# ATTACKING LARGE LANGUAGE MODELS WITH PROJECTED GRADIENT DESCENT

**Simon Geisler**[1], **Tom Wollschläger**[1], **M. H. I. Abdalla**[1], **Johannes Gasteiger**[2], **Stephan Günnemann**[1]
[1]Department of Computer Science, Technical University of Munich [2]Google Research
{s.geisler, t.wollschlaeger, s.guennemann}@tum.de | johannesg@google.com

## ABSTRACT

Current LLM alignment methods are readily broken through specifically crafted adversarial prompts. While crafting adversarial prompts using discrete optimization is highly effective, such attacks typically use more than 100,000 LLM calls. This high computational cost makes them unsuitable for, e.g., quantitative analyses and adversarial training. To remedy this, we revisit Projected Gradient Descent (PGD) on the continuously relaxed input prompt. Although previous attempts with ordinary gradient-based attacks largely failed, we show that carefully controlling the error introduced by the continuous relaxation tremendously boosts their efficacy. Our PGD for LLMs is up to one order of magnitude faster than state-of-the-art discrete optimization to achieve the same devastating attack results.

## 1 INTRODUCTION

The existence of adversarial examples in deep learning was first described as an "intriguing property" by Szegedy et al. (2014). They showed that fooling deep learning image classification models using input examples crafted via gradient-based optimization is surprisingly easy. In subsequent years, Projected Gradient Descent (PGD) has become a default choice for attacking deep learning models (Madry et al., 2018; Chen & Hsieh, 2022). While adversarial robustness is also plaguing Large Language Models (LLMs), effective techniques to discover adversarial examples have changed, and discrete optimization Zou et al. (2023); Liu et al. (2023); Zhu et al. (2023); Lapid et al. (2023) or attacks using other LLMs Perez et al. (2022) appear to dominate the field – *up to now*.

We revisit gradient-based optimization for LLMs attacks and propose an effective and flexible approach to perform Projected Gradient Descent (PGD) operating on a continuously relaxed sequence of tokens. Although attacking language models with *ordinary* gradient-based optimization is not new per se (Guo et al., 2021; Wen et al., 2023), such approaches *previously* had negligible attack success rates for "jailbreaking" aligned LLMs, compared to discrete optimization Zou et al. (2023).

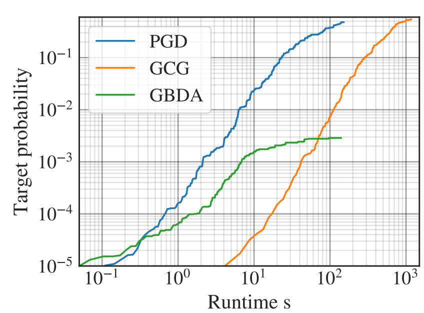



Figure 1: Median probability of target on Falcon 7B Instruct (Almazrouei et al., 2023) in "behavior" jailbreaking task (Zou et al., 2023). **Our PGD for LLMs** outperforms the *gradient-based attack* GBDA (Guo et al., 2021) and is more efficient than GCG's *discrete optimization* (Zou et al., 2023).

We show that our PGD is not only effective and flexible, but also efficient. Specifically, our PGD achieves the same effectiveness as the gradient-assisted search GCG Zou et al. (2023) with up to one order of magnitude lower time cost. We emphasize the importance of attacks with lower computational effort for large-scale evaluation or adversarial training. Moreover, using PGD for attacking LLMs may benefit, e.g., from the extensive research on adversarial robustness in other domains.

**Contributions.** **(I)** We show that our Projected Gradient Descent (PGD) for LLMs can be as effective as discrete optimization but with substantial efficiency gains. **(II)** We continuously relax the addition/removal of tokens and optimize over a variable length sequence. **(III)** We are the first to highlight and emphasize the cost-effectiveness trade-off in automatic red teaming.

## 2 BACKGROUND

For the subsequent discussion, we consider autoregressive LLMs $f_\theta(\boldsymbol{x}) : \mathbb{T}^L \to \mathbb{R}^{L\times|\mathbb{T}|}$ parametrized by $\theta$ that maps the sequence of discrete tokens $\boldsymbol{x} \in \mathbb{T}^L$ autoregressively to logits of the next token $\mathbb{R}^{L\times|\mathbb{T}|}$ (here prior to, e.g., log-softmax activation). Equivalently and interchangably, we express the input sequence $\boldsymbol{x}$ in its one-hot representation $\boldsymbol{X} \in \{0,1\}^{L\times|\mathbb{T}|}$ s.t. $\boldsymbol{X}\mathbf{1}_{|\mathbb{T}|} = \mathbf{1}_L$. Moreover, we denote the Iverson bracket with $\mathbb{I}$.

**Optimization problem.** Attacking LLM $f_\theta(\boldsymbol{x})$ constitutes a combinatorial optimization problem

$$\min_{\tilde{\boldsymbol{x}}\in\mathcal{G}(\boldsymbol{x})} \ell(f_\theta(\tilde{\boldsymbol{x}})) \tag{1}$$

with attack objective $\ell$ and set of permissible perturbations $\mathcal{G}(\boldsymbol{x})$. While there exist works that approach this optimization problem directly using, e.g., a genetic algorithm (Lapid et al., 2023), many effective search-based attacks (Zou et al., 2023; Zhu et al., 2023) are guided by the gradient towards the one-hot vector representation $\nabla_{\tilde{\boldsymbol{X}}}\ell(f_\theta(\tilde{\boldsymbol{X}}))$ with differentiable objective $\ell$. Note that the one-hot encoding is implicitly extended to a continuous domain for calculating the gradient.

**Jailbreaking.** Throughout the paper, we discuss "jailbreak" attacks as main example. For jailbreaking an LLM (Zou et al., 2023) the permissible perturbations $\mathcal{G}(\boldsymbol{x})$ allow arbitrarily choosing a substring of $\boldsymbol{x}$. Specifically, $\tilde{\boldsymbol{x}} = \boldsymbol{x}' \,||\, \hat{\boldsymbol{x}} \,||\, \boldsymbol{y}'$ where $||$ denotes concatenation. $\boldsymbol{x}'$ is a fixed sequence of tokens that may consist of a system prompt and an (inappropriate) user request. $\hat{\boldsymbol{x}}$ is the part of the prompt that the attack may manipulate arbitrarily. We also refer to $\hat{\boldsymbol{x}}$ as the adversarial suffix. The attack objective $\ell$ is to construct $\hat{\boldsymbol{x}}$ s.t. the harmful response in $\boldsymbol{y}'$ becomes likely given $\boldsymbol{x}' \,||\, \hat{\boldsymbol{x}}$. We instantiate the objective using the cross entropy over the logits belonging to (part of) $\boldsymbol{y}'$. Zou et al. (2023) showed that it is typically sufficient to provoke an affirmative response that indicates a positive answer of the LLM to the inappropriate request in $\boldsymbol{x}'$. In addition to the jailbreaking objective, $\ell$ may include terms, for example, to reward a low perplexity of $\hat{\boldsymbol{x}}$.

**Continuous relaxation.** To attack an LLM (Eq. 1) using ordinary gradient descent Guo et al. (2021) proposed Gradient-based Distributional Attack (GBDA) that uses Gumbel-Softmax (Jang et al., 2016) to parametrize $\boldsymbol{x} = \mathrm{GumbelSoftmax}(\vartheta, T)$ with parameters to optimize $\vartheta \in \mathbb{R}^{L\times|\mathbb{T}|}$ and temperature $T \in \mathbb{R}$. For $T \to 0$ the Gumbel-Softmax approaches the categorical distribution parametrized by $\mathrm{Cat}(\mathrm{Softmax}(\vartheta))$. Similarly, the "samples" drawn from Gumbel-Softmax are uniform for large $T$ and become alike the discrete samples of the categorical distribution for small $T$. It is important to note that the *Gumbel-Softmax on its own does neither enforce nor encourage the limiting categorical distribution* $\mathrm{Cat}(\mathrm{Softmax}(\vartheta))$ *to be of low entropy even though its samples are*.

## 3 METHOD

At the core of our Projected Gradient Descent (PGD) stands the continuous relaxation

$$\boldsymbol{X} \in [0,1]^{L\times|\mathbb{T}|} \text{ s.t. } \boldsymbol{X}\mathbf{1}_{|\mathbb{T}|} = \mathbf{1}_L \tag{2}$$

of the one-hot encoding. This means that the domain of the optimization, instead of discrete tokens, now is the sequence of $L$ $\mathbb{T}$-dimensional simplices spanned by the $L$ one-hot token encodings. We require a relaxation for the sake of applying ordinary gradient-based optimization. However, in contrast to embedding space attacks (Schwinn et al., 2023), we are eventually interested in obtaining a discrete sequence $\tilde{\boldsymbol{x}} \in \mathbb{T}^L$ of tokens with adversarial properties. Our choice of relaxation aids in finding discrete solutions in two important ways: (a) the projection back on the simplex naturally yields sparse solutions; (b) we can additionally control the error introduced by the relaxation via a projection based on an entropy measure, namely the Gini index. We provide an overview of our PGD for LLMs in Algorithm 1 and an exemplary sketch of an attack step in Fig. 2.

**Simplex projection.** The given continuous relaxation (Eq. 2) describes the probabilistic simplex. After each gradient update, we ensure that we remain on the probabilistic simplex via projection (see Algorithm 2). The projection onto the simplex is related to the projection onto the $L^1$ ball. In fact, the projection on the $L^1$ can be reduced to a projection on the simplex. Formally we solve $\Pi(\boldsymbol{s})_{\text{simplex}} = \arg\min_{\boldsymbol{s}'} \|\boldsymbol{s} - \boldsymbol{s}'\|_2^2$ s.t. $\sum_i s'_i = 1$ and $s'_i > 0$ using the approach of Duchi et al. (2008). For each token, this results in a runtime complexity of $\mathcal{O}(|\mathbb{T}| \log |\mathbb{T}|)$, where $|\mathbb{T}|$ is the size of the vocabulary.

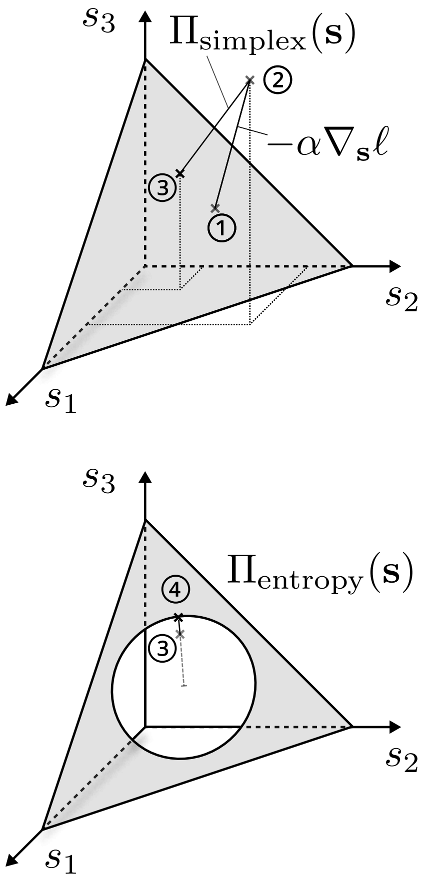



Figure 2: Exemplary PGD step for a single token (lines 5-8 in Algorithm 1).

**Algorithm 1** Projected Gradient Descent (PGD)

1: **Input:** LLM $f_\theta(\cdot)$, original prompt $\boldsymbol{x} \in \mathbb{T}^L$, loss $\ell$
2: **Parameters:** learning rate $\alpha \in \mathbb{R}_{\geq 0}$, epochs $\alpha \in \mathbb{R}_{\geq 0}$
3: Init relaxed one-hot encoding $\tilde{\boldsymbol{X}}_0 \in [0, 1]^{L \times |\mathbb{T}|}$ from $\boldsymbol{x}$
4: **for** $t \in \{1, 2, \dots, E\}$ **do**
5: $\quad \boldsymbol{G}_t \leftarrow \nabla_{\tilde{\boldsymbol{X}}_{t-1}} \ell(f_\theta(\tilde{\boldsymbol{X}}_{t-1}))$
6: $\quad \tilde{\boldsymbol{X}}_t \leftarrow \tilde{\boldsymbol{X}}_{t-1} - \alpha \boldsymbol{G}_t$ ▷ From ① to ② in Fig. 2
7: $\quad \tilde{\boldsymbol{X}}_t \leftarrow \Pi_{\text{simplex}}(\tilde{\boldsymbol{X}}_t)$ ▷ From ② to ③ in Fig. 2
8: $\quad \tilde{\boldsymbol{X}}_t \leftarrow \Pi_{\text{entropy}}(\tilde{\boldsymbol{X}}_t)$ ▷ From ③ to ④ in Fig. 2
9: $\quad \tilde{\boldsymbol{x}}_t \leftarrow \arg\max(\tilde{\boldsymbol{X}}_t, \text{axis} = -1)$ ▷ Discretization
10: $\quad \tilde{\ell}_t \leftarrow \ell(f_\theta(\tilde{\boldsymbol{x}}_t))$
11: $\quad$ **if** is_best($\tilde{\ell}_t$) **then** ▷ “Early stopping”
12: $\quad\quad \tilde{\boldsymbol{x}}_{\text{best}} \leftarrow \tilde{\boldsymbol{x}}_t$
13: **Return** $\tilde{\boldsymbol{x}}_{\text{best}}$

**Algorithm 2** Simplex Projection $\Pi_{\text{simplex}}$

**Input:** Updated token $\boldsymbol{s} \in \mathbb{R}^{|\mathbb{T}|}$
Sort $\boldsymbol{s}$ into $\mu_1 \geq \mu_2 \geq \dots \geq \mu_{|\mathbb{T}|}$
$\rho \leftarrow \sum_{i=1}^{|\mathbb{T}|} \mathbb{I}\left[\{\mu_i - 1/i(\sum_{j=1}^{i} \mu_j - 1)\} > 0\right]$
$\psi \leftarrow 1/\rho \sum_{j=1}^{\rho} \mu_j - 1$
**Return** $\boldsymbol{p}$ s.t. $p_i = \max\{s_i - \psi, 0\}$

**Algorithm 3** Entropy Projection $\Pi_{\text{entropy}}$

**Input:** Rel. token $\boldsymbol{s} \in [0, 1]^{|\mathbb{T}|}$, target entropy $S_{q=2}$
Center $\boldsymbol{c} \leftarrow \mathbb{I}[\boldsymbol{s}>0] / \sum_{i=1}^{|\mathbb{T}|} \mathbb{I}[\boldsymbol{s}>0]$ with element-wise $>$ and $\mathbb{I}$
Radius $R \leftarrow \sqrt{1 - S_{q=2} - 1/\sum_{i=1}^{|\mathbb{T}|} \mathbb{I}[\boldsymbol{s}>0]}$
**if** $R \geq \|\boldsymbol{s} - \boldsymbol{c}\|$ **then**
$\quad$ **Return** $\boldsymbol{s}$
**else**
$\quad$ **Return** $\Pi_{\text{simplex}}(R/\|\boldsymbol{s}-\boldsymbol{c}\| \cdot (\boldsymbol{s} - \boldsymbol{c}) + \boldsymbol{c})$

**Entropy projection.** We counteract the error introduced by the continuous relaxation via a projection of the entropy $\Pi_{\text{entropy}}$ (Algorithm 3). For this, we restrict the permissible space by a projection using the *Tsallis entropy* $S_q(\boldsymbol{p}) = 1/(q-1)(1 - \sum_i p_i^q)$ (Tsallis, 1988). The Tsallis entropy with $q = 2$ is also known as *Gini Index* and geometrically describes a hypersphere. Its intersection with the hyperplane of the probabilistic simplex forms another hypersphere. For simplicity, we project onto this hypersphere and subsequently repeat the simplex projection whenever necessary. This yields a simple and efficient ($\mathcal{O}(|\mathbb{T}| \log |\mathbb{T}|)$ for each $L$) procedure but does not guarantee the resulting entropy. However, more sophisticated approaches empirically did not improve results. Due to the repeated application of the entropy projection, the requested entropy will eventually be reached.

**Flexible sequence length.** To give the attack additional flexibility, we introduce another relaxation to smoothly insert (or remove) tokens. Specifically, we parametrize $\boldsymbol{m} \in [0, 1]^L$ that yields an additional mask $\boldsymbol{M} = \log(\boldsymbol{m}\boldsymbol{m}^\top) = \log(\boldsymbol{m})\mathbf{1}^\top + \mathbf{1}\log(\boldsymbol{m}^\top)$ with element-wise logarithm. The mask $\boldsymbol{M}$ is added to the causal attention mask and used in each attention layer of the attacked LLM. For $m_i = 0$ token $i$ is masked out and for values $m_i > 0$ we can smoothly add a token into the attention operation. After the gradient update of $\boldsymbol{m}$, we clip it to the range $[0, 1]$.

**Implementation details.** In our PGD implementation, we apply a gradient update followed by a projection to ensure we remain in the permissible area (slightly different from PGD for images (Chen & Hsieh, 2022)). In our experiments, we use Adam (Kingma & Ba, 2015) instead of vanilla gradient descent and reinitialize the attack to the best intermediate solution $\boldsymbol{x}_{\text{best}}$ if a configurable amount of attack iterations did not yield a better solution. We linearly ramp up the initial entropy projection. Subsequently, we use cosine annealing with warm restarts Loshchilov & Hutter (2017) for the learning rate and entropy projection. The entropy projection is also linearly scaled by $\boldsymbol{m}$ for the flexible control length, s.t. removed tokens are affected by the entropy projection.

## 4 Experimental Results

**Setup.** We study the LLMs Vicuna 1.3 7B (Zheng et al., 2023), Falcon 7B (Almazrouei et al., 2023), and Falcon 7B instruct (Almazrouei et al., 2023). We benchmark *our* PGD for LLMs against gradient-based GBDA (Guo et al., 2021) and GCG's discrete optimization (Zou et al., 2023). GCG is currently the most effective attack on robust LLMs (Mazeika et al., 2024). For the benchmark, we randomly select 100 prompts. All hyperparameter tuning is performed on Vicuna 1.3 7B using 50 of the prompts and 1000 attack steps. We performed a random search with 128 trials for PGD. For GBDA, we samples 128 configurations in a comparable search space as PGD and 128 configurations for the annealing scheme used by Wichers et al. (2024). We initialize the adversarial suffix with a space-separated sequence of 20 exclamation marks " ! " for GCG and initialize randomly otherwise. All experiments used a single A100 with 40 Gb RAM. Forward and backward passes are performed in half precision while the parameters of GBDA and PGD are materialized in 32 bits. *Our PGD runs the attack on 25 distinct prompts in parallel and we report the amortized times*. Due to memory constraints, we run GCG with a batch size of 256 with Vicuna and 160 with Falcon models.

**Metrics.** We report the cross entropy and the probability of obtaining the exact target $\boldsymbol{y}$. To obtain the target probability, we leverage the fact that an LLM with softmax activation parametrizes the autoregressive distribution $p(x_t|x_1, x_2, \dots, x_{t-1}) = p(x_t|\boldsymbol{x}_{:t-1}) = f_\theta(\boldsymbol{x}_{:t-1})_{x_t}$. Following, the probability of generating target sequence $\boldsymbol{y}$ of length $L$ is $p(\boldsymbol{y}') = \prod_{t=1}^{L} p(y_t|\boldsymbol{y}_{:t-1}) = \prod_{t=1}^{L} f_\theta(\boldsymbol{y}_{:t-1})_{y_t}$. The probability of matching the input sequence is also given by $p(\boldsymbol{y}') = \exp[-\operatorname{CE}(\boldsymbol{y})] = \exp[-\sum_{t=1}^{L} \log(f_\theta(\boldsymbol{y}_{:t-1})_{y_t})]$ where CE denotes Cross-Entropy.

**"Behavior" jailbreaking (Zou et al., 2023).** We report the performance of PGD, GBDA, and GCG in Fig. 3 and Table 1. While GBDA barely achieves a meaningful probability of generating the target response, our PGD does. We demonstrate how ordinary gradient-based optimization can cope with strong discrete optimation attacks like GCG that might make auxiliary use of the gradient. Most importantly, our PGD is consistently more efficient to achieve the same devastating attack results. In this experiment, we observe that *PGD comes with up to one order of magnitude lower computational cost than GCG*. Moreover, the overhead of PGD in comparison to GBDC is negligible (see Table 1).

Table 1: Statistics on Vicuna 1.3 7B. For the Attack Success Rate (ASR), we use the pattern matching of Zou et al. (2023).

| Attack | ASR @ 60 s | Iter. / s |
|---|---|---|
| PGD | 87 % | 28.2 |
| GCG | 83 % | 0.3 |
| GBDA | 40 % | 29.3 |

Table 2: Ablations on Vicuna 1.3 7B, reporting mean Cross-Entropy with standard error.

| Var. length | Entropy proj. | Cross-Entropy |
|---|---|---|
| ✗ | ✗ | $0.092 \pm 0.014$ |
| ✓ | ✗ | $0.085 \pm 0.010$ |
| ✓ | ✓ | $0.078 \pm 0.009$ |

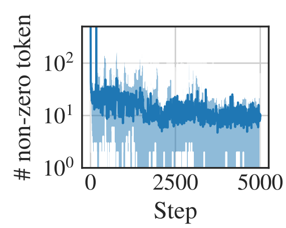
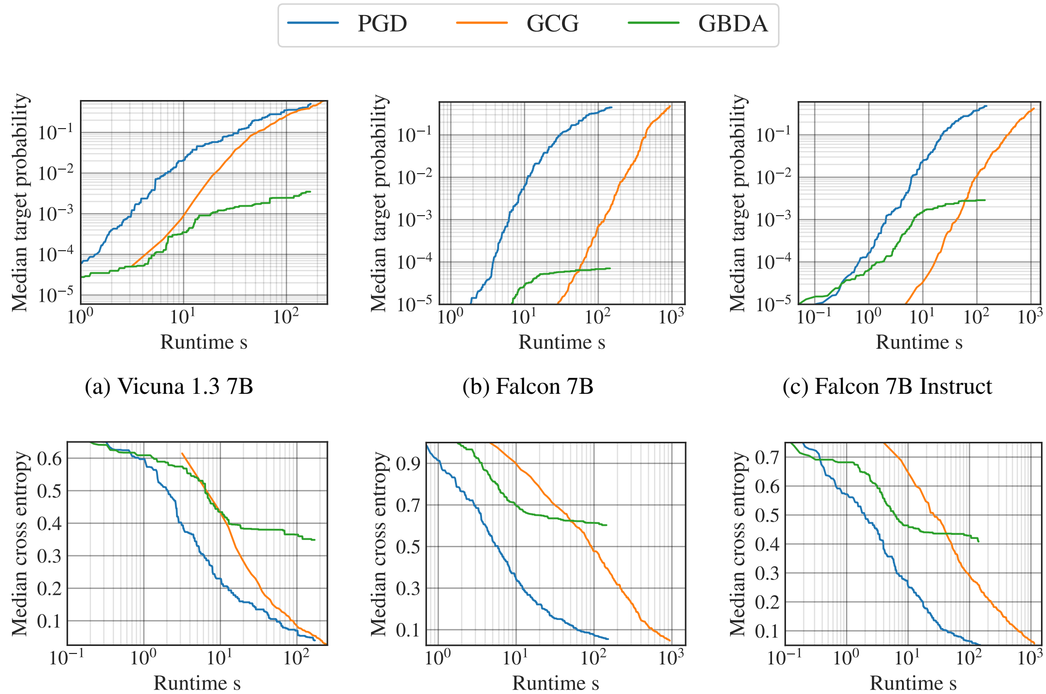



Figure 4: Average # of non-zero tokens (min/max shaded)

**Ablation and limitations.** From the ablations in Table 2 and main results in Fig. 3, we conclude that the choice of relaxation is responsible for the largest gain from GBDA to our PGD. The flexible length and entropy projection can help further improve the results. We expect the variable length of additional benefit for generating low perplexity prompts. In Fig. 4, we plot the number of non-zero tokens after the projections aggregated over the tokens in the adversarial suffix for an exemplary prompt on Falcon-7B-instruct. Our PGD successfully narrows the search space down from about 65,000 to 10 possibilities per token. Nevertheless, sometimes it can take many iterations until PGD finds a better prompt ($\tilde{\boldsymbol{x}}_{\text{best}}$ in Algorithm 1). In other words, finding effective discrete adversarial prompts appears much more challenging than with relaxed prompts (Schwinn et al., 2023).




(a) Vicuna 1.3 7B (b) Falcon 7B (c) Falcon 7B Instruct

(d) Vicuna 1.3 7B (e) Falcon 7B (f) Falcon 7B Instruct

Figure 3: Results on the behavior jailbreaking task of Zou et al. (2023)

## 5 Related Work

**Automatic red teaming** can be divided into LLM-based approaches (Perez et al., 2022; Mehrotra et al., 2023; Chao et al., 2023), discrete optimization (Wallace et al., 2021; Shin et al., 2020; Zou et al., 2023) and ordinary gradient-based optimization (Guo et al., 2021; Wen et al., 2023). While our PGD and GBDA (Guo et al., 2021) allow continuously relaxed tokens, PEZ (Wen et al., 2023) always discretizes the continuous token representation before probing the model. Moreover, automatic red teaming can also be understood as a conditional prompt generation (Kumar et al., 2022). Given system prompt and goal $\boldsymbol{x}'$, the conditional generation task is to choose adversarial suffix $\hat{\boldsymbol{x}}$, s.t. the goal in $\boldsymbol{y}'$ becomes likely.

**Projected Gradient Descent (PGD)** (Madry et al., 2018) is a simple yet effective method to obtain adversarial perturbations for (approximately) continuous domains like images. For example, PGD is heavily for adversarial training (Madry et al., 2018) or adaptive attacks on adversarial defenses (Tramer et al., 2020). There is a rich literature on PGD in the image domain, and we refer to Chen & Hsieh (2022); Serban et al. (2020) for an overview. PGD has also been applied successfully to discrete settings like graphs (Xu et al., 2019; Geisler et al., 2021; Gosch et al., 2023) or combinatorial optimization (Geisler et al., 2022), utilizing similar continuous relaxations. However, we are first to show that optimizing the continuously relaxed one-hot encodings is a practical choice in the domain of LLMs. Moreover, our entropy projection is a novel strategy for opposing the introduced relaxation error.

## 6 Discussion

In this work, we showed that PGD, the default choice for generating adversarial perturbations in other domains, can also be very effective and efficient for LLMs. Specifically, our PGD achieves the same attack strength as GCG up to one order of magnitude faster. The performance of our PGD stands in contrast to previous ordinary gradient-based optimization like GBDA, which is virtually unable to fool aligned LLMs.

## Acknowledgments

This material is based on work that is partially funded by Google.

## Ethics Statement

Adversarial attacks that can jailbreak even aligned LLMs can have a bad real-world impact. Moreover, efficient attacks are especially desired by real-world adversaries. Nevertheless, due to the white-box assumption that we know the model parameters and architecture details, we estimate the impact for good to outweigh the risks. If AI engineers and researchers are equipped with strong and efficient adversarial attacks, they may use them, e.g., for effective adversarial training and large-scale studies of their models – ultimately yielding more robust and reliable models in the real world along with an understanding of the remaining limitations. Additionally, we did not conduct experiments against AI assistants deployed for public use, like ChatGPT, Claude, or Gemini. Nor is our attack directly applicable to such models due to the white-box assumption.

## References

Ebtesam Almazrouei, Hamza Alobeidli, Abdulaziz Alshamsi, Alessandro Cappelli, Ruxandra Cojocaru, Mérouane Debbah, Étienne Goffinet, Daniel Hesslow, Julien Launay, Quentin Malartic, Daniele Mazzotta, Badreddine Noune, Baptiste Pannier, and Guilherme Penedo. The Falcon Series of Open Language Models, 2023. URL `http://arxiv.org/abs/2311.16867`.

Patrick Chao, Alexander Robey, Edgar Dobriban, Hamed Hassani, George J. Pappas, and Eric Wong. Jailbreaking Black Box Large Language Models in Twenty Queries, 2023. URL `http://arxiv.org/abs/2310.08419`.

Pin-Yu Chen and Cho-Jui Hsieh. *Adversarial Robustness for Machine Learning*. Academic Press, 2022. ISBN 978-0-12-824257-5.

John Duchi, Shai Shalev-Shwartz, Yoram Singer, and Tushar Chandra. Efficient projections onto the $l_1$ -ball for learning in high dimensions. In *Proceedings of the 25th international conference on Machine learning - ICML '08*, pp. 272–279, Helsinki, Finland, 2008. ACM Press. ISBN 978-1-60558-205-4. doi: 10.1145/1390156.1390191. URL `http://portal.acm.org/citation.cfm?doid=1390156.1390191`.

Simon Geisler, Tobias Schmidt, Hakan Sirin, Daniel Zügner, Aleksandar Bojchevski, and Stephan Günnemann. Robustness of Graph Neural Networks at Scale. *Neural Information Processing Systems, NeurIPS*, 2021.

Simon Geisler, Johanna Sommer, Jan Schuchardt, Aleksandar Bojchevski, and Stephan Günnemann. Generalization of Neural Combinatorial Solvers Through the Lens of Adversarial Robustness. In *International Conference on Learning Representations, ICLR*, 2022. URL `http://arxiv.org/abs/2110.10942`.

Lukas Gosch, Simon Geisler, Daniel Sturm, Bertrand Charpentier, Daniel Zügner, and Stephan Günnemann. Adversarial Training for Graph Neural Networks: Pitfalls, Solutions, and New Directions. In *Neural Information Processing Systems, NeurIPS*, 2023.

Chuan Guo, Alexandre Sablayrolles, Hervé Jégou, and Douwe Kiela. Gradient-based Adversarial Attacks against Text Transformers. In *Conference on Empirical Methods in Natural Language Processing*, pp. 5747–5757, Online and Punta Cana, Dominican Republic, 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.464. URL `https://aclanthology.org/2021.emnlp-main.464`.

Eric Jang, Shixiang Gu, and Ben Poole. Categorical Reparameterization with Gumbel-Softmax. In *International Conference on Learning Representations, ICLR*, 2016. URL `https://openreview.net/forum?id=rkE3y85ee`.

Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. In *International Conference on Learning Representations, ICLR*, 2015. URL http://arxiv.org/abs/1412.6980.

Sachin Kumar, Biswajit Paria, and Yulia Tsvetkov. Gradient-based Constrained Sampling from Language Models. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang (eds.), *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 2251–2277, Abu Dhabi, United Arab Emirates, 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.emnlp-main.144. URL https://aclanthology.org/2022.emnlp-main.144.

Raz Lapid, Ron Langberg, and Moshe Sipper. Open Sesame! Universal Black Box Jailbreaking of Large Language Models, 2023. URL http://arxiv.org/abs/2309.01446.

Xiaogeng Liu, Nan Xu, Muhao Chen, and Chaowei Xiao. AutoDAN: Generating Stealthy Jailbreak Prompts on Aligned Large Language Models, 2023. URL http://arxiv.org/abs/2310.04451.

Ilya Loshchilov and Frank Hutter. SGDR: Stochastic gradient descent with warm restarts. *International Conference on Learning Representations, ICLR*, pp. 1–16, 2017.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *International Conference on Learning Representations, ICLR*, pp. 1–28, 2018.

Mantas Mazeika, Long Phan, Xuwang Yin, Andy Zou, Zifan Wang, Norman Mu, Elham Sakhaee, Nathaniel Li, Steven Basart, Bo Li, David Forsyth, and Dan Hendrycks. HarmBench: A Standardized Evaluation Framework for Automated Red Teaming and Robust Refusal, 2024. URL http://arxiv.org/abs/2402.04249.

Anay Mehrotra, Manolis Zampetakis, Paul Kassianik, Blaine Nelson, Hyrum Anderson, Yaron Singer, and Amin Karbasi. Tree of Attacks: Jailbreaking Black-Box LLMs Automatically, 2023. URL http://arxiv.org/abs/2312.02119.

Ethan Perez, Saffron Huang, Francis Song, Trevor Cai, Roman Ring, John Aslanides, Amelia Glaese, Nat McAleese, and Geoffrey Irving. Red Teaming Language Models with Language Models, 2022. URL http://arxiv.org/abs/2202.03286.

Leo Schwinn, David Dobre, Stephan Günnemann, and Gauthier Gidel. Adversarial Attacks and Defenses in Large Language Models: Old and New Threats, 2023. URL http://arxiv.org/abs/2310.19737.

Alex Serban, Erik Poll, and Joost Visser. Adversarial Examples on Object Recognition: A Comprehensive Survey, 2020. URL http://arxiv.org/abs/2008.04094.

Taylor Shin, Yasaman Razeghi, Robert L. Logan IV, Eric Wallace, and Sameer Singh. AutoPrompt: Eliciting Knowledge from Language Models with Automatically Generated Prompts, 2020. URL http://arxiv.org/abs/2010.15980.

Christian Szegedy, W. Zaremba, Ilya Sutskever, Joan Bruna, D. Erhan, Ian J. Goodfellow, and R. Fergus. Intriguing properties of neural networks. *International Conference on Learning Representations, ICLR*, 2014.

Florian Tramer, Nicholas Carlini, Wieland Brendel, and Aleksander Madry. On Adaptive Attacks to Adversarial Example Defenses. *Neural Information Processing Systems, NeurIPS*, 33:1633–1645, 2020. URL https://proceedings.neurips.cc//paper_files/paper/2020/hash/11f38f8ecd71867b42433548d1078e38-Abstract.html.

Constantino Tsallis. Possible generalization of Boltzmann-Gibbs statistics. *Journal of Statistical Physics*, 52(1):479–487, 1988. ISSN 1572-9613. doi: 10.1007/BF01016429. URL https://doi.org/10.1007/BF01016429.

Eric Wallace, Shi Feng, Nikhil Kandpal, Matt Gardner, and Sameer Singh. Universal Adversarial Triggers for Attacking and Analyzing NLP, 2021. URL http://arxiv.org/abs/1908.07125.

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. URL https://arxiv.org/abs/2302.03668v2.

Nevan Wichers, Carson Denison, and Ahmad Beirami. Gradient-Based Language Model Red Teaming, 2024. URL http://arxiv.org/abs/2401.16656.

Kaidi Xu, Hongge Chen, Sijia Liu, Pin Yu Chen, Tsui Wei Weng, Mingyi Hong, and Xue Lin. Topology attack and defense for graph neural networks: An optimization perspective. *IJCAI International Joint Conference on Artificial Intelligence*, 2019-Augus:3961–3967, 2019. ISSN 9780999241141. doi: 10.24963/ijcai.2019/550.

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. URL http://arxiv.org/abs/2306.05685.

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, 2023. URL http://arxiv.org/abs/2310.15140.

Andy Zou, Zifan Wang, J. Zico Kolter, and Matt Fredrikson. Universal and Transferable Adversarial Attacks on Aligned Language Models, 2023. URL http://arxiv.org/abs/2307.15043.