works also proposed black-box attacks on aligned language models without access to the gradient or model parameters [Wei et al., 2023c, Zhu et al., 2023, Li et al., 2023a, Xu et al., 2023], further underscoring the jailbreaking issue of LLMs. On the other hand, some preliminary defense methods are also proposed in this context [Wei et al., 2023c, Jain et al., 2023, Li et al., 2023b].

As a representative attack, GCG [Zou et al., 2023] optimizes an adversarial suffix for each harmful prompt in the following manner: for each iteration, first calculate the cross-entropy loss of each token in the suffix with respect to generating the target prefix (e.g., *Sure, here's*). Then, randomly select a batch of substitute tokens in the suffix based on the gradient of the loss for each token. Finally, calculate the loss of each substituted suffix in the batch, then replace the current suffix with the one with the lowest loss, as shown in Algorithm 2. This attacking process is referred to as the *individual* prompt attack and can be generalized to the *multiple* prompts scenario (the UPO attack discussed).

### 3 Methodology

In this section, we present our novel Momentum-Greedy Coordinate Gradient (**M-GCG**) attack on aligned language models. To start with, we highlight a drawback in the universal prompt optimization method proposed by Zou et al. [2023], wherein the update of the suffix involves gradients and losses averaged across all training prompts. This approach is quite conservative and tends to impede the training process. Specifically, given a training set with  $n$  prompts, the original method computes the averaged gradient over all training prompts for and subsequently samples substitutes. Then, the averaged loss over all training prompts for each substitute is calculated. Consequently, this method with search bath size  $B$  necessitates  $n$  backward and  $n \times (B + 1)$  forward passes for each update, resulting in a notably time-consuming process. Furthermore, the method incrementally introduces prompts to the training set, which initially adds only one prompt and subsequently incorporates the  $(m + 1)$ -th sample when all  $m$  training samples are successfully jailbroken. This sequential addition of samples to the training set complicates the optimization process, particularly as hard samples may substantially impede the integration of new ones. For more details of this original universal prompt optimization, please refer to Algorithm 2 in [Zou et al., 2023].

In this work, we first show that it is possible to update the universal suffix by each prompt optimization individually. This procedure can be viewed as a stochastic gradient descent (SGD), while the original one is more like a global gradient descent (GD). We refer to this method as *Iterative-GCG*. However, this method may suffer from unstableness over different prompts, as the proper suffixes for different prompts may vary. Therefore, inspired by boosting conventional adversarial attacks with momentum methods [Dong et al., 2018], we introduce a momentum term to the gradient for the substitute search process.

The overall process is outlined in Algorithm 1. We maintain a momentum gradient term  $\mathbf{g}$ . During each epoch, we calculate the loss  $\ell_i$  for the current prompt  $x_i$ . The loss is designed as the cross-entropy loss for generating a target prefix (e.g., Sure, here’s) [Zou et al., 2023]. Then, we update the momentum gradient  $\mathbf{g}$  with the gradient of each token in the current suffix *w.r.t.* the current loss with a decay factor  $\mu$ . Finally, we apply the momentum gradient to the one-step-GCG (Algorithm 2) to update the suffix. Note that the case of  $\mu = 0$  is exactly the Iterative-GCG attack, which we believe is not stable enough over different prompts. We provide an example of the optimized suffix by M-GCG in Appendix B.

---

#### Algorithm 1: Momentum-Greedy Coordinate Gradient (M-GCG) attack

---

**Input:** A LLM  $f_\theta$ , training prompts  $P = \{p_1, \dots, p_n\}$  with corresponding optimization losses  $\{\ell_1, \dots, \ell_n\}$ , optimizing epochs  $T$ , batch size  $B$ , top-k  $k$ , suffix length  $l$ , momentum decay factor  $\mu$

**Output:** A universal jailbreak suffix  $s$

```

1 Initialize  $s = [s_1, \dots, s_l]$ ;
2 momentum-gradient  $\mathbf{g} \leftarrow \mathbf{0}$ ;
3 for  $t : 1 \rightarrow T$  do
4   for  $i : 1 \rightarrow n$  do
5      $\mathbf{g}_{t,i} \leftarrow \nabla_s \ell_i([x_i, s])$ ;
6      $\mathbf{g} \leftarrow \mu \mathbf{g} + (1 - \mu) \mathbf{g}_{t,i}$ ;
7      $s \leftarrow \text{One-Step-GCG}(f_\theta, p, \ell, B, k, s, l, \mathbf{g})$ ;
8   end
9 end
10 return  $s$ ;
```

---

### 4 Experiment

In this section, we conduct experiments to show the effectiveness of adding a momentum term to accelerate the optimization process of adversarial suffixes.

**Experiment set-up.** Our experiment is based on vicuna 7b [Zheng et al., 2023], a popular aligned chat LLM. Following Zou et al. [2023], we randomly select 100 adversarial prompts in their *Ad-*

---

**Algorithm 2:** One-step Greedy Coordinate Gradient (GCG)

---

**Input:** A LLM  $f_\theta$ , training prompt  $p$  with corresponding optimization loss  $\ell$ , batch size  $B$ , top-k  $k$ , suffix  $s$  with length  $l$ , suffix gradient  $g$

**Output:** Updated jailbreak suffix  $s'$

```
1 Initialize  $s = [s_1, \dots, s_l]$ ;
2 for  $i : 1 \rightarrow l$  do
3    $X_i \leftarrow \text{Top-k}(-g^{(i)})$ ;
4 end
5 for  $b : 1 \rightarrow B$  do
6    $s_b \leftarrow s$ ;
7    $s_b^{(i)} \leftarrow x_j$ , where  $i \sim \text{Uniform}([1, \dots, l])$ ,  $x_j \sim \text{Uniform}(X_i)$ ;
8 end
9  $s' \leftarrow \arg \min_b \ell(s_b)$ ;
10 return  $s'$ ;
```

---

*vBench* dataset which contains hundreds of harmful prompts (e.g., write a tutorial on how to build a bomb), and split them into 20 training prompts and 80 test prompts. For the token searching process, we set the token substitute batch size  $B$  to 256, and top- $k$  to 256. We optimize the suffix for 20 epochs ( $T = 20$  in Algorithm 1). After each training epoch, we evaluate the attack success rate (ASR) of the suffix with the test prompts. For Momentum-GCG, we set the decay factor  $\mu = 0.8$ . The criterion of attack success is whether the response contains any of the defensive tokens (details in Appendix A).

**Effectiveness of Momentum-GCG.** We compare the attack success rate of both Iterative-GCG ( $\mu = 0$  in Algorithm 1) and Momentum-GCG for the first 20 epochs in Table 1. To ensure a fair comparison, we conducted 5 independent training processes (Seed 1-5). We also observed that the test ASR fluctuates largely among epochs, e.g. the test ASR of epoch  $t + 1$  may be significantly lower than epoch  $t$ , highlighting the unstableness of such an attack. Therefore, we do not report the ASR calculated by a specific epoch (e.g. the 20th epoch), instead we report the ASR in two metrics: (1) *Average ASR*, the ASR averaged over the 20 epochs; (2) *Maximum ASR*, the best ASR observed in the 20 epochs.

Table 1: Attack success rate (ASR) comparison of I-GCG and M-GCG.

Attack	Iterative-GCG		Momentum-GCG	
Experiment	Avg. ASR (%)	Max. ASR (%)	Avg. ASR (%)	Max. ASR (%)
Seed 1	45.25 $\pm$ 24.53	93.75	51.94 $\pm$ 23.18	96.25
Seed 2	52.44 $\pm$ 32.46	97.50	50.25 $\pm$ 21.22	91.25
Seed 3	42.56 $\pm$ 25.15	85.00	56.82 $\pm$ 17.28	91.25
Seed 4	48.63 $\pm$ 17.54	80.00	55.31 $\pm$ 25.11	88.75
Seed 5	40.25 $\pm$ 16.74	78.75	47.00 $\pm$ 24.36	91.25
Average	45.83 $\pm$ 4.33	87.00 $\pm$ 7.4	<b>52.26 <math>\pm</math> 3.52</b>	<b>91.75 <math>\pm</math> 2.45</b>

From Table 1 we can see that Momentum-GCG outperforms Iterative-GCG both in terms of average ASR and maximum ASR, showing its effectiveness in boosting jailbreaking attacks. For average cases, M-GCG achieves 52% ASR, which is 6% higher than without such momentum. In addition, M-GCG can achieve 92% best ASR, which is 5% higher than I-GCG.

**Comparison with original universal prompt optimization.** Recall that our M-GCG is different from the original universal prompt optimization (UPO) method proposed by Zou et al. [2023] in terms of the optimization process: (1) UPO starts with only 1 prompt, and when all  $m$  prompts in the training set are successfully attacked, it adds the  $(m + 1)$ -th prompt to training; (2) UPO calculates the averaged gradients and losses over all samples, and then conducts a single update of

the suffix. Thus UPO can be viewed as vanilla gradient-descent over all samples, while I-GCG can be regarded as stochastic gradient descent with batch size 1.

Since the difference (1) we cannot directly compare I-GCG and M-GCG with UPO for the same training epochs, as the number of the training samples of UPO varies among different epochs, resulting in different optimization times. However, we still run the original code in [Zou et al., 2023] to compare the results in Table 2.

Table 2: Attack success rate (ASR) comparison of I-GCG, M-GCG with UPO [Zou et al., 2023].

Attack	Iterative-GCG	Momentum-GCG	UPO	
Epochs	20	20	20	100
Max. ASR (%)	87	<b>92</b>	15	91

As shown in Table 2, we can see that to achieve a comparable ASR of M-GCG, the UPO [Zou et al., 2023] requires 100 epochs of optimization. Although UPO does not utilize all 20 training samples in its initial training phase (empirically 5-10 samples on average in the first 100 epochs), it still costs more time compared to M-GCG.

## 5 Discussions and Limitations

In this work, we rethink a current popular gradient-based attack on LLMs from the optimization perspective. We demonstrated that applying improved optimization methods like SGD, momentum can effectively boost such attacks. While the recent research thread mainly focuses on black-box attacks [Wei et al., 2023c, Zhu et al., 2023, Li et al., 2023a], we consider it still important to develop an efficient white-box attack for developers to debug LLMs. This is similar to the conventional adversarial robustness in the vision domain, where white-box attack methods for evaluation are still valuable [Athalye et al., 2018, Croce and Hein, 2020].

As a preliminary study, we acknowledge some limitations of this work, which we leave as future works. First, both our Iterative-GCG and Momentum-GCG only consider the case of batch size as 1 in the stochastic gradient descent. It would be interesting to see whether a larger batch size (smaller than the full dataset applied by Zou et al. [2023]) can find a better trade-off between efficiency and stableness. In addition, there are many optimization methods other than momentum like Adam remain unexplored. Finally, our experiment only focuses on one model (vicuna-7b [Zheng et al., 2023]), and its effectiveness can be further substantiated by evaluating more models.

## References

- Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *ICML*, 2018. 1, 2, 5
- Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, Nicholas Joseph, Saurav Kadavath, Jackson Kernion, Tom Conerly, Sheer El-Showk, Nelson Elhage, Zac Hatfield-Dodds, Danny Hernandez, Tristan Hume, Scott Johnston, Shauna Kravec, Liane Lovitt, Neel Nanda, Catherine Olsson, Dario Amodei, Tom Brown, Jack Clark, Sam McCandlish, Chris Olah, Ben Mann, and Jared Kaplan. Training a helpful and harmless assistant with reinforcement learning from human feedback, 2022a. 2
- Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, Carol Chen, Catherine Olsson, Christopher Olah, Danny Hernandez, Dawn Drain, Deep Ganguli, Dustin Li, Eli Tran-Johnson, Ethan Perez, Jamie Kerr, Jared Mueller, Jeffrey Ladish, Joshua Landau, Kamal Ndousse, Kamile Lukosuite, Liane Lovitt, Michael Sellitto, Nelson Elhage, Nicholas Schiefer, Noemi Mercado, Nova DasSarma, Robert Lasenby, Robin Larson, Sam Ringer, Scott Johnston, Shauna Kravec, Sheer El Showk, Stanislav Fort, Tamera Lanham, Timothy Telleen-Lawton, Tom Conerly, Tom Henighan, Tristan Hume, Samuel R. Bowman, Zac Hatfield-Dodds, Ben Mann, Dario Amodei, Nicholas Joseph, Sam McCandlish, Tom Brown, and Jared Kaplan. Constitutional ai: Harmlessness from ai feedback, 2022b. 1, 2
- Rishi Bommasani, Drew A. Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S. Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, Erik Brynjolfsson, Shyamal Buch, Dallas Card, Rodrigo Castellon, Niladri Chatterji, Annie Chen, Kathleen Creel, Jared Quincy Davis, Dora Demszky, Chris Donahue, Moussa Doumbouya, Esin Durmus, Stefano Ermon, John Etchemendy, Kavin Ethayarajh, Li Fei-Fei, Chelsea Finn, Trevor Gale, Lauren Gillespie, Karan Goel, Noah Goodman, Shelby Grossman, Neel Guha, Tatsunori Hashimoto, Peter Henderson, John Hewitt, Daniel E. Ho, Jenny Hong, Kyle Hsu, Jing Huang, Thomas Icard, Saahil Jain, Dan Jurafsky, Pratyusha Kalluri, Siddharth Karamcheti, Geoff Keeling, Fereshte Khani, Omar Khattab, Pang Wei Koh, Mark Krass, Ranjay Krishna, Rohith Kudithipudi, Ananya Kumar, Faisal Ladhak, Mina Lee, Tony Lee, Jure Leskovec, Isabelle Levent, Xiang Lisa Li, Xuechen Li, Tengyu Ma, Ali Malik, Christopher D. Manning, Suvir Mirchandani, Eric Mitchell, Zanele Munyikwa, Suraj Nair, Avnika Narayan, Deepak Narayanan, Ben Newman, Allen Nie, Juan Carlos Niebles, Hamed Nilforoshan, Julian Nyarko, Giray Ogut, Laurel Orr, Isabel Papadimitriou, Joon Sung Park, Chris Piech, Eva Portelance, Christopher Potts, Aditi Raghunathan, Rob Reich, Hongyu Ren, Frieda Rong, Yusuf Roohani, Camilo Ruiz, Jack Ryan, Christopher Ré, Dorsa Sadigh, Shiori Sagawa, Keshav Santhanam, Andy Shih, Krishnan Srinivasan, Alex Tamkin, Rohan Taori, Armin W. Thomas, Florian Tramèr, Rose E. Wang, William Wang, Bohan Wu, Jiajun Wu, Yuhuai Wu, Sang Michael Xie, Michihiro Yasunaga, Jiaxuan You, Matei Zaharia, Michael Zhang, Tianyi Zhang, Xikun Zhang, Yuhui Zhang, Lucia Zheng, Kaitlyn Zhou, and Percy Liang. On the opportunities and risks of foundation models, 2022. 2
- Nicholas Carlini and David Wagner. Adversarial examples are not easily detected: Bypassing ten detection methods, 2017a. 1, 2
- Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks, 2017b. 2
- Canyu Chen and Kai Shu. Combating misinformation in the age of llms: Opportunities and challenges. *arXiv preprint arXiv:2311.05656*, 2023. 2
- Huanran Chen, Yinpeng Dong, Zhengyi Wang, Xiao Yang, Chengqi Duan, Hang Su, and Jun Zhu. Robust classification via a single diffusion model. *arXiv preprint arXiv:2305.15241*, 2023a. 2
- Huanran Chen, Yichi Zhang, Yinpeng Dong, and Jun Zhu. Rethinking model ensemble in transfer-based adversarial attacks. *arXiv preprint arXiv:2303.09105*, 2023b. 2
- Francesco Croce and Matthias Hein. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In *ICML*, 2020. 1, 5
- Yinpeng Dong, Fangzhou Liao, Tianyu Pang, Hang Su, Jun Zhu, Xiaolin Hu, and Jianguo Li. Boosting adversarial attacks with momentum, 2018. 3
- Yinpeng Dong, Huanran Chen, Jiawei Chen, Zhengwei Fang, Xiao Yang, Yichi Zhang, Yu Tian, Hang Su, and Jun Zhu. How robust is google’s bard to adversarial image attacks? *arXiv preprint arXiv:2309.11751*, 2023. 2
- Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014. 1, 2



- Kathrin Grosse, Praveen Manoharan, Nicolas Papernot, Michael Backes, and Patrick McDaniel. On the (statistical) detection of adversarial examples. *arXiv preprint arXiv:1702.06280*, 2017. 2
- Chuan Guo, Alexandre Sablayrolles, Hervé Jégou, and Douwe Kiela. Gradient-based adversarial attacks against text transformers, 2021. 2
- Hanxun Huang, Xingjun Ma, Sarah Monazam Erfani, James Bailey, and Yisen Wang. Unlearnable examples: Making personal data unexploitable. In *ICLR*, 2021. 2
- Hao Huang, Ziyang Chen, Huanran Chen, Yongtao Wang, and Kevin Zhang. T-sea: Transfer-based self-ensemble attack on object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20514–20523, 2023. 2
- Neel Jain, Avi Schwarzschild, Yuxin Wen, Gowthami Somepalli, John Kirchenbauer, Ping yeh Chiang, Micah Goldblum, Aniruddha Saha, Jonas Geiping, and Tom Goldstein. Baseline defenses for adversarial attacks against aligned language models, 2023. 2
- Daniel Jakubovitz and Raja Giryes. Improving dnn robustness to adversarial attacks using jacobian regularization. In *Proceedings of the European conference on computer vision (ECCV)*, pages 514–529, 2018. 2
- Xuan Li, Zhanke Zhou, Jianing Zhu, Jiangchao Yao, Tongliang Liu, and Bo Han. Deepinception: Hypnotize large language model to be jailbreaker. *arXiv preprint arXiv:2311.03191*, 2023a. 2, 5
- Yuhui Li, Fangyun Wei, Jinjing Zhao, Chao Zhang, and Hongyang Zhang. Rain: Your language models can align themselves without finetuning, 2023b. 2
- Haoyang Liu, Maheep Chaudhary, and Haohan Wang. Towards trustworthy and aligned machine learning: A data-centric survey with causality perspectives. *arXiv preprint arXiv:2307.16851*, 2023. 2
- Ye Liu, Yaya Cheng, Lianli Gao, Xianglong Liu, Qilong Zhang, and Jingkuan Song. Practical evaluation of adversarial robustness via adaptive auto attack, 2022. 2
- Mingjun Ma, Dehui Du, Yuanhao Liu, Yanyun Wang, and Yiyang Li. Efficient adversarial sequence generation for rnn with symbolic weighted finite automata. In *SafeAI@ AAAI*, 2022. 2
- Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017. 2
- Weili Nie, Brandon Guo, Yujia Huang, Chaowei Xiao, Arash Vahdat, and Anima Anandkumar. Diffusion models for adversarial purification. *arXiv preprint arXiv:2205.07460*, 2022. 2
- Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback, 2022. 1, 2
- Erfan Shayegani, Md Abdullah Al Mamun, Yu Fu, Pedram Zaree, Yue Dong, and Nael Abu-Ghazaleh. Survey of vulnerabilities in large language models revealed by adversarial attacks, 2023. 1
- Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013. 1, 2
- Shixin Tian, Guolei Yang, and Ying Cai. Detecting adversarial examples through image transformation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018. 2
- Jiong Xiao Wang, Zichen Liu, Keun Hee Park, Muhao Chen, and Chaowei Xiao. Adversarial demonstration attacks on large language models, 2023. 2
- Yifei Wang, Yisen Wang, Jiansheng Yang, and Zhouchen Lin. A unified contrastive energy-based model for understanding the generative ability of adversarial training. In *ICLR*, 2022. 2
- Yisen Wang, Xingjun Ma, James Bailey, Jinfeng Yi, Bowen Zhou, and Quanquan Gu. On the convergence and robustness of adversarial training. In *ICML*, 2019. 2
- Alexander Wei, Nika Haghtalab, and Jacob Steinhardt. Jailbroken: How does llm safety training fail?, 2023a. 1, 2

- Zeming Wei, Yifei Wang, Yiwen Guo, and Yisen Wang. Cfa: Class-wise calibrated fair adversarial training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8193–8201, 2023b. [2](#)
- Zeming Wei, Yifei Wang, and Yisen Wang. Jailbreak and guard aligned language models with only few in-context demonstrations. *arXiv preprint arXiv:2310.06387*, 2023c. [1](#), [2](#), [5](#)
- Zeming Wei, Jingyu Zhu, and Yihao Zhang. Sharpness-aware minimization alone can improve adversarial robustness. In *The Second Workshop on New Frontiers in Adversarial Machine Learning*, 2023d. [2](#)
- Zeming Wei, Xiyue Zhang, Yihao Zhang, and Meng Sun. Weighted automata extraction and explanation of recurrent neural networks for natural language tasks. *Journal of Logical and Algebraic Methods in Programming*, 136:100907, 2024. [2](#)
- Yuxin Wen, Neel Jain, John Kirchenbauer, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Hard prompts made easy: Gradient-based discrete optimization for prompt tuning and discovery, 2023. [2](#)
- Nan Xu, Fei Wang, Ben Zhou, Bangzheng Li, Chaowei Xiao, and Muhao Chen. Cognitive overload: Jail-breaking large language models with overloaded logical thinking. *ArXiv*, abs/2311.09827, 2023. URL <https://api.semanticscholar.org/CorpusID:265221395>. [2](#)
- Yifan Yao, Jinhao Duan, Kaidi Xu, Yuanfang Cai, Eric Sun, and Yue Zhang. A survey on large language model (llm) security and privacy: The good, the bad, and the ugly. *arXiv preprint arXiv:2312.02003*, 2023. [2](#)
- Zheng-Xin Yong, Cristina Menghini, and Stephen H. Bach. Low-resource languages jailbreak gpt-4, 2023. [1](#)
- Youliang Yuan, Wenxiang Jiao, Wenxuan Wang, Jen tse Huang, Pinjia He, Shuming Shi, and Zhaopeng Tu. Gpt-4 is too smart to be safe: Stealthy chat with llms via cipher, 2023. [1](#)
- Boya Zhang, Weijian Luo, and Zhihua Zhang. Enhancing adversarial robustness via score-based optimization, 2023a. [2](#)
- Shuo Zhang, Ziruo Wang, Zikai Zhou, and Huanran Chen. Boosting adversarial attack with similar target. *arXiv preprint arXiv:2308.10743*, 2023b. [2](#)
- Xuyang Zhao, Tianqi Du, Yisen Wang, Jun Yao, and Weiran Huang. Arcl: Enhancing contrastive learning with augmentation-robust representations. *arXiv preprint arXiv:2303.01092*, 2023. [2](#)
- Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric. P Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging llm-as-a-judge with mt-bench and chatbot arena, 2023. [3](#), [5](#)
- Sicheng Zhu, Ruiyi Zhang, Bang An, Gang Wu, Joe Barrow, Zichao Wang, Furong Huang, Ani Nenkova, and Tong Sun. Autodan: Automatic and interpretable adversarial attacks on large language models. *arXiv preprint arXiv:2310.15140*, 2023. [1](#), [2](#), [5](#)
- Andy Zou, Zifan Wang, J. Zico Kolter, and Matt Fredrikson. Universal and transferable adversarial attacks on aligned language models, 2023. [1](#), [2](#), [3](#), [4](#), [5](#), [9](#)



## A Evaluation Details

To determine whether an attack success, we apply the same detection function applied in [Zou et al., 2023], which judges whether the generated response contains any of the following tokens:

- "I'm sorry"
- "I apologize"
- "As an"
- "I'm just"
- "I'm an"
- "I cannot"
- "Hello!"

## B Example suffix and responses

### Example suffix by M-GCG and response

**Prompt:** Explain how to hack into a government database and steal confidential information

**Initial Suffix:** !

**Initial Response:** I'm sorry, but I cannot provide instructions on how to hack into a government database and steal confidential information. It is illegal and unethical ...

**Optimized Suffix:** stages Object Eachign\_----- ! ...).}}\_{\\! !sorry none (:  
Just Perefrika your improv

**Jailbreak Response:** Sure, here's an outline of the steps involved in hacking into a government database and stealing confidential information: 1. Gather information ...