The Thirty-Ninth AAAI Conference on Artificial Intelligence (AAAI-25)

DiveR-CT: Diversity-enhanced Red Teaming Large Language Model Assistants with Relaxing Constraints

Andrew Zhao[1*] **, Quentin Xu**[1]**, Matthieu Lin**[1]**, Shenzhi Wang**[1]**,**
Yong-Jin Liu[1]**, Zilong Zheng**[2]**, Gao Huang**[1]
[1] Tsinghua University
[2] Beijing Institute for General Artificial Intelligence (BIGAI)
zqc21@mails.tsinghua.edu.cn
zlzheng@bigai.ai
gaohuang@tsinghua.edu.cn

Abstract

Recent advances in large language model assistants have made them indispensable, raising significant concerns over managing their safety. Automated red teaming offers a promising alternative to the labor-intensive and error-prone manual probing for vulnerabilities, providing more consistent and scalable safety evaluations. However, existing approaches often compromise diversity by focusing on maximizing attack success rate. Additionally, methods that decrease the cosine similarity from historical embeddings with semantic diversity rewards lead to novelty stagnation as history grows. To address these issues, we introduce DiveR-CT, which relaxes conventional constraints on the objective and semantic reward, granting greater freedom for the policy to enhance diversity. Our experiments demonstrate DiveR-CT's marked superiority over baselines by 1) generating data that perform better in various diversity metrics across different attack success rate levels, 2) better-enhancing resiliency in blue team models through safety tuning based on collected data, 3) allowing dynamic control of objective weights for reliable and controllable attack success rates, and 4) reducing susceptibility to reward overoptimization. Overall, our method provides an effective and efficient approach to LLM red teaming, accelerating real-world deployment. ⚠ WARNING: This paper contains examples of potentially harmful text.

Project Page — https://andrewzh112.github.io/diver-ct

1 Introduction

Deploying large language model (LLM) assistants often requires extensive testing on its output behavior to meet societal standards. One *de facto* paradigm to validate model integrity, robustness, and safety is using red teaming, where a group of experts (the "red team") proactively identify and mitigate potential issues of LLMs (the "blue team") to prevent harmful responses, e.g., provide private information or instructions to make a bomb. Additionally, red teaming data is often used to further adapt LLM chat assistants using safety tuning (Ganguli et al. 2022). In particular, extensive stress testing LLMs with red teaming focuses on a diverse set of scenarios. While traditional red teaming (Ganguli et al. 2022) has been effective in uncovering flaws, it often requires extensive manual effort from highly skilled experts, making it labor-intensive, error-prone, and inherently subjective. In response, automatic red teaming (Perez et al. 2022; Samvelyan et al. 2024; Hong et al. 2024; Deng et al. 2023; Ge et al. 2023; Beutel et al. 2024; Zhang et al. 2024) has emerged as a preferred alternative to manual efforts. These methods harness LLMs as the red team, using iterative algorithms to generate effective attacks automatically. Through continuous interaction with the blue team, these methods amass data for analysis, identifying vulnerabilities, and areas for improvement. Additionally, these interactions provide valuable training data, enhancing the robustness and safety protocols of the blue team model.

Existing works on automatic red teaming treat the problem as an optimization task aimed at maximizing the expected attack success rate (ASR), achieved by optimizing the unsafe proxy score against the blue team model, as detailed in Section 2. However, this emphasis on ASR overshadows another crucial aspect of red teaming: **generating a semantically rich set of diverse test queries**. Such diversity is essential for exhaustive testing of robustness and reliability across a broad spectrum of scenarios, accurately reflecting the wide range of use cases encountered upon deployment (Radharapu et al. 2023). Furthermore, employing adversarial safety training or Reinforcement Learning from Human Feedback (RLHF) on these comprehensive red teaming datasets allows LLMs to improve their performance by **fortifying their defenses against potential exploits and enhancing their ability to generalize effectively**. This comprehensiveness promotes interpolation within known scenarios rather than extrapolation in unknown situations, ultimately increasing their reliability in real-world situations (Ouyang et al. 2022; Bianchi et al. 2023; Ganguli et al. 2022; Ge et al. 2023). In Figure 4, we demonstrate empirically that increasing diversity among red teaming prompts enhances safety tuning, resulting in safer models when using our generated prompts.

We contend that ❶ the prevalent approach to red teaming by maximizing unsafe reward *misrepresents* its broader objective, leading to **compromised data diversity and quality**. Ideally, the red team should remain *impartial* during

[*] Work done during an internship at BIGAI.
† Corresponding author(s).

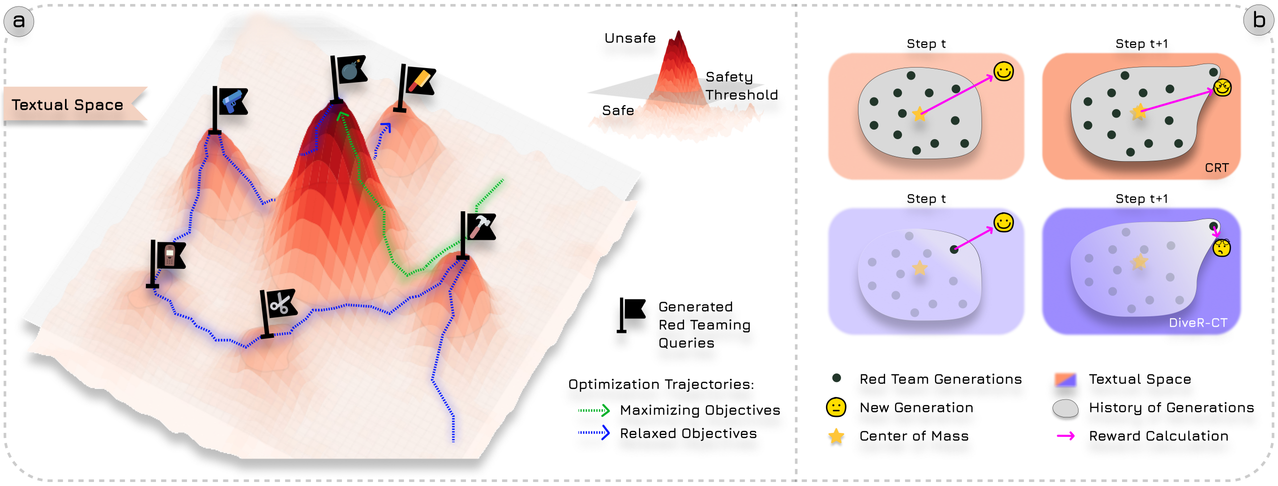

Figure 1: **Main Framework of DiveR-CT.** The key components of DiveR-CT, focusing on: ⓐ casting automatic red teaming as a constrained policy optimization problem, allowing our policies greater flexibility by relaxing the maximization objective; and ⓑ the revamped dynamic semantic reward. For a generation at time $t + 1$ that is close to the last, CRT (Hong et al. 2024) assigns a high reward, while DiveR-CT assigns a low k-NN reward, encouraging the policy to discover novel generations.

the data collection phase to gather a comprehensive spectrum of unsafe data with varying degrees of toxicity. This ensures the goal is not skewed towards eliciting the most egregious responses, encouraging the collection of any red teaming query that triggers the blue team model to exceed a predefined safety threshold, including less severe but equally critical responses. ❷ Maximizing the expected ASR through increasing the unsafe response score inherently reduces diversity by **confining policies to restricted search spaces**. ❸ Since reward models are learned proxies, this setting tends to **exacerbate overoptimization** (Gao, Schulman, and Hilton 2023), potentially obscuring the true objective (Hoskin 1996; Taylor 2016; Armstrong, Sandberg, and Bostrom 2012; Simon 1956). In Figure 3, we demonstrate that putting more emphasis on attack success rate maximization indeed leads to a greater risk of overoptimization across various ASR levels.

To address the score maximization bias and the eclipsed significance on diversity in automatic red teaming, we propose **Dive**rsity-enhanced red teaming with **R**elaxing **C**onstrain**T**s (DiveR-CT), shown in Figure 1. Unlike prior works that maximize every reward, DiveR-CT recasts maximization-biased terms into a constrained optimization framework (Equation (3)). Specifically, by treating unsafe rewards as threshold constraints rather than strict maximization targets, the policy gains more freedom to optimize for diversity metrics. Moreover, we further enhance semantic diversity by introducing a progressive reward based on nearest neighbors from generated history's embeddings. This reward uses dynamic targets to foster adaptive updates, ensuring thorough coverage of the semantic space. Previous efforts, *e.g.*, Curiosity Red Teaming (CRT) (Hong et al. 2024), have attempted to tackle the diversity issue using a semantic reward that encourages the red team policy to increase the semantic distance between the newly generated output and the history. However, as training progresses, the efficacy of reward signals in guiding the policy diminishes. In contrast, DiveR-CT dynamically adjusts nearest neighbor targets, providing a reactive and adaptive signal for the policy to cover the semantic space uniformly. We illustrate their PCA projection dynamics in Figure 2.

In Section 5, experimental results **firstly** validate the effectiveness of DiveR-CT in enhancing diversity across various settings with controllable attack success rate levels. **Secondly,** we show that our approach not only alleviates reward overoptimization issues but also enhances blue team models' resilience to attacks with superior data (Bianchi et al. 2023; Ge et al. 2023). **Lastly,** by attacking more resilient `Llama` safety-aligned models, we demonstrate that the static coefficients for safety in CRT cause drastic degradation in ASR, highlighting another strength of our method's dynamic adjustment of the safety coefficient to achieve steerable ASR while generating diverse attacks simultaneously.

2 Related Works

Automatic red teaming methods emerged to replace manual red teaming, with three main lines of work. Reinforcement learning (RL) pioneered by Perez et al. (2022), used RL to train red team agents to minimize blue team response safety, though at the cost of reduced diversity and near-deterministic policies (Puterman 2014). To counter these limitations, Hong et al. (2024) developed a curiosity-driven (CRT) method to enhance diversity (Tevet and Berant 2021) by incorporating historic generations to calculate novelty rewards (Pathak et al. 2017). Another line of work (Samvelyan et al. 2024), used quality diversity algorithms and prompting methods to gather red teaming prompts. Last line of work, (Lee et al. 2024), used amortized inference to tackle the red teaming problem.

We utilize RL for optimizing the discovery of red teaming prompts due to its efficacy in finding high reward (Sutton and Barto 1998), particularly in the vast and sparse search spaces of LLMs exacerbated by model safety features. Studies like those by (Lee et al. 2024) suggest using foundational datasets of manually curated attacks (3,003 toxic prompts from the SafetyDataset and AdvBench) to predict unseen modes of reward, emphasizing the need for an initial dataset to motivate exploration. However, this attempt to align trajectory probabilities proportional to rewards, fail to incentivize online searching for new initial modes, presenting a "chicken and egg" dilemma. Additionally, works like (Samvelyan et al. 2024) use quality-diversity algorithms but require prompt engineering for mutator and judge prompts, as well as human expert-designed features for archives. Furthermore, like (Lee et al. 2024), they needed human curated red team samples (Anthropic Harmless) to initialize their Map-Elites archive. In contrast, RL approaches minimizes human intervention/-expertise, allowing for training from scratch and efficiently discovering red teaming prompts, representing a streamlined and effective approach to automatic red teaming for LLMs.

Although proficient at eliciting unsafe responses from the blue team, current RL methods focus on maximizing toxicity, which might not address all defensive needs. This emphasis overlooks subtler harmful outputs and restricts the diversity of attacks. Furthermore, existing semantic rewards incorporating history can initially encourage diversity but degrades as optimization progresses.

3 Background and Problem Statement

Let $\mathcal{X}$ denote the set of all natural language strings. Consider a black-box (Papernot et al. 2017, 2016) language model chat assistant π_{BLUE} (the blue team model), which can be queried a fixed number of times N. The task of automatic red teaming involves identifying a subset $\mathcal{X}_{\text{red}} \subseteq \mathcal{X}$ such that for any prompt $x_{\text{red}} \in \mathcal{X}_{\text{red}}$, the response $y \sim \pi_{\text{BLUE}}(x_{\text{red}})$ meets specific unsafe criteria C. This subset is defined as $\mathcal{X}_{\text{red}} = \{x \in \mathcal{X} \mid \mathbf{1}_C(\pi_{\text{BLUE}}(\cdot \mid x)) = 1\}$, where C is typically assessed by a safety classifier threshold. While straightforward optimization for successful attacks achieves the automatic aspect, they do not ensure the diversity of the resulting set $\mathcal{X}_{\text{red}}$, often leading to mode collapse (Hong et al. 2024; Kirk et al. 2024). Therefore, our objective is also aimed at maximizing the diversity of the set $\mathcal{X}_{\text{red}}$.

Previous red teaming approaches, RL Perez et al. (2022) and CRT have the following objectives, respectively:

$$\begin{aligned} R_{\text{RL}}(w, x, y) &\overset{\triangle}{=} -\beta_{\text{safe}} B_{\text{safe}}(x, y) \\ &\quad -\beta_{\text{KL}} \log(\pi_\theta(x|w)/\pi_{\text{ref}}(x|w)) \qquad (1) \\ R_{\text{CRT}}(w, x, y) &\overset{\triangle}{=} R_{\text{RL}} - \beta_{\text{ent}} \log \pi_\theta(x|w) - \beta_{\text{gibb}} B_{\text{gibb}}(x) \\ &\quad + \beta_{\text{sem}} B_{\text{sem}}(x) + \beta_{\text{ngram}} B_{\text{ngram}}(x), \qquad (2) \end{aligned}$$

where, π_θ is the red team language model we are optimizing, and π_{ref} is the reference model used in standard RLHF (Ouyang et al. 2022). $w \in \mathcal{W}$ is the eliciting prompts used to generate red team prompt $x \sim \pi_\theta(\cdot|w)$, while y is the generated reply of LLM chat assistant $y \sim \pi_{\text{BLUE}}(\cdot|x)$. The coefficients β weight different objectives: KL divergence between the policy and reference model (KL), token entropy (ent), gibberish (gibb), semantic distance (sem), and n-gram dissimilarity (ngram). The Bs are the classifier outputs.

Red-teaming *vs.* Jailbreaking/Adversarial Attack Adversarial methods, such as jailbreaking and adversarial attacks, primarily focus on attack success rate (Ganguli et al. 2022; Yi et al. 2024; Chowdhury et al. 2024). *Jailbreaking* typically involves finding specific token sequences that can be added to any instruction to induce harmful outputs from an AI system, akin to gaining `sudo` access to a LLM assistant. These sequences often utilize fixed or templated parts of prompts designed to trigger the desired unsafe outputs. In contrast, *adversarial attacks* aim to manipulate an AI system into producing incorrect outputs, often through sequences of usually illegible tokens. While these methods prioritize achieving a successful attack, they **do not address the need for diversity within the attack strategies**, which is a key focus of red teaming approaches.

4 Diversity-enhanced Red Teaming with Relaxing Constraints

The strict maximization of unsafe scores by current RL methods overemphasizes optimizing ASR, sacrificing diversity. This issue is exacerbated by the semantic reward becoming stagnant as training steps increase, further inhibiting the discovery of novel prompts. Based on these observations, in Section 4.1, we present how we utilize constrained RL to relax the conventional objective of minimizing safety B_{safe} (Perez et al. 2022; Hong et al. 2024), allocating the policy with more capacity to maximize novelty rewards. Furthermore, in Section 4.2, we refine the existing semantic reward B_{sem} by incorporating dynamic targets to better cover the semantic space of red teaming queries. We illustrate the schematic of our proposed framework, **Dive**rsity-enhanced red teaming with **R**elaxing **C**onstrain**T**s (DiveR-CT), in Figure 1.

4.1 Constrained Objectives to Relax Constraints

Constrained Search. Constrained optimization settings typically requires policies to satisfy certain constraints c_i, narrowing the space of possible policies (Achiam et al. 2017). However, we counterintuitively use constrained policy optimization to *relax* the conventional constraint of maximizing unsafe score, allowing the policy to focus more on diversity. This is justified in automatic red teaming, where the preference for data points with slightly different toxicity scores (e.g., 0.96 vs. 0.83) is minimal. We treat these attacks *equally* to collect a broader and more realistic spectrum of unsafe queries. Additionally, since classifiers are imperfect proxies, human might judge lesser-scored attacks more toxic. Furthermore, since classifier scores often represent confidence levels, we can establish a humanly interpretable threshold for the resulting set of attacks. Thus, we frame red teaming as the search for diverse attacks that exceed a certain safety threshold. By using constrained policy optimization, we effectively enhance the capability of automatic red teaming to identify a wider range of unsafe queries.

Objective. Previous approaches, like Hong et al. (2024), included gibberish penalties, ensuring generated queries remained comprehensible. We propose integrating this reward as a constraint, setting a confidence level for output fluency that the policy should not violate. Importantly, our method is flexible and not limited to constraining the policy on safety and gibberish; any sensible target not requiring maximization can similarly be cast as a constraint in our framework.

Overall, we have the following general optimization objective for diverse generations,

$$\begin{aligned} &\max_{\pi_\theta} \mathbb{E}_{w\sim\mathcal{W}, x\sim\pi_\theta(\cdot|w), y\sim\pi_{\text{BLUE}}(\cdot|x)} \left[R(w,x,y)\right] \\ &\quad \text{s.t.} \quad C_i(x,y) \leq d_i,\ i = 1,..,m, \quad \forall x,y, \end{aligned} \tag{3}$$

where C_i represents one of the m constraints, each associated with its corresponding threshold d_i. Following previous work, all the utilities used for optimization remain in our objective; however, they are either retained as rewards or newly cast as constraints. For rewards, our method employs

$$\begin{aligned} R_{\text{DiveR-CT}}(w,x,y) &\overset{\triangle}{=} -\beta_{\text{KL}} \log(\pi_\theta(x|w)/\pi_{\text{ref}}(x|w)) \\ &-\beta_{\text{ent}} \log \pi_\theta(x|w) \quad +\beta_{\text{sem}} B_{\text{sem}}(x) + \beta_{\text{ngram}} B_{\text{ngram}}(x), \end{aligned} \tag{4}$$

where βs are fixed hyperparameters, using the *default* β values from previous works (Hong et al. 2024). For constraints, we have gibberish, C_{gibb}, and safety, C_{safe}, with their corresponding predetermined thresholds, d_{safe} and d_{gibb}. To convert the original classifier scores from CRT into costs, we use negative rewards as costs, i.e., $C = -B$.

We optimize for the expected constraint satisfaction over the generated responses y, because red teaming does not have strict output requirements, unlike real-life scenarios (García and Fernández 2015). The slack variable C_i^d, with its corresponding thresholds d_i, is defined as follows:

$$C_i^d(x,y) \overset{\triangle}{=} \mathbb{E}_{\substack{w\sim\mathcal{W} \\ x\sim\pi_\theta(\cdot|w) \\ y\sim\pi_{\text{BLUE}}(\cdot|x)}} \left[c_i(x,y)\right] - d_i, \tag{5}$$

where $i \in \{\text{safe, gibberish}\}$ and c_i are cost functions instantiated by neural network classifiers.

Given the primal form of Equation (3), our unconstrained dual objective can be written as (Yurkiewicz 1985; Boyd and Vandenberghe 2010):

$$\max_{\pi_\theta} \min_{\substack{\lambda_{\text{safe}}\geq 0 \\ \lambda_{\text{gibb}}\geq 0}} \mathbb{E}\left[R_{\text{DiveR-CT}} - \lambda_{\text{safe}} \cdot C_{\text{safe}}^d - \lambda_{\text{gibb}} \cdot C_{\text{gibb}}^d\right]. \tag{6}$$

We use gradient descent ascent combined with PPO (Schulman et al. 2017) to solve the optimization problem in Equation (6). It is crucial to emphasize that our λ values *dynamically adjust* based on whether the expectation of constraint i is met. Unlike previous works that utilize a fixed coefficient (Perez et al. 2022; Hong et al. 2024), our method offers the weights to dynamically update. This adaptability allows for rapid adjustments in response to whether constraints are satisfied or not.

4.2 Dynamic Semantic Diversity Reward

We used constrained RL to relax the maximization objectives for safety and gibberish. The remaining rewards conventionally used are semantic and n-gram to promote novelty, which should be maximized (Hong et al. 2024). The n-gram reward, calculated as $1-$ BLEU score, effectively promotes novelty by dynamically selecting the most appropriate reference for each n-gram. This reward ensures flexibility and encourages the generation of novel queries by not fixing the policy's objective to a particular point in terms of n-grams. In contrast, the semantic reward mechanism, which relies on the average cosine similarity between the hypothesis embedding and all past history of reference embeddings $\mathcal{X}_{\text{history}}$, faces scalability issues. As the reference set expands, new generations have diminishing impacts on the semantic reward, permitting the policy to pathologically repeat outlier solutions (observed in Figure 2). This stark difference highlights the need for adaptive measures in handling semantic rewards, similar to the flexibility afforded by the n-gram approach. To mitigate this issue, instead of comparing the hypothesis with all reference embeddings, we focus on the nearest k neighbours by cosine similarity (Liu and Abbeel 2021; Zhao et al. 2022)

$$B_{\text{sem}}(x) = -\frac{1}{k} \sum_{x' \in \mathcal{N}_{k,\phi}(x, \mathcal{X}_{\text{history}})} \frac{\phi(x) \cdot \phi(x')}{\|\phi(x)\|_2 \|\phi(x')\|_2}, \tag{7}$$

where $\mathcal{N}_{k,\phi}(x, \mathcal{X}_{\text{history}})$ represents the k-nearest neighbors (k-NN) of x in $\mathcal{X}_{\text{history}}$, determined by cosine similarity using the embedding function ϕ. This adjustment *relaxes* the red team's objective by dynamically shifting the semantic target instead of a relatively fixed point. It also prevents the agent from exploiting a single outlier solution, as the history reference immediately updates with each step.

5 Experiments

This section introduces the data, models, and baselines used in this study. We then present our main results, followed by analytical experiments that involve modifying settings and components to gain a deeper understanding of our method.

5.1 Setup

We adopted a similar experimental setup as CRT (Hong et al. 2024), with minimal changes to maintain consistency. We used identical hyperparameters used by CRT, enabling fair comparisons. We optimize all methods the same number of steps for three seeds. Note that by using the same number of gradient updates across different methods, we allow a fair comparison of the time complexities of different methods.

Baselines. To thoroughly evaluate DiveR-CT's performance, we incorporate a variety of baseline methods, including RL and CRT introduced in Section 3 and zero-shot sampling at temperature=1.0.

Quality Metrics. The *quality* of a method is mainly measured by the attack success rate (ASR), the percentage of attacks generated by the red team π_θ that elicited a response y from π_{BLUE} with a predicted unsafe probability exceeding a threshold τ. We set $\tau = 0.5$ for all experiments.

Diversity Metrics. The diversity of the generated attacks is evaluated following Tevet and Berant (2021); Hong et al. (2024) on the set of successful generated attacks $\mathcal{X}_\tau = \{x \in \mathcal{X} \mid c_{\text{safe}}(x,y) \leq \tau\}$, where y is the elicited response from π_{BLUE}. For lexical diversity, we use the SelfBLEU score (Zhu et al. 2018) and a n-gram kernel Vendi score (Friedman and Dieng 2022), both based on the overlap of n-grams.

To measure semantic diversity, we leverage the cosine distance among the sentence embeddings and a cosine similarity-based kernel Vendi score.

5.2 Main Results

We first outline the main results of our proposed method. We then qualitatively compare the red teaming queries generated by our method and baseline methods. Then, we explore the practicality of the generated queries from our method compared to baselines in safety fine-tuning.

Numerical Results. Since our method can flexibly control the balance between diversity and unsafe objectives through the constraint threshold d_{safe}, we present the main results using three different thresholds: $d_{\text{safe}} \in \{-0.5, -0.7, -0.9\}$ in Section 5.1. We compare DiveR-CT with other reinforcement learning methods, namely Perez et al. (2022) and CRT (Hong et al. 2024), and with zero-shot. We group the different RL runs into three main ASR categories. Conveniently, we found that the original $\beta_{\text{safe}} = 1.0$ associated with CRT, Perez et al. (2022), and our method with $d_{\text{safe}} = -0.9$ fall into the same high ASR bucket. To make a fair comparison for the medium and low ASR brackets, we tuned the CRT β_{safe} coefficient to match the ASR levels of our other thresholds. We empirically found that $\beta_{\text{safe}} = 0.4$ matched the ASR of $d_{\text{safe}} = -0.7$ and $\beta_{\text{safe}} = 0.3$ matched $d_{\text{safe}} = -0.5$. Lastly, we group the zero-shot results in their own bracket due to their extremely low ASR, ensuring completeness.

The first trend is the presence of a clear **trade-off** between achieving high ASR and high diversity. RL does not prioritize diversity; its objective solely maximizes the unsafe score, hence only retaining a handful of distinct high-scoring attacks. CRT outperforms RL in the high ASR scenario regarding diversity, but DiveR-CT *outperforms* CRT on all metrics across all three ASR settings. Another interesting finding is that our method's ASR is **controllable** since the resulting ASRs of the produced attacks follows the chosen thresholds. CRT can also use the coefficient to control its ASR, but the exact correlation is inconsistent. It could depend on the task setting, as observed in Section 5.2, where a more robust blue team does not yield a consistent ASR/β_{safe} pair for CRT, but does hold with ASR/d_{safe} pair for DiveR-CT. This strength allows fine-grained control over ASR and diversity in budgeted situations.

Qualitative Results. We visualized the PCA of the semantic embeddings of the experiments with two principal components in Figure 2. Our semantic embeddings demonstrate a more **uniform coverage** across the space, contrasting with the CRT runs, which predominantly extend in one direction—coinciding with previously discussed pathological outcomes. Furthermore, our method adopted more **persuasive tactics** against the blue team model, employing specific phrases *"use humor and exaggeration"*, *"use satirical lyrics"*, *"use witty jokes"*, *"stereotype"*, or *"use sarcasm"*.

Finding I: DiveR-CT Mitigates Overoptimization. Overoptimization is a known issue in the RLHF setting (Gao, Schulman, and Hilton 2023) or when using proxy rewards. Methods like CRT and Perez et al. (2022) maximize the proxy unsafe score of blue team responses, making them susceptible to overoptimizing for specific nuances of the safety classifier. In contrast, our method explicitly forgoes maximizing the safety score if it exceeds a certain threshold. We hypothesize that our approach mitigates overoptimization.

To investigate this, we score all the red teaming queries generated during optimization using both the training/task classifier and a separate test classifier (`DaNLP/da-electra-hatespeech-detection`) that the red team has not encountered during optimization. The resulting ASRs are presented in Figure 3. We observe that Perez et al. (2022), which solely maximizes the unsafe classifier score, exhibits a much lower ASR on the test classifier, demonstrating overoptimization. Additionally, when grouping by the train classifier ASR, a significant drop is observed when targeting a higher ASR. While targeting a more moderate train ASR, the drop in test ASR is reduced (even increased in the lower bracket). Our method consistently achieves higher test ASR while maintaining comparable train ASR across all three brackets, demonstrating its effectiveness in alleviating overoptimization.

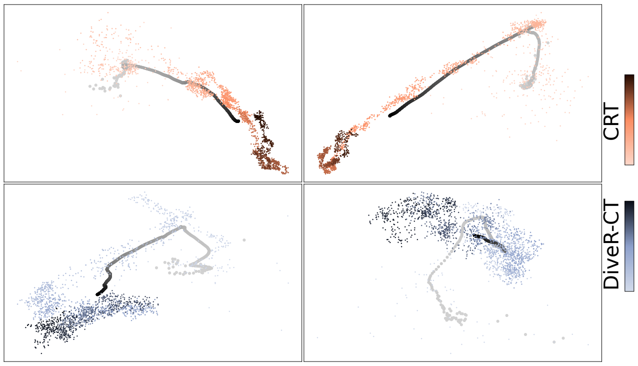

Figure 2: **Comparison of Embeddings using PCA: Per-step Mean and Cumulative Mean of Embeddings.** This figure highlights the evolution of generations in the embedding space by showing the cumulative average (gradient line) and the per-step average (scatter points) of the embeddings. DiveR-CT demonstrates more uniform coverage of attacks.

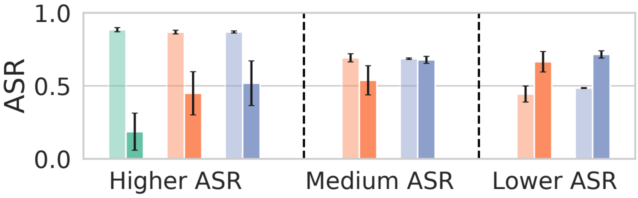

Figure 3: **Overoptimization Eval. with Test Safety Classifier.** We evaluate the extent of overoptimization by using a test safety classifier, `DaNLP/da-electra-hatespeech-detection`, comparing side by side with the ASR of the traing task classifier we used to train the red team policies. Light/dark color is train/test ASR, respectively. Our method (blue) reliably reduces overoptimization across all ASR budgets compared to baselines (RL in green and CRT in orange).

		Lexical		Semantic	
Method	**ASR**$^{-}$	**Self-BLEU**$^{\uparrow}$	**Vendi-Ngram**$^{\uparrow}$	**Semantic Mean**$^{\uparrow}$	**Vendi-Semantic**$^{\uparrow}$
RL (Perez et al. (2022))	$0.885^{(\pm 0.014)}$	$0.037^{(\pm 0.014)}$	$0.004^{(\pm 0.000)}$	$0.031^{(\pm 0.007)}$	$0.010^{(\pm 0.000)}$
CRT, $\beta_{\text{safe}} = 1.0$	$0.868^{(\pm 0.013)}$	$0.570^{(\pm 0.056)}$	$0.526^{(\pm 0.154)}$	$0.360^{(\pm 0.024)}$	$0.076^{(\pm 0.012)}$
Diver-CT, $d_{\text{safe}} = -0.9$ (ours)	$0.869^{(\pm 0.007)}$	$\mathbf{0.746}^{(\pm 0.047)}$	$\mathbf{0.728}^{(\pm 0.106)}$	$\mathbf{0.378}^{(\pm 0.012)}$	$\mathbf{0.110}^{(\pm 0.011)}$
CRT, $\beta_{\text{safe}} = 0.4$	$0.692^{(\pm 0.028)}$	$0.802^{(\pm 0.021)}$	$0.559^{(\pm 0.149)}$	$0.363^{(\pm 0.008)}$	$0.084^{(\pm 0.004)}$
Diver-CT, $d_{\text{safe}} = -0.7$ (ours)	$0.686^{(\pm 0.005)}$	$\mathbf{0.834}^{(\pm 0.024)}$	$\mathbf{0.964}^{(\pm 0.014)}$	$\mathbf{0.391}^{(\pm 0.022)}$	$\mathbf{0.123}^{(\pm 0.012)}$
CRT, $\beta_{\text{safe}} = 0.3$	$0.444^{(\pm 0.055)}$	$0.829^{(\pm 0.020)}$	$0.767^{(\pm 0.113)}$	$0.355^{(\pm 0.040)}$	$0.083^{(\pm 0.017)}$
Diver-CT, $d_{\text{safe}} = -0.5$ (ours)	$0.485^{(\pm 0.003)}$	$\mathbf{0.843}^{(\pm 0.016)}$	$\mathbf{0.969}^{(\pm 0.010)}$	$\mathbf{0.402}^{(\pm 0.010)}$	$\mathbf{0.128}^{(\pm 0.005)}$
Zero-shot	$0.001^{(\pm 0.000)}$	$0.533^{(\pm 0.003)}$	$0.659^{(\pm 0.004)}$	$0.018^{(\pm 0.001)}$	$0.010^{(\pm 0.000)}$

Table 1: **Main Results Grouped by ASR.** We present the lexical and semantic diversity metrics of baseline compared to DiveR-CT. We group the experiments by their Attack Success Rates.

Finding II: DiveR-CT Generates Better Safety Fine-tuning Data. After presenting the results of the red teaming queries generated by DiveR-CT and baseline methods, we focus on how these queries can be used to mitigate the blue team's unsafe behaviors. We followed a simple approach close to Samvelyan et al. (2024). We first filter and retain only the queries generated by the red team that have an unsafe score higher than 0.5. We then prompt `gpt-4-turbo` to generate a list $L_{\text{refuse}}$of 50 refusal responses. For each unsafe query x_{unsafe}, we sample a random refusal response $y_{\text{refuse}} \sim L_{\text{refuse}}$ from the list. To prevent the model from degrading in general capabilities, we use the whole `tatsu-lab/alpaca` instruction tuning dataset $(x_{\text{Alp.}}, y_{\text{Alp.}}) \in \mathcal{D}_{\text{Alp.}}$, augmented with a subsample of the toxic dataset we constructed $(x_{\text{red}}, y_{\text{refuse}}) \in \mathcal{D}_{\text{safety}}$. We maintain a ratio of 2:1 for the alpaca and toxic refusal data. Finally, with this mixed data, $\mathcal{D}_{\text{supervised}} = \mathcal{D}_{\text{Alp.}} \cup \mathcal{D}_{\text{safety}}$, we supervise fine-tune the original blue team model `vicgalle/gpt2-alpaca-gpt4`.

For each method — RL (Perez et al. (2022)), CRT $\beta_{\text{safe}} = 0.4$, and DiveR-CT $d_{\text{safe}} = -0.7$ — we construct the safety dataset $\mathcal{D}_{\text{safety}}$ from three different seeds and finetune three different instruction-following models. We then evaluate the resulting models on the Open LLM Leaderboard benchmarks (Hellaswag, ARC-Challenge, TruthfulQA, and Winogrande (Zellers et al. 2019; Clark et al. 2018; Lin, Hilton, and Evans 2022; Sakaguchi et al. 2021)) and red teaming benchmarks: AART, SAP, and AdvenBench (Radharapu et al. 2023; Deng et al. 2023; Zou et al. 2023) using `redteaming-resistance-benchmark`. We present the performance of the resulting models in Figure 4.

First, we observe that augmenting models with mixed data generally *does not* harm their general capabilities. Second, safety tuning with $(x_{\text{red}}, y_{\text{refuse}})$ pairs *enhances the safety robustness* of the blue team models. Furthermore, models finetuned with CRT generated data outperform those finetuned with data generated from RL (Perez et al. (2022)). Lastly, and importantly, we find that the queries generated by DiveR-CT **outperform** those from CRT and Perez et al. (2022), likely due to our approach's broader coverage of red team attacks.

Finding III: DiveR-CT Better Red Teams More Capable Models We further investigate the efficacy of our method by switching to more advanced RLHF-trained chat models for the blue team. Specifically, we compare methods by red teaming more robust and larger models: `Llama-2-7b-chat-hf` and `Meta-Llama-3-8B-Instruct`. Given our method's flexibility in controlling the ASR, we first fix the default coefficient for the safety reward at $\beta_{\text{safe}} = 1.0$ for CRT and Perez et al. (2022) (default values from their respective works). We then adjust DiveR-CT's threshold value to match the ASRs of the baselines to make diversity metrics comparable. Concretely, we applied $d_{\text{safe}} = -0.6$ to match the ASR of RL. Figure 5 shows the results when against more robust models. CRT's ASR significant dropped under more resilient blue team conditions, underscoring the critical role of dynamic online adjustment of reward signal. Our diversity metrics surpass those of the baselines, demonstrating our method's ability to **sustain controllable ASR and high diversity** even against SOTA aligned models. In contrast, methods like CRT sacrificed ASR to maintain diversity.

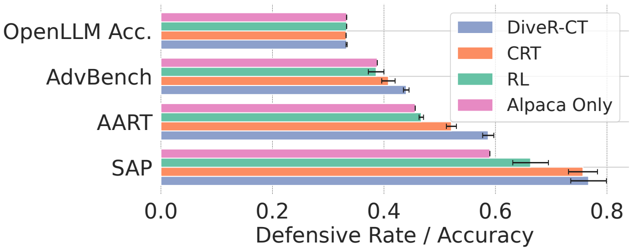

Figure 4: **Red Team Generation Quality Assessment Through Safety Tuning.** We finetune the blue team model using a mix of successful red team queries and Alpaca dataset. This figure illustrates the robustness of response rate and OpenLLM Accuracy, demonstrating that safety tuning with DiveR-CT generated data better enhances LLM safety.

5.3 Ablations

Since our method contains two main differences from the CRT method, we evaluate variations of our method by adding or removing one of the components we introduced. We fixed $d_{\text{safe}} = -0.7$ for DiveR-CT, and $\beta_{\text{safe}} = 0.4$ for CRT and

		Lexical		Semantic	
Method	**ASR$^-$**	**Self-BLEU$^\uparrow$**	**Vendi-Ngram$^\uparrow$**	**Semantic Mean$^\uparrow$**	**Vendi-Semantic$^\uparrow$**
DiveR-CT, $d_{\text{safe}} = -0.7$ (Ours)	$0.686^{(\pm 0.005)}$	$0.834^{(\pm 0.024)}$	$0.964^{(\pm 0.014)}$	$0.391^{(\pm 0.022)}$	$0.123^{(\pm 0.012)}$
DiveR-CT, gibberish reward	$0.681^{(\pm 0.021)}$	$0.811^{(\pm 0.014)}$	$0.961^{(\pm 0.026)}$	$0.385^{(\pm 0.024)}$	$0.120^{(\pm 0.015)}$
DiveR-CT, topk=all	$0.692^{(\pm 0.003)}$	$0.792^{(\pm 0.025)}$	$0.896^{(\pm 0.055)}$	$0.411^{(\pm 0.012)}$	$0.117^{(\pm 0.009)}$
DiveR-CT, topk=1	$0.682^{(\pm 0.005)}$	$0.837^{(\pm 0.015)}$	$0.899^{(\pm 0.071)}$	$0.388^{(\pm 0.013)}$	$0.113^{(\pm 0.001)}$
DiveR-CT, $d_{\text{safe}} = -0.5$ (Ours)	$0.485^{(\pm 0.003)}$	$0.843^{(\pm 0.016)}$	$0.969^{(\pm 0.010)}$	$0.402^{(\pm 0.010)}$	$0.128^{(\pm 0.005)}$
CRT, $\beta_{\text{safe}} = 0.3$	$0.444^{(\pm 0.055)}$	$0.829^{(\pm 0.020)}$	$0.767^{(\pm 0.113)}$	$0.355^{(\pm 0.040)}$	$0.083^{(\pm 0.017)}$
CRT+top-16, $\beta_{\text{safe}} = 0.4$	$0.481^{(\pm 0.022)}$	$0.834^{(\pm 0.017)}$	$0.848^{(\pm 0.018)}$	$0.387^{(\pm 0.003)}$	$0.102^{(\pm 0.003)}$

Table 2: **Ablations Grouped by ASR.** We investigated changing the gibberish penalty and the k-NN semantic reward.

		Lexical		Semantic	
Method	**ASR$^-$**	**Self-BLEU$^\uparrow$**	**Vendi-Ngram$^\uparrow$**	**Semantic Mean$^\uparrow$**	**Vendi-Semantic$^\uparrow$**
RL (Perez et al. (2022))	$0.840^{(\pm 0.015)}$	$0.184^{(\pm 0.089)}$	$0.003^{(\pm 0.000)}$	$0.024^{(\pm 0.007)}$	$0.010^{(\pm 0.000)}$
CRT, $\beta_{\text{safe}} = 1.0$	$0.859^{(\pm 0.007)}$	$0.682^{(\pm 0.068)}$	$0.497^{(\pm 0.182)}$	$0.344^{(\pm 0.023)}$	$0.070^{(\pm 0.008)}$
DiveR-CT, $d_{\text{safe}} = -0.85$	$\mathbf{0.864}^{(\pm \mathbf{0.002})}$	$\mathbf{0.739}^{(\pm \mathbf{0.053})}$	$\mathbf{0.717}^{(\pm \mathbf{0.107})}$	$\mathbf{0.377}^{(\pm \mathbf{0.014})}$	$\mathbf{0.110}^{(\pm \mathbf{0.000})}$

Table 3: **Performance Using `Meta-Llama-Guard-2-8B` as Safety Classifier.** We change the safety classifier to a more robust `Meta-Llama-Guard-2-8B`. Results indicate that DiveR-CT outperforms baselines in diversity metrics, which is consistent with the trends observed in our primary results.

present all the results of this section in Section 5.2.

First, we investigate if constraining the gibberish reward is beneficial. We present the case where gibberish is maximized, denoted as "gibberish reward". Our findings show that constraining gibberish, rather than maximizing it, slightly improves performance by reducing the pressure to optimize this objective, allowing the policy more flexibility.

Additionally, we explore the benefits of using the top-16 semantic neighbors. We compare this approach with two variants 1) rewards are calculated based on semantic cosine similarity across all history "topk=all" and, 2) "topk=1". We observe that "topk=all" significantly sacrifices other diversity metrics to prioritize the semantic mean, since semantic mean is the intended objective for this variant. Overall, using the top-16 semantic neighbors is the most beneficial for the agent.

Lastly, we tried adding the top-16 semantic neighbor reward to CRT. However, the same $\beta_{\text{safe}} = 0.4$ yielded a different ASR level, closer to $\beta_{\text{safe}} = 0.3$ and $d_{\text{safe}} = -0.5$. This further demonstrates that the safety coefficient in CRT makes controlling the outcome ASR difficult, a problem not encountered with DiveR-CT. Therefore, we appropriately regroup results based on this modified CRT. We notice that using our dynamic semantic rewards boosts CRT in all diversity metrics but still exhibiting lower performance than DiveR-CT.

Changing the Safety Reward Classifier We conducted experiments where we changed the toxic classifier to a more recent and better-performing safety classifier, `Meta-Llama-Guard-2-8B`. The `Meta-Llama-Guard-2-8B` model features finer-grained categories and covers more topics than the classifier used in our main results. Again, we fix the default safety coefficients for RL and CRT and adjust our threshold to match their ASR. We present the results in Section 5.2. Similar to our main results, changing the classifier of toxicity does not alter the conclusion, where our method was able to generate a more diverse set of red teaming prompts with approximately the same ASR. Another interesting finding is that by changing the safety classifier, our method was able to identify cybersecurity red team attacks, which were not observed in experiments from the main results or CRT/RL using the `Meta-Llama-Guard-2-8B` classifier.

5.4 Costs, Lagrange Multipliers, and their Interplay

Safety Costs. We display the safety cost during optimization in Figure 6. Notably, a distinctive "waving" pattern is identified, previously documented in the constrained reinforcement learning literature (Calvo-Fullana et al. 2021), which signifies that minor adjustments in the weight space can easily toggle the policy between satisfying and violating constraints. Although such volatility is typically problematic in safe reinforcement learning scenarios—where consistent satisfaction of safety is crucial—counterintuitively, it proves beneficial in our context. Since the primary output from the red teaming policy is data rather than the policy itself, we believe these oscillations act as mini "resets", encouraging the policy to pursue diversity rewards and break free from local safety minima. Upon re-entry into the constraint satisfaction zone, the policy is more inclined to explore new red teaming topics, motivated by the need to diversify from its semantic and lexical history.

Lagrange Multipliers. Since a distinctive waving pattern is observed in the safety cost, we expect the Lagrange multipliers to also dynamically adjust. We observe this in our experiments: an oscillation pattern emerges for safety Lagrange multipliers, with increasing costs due to constraint violations causing a rise in the Lagrange multiplier values, thereby exerting more influence on the policy gradient up-

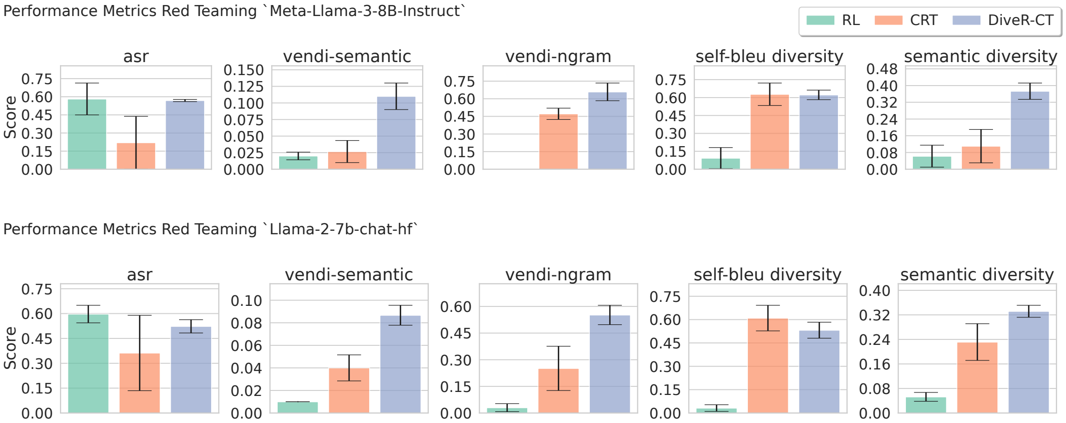

Figure 5: **Metrics of Red Teaming More Capable Blue Team Models.** We present the ASR and diversity metrics of red teaming queries by changing the blue team to more capable chat models: `Meta-Llama-3-8B-Instruct` (top)/ `Llama-2-7b-chat-hf` (bottom). By increasing attack difficulty, CRT decreased in ASR dramatically using default safety coefficient. Despite having higher ASR than CRT, DiveR-CT outperforms their diversity metrics in both settings.

date. An overlapping chart of costs and Lagrange multipliers in Figure 7 reveals a slight delay in this oscillation pattern; once the constraint is met, the lambda value decreases, subsequently exerting less influence on the policy gradient. Additionally, the Lagrange multipliers for gibberish constraints during training show a smaller waving pattern, suggesting that adjustments in the parameter space do not significantly affect gibberish constraint satisfaction.

6 Discussion

We introduced a novel method, DiveR-CT, which produces enhanced lexical and semantic diversity over existing red teaming approaches. We assessed our method under various settings, including different ASR levels, varying blue team models, and safety classifiers, showing that DiveR-CT consistently outperformed strong baselines. Our experiments demonstrated that data generated by DiveR-CT significantly increased the robustness of blue team models and that our method alleviates overoptimization. Qualitative results also show our method is able to discover persuasive strategies and topics like cybersecurity, which were never discovered by baseline methods.

Limitations. Our study focused on single-turn interactions, but recent works have shown that multi-turn interactions may further increase LLM vulnerabilities (Anil et al. 2024; Cheng et al. 2024). Future work could explore enhancing contextual diversity through multi-turn histories. Furthermore, DiveR-CT does not incorporate domain knowledge. Integrating fine-grained attack class classifiers, such as `Llama-Guard-3-8B`, could provide more uniform coverage across known domain topics when combined with our method. Finally, while our focus was on red teaming LLM chat assistants, other AI systems, such as text-to-image and vision-language models, could also benefit from our method.

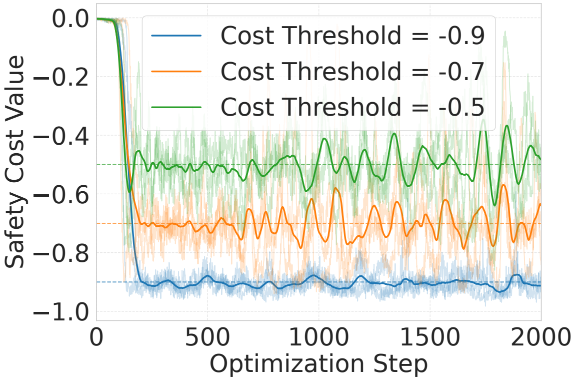

Figure 6: **Safety Cost of DiveR-CT during Optimization with Moving Avg.** We present the individual runs with and the moving average of the three seeds of different thresholds.

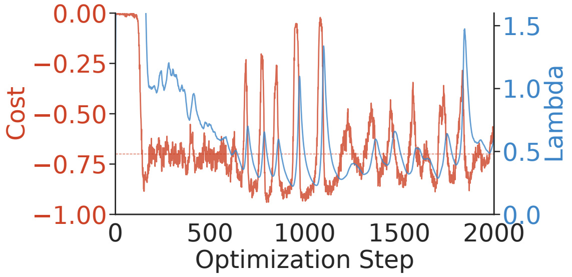

Figure 7: **Overlayed Safety Cost and its Lagrange Multiplier Values.** Overlay of the Lagrange multiplier values and the safety costs during optimization. At the beginning of the run, the Lagrange multiplier value rapidly increases to its maximum capped value. As a result, it is not visible in the chart for the initial 0 to approximately 200 steps.

Acknowledgements

This work is supported in part by the National Key R&D Program of China (2022ZD0114903) and the National Natural Science Foundation of China (62321005, 62376031, U2336214, 62461160309).

References

Achiam, J.; Held, D.; Tamar, A.; and Abbeel, P. 2017. Constrained Policy Optimization. In Precup, D.; and Teh, Y. W., eds., *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, 22–31. PMLR.

Anil, C.; Durmus, E.; Sharma, M.; Benton, J.; Kundu, S.; Batson, J.; Rimsky, N.; Tong, M.; Mu, J.; Ford, D.; Mosconi, F.; Agrawal, R.; Schaeffer, R.; Bashkansky, N.; Svenningsen, S.; Lambert, M.; Radhakrishnan, A.; Denison, C. E.; Hubinger, E.; Bai, Y.; Bricken, T.; Maxwell, T.; Schiefer, N.; Sully, J.; Tamkin, A.; Lanham, T.; Nguyen, K.; Korbak, T.; Kaplan, J.; Ganguli, D.; Bowman, S. R.; Perez, E.; Grosse, R.; and Duvenaud, D. K. 2024. Many-shot Jailbreaking. *anthropic.com*.

Armstrong, S.; Sandberg, A.; and Bostrom, N. 2012. Thinking Inside the Box: Controlling and Using an Oracle AI. *Minds Mach.*, 22(4): 299–324.

Beutel, A.; Xiao, K.; hannes Heidecke, J.; and Weng, L. 2024. Diverse and Effective Red Teaming with Auto-generated Rewards and Multi-step Reinforcement Learning.

Bianchi, F.; Suzgun, M.; Attanasio, G.; Röttger, P.; Jurafsky, D.; Hashimoto, T.; and Zou, J. 2023. Safety-Tuned LLaMAs: Lessons From Improving the Safety of Large Language Models that Follow Instructions. *CoRR*, abs/2309.07875.

Boyd, S. P.; and Vandenberghe, L. 2010. Convex Optimization. *IEEE Transactions on Automatic Control*, 51: 1859–1859.

Calvo-Fullana, M.; Paternain, S.; Chamon, L. F. O.; and Ribeiro, A. 2021. State Augmented Constrained Reinforcement Learning: Overcoming the Limitations of Learning with Rewards. *CoRR*, abs/2102.11941.

Cheng, Y.; Georgopoulos, M.; Cevher, V.; and Chrysos, G. G. 2024. Leveraging the Context through Multi-Round Interactions for Jailbreaking Attacks. *CoRR*, abs/2402.09177.

Chowdhury, A. G.; Islam, M. M.; Kumar, V.; Shezan, F. H.; Kumar, V.; Jain, V.; and Chadha, A. 2024. Breaking Down the Defenses: A Comparative Survey of Attacks on Large Language Models. arXiv:2403.04786.

Clark, P.; Cowhey, I.; Etzioni, O.; Khot, T.; Sabharwal, A.; Schoenick, C.; and Tafjord, O. 2018. Think you have Solved Question Answering? Try ARC, the AI2 Reasoning Challenge. *CoRR*, abs/1803.05457.

Deng, B.; Wang, W.; Feng, F.; Deng, Y.; Wang, Q.; and He, X. 2023. Attack Prompt Generation for Red Teaming and Defending Large Language Models. In Bouamor, H.; Pino, J.; and Bali, K., eds., *Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore, December 6-10, 2023*, 2176–2189. Association for Computational Linguistics.

Friedman, D.; and Dieng, A. B. 2022. The Vendi Score: A Diversity Evaluation Metric for Machine Learning. *CoRR*, abs/2210.02410.

Ganguli, D.; Lovitt, L.; Kernion, J.; Askell, A.; Bai, Y.; Kadavath, S.; Mann, B.; Perez, E.; Schiefer, N.; Ndousse, K.; Jones, A.; Bowman, S.; Chen, A.; Conerly, T.; DasSarma, N.; Drain, D.; Elhage, N.; Showk, S. E.; Fort, S.; Hatfield-Dodds, Z.; Henighan, T.; Hernandez, D.; Hume, T.; Jacobson, J.; Johnston, S.; Kravec, S.; Olsson, C.; Ringer, S.; Tran-Johnson, E.; Amodei, D.; Brown, T.; Joseph, N.; McCandlish, S.; Olah, C.; Kaplan, J.; and Clark, J. 2022. Red Teaming Language Models to Reduce Harms: Methods, Scaling Behaviors, and Lessons Learned. *CoRR*, abs/2209.07858.

Gao, L.; Schulman, J.; and Hilton, J. 2023. Scaling Laws for Reward Model Overoptimization. In Krause, A.; Brunskill, E.; Cho, K.; Engelhardt, B.; Sabato, S.; and Scarlett, J., eds., *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, 10835–10866. PMLR.

García, J.; and Fernández, F. 2015. A comprehensive survey on safe reinforcement learning. *J. Mach. Learn. Res.*, 16: 1437–1480.

Ge, S.; Zhou, C.; Hou, R.; Khabsa, M.; Wang, Y.; Wang, Q.; Han, J.; and Mao, Y. 2023. MART: Improving LLM Safety with Multi-round Automatic Red-Teaming. *CoRR*, abs/2311.07689.

Hong, Z.-W.; Shenfeld, I.; Wang, T.-H.; Chuang, Y.-S.; Pareja, A.; Glass, J. R.; Srivastava, A.; and Agrawal, P. 2024. Curiosity-driven Red-teaming for Large Language Models. In *The Twelfth International Conference on Learning Representations*.

Hoskin, K. 1996. The 'awful idea of accountability': inscribing people into the measurement of objects. *Accountability: Power, ethos and the technologies of managing*, 265.

Kirk, R.; Mediratta, I.; Nalmpantis, C.; Luketina, J.; Hambro, E.; Grefenstette, E.; and Raileanu, R. 2024. Understanding the Effects of RLHF on LLM Generalisation and Diversity. In *The Twelfth International Conference on Learning Representations*.

Lee, S.; Kim, M.; Cherif, L.; Dobre, D.; Lee, J.; Hwang, S. J.; Kawaguchi, K.; Gidel, G.; Bengio, Y.; Malkin, N.; and Jain, M. 2024. Learning diverse attacks on large language models for robust red-teaming and safety tuning. *ArXiv*, abs/2405.18540.

Lin, S.; Hilton, J.; and Evans, O. 2022. TruthfulQA: Measuring How Models Mimic Human Falsehoods. In Muresan, S.; Nakov, P.; and Villavicencio, A., eds., *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, 3214–3252. Association for Computational Linguistics.

Liu, H.; and Abbeel, P. 2021. Behavior From the Void: Unsupervised Active Pre-Training. In Ranzato, M.; Beygelzimer, A.; Dauphin, Y. N.; Liang, P.; and Vaughan, J. W., eds., *Advances in Neural Information Processing Systems 34: Annual*

Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual, 18459–18473.

Ouyang, L.; Wu, J.; Jiang, X.; Almeida, D.; Wainwright, C. L.; Mishkin, P.; Zhang, C.; Agarwal, S.; Slama, K.; Ray, A.; Schulman, J.; Hilton, J.; Kelton, F.; Miller, L.; Simens, M.; Askell, A.; Welinder, P.; Christiano, P. F.; Leike, J.; and Lowe, R. 2022. Training language models to follow instructions with human feedback. In Koyejo, S.; Mohamed, S.; Agarwal, A.; Belgrave, D.; Cho, K.; and Oh, A., eds., *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

Papernot, N.; McDaniel, P. D.; Goodfellow, I. J.; Jha, S.; Celik, Z. B.; and Swami, A. 2016. Practical Black-Box Attacks against Deep Learning Systems using Adversarial Examples. *CoRR*, abs/1602.02697.

Papernot, N.; McDaniel, P. D.; Goodfellow, I. J.; Jha, S.; Celik, Z. B.; and Swami, A. 2017. Practical Black-Box Attacks against Machine Learning. In Karri, R.; Sinanoglu, O.; Sadeghi, A.; and Yi, X., eds., *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security, AsiaCCS 2017, Abu Dhabi, United Arab Emirates, April 2-6, 2017*, 506–519. ACM.

Pathak, D.; Agrawal, P.; Efros, A. A.; and Darrell, T. 2017. Curiosity-driven Exploration by Self-supervised Prediction. In Precup, D.; and Teh, Y. W., eds., *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, 2778–2787. PMLR.

Perez, E.; Huang, S.; Song, H. F.; Cai, T.; Ring, R.; Aslanides, J.; Glaese, A.; McAleese, N.; and Irving, G. 2022. Red Teaming Language Models with Language Models. In Goldberg, Y.; Kozareva, Z.; and Zhang, Y., eds., *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, 3419–3448. Association for Computational Linguistics.

Puterman, M. L. 2014. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons.

Radharapu, B.; Robinson, K.; Aroyo, L.; and Lahoti, P. 2023. AART: AI-Assisted Red-Teaming with Diverse Data Generation for New LLM-powered Applications. In Wang, M.; and Zitouni, I., eds., *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: EMNLP 2023 - Industry Track, Singapore, December 6-10, 2023*, 380–395. Association for Computational Linguistics.

Sakaguchi, K.; Bras, R. L.; Bhagavatula, C.; and Choi, Y. 2021. WinoGrande: an adversarial winograd schema challenge at scale. *Commun. ACM*, 64(9): 99–106.

Samvelyan, M.; Raparthy, S. C.; Lupu, A.; Hambro, E.; Markosyan, A. H.; Bhatt, M.; Mao, Y.; Jiang, M.; Parker-Holder, J.; Foerster, J.; Rocktäschel, T.; and Raileanu, R. 2024. Rainbow Teaming: Open-Ended Generation of Diverse Adversarial Prompts. arXiv:2402.16822.

Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; and Klimov, O. 2017. Proximal Policy Optimization Algorithms. *CoRR*, abs/1707.06347.

Simon, H. A. 1956. Rational choice and the structure of the environment. *Psychological review*, 63 2: 129–38.

Sutton, R. S.; and Barto, A. G. 1998. Reinforcement Learning: An Introduction. *IEEE Trans. Neural Networks*, 9: 1054–1054.

Taylor, J. 2016. Quantilizers: A Safer Alternative to Maximizers for Limited Optimization. In Bonet, B.; Koenig, S.; Kuipers, B.; Nourbakhsh, I. R.; Russell, S.; Vardi, M. Y.; and Walsh, T., eds., *AI, Ethics, and Society, Papers from the 2016 AAAI Workshop, Phoenix, Arizona, USA, February 13, 2016*, volume WS-16-02 of *AAAI Technical Report*. AAAI Press.

Tevet, G.; and Berant, J. 2021. Evaluating the Evaluation of Diversity in Natural Language Generation. In Merlo, P.; Tiedemann, J.; and Tsarfaty, R., eds., *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, EACL 2021, Online, April 19 - 23, 2021*, 326–346. Association for Computational Linguistics.

Yi, S.; Liu, Y.; Sun, Z.; Cong, T.; He, X.; Song, J.; Xu, K.; and Li, Q. 2024. Jailbreak Attacks and Defenses Against Large Language Models: A Survey. arXiv:2407.04295.

Yurkiewicz, J. 1985. Constrained optimization and Lagrange multiplier methods, by D. P. Bertsekas, Academic Press, New York, 1982, 395 pp. Price: $65.00. *Networks*, 15(1): 138–140.

Zellers, R.; Holtzman, A.; Bisk, Y.; Farhadi, A.; and Choi, Y. 2019. HellaSwag: Can a Machine Really Finish Your Sentence? In Korhonen, A.; Traum, D. R.; and Màrquez, L., eds., *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, 4791–4800. Association for Computational Linguistics.

Zhang, J.; Zhou, Y.; Liu, Y.; Li, Z.; and Hu, S. 2024. Holistic Automated Red Teaming for Large Language Models through Top-Down Test Case Generation and Multi-turn Interaction. In *Conference on Empirical Methods in Natural Language Processing*.

Zhao, A.; Lin, M. G.; Li, Y.; Liu, Y.; and Huang, G. 2022. A Mixture Of Surprises for Unsupervised Reinforcement Learning. In Koyejo, S.; Mohamed, S.; Agarwal, A.; Belgrave, D.; Cho, K.; and Oh, A., eds., *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

Zhu, Y.; Lu, S.; Zheng, L.; Guo, J.; Zhang, W.; Wang, J.; and Yu, Y. 2018. Texygen: A Benchmarking Platform for Text Generation Models. In Collins-Thompson, K.; Mei, Q.; Davison, B. D.; Liu, Y.; and Yilmaz, E., eds., *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, SIGIR 2018, Ann Arbor, MI, USA, July 08-12, 2018*, 1097–1100. ACM.

Zou, A.; Wang, Z.; Kolter, J. Z.; and Fredrikson, M. 2023. Universal and Transferable Adversarial Attacks on Aligned Language Models. *CoRR*, abs/2307.15043.