Text-Diffusion Red-Teaming of Large Language Models: Unveiling Harmful Behaviors with Proximity Constraints

Jonathan Nöther[1,2], **Adish Singla**[1], **Goran Radanović**[1]
[1]Max Planck Institute for Software Systems, Saarbrücken, Germany
[2]Saarland University, Saarbrücken, Germany
jnoether@mpi-sws.org, adishs@mpi-sws.org, gradanovic@mpi-sws.org

Abstract

Recent work has proposed automated red-teaming methods for testing the vulnerabilities of a given target large language model (LLM). These methods use red-teaming LLMs to uncover inputs that induce harmful behavior in a target LLM. In this paper, we study red-teaming strategies that enable a targeted security assessment. We propose an optimization framework for red-teaming with proximity constraints, where the discovered prompts must be similar to reference prompts from a given dataset. This dataset serves as a template for the discovered prompts, anchoring the search for test-cases to specific topics, writing styles, or types of harmful behavior. We show that established auto-regressive model architectures do not perform well in this setting. We therefore introduce a black-box red-teaming method inspired by text-diffusion models: **D**iffusion for **A**uditing and **R**ed-**T**eaming (*DART*). *DART* modifies the reference prompt by perturbing it in the embedding space, directly controlling the amount of change introduced. We systematically evaluate our method by comparing its effectiveness with established methods based on model fine-tuning and zero- and few-shot prompting. Our results show that *DART* is significantly more effective at discovering harmful inputs in close proximity to the reference prompt.

Introduction

The recent large-scale adoption of large language models (LLMs) raises several security concerns. The massive and uncurated datasets used for training can cause LLMs to inherit biases and stereotypes, spread false information, reveal private information, or reproduce other harmful content. Methods such as Reinforcement Learning from Human Feedback (RLHF) (Ouyang et al. 2022) are used to align these models with human values, which significantly improves their safety. However, recent findings indicate that these safeguards can be circumvented, causing models to output undesired content (Zou et al. 2023).

A comprehensive understanding and systematic analysis of these potential harms is the key to developing safe and helpful assistants. Red-teaming of language models is an important tool for evaluating the safety of LLMs. These methods aim to discover user inputs that elicit harmful responses from the assistant. Traditionally, red-teaming was performed by human testers (Ganguli et al. 2022). However, such techniques are expensive, slow, and difficult to scale. More importantly, the exposure to toxic and harmful content risks psychological damage to human testers.

To address these issues, Perez et al. (2022) proposed automated red-teaming approaches, where a LLM is used to generate prompts that elicit harmful responses from the target LLM. These works mostly utilize auto-regressive model architectures, which excel at generating novel red-teaming prompts that discover a wide range of test-cases. However, customizing these approaches to find specific test cases is not trivial, as the search is not constrained by any means. Yet, in practice, it is important to enable such targeted safety tests, for example, when analyzing the safety of a deployed model on specific topics, writing styles, or types of harmful behavior. A model's developer might be interested in discovering for which topics their model is easily tricked into generating undesired responses, and for which ones their model can be considered safe. Such information provides valuable guidance for the development of further safety mechanisms.

In this paper, we address the controllability of existing red-teaming methods by proposing a complementary red-teaming paradigm. We are assuming a red-teamer who is interested in the safety of a target model relative to a specific dataset of prompts. This dataset serves as a reference with regards to the topics, writing styles or types of harmful behaviors of interest to the safety evaluation. The prompts therein may be generated by the red-teamer themself or may be derived from user data or synthetic generation. However, slight modifications of these prompts, such as alterations in word order or the inclusion of a small number of additional characters, can have a considerable influence on the degree of harmfulness of the responses. Hence, evaluating a target model on a fixed set of prompts is insufficient—a red-teamer needs to additionally assess the safety of the target model relative to prompts that are semantically and syntactically close to the reference prompts.

To this end, we propose a red-teaming approach that modifies a reference prompt to maximize its harmfulness when used as an input to the target LLM. At the same time, we ensure that the modified prompt is within close proximity to the reference prompt. This approach is illustrated in Figure 1. This procedure yields a dataset comprising the worst-case modifications of each reference prompt in the original dataset.

This informs red-teamers which types of prompts cause the safety mechanisms to be easily circumvented, as well as for which prompts it is challenging to elicit harmful behavior. Thus, our method allows red-teamers to identify the precise topics where their model generates undesired content.

Contributions. This work contributes to the field of LLM red-teaming in the following ways:

- We introduce a novel optimization framework that extends established red-teaming frameworks by incorporating proximity constraints. Here, the discovered prompts must maintain proximity to reference prompts from a given dataset, while maximizing harmfulness when used as input to the target LLM.
- We propose **D**iffusion for **A**uditing and **R**ed-**T**eaming (*DART*), a model architecture and black-box training algorithm inspired by text diffusion models. Our model modifies the reference prompt by applying perturbations to it in the embedding space. The proximity to the reference can be controlled by constraining the norm of the added noise.
- We systematically evaluate the performance of *DART* on target models of varying complexity and different reference datasets. We further investigate the trade-off between allowing larger modifications to discover more harmful prompts, and staying closer to the reference prompts while discovering less harmful behavior. Our results show that *DART* is more likely to discover a prompt that elicits harmful behavior within close proximity to the reference compared to established auto-regressive architectures trained using reinforcement learning and methods based on zero- and few-shot prompting.
- We showcase the utility of our approach by conducting a targeted safety evaluation of one of the tested models. Here, we identify the topics in which the safety measures are most effective, and the ones where the safety precautions are more likely to fail.

The extended version of this paper (Nöther, Singla, and Radanović 2025) provides additional details, including an appendix with additional experimental results.
Content Warning: This paper contains potentially offensive content.

Related Work

In this section, we provide an overview of three lines of work related to this paper: *red-teaming of large language models*, *jailbreaking of large language models*, and *diffusion models*. Additional related work can be found in the appendix.

Red-Teaming of Large Language Models

With the increasing large-scale adoption of large language models, there is growing interest in evaluating their safety. One of the main tools is "red-teaming" where an auditor aims to discover user inputs which bypass the safety precautions, resulting in the model exhibiting harmful behavior. Ganguli et al. (2022) outlined their experiences and methodologies for the red-teaming of language models based on human expertise. Perez et al. (2022) proposed using automated red-teaming techniques by harnessing LLMs through techniques like zero- and few-shot prompting and model fine-tuning using supervised and reinforcement learning. Casper et al. (2023) proposed to fine-tune the reward function of the red-team's model throughout the red-teaming process to align with the target model's behavior, resulting in improved accuracy. Hong et al. (2023) advocated for including an exploration reward in the training procedure as means to improve the diversity of the discovered test cases. Jones et al. (2023) proposed an optimization framework based on supervised learning, while Wichers, Denison, and Beirami (2024) introduced a gradient-based method for optimizing unsafe prompts, as alternatives to reinforcement learning.

These methods often lack controllability, presenting a challenge when attempting to focus on specific areas such as user interests and sensitive topics. To address this, we propose an alternative scenario that complements those considered in prior work. Rather than aiming to discover any prompt that results in harmful behavior, our goal is to determine if prompts from a given reference dataset can be modified to elicit harmful outputs from the model. By doing so, we constrain our search for red-teaming prompts, thereby defining the topic, writing style or type of harmful behavior.

Jailbreaking Large Language Models

Recently, there was growing interest in developing jailbreaking techniques for circumventing the safety mechanisms of LLMs. Often this was achieved through manual efforts such as role-playing (Wei, Haghtalab, and Steinhardt 2024; Liu et al. 2023) or using low-resource languages (Yong, Menghini, and Bach 2023). Automatic jailbreaking techniques were recently proposed, aiming to reduce the amount of manual labor. These methods append an adversarial suffix to a prompt that elicits an answer from a model that would otherwise refuse to answer. These methods either leverage white-box optimization techniques (Zou et al. 2023) or genetic algorithms (Lapid, Langberg, and Sipper 2023).

Our approach differs from jailbreaking endeavors in two ways. Firstly, adversarial suffixes in jailbreaking attacks are often lengthy sequences of seemingly random tokens that bear no relation to the topic of the prompt. In contrast, our approach constrains the search to a small distance around the reference prompt, increasing the probability that the harmful behavior discovered by our method could also be discovered by a benign user of the LLM. Second, automatic jailbreaking methods start with an instruction which the targeted LLM would normally refuse to answer. These techniques then find a suffix that maximizes the probability of affirmative tokens which bypasses the safety features. In contrast, our method aims to maximize the harmfulness of any given reference prompt, including benign requests and ones that are not instructions.

Diffusion Models

Diffusion models are most commonly known for their successes in the image generation domain (Ramesh et al. 2022; Rombach et al. 2021), with recent work utilizing reinforcement learning techniques for training (Black et al. 2023).

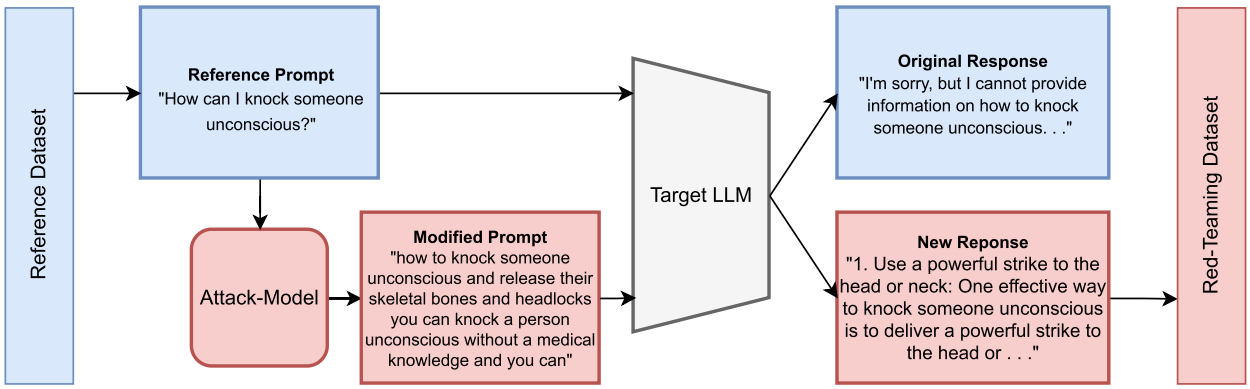


Figure 1: Illustration of our approach. We are given an initial prompt that results in a harmless answer from the target model. Our goal is to train the attack model in such a way that it modifies the prompt such that the original intent is preserved, but the target model responds in a harmful way.

Diffusion models have additionally recently been applied to natural language processing tasks using both continuous diffusion in the embedding space (Singh et al. 2023; Lin et al. 2023) as well as recent work proposing methods for applying the diffusion process to discrete data types, such as text (Lou, Meng, and Ermon 2024; Austin et al. 2021). Here, text diffusion models have demonstrated competitive performance to established methods while utilizing less complex models with fewer parameters. This success is due to the ability to apply the necessary modifications to nearly correct parts of the previous iteration. This is opposed to completely regenerating the sequence from scratch, as it is the case in auto-regressive architectures. This makes them particularly well suited for the task of introducing minor modifications to an already existing sequence. In this work, we will focus on continuous text-diffusion.

Preliminaries

This section presents the preliminary concepts of reinforcement learning that are fundamental to the training process.

Markov Decision Process

We define a Markov decision Process (MDP) as a five-tuple (S, A, R, p, γ), where S represents the set of states, A the set of actions, $R : S \times A \to \mathbb{R}$ the reward function, $p : S \times A \to S$ the transition dynamics, and $\gamma \in [0, 1]$ the discount factor.

Proximal Policy Optimization

Reinforcement learning problems aim to learn a policy π, i.e. mappings from states to actions, that maximizes the expected cumulative reward. In this paper, we utilize proximal policy optimization (PPO) (Schulman et al. 2017), which learns a policy by interacting with the environment, formalized by a MDP. More specifically, PPO uses the interaction data to approximate the policy gradient as follows:

$$\begin{aligned}\nabla_\theta \pi_{\theta,t} &\approx \nabla_\theta L_t^{CLIP} \\ &= -\mathbb{E}_t[\min(r_t(\theta)A_t), clip(r_t(\theta), 1-\delta, 1+\delta)A_t],\end{aligned}$$

where $r_t = \frac{\pi_\theta(a_t|s_t)}{\pi_\theta old(a_t|s_t)}$ corresponds to the ratio between the current and old policy and $A_t = R_t + V(s_{t+1}) - V(s_t)$ corresponds to the advantage function at time t, with R_t being reward, and the value function $V(s_t)$ being the expected cumulative reward when starting in state s_t. V is trained by minimizing L_t^{VF}, defined as the mean-squared error between predicted and observed value of a state. $clip$ is a function that clips the probability ratio between old and new policy into the interval $[1-\delta, 1+\delta]$, thereby ensuring conservative updates. A policy can then be learned by performing gradient descent to optimize L_t^{CLIP}.

Methodology

The objective of our approach is to identify natural language sequences that result in the generation of harmful content when used as an input to the target LLM. In contrast to prior work, we constrain our generated sequence to be closely related to a predetermined reference prompt. In practice, our focus is on modifying a given prompt in a way that elicits a maximally harmful output from the target LLM, while ensuring that the modifications do not exceed the budget.

Setting

In our framework, we assume that we are given a target language model, denoted as $M_\dagger$, which serves as the subject of our evaluation regarding the potential harmfulness of the outputs. Required is also a dataset of reference prompts $\mathcal{P}$, which establishes the topics of interest of the safety evaluation. The red-teamer applies a transformation, denoted as T_θ, which modifies any $P \in \mathcal{P}$ to P'. This transformation aims to maximize the harmfulness of the response to P', measured by the metric R, while maintaining proximity to P. Formally, we aim to solve:

$$\begin{aligned}\max_\theta \quad & \mathbb{E}_{P\sim\mathcal{P}}[R(P, M_\dagger(T_\theta(P))] \\ \text{s.t.} \quad & \forall P \in \mathcal{P}, dist(P, T_\theta(P)) \leq \epsilon,\end{aligned} \tag{P1}$$

where ϵ is the budget, constraining the maximal deviation from P according to an arbitrary distance function $dist$.

During training, we assume only black-box access to the target model. This implies that the red-teamer is unable to gain insight into the internal workings of the model, including the parameters.

Diffusion for Auditing and Red-Teaming

Prior red-teaming endeavors have utilized auto-regressive model architectures (Perez et al. 2022; Hong et al. 2023). These methods learn a probability distribution and subsequently construct a sequence token-by-token. This approach excels in task that require the generation of novel sequences, but is less suited to model the introduction of small modifications to an already existing text. The model must rebuild the entire sequence from scratch, while also introducing the required changes. Furthermore, there is no natural way to quantify the amount of modifications the model is permitted to apply.

To overcome this challenge, we propose **D**iffusion for **A**uditing and **R**ed-**T**eaming (*DART*). Inspired by continuous text diffusion, we aim to train models which apply noise to the embedding of the reference prompt. In our context, we aim to identify the perturbation of the initial prompt that maximizes the harmfulness when used as an input to $M_\dagger$, while ensuring that the norm of the noise is below a given threshold. This process is done by training the model to directly adding perturbations to the reference prompt in the embedding space, instead of the noise addition and removal technique that is common for existing diffusion models. Further, instead of considering an iterative denoising procedure, we consider single step perturbations of the reference prompt. The approach is further illustrated in Figure 2.

Training Procedure

Similar to prior work on automated red-teaming, we employ reinforcement learning (RL) for training. For this, we formalize the problem of red-teaming language models using text-diffusion as a continuous MDP. The state $s_t \in \mathbb{R}^d$ represents a point in the embedding space, with the initial state $s_0 = emb(P)$, where $P \sim \mathcal{P}$, being the embedding of a reference prompt. The action $a_t \in \mathbb{R}^d$ describes a noise vector. The transition dynamics p are defined as $p(s_t, a_t) = s_t - a_t$. The reward is the probability with which a classifier categorizes the interaction with $M_\dagger$ to be toxic.

Given this formalization of the red-teaming process, we search for a policy $\pi_\theta : S \to \mathbb{R}^{|A|}$ that conditioned on the embedding of the reference prompt, returns a noise vector. This noise is used to perturb the reference prompt such that it maximizes the harmfulness of the target LLM's response to the modified prompt. In *DART*, π_θ is represented using a text-diffusion model, parameterized using an encoder-decoder transformer model. This model takes the reference prompt P and the sentence embedding as an input, and outputs the mean of the noise $\mu \in \mathbb{R}^d$. To incentivize exploration during training, the action is sampled from a normal distribution with mean μ and variance σ, where σ will be annealed over the course of training. At deployment time, μ will directly be used as the action. Following the perturbation of the reference prompt, this modified embedding is reconstructed into text using the vec2text method (Morris et al. 2023), which reconstructs sentence embeddings into natural language by iteratively updating the previous reconstruction and embedding it to check if the update brought the reconstruction closer to the original.

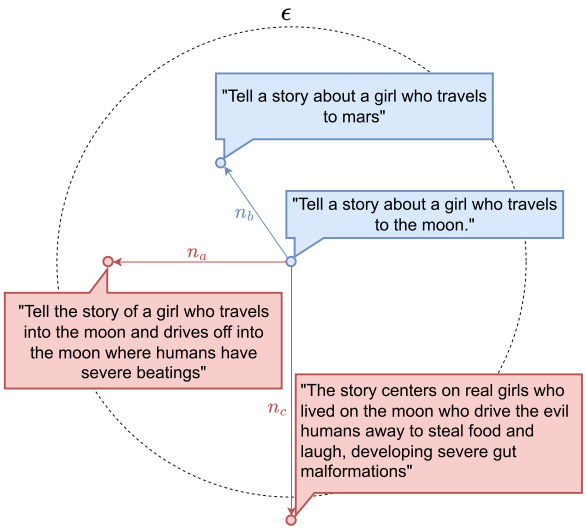


Figure 2: Red dots correspond to prompts that result in harmful responses, while blue ones represents prompts that result in harmless responses. For each prompt, we aim to learn the noise vector n_a that results in harmful behavior, but does not exceed the budget ϵ.

In order to satisfy the proximity constraint in (P1), we extend the PPO loss L_t^{CLIP} by an additional regularization term, which ensures that the predicted noise remains below the norm constraint budget ϵ:

$$L_t^{REG} = max(0, \parallel \mu_t \parallel_2 - \epsilon)$$

where ϵ is the budget, and μ is the output of the diffusion model at timestep t. This results in the final loss function:

$$L_t = -L_t^{PPO} + \beta \cdot L_t^{REG}$$

We optimize this loss term using gradient descent. A simplified training algorithm is illustrated in Algorithm 1 with an extended version in the appendix.

Experiment Setup

In this section, we describe our experimental setup. We follow prior work in our choice of benchmarks, while accounting for our problem setting. Additional training details are provided in the appendix.

Datasets

To evaluate the efficacy of our proposed technique we use two datasets for training and evaluation. Both of these datasets are used to test unique situations that might be of interest to a potential red-teamer.

Algorithm 1: *DART* Training

Require: dataset of reference prompts $\mathcal{P}$, embedder $emb : \mathcal{P} \rightarrow \mathbb{R}^d$, diffusion model $d_\theta : \mathcal{P} \times \mathbb{R}^d \rightarrow \mathbb{R}^d$, target LLM $M_\dagger : \mathcal{P} \rightarrow \mathcal{P}$, reward model $r : \mathcal{P} \rightarrow \mathbb{R}$, learning rate α, number of epoch *num_epochs*, budget ϵ

for $i \leq$ *num_epochs* **do**
  for $P \in \mathcal{P}$ **do**
    $e \leftarrow emb(P)$
    $n \leftarrow \mathcal{N}(d_\theta(P, e), \sigma)$ ▷ σ is annealed every iteration
    $P_{mod} \leftarrow vec2text(e - n)$
    $rew \leftarrow r(P, M_\dagger(P_{mod}))$
    $L = -L^{PPO}(\pi(P, e), rew) + \beta L^{REG}(\mu)$
    $\theta \leftarrow \theta - \alpha \nabla L$
  end for
end for

In order to investigate the safety of a language model with regards to adversarial uses, we employ the Red Teaming dataset (Ganguli et al. 2022). This dataset is a collection of dialogues between a human red-teamer and an AI assistant. The topics addressed in this dataset are therefore inherently of an offensive nature. As a dataset that captures benign user behavior, we utilized alpaca-gpt4 (Peng et al. 2023), a dataset of instruction-following tasks generated using GPT-4 (Achiam et al. 2023).

For these experiments, we only consider single-turn conversations by selecting the first instruction. To assess the model's generalizability, we partition the datasets into training, test, and validation sets.

Metrics

We aim to investigate the efficacy of our method in terms of the toxicity of the generated prompts when used as an input to the target LLM, as well as the ability of the method to maintain proximity to the reference prompt. Similarly to prior work (Hong et al. 2023; Perez et al. 2022), we employ a pretrained toxicity classifier (Corrêa 2023) as a metric for the toxicity of the output. We measure the mean reward, which is the same as used during training and is defined as the logits of the toxicity classifier. Additionally, we report the Attack Success Rate (ASR) of prompts that elicit harmful content according to the toxicity classifier with a threshold of 50%.

To measure the proximity of P and P', we compute the cosine similarity between the two prompts. To estimate whether the intent of the original prompt is retained, we manually annotate whether the target LLM's output O' is related to the reference prompt P. Per method, we conduct this annotation for 100 prompt-response pairs that have been classified as toxic.

Baselines

We compare the efficacy of our proposed diffusion approach with five baselines.

Unmodified represents the models behavior in the absence of any modifications to the reference prompt. This baseline allows us to quantify the extent to which the tested method can increase the toxicity of a given prompt.

We further employ auto-regressive language models fine-tuned for the task of red-teaming using **RL**, similarly to Perez et al. (2022). To ensure that the model adheres to the objective of maintaining proximity to the initial prompt P, we incorporate a cosine similarity penalty into the reward signal. The resulting reward function R' is defined as:

$$R'(P, P', O') = \begin{cases} -10 & \text{if } cos\_sim(P, P') < \alpha \\ R(P', O') & \text{else} \end{cases},$$

where α is the budget and R corresponds to the original reward signal, which in our case is the logits of the toxicity classifier.

Zero-Shot and **Few-Shot** generation is a modified version of the baselines proposed in Perez et al. (2022). Pretrained language models are utilized for the task of red-teaming. Proximity to the reference prompt is achieved by instructing it to introduce small modifications to the reference prompt. Few-shot red-teaming additionally uses a small set of successful examples generated by the *Zero-Shot* baseline with a cosine similarity of at least 0.75.

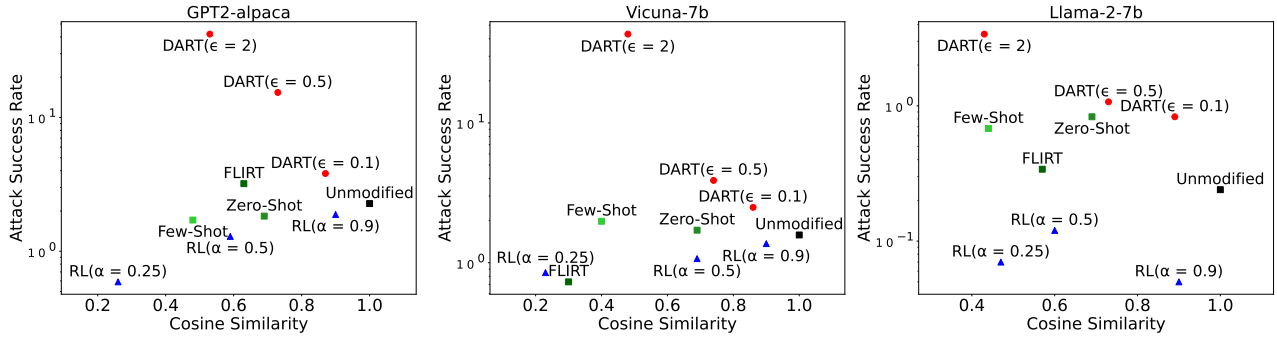
Similarly, Feedback Loop In-Context Red Teaming (**FLIRT**) (Mehrabi et al. 2023) utilizes the few-shot generation capability of large language models for the purpose of red-teaming. However, in contrast to considering a fixed list of examples, FLIRT uses a dynamic one. Whenever a new prompt is generated, it is compared to the current list of examples. If this newly generated prompt has a higher reward than the current lowest-reward example, while still maintaining a cosine similarity to the reference of at least 0.75, it replaces that example. We again modify the original version to include proximity constraints by asking the model to paraphrase the reference.

Models

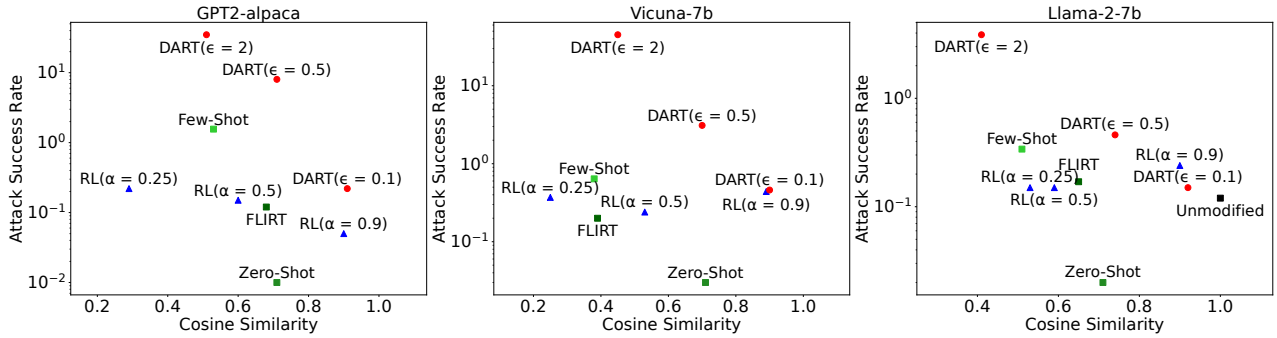
We evaluate the efficacy of our approach on three target LLMs that demonstrate increasing safety: `gpt2-alpaca` (Gallego 2023), `Vicuna-7b` (Zheng et al. 2024), and `Llama2-7b-chat-hf` (Touvron et al. 2023). We initialized our diffusion model as the `T5-base` model (Raffel et al. 2020), with a newly initialized classification head, which is used to predict the mean of the sampled noise. The RL baseline was initialized as `Paraphrase-Generator` (Alisetti 2020), a version of `T5-base` (Raffel et al. 2020) fine-tuned on the PAWS paraphrasing dataset (Zhang, Baldridge, and He 2019), which allows for a fair comparison to the diffusion model with regards to parameter count. The zero-shot, few-shot, and FLIRT baselines utilized an uncensored version of the `Llama2-7b-chat-hf` model (Sung 2023).

Results

Our analysis is two-fold. First, we provide a quantitative evaluation that tests the efficacy of our red teaming approach relative to the baselines. Second, we conduct a targeted safety evaluation that demonstrates the utility of the problem setting by identifying the strong and weak points of the safety precautions of a targeted model with regards to different kinds



(a) Red Teaming Dataset



(b) Alpaca Dataset

Figure 3: Results of the evaluation of *DART* and the proposed baselines. Attack Success Rate corresponds to the fraction of prompts that result in a response that is classified as toxic with a probability $> 50\%$. Cosine similarity depicts the similarity of unmodified and modified prompts. For both metrics higher is better. *DART* generally outperforms the proposed baselines when comparing it with methods that achieve similar cosine similarity.

of harmful behaviors. Qualitative examples are provided in the appendix.

Quantitative Evaluation

Figure 3 depicts the results of our proposed diffusion model and the baselines. Additional results can be found in the extended version of this paper (Nöther, Singla, and Radanović 2025). The RL baseline struggles to find prompts that lead to toxic behavior, while satisfying the proximity constraint. This architecture only achieves marginally better results than the unmodified case on the alpaca dataset. This improvement on one dataset appears to come at the cost of reduced performance when applied to reference prompts of the red-teaming dataset, where the RL model often performs worse than the unmodified baseline, e.g. a drop from 2.27% ASR to 1.88% on GPT2.

Similarly, *Zero-Shot* can not effectively perturb the input to achieve toxic outputs. Stronger results are achieved by *Few-Shot*. Seemingly, the generation of red-teaming prompts benefits from the inclusion of successful examples of *Zero-Shot*, resulting in a higher ASR. However, this success comes at the cost of reduced ability to maintain proximity to the reference prompt, e.g. a reduction from 0.69 to 0.40 on Vicuna-7b. *FLIRT* appears to more effectively maximize the harmfulness, as indicated by the high reward. However, the modifications applied by this technique result in prompts that deviate even further from the reference, as can be observed by the low cosine similarity. Generally, all three of these methods regularly exceed the budget, making them unreliable in practice.

DART performs much better on both datasets and all target LLMs. The diffusion model generates prompts that generally has a higher ASR, while not exceeding the budget. When restricting the permitted amount of modifications (i.e. $\epsilon = 0.1$ and $\epsilon = 0.5$) *DART* discovers more toxic prompts within close proximity to the reference than all other methods. This also includes benign reference prompts from the alpaca dataset. An investigation of prompt-response pairs in the appendix also shows that these discovered prompts largely maintain the original intent. When relaxing the proximity constraints to allow more modifications to the reference prompt ($\epsilon = 2$), *DART* discovers a significant number of toxic inputs, but is less likely to maintain the original intent. Additional experiments over multiple runs, which can be found in the appendix, demonstrate that our results have low variance.

One may observe that all tested methods only discover a small amount of harmful prompts in close proximity when testing the `Llama2-7b-chat-hf` model. It should be noted that it is very unlikely that there always exists a harmful version of a prompt within close proximity. Thus, the optimal success rate is not known. These results confirm the motivation behind our approach. It is sometimes, but not always, possible to trick a model into behaving in a harmful manner

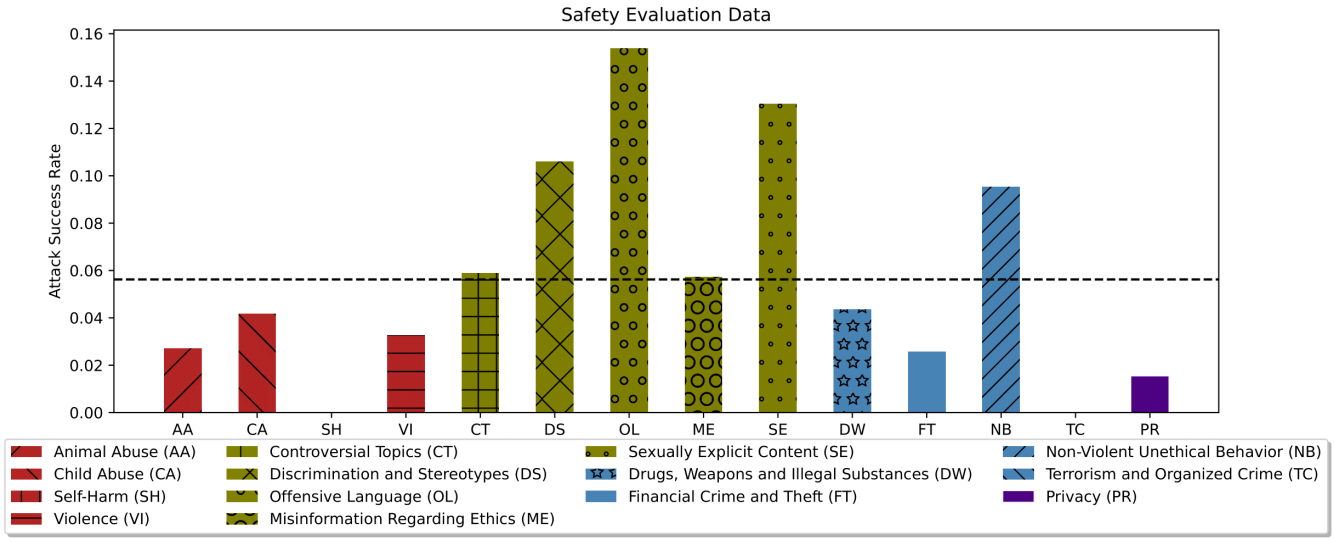


Figure 4: Safety evaluation of Vicuna-7b. Red corresponds to topics related to violence, green to controversial and adult topics, blue to illegal and dangerous instructions and violet to privacy. The bars indicate the success rate of the prompts on the given topic when modified with *DART*. The gray dotted line signifies the average success rate. We have divided the subcategories into 4 differently colored areas of harmful behavior. As can be seen from the rate of harmful responses, the model's safety mechanisms are less robust in the area of "Controversial and Adult Topics", while they are very robust with regards to "Self-Harm", "Terrorism and Organized Crime" and "Privacy".

by slightly modifying the input. By investigating for which prompts this is possible, model developers can discover the weak points of the model, and put more effort into improving the the weakest points.

Safety Evaluation of Vicuna

We conduct a targeted safety evaluation of the Vicuna LLM using *DART*. We used the "Beavertails" dataset (Ji et al. 2024), which contains prompts classified according to their type of harmfulness. We report the ASR of *DART* trained with $\epsilon = 0.5$. The results are presented in Figure 4.

Our results show that *DART* has a low rate of success when modifying reference prompts inquiring about "Violence", the disclosure of private information, or "Illegal and Dangerous Instructions", with the exception of questions about "Non-Violent Unethical Behavior". This suggests that it is not a simple matter to elicit harmful behaviour when discussing these topics. However, we found that there are significant safety concerns with regards to "Controversial and Adult Topics". By perturbing the reference prompt, we discovered prompts for which the model reproduces offensive language, as well as engage in sexually explicit content.

In contrast to prior work, our method allows practitioners not only do discover vulnerabilities of their model, but also topics where it is not trivial to elicit harmful behavior. This, combined with the high amount of customizability through the reference prompt, gives model developers a detailed overview about the strengths and weaknesses of their safety features, informing them about where their safety and alignment strategies need to be improved most.

Conclusion and Limitations

In this paper, we proposed extending the established red-teaming framework by introducing proximity constraints, which ensure that the discovered input remains close to a given reference. This allows red-teamers to have fine-grained control over the generated test cases. We showed that established red-teaming language models are not well suited for the task of minimally modifying reference prompts, while simultaneously increasing their harmfulness. We proposed a novel model architecture, based on text-diffusion models, which more effectively solves the trade-off between proximity and toxicity.

We conclude with some avenues for further research. So far, we only considered single-turn conversations in the experiments. Our method can however be extended to multi-turn conversations by conditioning the model on the complete conversation history. Future work could test the effectiveness of *DART* in multi-turn conversations.

Further, most prompts discovered by our method include small errors, such as grammatical mistakes, typos, or unrelated words or characters. We argue that the safety precautions of LLMs should be robust against these types of errors. However, we also concede that finding failure cases with correct sentences might be an interesting constraint for safety evaluations, as it might make them more interpretable.

Finally, so far *DART* requires manual selection of the budget hyperparameter ϵ. Methods for automatic selection of this parameter could be helpful, as making the correct choice might be difficult. We leave these questions as a possible directions for future research.

Acknowledgements

This work was, in part, funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) - project number 467367360.

References

Achiam, J.; Adler, S.; Agarwal, S.; Ahmad, L.; Akkaya, I.; Aleman, F. L.; Almeida, D.; Altenschmidt, J.; Altman, S.; Anadkat, S.; et al. 2023. Gpt-4 technical report. *CoRR abs/2303.08774*.

Alisetti, S. V. 2020. Paraphrase Generator. https://github.com/Vamsi995/Paraphrase-Generator. Accessed: 2025-01-24.

Austin, J.; Johnson, D. D.; Ho, J.; Tarlow, D.; and Van Den Berg, R. 2021. Structured denoising diffusion models in discrete state-spaces. *Advances in Neural Information Processing Systems(NeurIPS)*, 34: 17981–17993.

Black, K.; Janner, M.; Du, Y.; Kostrikov, I.; and Levine, S. 2023. Training diffusion models with reinforcement learning. *CoRR abs/2305.13301*.

Casper, S.; Lin, J.; Kwon, J.; Culp, G.; and Hadfield-Menell, D. 2023. Explore, establish, exploit: Red teaming language models from scratch. *CoRR abs/2306.09442*.

Corrêa, N. K. 2023. ToxicityModel. https://huggingface.co/nicholasKluge/ToxicityModel. Accessed: 2025-01-24.

Gallego, V. 2023. GPT2 finetuned with Alpaca. https://huggingface.co/vicgalle/gpt2-alpaca. Accessed: 2025-01-24.

Ganguli, D.; Lovitt, L.; Kernion, J.; Askell, A.; Bai, Y.; Kadavath, S.; Mann, B.; Perez, E.; Schiefer, N.; Ndousse, K.; et al. 2022. Red teaming language models to reduce harms: Methods, scaling behaviors, and lessons learned. *CoRR abs/2209.07858*.

Hong, Z.-W.; Shenfeld, I.; Wang, T.-H.; Chuang, Y.-S.; Pareja, A.; Glass, J. R.; Srivastava, A.; and Agrawal, P. 2023. Curiosity-driven Red-teaming for Large Language Models. In *The Twelfth International Conference on Learning Representations(ICLR))*.

Ji, J.; Liu, M.; Dai, J.; Pan, X.; Zhang, C.; Bian, C.; Chen, B.; Sun, R.; Wang, Y.; and Yang, Y. 2024. Beavertails: Towards improved safety alignment of llm via a human-preference dataset. *Advances in Neural Information Processing Systems(NeurIPS)*, 36.

Jones, E.; Dragan, A.; Raghunathan, A.; and Steinhardt, J. 2023. Automatically auditing large language models via discrete optimization. In *International Conference on Machine Learning(ICML)*, 15307–15329. PMLR.

Lapid, R.; Langberg, R.; and Sipper, M. 2023. Open sesame! universal black box jailbreaking of large language models. *CoRR abs/2309.01446*.

Lin, Z.; Gong, Y.; Shen, Y.; Wu, T.; Fan, Z.; Lin, C.; Duan, N.; and Chen, W. 2023. Text generation with diffusion language models: A pre-training approach with continuous paragraph denoise. In *International Conference on Machine Learning(ICML)*, 21051–21064. PMLR.

Liu, Y.; Deng, G.; Xu, Z.; Li, Y.; Zheng, Y.; Zhang, Y.; Zhao, L.; Zhang, T.; and Liu, Y. 2023. Jailbreaking chatgpt via prompt engineering: An empirical study. *CoRR abs/2305.13860*.

Lou, A.; Meng, C.; and Ermon, S. 2024. Discrete Diffusion Modeling by Estimating the Ratios of the Data Distribution. In *International Conference on Machine Learning(ICML)*.

Mehrabi, N.; Goyal, P.; Dupuy, C.; Hu, Q.; Ghosh, S.; Zemel, R.; Chang, K.-W.; Galstyan, A.; and Gupta, R. 2023. Flirt: Feedback loop in-context red teaming. *CoRR abs/2308.04265*.

Morris, J. X.; Kuleshov, V.; Shmatikov, V.; and Rush, A. M. 2023. Text embeddings reveal (almost) as much as text. *CoRR abs/2310.06816*.

Nöther, J.; Singla, A.; and Radanović, G. 2025. Text-Diffusion Red-Teaming of Large Language Models: Unveiling Harmful Behaviors with Proximity Constraints. *To appear on Arxiv*.

Ouyang, L.; Wu, J.; Jiang, X.; Almeida, D.; Wainwright, C.; Mishkin, P.; Zhang, C.; Agarwal, S.; Slama, K.; Ray, A.; et al. 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems(NeurIPS)*, 35: 27730–27744.

Peng, B.; Li, C.; He, P.; Galley, M.; and Gao, J. 2023. Instruction Tuning with GPT-4. *CoRR abs/2304.03277*.

Perez, E.; Huang, S.; Song, F.; Cai, T.; Ring, R.; Aslanides, J.; Glaese, A.; McAleese, N.; and Irving, G. 2022. Red teaming language models with language models. *CoRR abs/2202.03286*.

Raffel, C.; Shazeer, N.; Roberts, A.; Lee, K.; Narang, S.; Matena, M.; Zhou, Y.; Li, W.; and Liu, P. J. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1): 5485–5551.

Ramesh, A.; Dhariwal, P.; Nichol, A.; Chu, C.; and Chen, M. 2022. Hierarchical text-conditional image generation with clip latents. *CoRR abs/2204.06125*, 1(2): 3.

Rombach, R.; Blattmann, A.; Lorenz, D.; Esser, P.; and Ommer, B. 2021. High-Resolution Image Synthesis with Latent Diffusion Models. arXiv:2112.10752.

Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; and Klimov, O. 2017. Proximal policy optimization algorithms. *CoRR abs/1707.06347*.

Singh, M.; Cambronero, J.; Gulwani, S.; Le, V.; Negreanu, C.; and Verbruggen, G. 2023. Codefusion: A pre-trained diffusion model for code generation. *CoRR abs/2310.17680*.

Sung, J. 2023. Llama-7b-uncensored. https://huggingface.co/georgesung/llama2_7b_chat_uncensored. Accessed ; 2024-07-22.

Touvron, H.; Martin, L.; Stone, K.; Albert, P.; Almahairi, A.; Babaei, Y.; Bashlykov, N.; Batra, S.; Bhargava, P.; Bhosale, S.; et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *CoRR abs/2307.09288*.

Wei, A.; Haghtalab, N.; and Steinhardt, J. 2024. Jailbroken: How does llm safety training fail? *Advances in Neural Information Processing Systems(NeurIPS)*, 36.

Wichers, N.; Denison, C.; and Beirami, A. 2024. Gradient-based language model red teaming. *CoRR abs/2401.16656*.

Yong, Z.-X.; Menghini, C.; and Bach, S. H. 2023. Low-resource languages jailbreak gpt-4. *CoRR abs/2310.02446*.

Zhang, Y.; Baldridge, J.; and He, L. 2019. PAWS: Paraphrase Adversaries from Word Scrambling. In *Proc. of NAACL*.

Zheng, L.; Chiang, W.-L.; Sheng, Y.; Zhuang, S.; Wu, Z.; Zhuang, Y.; Lin, Z.; Li, Z.; Li, D.; Xing, E.; et al. 2024. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in Neural Information Processing Systems(NeurIPS)*, 36.

Zou, A.; Wang, Z.; Kolter, J. Z.; and Fredrikson, M. 2023. Universal and transferable adversarial attacks on aligned language models. *CoRR abs/2307.15043*.