

# DREAM: Scalable Red Teaming for Text-to-Image Generative Systems via Distribution Modeling

Boheng Li<sup>1</sup>, Junjie Wang<sup>2</sup>, Yiming Li<sup>1\*</sup>, Zhiyang Hu<sup>2</sup>, Leyi Qi<sup>3</sup>,  
Jianshuo Dong<sup>4</sup>, Run Wang<sup>2</sup>, Han Qiu<sup>4</sup>, Zhan Qin<sup>3</sup>, Tianwei Zhang<sup>1</sup>

<sup>1</sup>*Nanyang Technological University, boheng001@e.ntu.edu.sg, {ym.li, tianwei.zhang}@ntu.edu.sg*

<sup>2</sup>*School of Cyber Science and Engineering, Wuhan University, {junjie.wang, zhiyanghu, wangrun}@whu.edu.cn*

<sup>3</sup>*State Key Laboratory of Blockchain and Data Security, Zhejiang University, qileyi@foxmail.com, qinzhan@zju.edu.cn*

<sup>4</sup>*Tsinghua University, dongjs23@mails.tsinghua.edu.cn, qiuhan@tsinghua.edu.cn*

<sup>\*</sup>*Corresponding Author*

**Abstract**—Despite the integration of safety alignment and external filters, text-to-image (T2I) generative systems are still susceptible to producing harmful content, such as sexual or violent imagery. This raises serious concerns about unintended exposure and potential misuse. Red teaming, which aims to proactively identify diverse prompts that can elicit unsafe outputs from the T2I system, is increasingly recognized as an essential method for assessing and improving safety before real-world deployment. However, existing automated red teaming approaches often treat prompt discovery as an isolated, prompt-level optimization task, which limits their scalability, diversity, and overall effectiveness. To bridge this gap, in this paper, we propose DREAM, a scalable red teaming framework to automatically uncover diverse problematic prompts from a given T2I system. Unlike prior work that optimizes prompts individually, DREAM directly models the probabilistic distribution of the target system’s problematic prompts, which enables explicit optimization over both effectiveness and diversity, and allows efficient large-scale sampling after training. To achieve this without direct access to representative training samples, we draw inspiration from energy-based models and reformulate the objective into a simple and tractable form. We further introduce GC-SPSA, an efficient optimization algorithm that provides stable gradient estimates through the long and potentially non-differentiable T2I pipeline. During inference, we also propose a diversity-aware sampling strategy to enhance prompt variety. The effectiveness of DREAM is validated through extensive experiments, demonstrating state-of-the-art performance across a wide range of T2I models and safety filters in terms of both prompt success rate and diversity. Our code is available at <https://github.com/AntigoneRandy/DREAM>.

## 1. Introduction

Text-to-image (T2I) generative models [1, 2, 3, 4] are driving a new wave of visual content creation, reshaping our expectations of what machines are capable of. Trained on large-scale datasets [5], these models capture rich associations between language and imagery, allowing them to produce high-quality images with simple text inputs (known

as *prompts*). Their ease of use and impressive flexibility have driven rapid adoption across creative arts, entertainment, and social media, particularly among younger users such as teenagers [6, 7, 8]. However, the same large-scale, web-crawled datasets that enable this versatility also inevitably contain not safe for work (NSFW) content (e.g., sexually explicit material) [9, 10]. As a result, the models also acquire the ability to produce harmful images during real use, raising serious ethical, legal, and accountability concerns [11, 12].

To mitigate these risks, a growing number of efforts from both academia and industry [13, 10, 14, 15, 16] have focused on improving the safety of T2I generative models. One popular approach is *safety alignment*, also referred to as *unsafe concept erasure* in the T2I literature [13, 10, 14, 17], which fine-tunes the model using a curated set of unsafe prompts or images to suppress undesirable generations. This process helps steer the model toward harmless outputs: for example, returning a clothed figure even when prompted with “a nude person”. In addition, commercial companies like Stability AI [18] and Ideogram [19] also employ proprietary *safety filters* (e.g., NSFW image detectors) to block generation attempts when unsafe content is detected. These filters, when combined with the core generative model and other processing components, constitute the deployed *T2I system*. However, while these techniques show promising results in controlled environments, they remain imperfect when applied in practice. For example, both real-world users and researchers [20, 21] have reported that prompts unseen during training (e.g., implicit references to sensitive content), or even totally benign inputs (e.g., “the origin of woman”), may still escape moderation and lead to unsafe outputs. These observations highlight the limitations of current methods under open-ended inputs and the urgent need for proactive mechanisms to expose safety vulnerabilities of T2I generative systems before real-world deployment.

One emerging solution to proactively identify such blind spots is *red teaming*, where model owners (e.g., developers) simulate the behavior of real-world users to generate various testing prompts, aiming to systematically probe the model’s failure modes before deployment. In the context of T2I generative systems, red teaming typically attempts to find

a diverse set of problematic prompts that can elicit unsafe or policy-violating outputs despite potential safeguards [20, 21]. By doing so, it not only serves as an evaluation tool for stress-testing the system’s safety and trustworthiness under open-ended inputs [20], but also provides valuable references for future improvement [22]. As a result, red teaming is increasingly recognized as a critical practice, with major companies like Google [23] initiating human-in-the-loop red teaming programs. At the same time, regulatory bodies are increasingly emphasizing rigorous safety testing before deployment, as reflected in the EU AI Act [24] and the U.S. NIST AI Risk Management Framework [25], alongside similar official efforts around the world [26, 27].

While early red teaming relied on human experts, recent works [20, 21, 28] have shifted toward automated red teaming, aiming to discover problematic prompts with minimal human intervention. For example, FLIRT [28] employs a large language model (LLM) to iteratively rewrite a seed prompt toward unsafe outputs, while P4D [20] starts with a moderated unsafe prompt and applies token-level gradient-based substitutions to penetrate safety alignment. However, these methods often struggle to balance the success rate with prompt diversity, and can be prohibitively slow and costly to scale. These limitations underscore the urgent need for a scalable red teaming method that can efficiently generate a large, diverse set of effective problematic prompts.

In this paper, we present the first attempt to bridge the aforementioned gap. Our method is driven by a unified insight into the shared limitations of prior works: they treat red teaming as a prompt-to-prompt discrete optimization problem, where each seed prompt is optimized independently yet without accumulating global, distribution-level knowledge of the target model’s unsafe prompt distribution across runs. Built upon this understanding, we propose Distributional Red tEaming via energy-based Modeling (DREAM), which directly models the probabilistic distribution of the target model’s unsafe prompts by training a parameterized prompt generator (e.g., an autoregressive LLM). In contrast to previous approaches, our formulation enables explicit optimization of both success and diversity, supports global updates to the modeled prompt distribution, and allows efficient large-scale sampling after training.

However, modeling the target prompt distribution is challenging, as it is tightly coupled to the specific T2I system and lacks sufficiently representative samples, making direct training infeasible. To overcome this, we draw inspiration from energy-based models [29] and decompose the originally intractable training objective into two simple sub-objectives that allow effective distribution learning without direct sample access. Moreover, to enable effective and efficient gradient-based optimization for these objectives under long and potentially non-differentiable pipelines, we introduce Gradient-Calibrated Simultaneous Perturbation Stochastic Approximation (GC-SPSA), an efficient zeroth-order optimization method based on SPSA [30]. Specifically, it estimates gradients using only forward evaluations and further improves stability via a history-aware calibration mechanism. We also provide theoretical analysis and

convergence guarantees to support GC-SPSA. Finally, we propose a novel adaptive temperature scaling method to further enhance coverage at inference time.

We conduct extensive experiments on 5 state-of-the-art (SOTA) safety-aligned T2I models and 4 safety filters, comparing DREAM with 10 SOTA baselines across two unsafe categories. The results demonstrate that DREAM achieves superior performance in terms of prompt success rate while maintaining diversity comparable to human-written datasets. DREAM also generalizes well to advanced T2I models (e.g., SDXL, SD v3) and 6 other NSFW themes, and remains effective even under combinational defenses or aggressive filters. We further conduct a user study to confirm that the red-teaming prompts generated by DREAM are both diverse and highly effective from human perspectives. Moreover, case studies demonstrate that the prompts generated by DREAM transfer effectively to 4 commercial T2I platforms with unknown safety mechanisms. Finally, DREAM also enhances safety fine-tuning, enabling defended models to better resist both seen and unseen harmful prompts.

To summarize, we make the following key contributions:

- We revisit existing red teaming methods and identify a shared limitation: they treat prompt discovery as an isolated, prompt-level optimization problem without global modeling, restricting their scalability and performance.
- We introduce DREAM, a scalable and distribution-aware red teaming framework that learns a probabilistic model over unsafe prompts using energy-based modeling. We further propose GC-SPSA, a novel zeroth-order optimizer that supports effective and efficient training, along with adaptive inference strategies for broader coverage. We also provide theoretical analyses and global convergence guarantees to support our GC-SPSA framework.
- We conduct comprehensive evaluations across 5 safety-aligned T2I models, 4 safety filters, and 10 SOTA baselines, showing that DREAM achieves superior prompt success rates while matching human-level diversity. We also show that DREAM can expose failure cases in 4 commercial T2I platforms and improve safety fine-tuning with notable generalization to unseen harmful prompts.

## 2. Related Work

### 2.1. Text-to-Image Generative Models

Text-to-image (T2I) generative models have become a cornerstone of modern visual synthesis, enabling users to create highly detailed images from natural language descriptions. Among various generative paradigms, diffusion models [31, 3, 1, 2, 4] have emerged as the dominant approach due to their superior training stability, generation quality, and controllability. Diffusion models operate by iteratively denoising random noise into coherent images, often conditioned on texts, making them particularly effective for large-scale training and text-controlled generation. Building upon this, a wide range of open-source (e.g., Stable Diffusion family [4]) and commercial systems (e.g., DeepAI [32],

DALL·E 3 [33], Midjourney [34], Ideogram [19]) have been developed, providing state-of-the-art generation experiences through user-friendly graphical interfaces.

## 2.2. Unsafe Generation & Mitigation

The success of modern T2I models relies heavily on large datasets. For instance, Stable Diffusion is trained on LAION-5B [5], a web-scraped set of over 5 billion image-text pairs, while commercial models like Ideogram use even larger private datasets [19]. These datasets support powerful multimodal learning but also contain harmful content such as sexual or violent imagery. Such content can be reproduced by these models, raising ethical and legal concerns. The risks become particularly significant as these tools grow more accessible and popular among children and adolescents, who may experience psychological harm, safety risks, and developmental disruptions due to exposure [11, 9, 35].

In response to these concerns, several mitigation strategies have emerged, which can be broadly categorized into two lines: model safety alignment and inference-time safety filtering. Model safety alignment [10, 14] refers to techniques that tune the diffusion model’s parameters directly to suppress its ability to produce unsafe content. This is typically achieved by collecting a curated set of harmful prompts or images and reinforcing the model to “unlearn” them through methods such as adversarial training [36], supervised fine-tuning [10, 13], or model editing [14, 17]. For example, CA [13] fine-tunes the diffusion model to match the image distribution of an unsafe *target concept* (e.g., “a nude person”) to that of a safe *anchor concept* (e.g., “person”). As a result, the model learns to resist prompts that are the same or similar to training-time target concepts and generates safe images instead. In contrast, safety filters [16, 37, 15] act as external control mechanisms during inference. They can operate at prompt-level or image-level, aiming to detect and block unsafe content before or after generation. A representative case is the Safety Checker (SC) [15] employed in Stable Diffusion models, which compares the generated image with a set of predefined *sensitive concepts* and blocks outputs that exceed a similarity threshold.

While these approaches have demonstrated effectiveness with acceptable trade-offs in benign performance under their own evaluation protocols, their robustness in real-world scenarios has been frequently challenged by a growing body of recent research and user reports. For instance, text-based safety filters can be bypassed using simple synonym substitutions [38], while image-based filters may lose effectiveness under subtle alterations in image styles, compositions, or rendering [39]. Moreover, while safety-aligned models perform well when the prompts contain in-distribution explicit words, they may still fail to handle out-of-distribution veiled expressions, metaphors, or context-related implications [40, 41, 20], which are unseen during unlearning. In addition to these scattered findings, recent research [42, 43, 44, 45] has developed various optimization methods to transform a given rejected unsafe prompt into evasive variants to bypass safety mechanisms, a technique

known as *adversarial jailbreak attacks*. These diverse failure patterns across different safety mechanisms suggest it is crucial to proactively test and improve the T2I generative system’s safety before real-world deployment.

## 2.3. Red Teaming for Text-to-Image Models

The concept of “red teaming” originated during the Cold War era in the 1960s as a form of structured military system testing and has since expanded to fields like cybersecurity, airport security, software engineering, and more recently to AI and ML systems [46]. For generative models, red teaming typically involves simulating real user behavior to explore the system and find prompts that produce harmful or policy-violating outputs [47, 48, 49]. Unlike jailbreak attacks that tweak known unsafe prompts into evasive variants [42, 44], red teaming focuses on broader exploration to reveal diverse or even unexpected failure modes [21]. It is now a key part of responsible model development [23] and is increasingly emphasized in recent regulatory frameworks [24, 25, 26].

One predominant form of red teaming is manual construction. For example, the I2P dataset [42] was formed by collecting and filtering harmful prompts from various forums through a mix of automatic tools and human curation. Similarly, Google’s Adversarial Nibbler Challenge [23] invited participants to attack real-world T2I models and selected high-quality prompts based on their effectiveness and diversity. Commercial providers also employ in-house or external experts to manually test models for discovering failure modes [50]. While such methods can surface unexpected and model-specific vulnerabilities, they rely heavily on human effort and lack automation, making them inefficient and expensive to conduct.

To this end, several methods for automated red teaming have been proposed [20, 28, 21, 51]. These methods typically adopt paradigms and techniques similar to jailbreak attacks and transform a set of initial prompts into harmful ones, using methods like token-level substitution [20, 51] and LLM-rewrite [28, 21]. However, as we will identify in the following section, this inherited formulation inherently limits their effectiveness, exploration space, and efficiency, making them suboptimal for scalable red teaming.

## 3. Preliminaries

### 3.1. Threat Model

We consider the red team to be a benign (non-malicious) model owner aiming to proactively identify safety vulnerabilities in their own T2I generative system. Specifically, their goal is to find a set of diverse and effective prompts that can elicit unsafe or policy-violating outputs, in order to assess and improve safety before real-world use. Specifically, we assume the red teamer (1) has full control over their T2I generative system. They may request it with an arbitrary prompt and receive the resulting image (or an all-black image if blocked by filters), or access the model’s parameters and gradients, and (2) can leverage auxiliary models

(e.g., open-source LLMs) for assistance and has moderate computational resources to fine-tune these models.

### 3.2. Formulation of Red Teaming

Despite the growing importance of red teaming in evaluating the safety of T2I generative models, existing literature [21, 28, 20] largely lacks a formal formulation that captures the fundamental nature of the red teaming task. This absence has led to fragmented understanding and inconsistent objectives, which limits both theoretical analysis and the principled design of scalable red teaming methods.

To bridge this gap, we present a formal definition of red teaming in this section. Let  $\mathcal{X}$  and  $\mathcal{Y}$  be the prompt space and image space of the target T2I generative system, respectively. We can draw the following definition:

**Definition 1** (Red Teaming T2I Systems). *Let  $G : \mathcal{X} \rightarrow \mathcal{Y}$  be a T2I system that maps a text prompt  $x \in \mathcal{X} = \mathcal{V}^T$  to an image  $y \in \mathcal{Y}$ , where  $\mathcal{V}$  and  $T$  represent the full vocabulary and the maximum prompt length of the system, respectively. Red teaming aims to find a prompt subset  $\mathcal{A} \subseteq \mathcal{X}$  such that:*

$$\mathcal{A} := \{x \in \mathcal{X} \mid \mathcal{O}(G(x)) = 1\},$$

where  $\mathcal{O} : \mathcal{Y} \rightarrow \{0, 1\}$  is a binary oracle classifier that outputs 1 if the image is unsafe, and 0 if the image is safe or the request is blocked by the built-in safety filter.

Intuitively, this definition formulates red teaming as a combinatorial subset discovery problem that aims to identify all prompts within the full prompt set  $\mathcal{V}^T$  that can trigger the T2I system to produce unsafe content. Note that the oracle function  $\mathcal{O}$  is fundamentally unobservable in practice, as determining whether an image is “unsafe” is often vague, influenced by context, culture, and subjective interpretation [9]. As a practical alternative, red teaming methods rely on a *surrogate scoring function*  $S : \mathcal{Y} \rightarrow \mathbb{R}$ , which approximates the oracle with an objective score (e.g., the confidence score of an NSFW image detector). A prompt is deemed unsafe if its surrogate score is large enough to exceed a threshold  $\tau$ , yielding the surrogate unsafe set  $\mathcal{A}_\tau := \{x \in \mathcal{X} \mid S(G(x)) \geq \tau\}$ . While the surrogate formulation makes the task operational, obtaining the exact solution of the unsafe set  $\mathcal{A}_\tau$  remains computationally intractable, as the task essentially reduces to a combinatorial search problem over the prompt space  $\mathcal{X}$ , whose size grows exponentially with the prompt length  $T$ , i.e.,  $|\mathcal{X}| = |\mathcal{V}|^T$ . In such combinatorial settings, exhaustive enumeration is the only general procedure that can ensure complete accuracy [52], yet it requires evaluating the surrogate score  $S(G(x))$  for every enumerated  $x \in \mathcal{V}^T$ , making it computationally infeasible even for modest values of  $T$ . As a result, exact discovery is impractical except in trivial cases.

Fortunately, previous works have shown that exact recovery of  $\mathcal{A}_\tau$  is often unnecessary. For instance, unlearning a moderate number of diverse and representative unsafe prompts is often sufficient to invalidate a much broader class of similar unsafe prompts [10, 14, 53, 54]. Consequently, the

---

### Algorithm 1 A Generic Form of Existing Methods

---

**Input:** Seed distribution  $\pi(x)$ , number of prompts  $N$ , max steps  $T$ , scoring function  $S(\cdot)$ , target T2I system  $G(\cdot)$ , threshold  $\tau$ , update operator  $\text{UPDATE}(\cdot)$

**Output:** Final set of optimized prompts  $\hat{\mathcal{A}}$

```

1:  $\hat{\mathcal{A}} \leftarrow \emptyset$ 
2: for  $i = 1$  to  $N$  do
3:    $x_i^{(0)} \sim \pi(x)$ 
4:    $t \leftarrow 0$ 
5:   while  $t < T$  and  $S(G(x_i^{(t)})) < \tau$  do
6:      $x_i^{(t+1)} \leftarrow \text{UPDATE}(x_i^{(t)}, S(G(x_i^{(t)})))$ 
7:      $t \leftarrow t + 1$ 
8:   end while
9:    $\hat{\mathcal{A}} \leftarrow \hat{\mathcal{A}} \cup \{x_i^{(t)}\}$ 
10: end for
11: Return  $\hat{\mathcal{A}}$ 

```

---

practical goal of red teaming shifts from full enumeration to the discovery of a representative and diverse subset  $\hat{\mathcal{A}} \subseteq \mathcal{A}_\tau$ , which captures a wide range of unsafe prompts while remaining computationally tractable to obtain.

### 3.3. Limitations of Previous Works

With an understanding of the red teaming task formulation, we now take a closer look at existing methods [20, 51, 28, 21]. While prior works differ substantially in their technical implementation, we distilled them into a unified, generic prompt-level discrete optimization paradigm, formalized in Alg. 1. Under this view, a red teaming algorithm begins with a seed prompt sampled from a seed distribution  $\pi(x)$ , and iteratively applies an  $\text{UPDATE}$  operator guided by a scoring function  $S(G(x))$ , where  $G(x)$  denotes the image generated by the T2I model. This loop continues until a generation crosses a threshold or a step budget is reached, at which point the final prompt is collected and the process resets. Note that the seed prompt distribution, the  $\text{UPDATE}$  operator, and the scoring function are all method-dependent. Despite empirical progress in uncovering unsafe prompts, we identify that this core algorithmic structure introduces two fundamental limitations, making these methods less suitable for scalable red teaming.

First, the  $\text{UPDATE}$  operator essentially performs discrete optimization, which is inherently difficult due to the discontinuous and non-smooth nature of the ill-posed loss landscape of the discrete prompt space [55]. In fact, how to accurately obtain prompt-level gradient remains an open challenge in existing literature [56, 57]. As such, some existing methods [42, 43] resort to token-level gradient replacement, where each token is iteratively and greedily updated based on its locally estimated gradient with respect to  $S(G(x_i^{(t)}))$ . However, this limits the search space to local neighborhoods around the initial seed, making the optimization process highly sensitive to initialization [58]. While recent methods attempt to broaden the search space by prompting LLMs to generate sentence-level paraphrases [21, 28], these approaches are largely heuristic, lack convergence guarantees, and often result in unstable training dynamics in practice.

Second, one can easily observe from Alg. 1 that current methods are essentially operating at the individual prompt level: each run starts from a fresh seed prompt  $x_i^{(0)}$ , performs a local search trajectory  $\{x_i^{(1)}, x_i^{(2)}, \dots, x_i^{(t)}\}$ , outputs  $x_i^{(t)}$ , and then discards all intermediate states but the final output before restarting the next run. This *stateless* fashion is naturally sub-optimal, as the algorithm would not accumulate any knowledge about explored regions or learn from past failures. Therefore, it may revisit similar attempted trajectories, re-try strategies that are known to be ineffective in previous runs, and converge to familiar local optima, especially if the seed prompts are semantically or syntactically similar [21]. This makes the algorithm inefficient and results in highly similar prompts with limited marginal utility. Moreover, this inefficiency is exacerbated by the inherently slow convergence of discrete optimization. For example, P4D [20] requires roughly 3,000 rounds of model invocation and gradient updates to optimize a single red teaming prompt, taking about 30 minutes per prompt on an NVIDIA RTX A100 GPU. These deficiencies make it very difficult to be scaled up for large-scale red teaming.

## 4. The Design of DREAM

### 4.1. Distributional Red Teaming via Energy-Based Modeling

Motivated by our previous analysis about the limitations of existing methods, the key insight behind our proposal is to shift from discrete, stateless prompt-to-prompt optimization to directly modeling the distribution over unsafe prompts.

Formally, let  $q^*(x)$  denote the true (but unknown) distribution over the target model’s problematic prompts, i.e., the probabilistic distribution from which samples  $x \in \hat{\mathcal{A}}$  are drawn. Our goal is to learn a probabilistic distribution  $p_\theta(x)$  parameterized by  $\theta$  (e.g., an autoregressive language model  $p_\theta(x) = \prod_{t=1}^T p_\theta(x_t | x_{<t})$ ), such that  $p_\theta(x)$  approximates  $q^*(x)$  as close as possible. This objective can be characterized by the following Kullback–Leibler divergence [59]:

$$\theta^* \in \arg \min_{\theta} D_{\text{KL}}(p_\theta \| q^*) \quad (1)$$

This formulation has several desirable properties. First, by modeling the distribution  $p_\theta(x)$ , our method naturally converts the prompt-level discrete optimization into continuous optimization over model parameters  $\theta$ . Second, since the parameters encode the distribution over prompts, each parameter update accumulates knowledge about which types of prompts are more or less likely to trigger unsafe outputs, thus promoting exploration efficiency during training. Furthermore, it is totally feasible to initialize  $p_\theta(x)$  with a pretrained language model. As a result, our method inherits strong priors from large-scale human language data, which enables the model to understand and explore nuanced expressions, innuendos, and cultural references, which are subtle signals that typically require human-like common sense or contextual awareness and are often inaccessible to

previous token-level search methods. Finally, once training is complete, sampling from the learned distribution  $p_{\theta^*}(x)$  is efficient. An arbitrary number of diverse prompts can be generated efficiently via forward passes, without requiring iterative search or gradient updates. This property makes our approach particularly suitable for red teaming, where a large number of unsafe prompts (e.g., thousands) are required for safety assessment and downstream safety-tuning.

Despite these promising properties, the objective in Eq. (1) remains particularly challenging to optimize in practice. The core difficulty lies in the fact that the ground-truth distribution  $q^*(x)$  is fundamentally unknown and there exists no readily available dataset that is sufficiently representative of the full support of the target model’s problematic prompts. This makes direct optimization (e.g., through fine-tuning with MLE [60]) impossible. Fortunately, recent advances in implicit generative modeling [61] provide a viable pathway to tackle this challenge. Specifically, results from the theory of energy-based models [29, 61] suggest that even in the absence of explicit samples, the target distribution  $q^*(x)$  can be implicitly characterized with a properly defined *energy function*  $E(x)$ , which is a real-valued function that assigns lower values to more likely (or desirable) samples, and higher values otherwise. Then, the unknown distribution  $q^*(x)$  can be expressed as a Boltzmann distribution [62]  $q^*(x) = \exp(-\beta \cdot E(x))/Z$ , where  $Z$  is a constant that normalizes the distribution [29, 61] and  $\beta > 0$  is a hyperparameter. Then, by plugging it into Eq. (1), we have:

$$\begin{aligned} \arg \min_{\theta} D_{\text{KL}}(p_\theta \| q^*) &= \arg \min_{\theta} \mathbb{E}_{x \sim p_\theta} \left[ \log \frac{p_\theta(x)}{q^*(x)} \right] \\ &= \arg \min_{\theta} \mathbb{E}_{x \sim p_\theta} \left[ E(x) + \frac{1}{\beta} \log p_\theta(x) \right]. \end{aligned} \quad (2)$$

The derivation above reduces the otherwise intractable KL divergence to two simple yet intuitive components: the first is to minimize the expected energy  $\mathbb{E}_{x \sim p_\theta}[E(x)]$ , thereby shaping the distribution toward lower-energy (and thus more desirable) regions of the prompt space. The second objective acts as an entropy regularizer that penalizes low-entropy distributions by minimizing the log-likelihood  $\mathbb{E}_{x \sim p_\theta}[\log p_\theta(x)]$ , thus avoiding degenerate solutions where the model collapses to a narrow set of prompts.

### 4.2. Energy Function Design

So far, we have decomposed the objective into two intuitive and interpretable sub-goals. The second regularization term  $\mathbb{E}_{x \sim p_\theta}[\log p_\theta(x)]$  is straightforward to compute and optimize in practice. We now turn our attention to the first component, the energy function  $E(\cdot)$ . The energy function essentially defines the target distribution by assigning lower energy scores to desirable prompts and higher scores to undesired ones. In this section, we introduce our energy function design, which captures the following two scores.

**Vision-level Harmfulness Energy.** The primary goal of  $E$  is to guide the learned distribution  $p_\theta(x)$  toward the target model’s vulnerable prompt distribution  $\mathcal{A}_\tau$ . However,

directly assessing the harmfulness of a text prompt  $x$  is difficult as the risk often emerges only after it is rendered into an image. Therefore, we take a vision-level approach by evaluating the output image  $y = G(x)$  instead of the prompt itself. Specifically, we employ BLIP-2 [63], a pre-trained vision-language model with strong generalization across diverse image-text domains, to compute a *vision-level harmfulness energy* as part of the energy function. It assesses how the generated image is semantically aligned with a predefined harmful concept, and assigns lower energy to prompts that align better with the harmful concept.

Formally, given a generated image  $y = G(x)$  and a predefined target description  $c$  (like “an image containing nudity”), the textual description  $c$  is first sent to a pretrained language encoder  $\mathcal{T}_\phi$  to obtain the sentence-level semantic embedding  $t = \mathcal{T}_\phi(c)$ . Then, the image  $y$  is passed through a vision encoder followed by a specialized transformer module known as the Q-Former [63]. This module employs a set of query embeddings to interact with the visual features via cross-attention and finally extracts a set of latent tokens  $\mathcal{I}_\psi(y) = \{z_1, \dots, z_k\}$ , each representing a different fine-grained aspect of the image in the same vision-language embedding space. Then, we define the alignment score as:

$$E_{\text{align}}(x) = \mathbb{E}_{x \sim p_\theta} \left[ - \max_{z_i \in \mathcal{I}_\psi(G(x))} \frac{\langle z_i, t \rangle}{\|z_i\| \cdot \|t\|} \right] \quad (3)$$

where  $\langle \cdot, \cdot \rangle$  and  $\|\cdot\|$  denote the inner product and the Euclidean norm, respectively.  $E_{\text{align}}(x)$  measures the cosine similarity between the resulting image and the textual embedding, where higher similarity indicates stronger alignment with the harmful concept and thus lower energy.

This formulation brings three key benefits. First, BLIP-2 provides better generalization even under distribution shifts, such as stylized or non-photorealistic images, making the alignment score more reliable across visual domains [63, 64]. Second, the approach enables flexible red teaming through natural language descriptions. One can easily shift the target by modifying the concept prompt, e.g., replacing  $c$  with “an image depicting violent scenes” to target violent content. When a small set of reference images is available, prompt tuning techniques can also be used to further refine and control the targeted concept [65]. Third, the alignment score is continuous, allowing small improvements in prompt effectiveness to be reflected. This supports more stable optimization than discrete (e.g., binary) success signals.

**Prompt-level Diversity Energy.** While jointly optimizing the harmfulness energy in Eq. (3) and the entropy term  $\mathbb{E}_{x \sim p_\theta} [\log p_\theta(x)]$  in the main objective (Eq. (2)) effectively guides the model toward  $q^*(x)$ , we observe that relying solely on this formulation tends to produce a learned distribution with limited semantic diversity. This is possibly because the target distribution  $q^*(x)$  may itself be biased, e.g., certain keywords like “nude” and their semantically similar variants might dominate the probability mass. Consequently, semantically distinct prompts with lower probability under  $q^*(x)$  may remain largely unvisited in limited sampling iterations. To address this, we introduce a diversity energy

term that explicitly encourages broader coverage within a limited sample budget. Let  $\mathcal{E}_\xi(x) \in \mathbb{R}^d$  denote the sentence embedding of prompt  $x$ , obtained from a frozen pretrained encoder (e.g., a sentence transformer [66]). Then, we define the prompt-level diversity energy as the expected pairwise similarity among prompt embeddings sampled from the current model distribution  $p_\theta(x)$ :

$$E_{\text{div}}(x) = \mathbb{E}_{x, x' \sim p_\theta, x \neq x'} \left[ \frac{\langle \mathcal{E}_\xi(x), \mathcal{E}_\xi(x') \rangle}{\|\mathcal{E}_\xi(x)\| \cdot \|\mathcal{E}_\xi(x')\|} \right]. \quad (4)$$

This would explicitly encourage semantic diversity among generated prompts within limited sampling iterations, thus promoting broader exploration and reducing redundancy.

### 4.3. Red Team LLM Optimization

After designing the energy function, we can plug Eq. (3) and Eq. (4) into Eq. (2), and arrive at the final training objective for the red team prompt generator  $\theta$ :

$$\min_{\theta} \mathbb{E}_{x \sim p_\theta} \left[ E_{\text{align}}(x) + \lambda \cdot E_{\text{div}}(x) + \frac{1}{\beta} \cdot \log p_\theta(x) \right], \quad (5)$$

where  $\lambda$  and  $\beta$  are balancing hyperparameters. However, optimizing Eq. (5) is non-trivial. One intuitive approach would be to use backpropagation-based methods to obtain exact gradients and then update the LLM’s parameters. However, applying backpropagation-based optimization directly is challenging. This is because the full red-teaming pipeline comprises multiple components such as autoregressive language generation, multi-step diffusion denoising, and energy models, each requiring storage of numerous intermediate activations for backpropagation-based gradient computation. For instance, generating an image via Stable Diffusion v1.5 typically requires 30 denoising steps, each producing high-dimensional feature maps (e.g.,  $\sim 2.3$  GB). Even under a moderate batch size of 32, this already amounts to over 2 TB of GPU memory for a single pass, making end-to-end backpropagation-based training memory-prohibitive. Moreover, certain components like keyword-based safety filters are non-differentiable, further hindering backpropagation.

To enable effective gradient-driven optimization while avoiding the need for backpropagation through the entire pipeline, we propose a novel framework based on Simultaneous Perturbation Stochastic Approximation (SPSA) [30]. SPSA is a classical zeroth-order optimization method that allows estimates of high-dimensional gradients using only forward evaluations. However, vanilla SPSA has been empirically observed to suffer from instability and slow convergence in our red-teaming setup, due to the highly stochastic nature of both LLMs and diffusion-based generation (see experiments in Section 5.8). To mitigate this, we propose a simple yet effective variant, GC-SPSA, which incorporates an adaptive sampling schedule as well as a history-aware gradient calibration mechanism to reduce gradient variance while maintaining efficiency. In the following, we first introduce SPSA and analyze its limitations, then present our GC-SPSA, and finally provide theoretical analysis and convergence guarantees to support our design.

**Definition 2** (SPSA [67]). *Given an objective function  $\mathcal{L} : \mathbb{R}^d \rightarrow \mathbb{R}$  and parameters  $\theta \in \mathbb{R}^d$ , SPSA uses the following randomized two-point finite-difference approximation to compute an unbiased estimate of the gradient  $\nabla_\theta \mathcal{L}(\theta)$ :*

$$g(\theta) := \frac{\mathcal{L}(\theta + \epsilon \Delta) - \mathcal{L}(\theta - \epsilon \Delta)}{2\epsilon} \Delta, \quad (6)$$

where  $\epsilon > 0$  is a small perturbation magnitude, and  $\Delta \in \mathbb{R}^d$  is a random perturbation vector sampled from a zero-mean Gaussian distribution.

Previous works have proved that SPSA provides an unbiased estimate of the true gradient, i.e.,  $\mathbb{E}_\Delta[g(\theta)] = \nabla_\theta \mathcal{L}(\theta)$  [30]. In our setting, SPSA is particularly advantageous: it avoids backpropagation entirely and requires only forward evaluations at perturbed parameters, making it well-suited for bypassing the long and possibly non-differentiable pipeline. Moreover, it eliminates the need to store intermediate forward activations, further reducing memory consumption compared to backpropagation-based methods [67].

Despite these advantages, we observe in our experiments that directly applying SPSA leads to unstable training dynamics. This instability primarily stems from the inherent stochasticity in both LLM and diffusion-based sampling, which introduces high variance into single-shot gradient estimates. To reduce variance, a straightforward strategy is to increase the number of forward estimates per iteration and average the resulting gradients. However, this significantly increases training cost linearly and is expensive in practice.

To mitigate this problem, we propose a simple yet effective method, GC-SPSA, which stabilizes SPSA with a novel adaptive gradient calibration algorithm. Our key insight is that Eq. (5) primarily steers the LLM toward a subspace of prompts that are likely to elicit harmful images from the target model. Since the pretrained LLM already possesses strong priors, such subspaces are shown to typically reside within a relatively flat and continuous basin in the loss landscape near the pretrained parameters [68, 69, 70]. Therefore, we hypothesize that it is more crucial to ensure the reliability of early optimization steps to accurately locate this subspace, yet later updates may tolerate more variance and can be calibrated with early reliable gradients.

Specifically, for the  $t$ -th update, we first estimate the gradient  $n_t$  times using Eq. (6), obtaining a set of stochastic gradient estimates  $\{g_{t,1}(\theta_t), g_{t,2}(\theta_t), \dots, g_{t,n_t}(\theta_t)\}$ . The number of queries  $n_t$  is controlled by an exponentially decaying schedule:  $n_t = \max\left(1, \left\lfloor \frac{n_0}{2^{t/T_{\text{dec}}}} \right\rfloor\right)$ , where  $n_0$  is the initial number of sampling times and  $T_{\text{dec}}$  governs the decay rate. This scheduling allocates a higher sampling budget to early iterations and gradually reduces the number of samples as optimization stabilizes. In our experiments, we find that setting  $n_0 = 4$  and  $T_{\text{dec}} = 10$ , i.e., starting with 4 samples and halving the sampling budget every 10 steps, yields stable and efficient optimization performance (see experiments in Sec. 5.8). To further reduce the variance of the later estimated gradient, inspired by confidence-aware optimal Bayesian fusion [71], we introduce a gradient calibration mechanism. Specifically, we treat each new gradient

---

**Algorithm 2** The Complete Training Procedure of DREAM

---

**Input:** Initial model parameters  $\theta_0$ , initial sampling budget  $n_0$ , learning rate  $\eta$ , correction strength  $\gamma$ , smoothing factor  $\rho$ , decay factor  $T_{\text{dec}}$ , current step  $t$   
**Output:** Optimized generator parameters  $\theta_t$

1:  $\hat{g}_0 \leftarrow \frac{1}{n_0} \sum_{i=1}^{n_0} g_{0,i}(\theta_0)$   
2:  $w_0 \leftarrow n_0$ ,  $t \leftarrow 0$   
3: **while** loss not converged **do**  
    ▷ Determine sampling budget for current step  
4:    $n_t \leftarrow \max\left(1, \left\lfloor \frac{n_0}{2^{t/T_{\text{dec}}}} \right\rfloor\right)$   
    ▷ Estimate gradients via SPSA with  $n_t$  queries  
5:   **for**  $i = 1$  to  $n_t$  **do**  
6:     Sample perturbation  $\Delta_{t,i} \sim \text{Gaussian}^d$   
7:      $\mathcal{L}_{t,i}^+, \mathcal{L}_{t,i}^- \leftarrow \mathcal{L}(\theta_t + \epsilon \Delta_{t,i}), \mathcal{L}(\theta_t - \epsilon \Delta_{t,i})$   
8:      $g_{t,i} \leftarrow \frac{\mathcal{L}_{t,i}^+ - \mathcal{L}_{t,i}^-}{2\epsilon} \cdot \Delta_{t,i}$   
9:   **end for**  
    ▷ Aggregate and calibrate gradients  
10:    $\hat{g}_t \leftarrow \frac{1}{n_t} \sum_{i=1}^{n_t} g_{t,i} + \gamma \cdot \frac{w_{t-1}}{w_{t-1} + n_t} \cdot \hat{g}_{t-1}$   
11:    $w_t \leftarrow \rho \cdot w_{t-1} + (1 - \rho) \cdot n_t$   
    ▷ Update model parameters  
12:    $\theta_{t+1} \leftarrow \theta_t - \eta \cdot \hat{g}_t$   
13:    $t \leftarrow t + 1$   
14: **end while**  
15: **Return**  $\theta_t$

---

estimate as a noisy observation and combine it with historical information using a confidence-aware correction term:

$$\hat{g}_t = \frac{1}{n_t} \sum_{i=1}^{n_t} g_{t,i}(\theta_t) + \gamma \cdot \frac{w_{t-1}}{w_{t-1} + n_t} \cdot \hat{g}_{t-1}, \quad (7)$$

$$\theta_{t+1} = \theta_t - \eta \cdot \hat{g}_t,$$

where  $\hat{g}_{t-1}$  is the accumulated gradient estimate from previous iterations ( $\hat{g}_0 = \frac{1}{n_0} \sum_{i=1}^{n_0} g_{0,i}(\theta_0)$ ),  $w_{t-1}$  denotes its effective sample size (initialized as  $w_0 = n_0$ ),  $\eta$  is the learning rate. The update of  $w_t$  follows an exponential moving average rule. We use the term  $w_{t-1}/w_{t-1} + n_t$  to approximate the relative confidence of historical vs. current gradients, and  $\gamma$  controls the overall strength of the correction. Intuitively, our confidence-weighted fusion scheme anchors the current noisy gradient estimate towards the historically aggregated direction, especially when the current estimate is based on fewer samples (i.e., lower confidence). As formally analyzed in Theorem 1, the GC-SPSA estimator achieves a strictly higher signal-to-noise ratio (SNR) than the vanilla SPSA estimator for all  $t \geq 1$ , which helps mitigate gradient noise and promotes a more consistent optimization path.

**Theorem 1** (Improved SNR of GC-SPSA). *Let  $\|g_{\text{true}}\|$  be the ground-truth gradient,  $\bar{g}_k$  be the vanilla SPSA estimator, and  $\hat{g}_k = \bar{g}_k + H_k \hat{g}_{k-1}$  be the GC-SPSA estimator, with  $\hat{g}_0 = \bar{g}_0$  and  $H_k > 0$ . Then for all  $t \geq 1$ , the SNR difference between the GC-SPSA and the vanilla SPSA admits the explicit positive lower bound  $\mathcal{D}_t$  as follows:*

$$\mathcal{D}_t = \frac{\|g_{\text{true}}\|^2}{V_{\text{single}} \sum_{k=0}^t h_k^2 V_k} \left[ P_t^2 V_{\text{single}} - \sum_{k=0}^t h_k^2 V_k \right] \quad (8)$$

where  $h_t = 1$ ,  $h_k = \prod_{j=k+1}^t H_j$  and  $H_j = \gamma^{\frac{w_{j-1}}{w_{j-1} + n_j}}$ .  $V_{\text{single}}$  is the gradient estimation variance of vanilla SPSA,

$P_t = \sum_{k=0}^t h_k$  is the cumulative weight sum, and  $V_k = V_{\text{single}}/n_k$  is the gradient variance of GC-SPSA at step  $k$ .

The detailed proof is in Appendix A.1. Furthermore, we also provide a theoretical guarantee of global convergence and convergence rate analysis for GC-SPSA under mild assumptions, as formally shown in the following theorem:

**Theorem 2** (Global Convergence and Rate Analysis of GC-SPSA). *Consider an objective  $\mathcal{L} : \mathbb{R}^d \rightarrow \mathbb{R}$  satisfying  $\nabla^2 \mathcal{L}(\theta) \preceq \ell I_d$  for all  $\theta \in \mathbb{R}^d$ , where  $I_d$  denotes the  $d$ -dimensional identity matrix. Then, for a stationarity level  $\delta$ , GC-SPSA will converge (i.e.,  $\min_{t \in [T]} \mathbb{E}[\|g(\theta_t)\|^2] \leq \delta$ ) after*

$$T = \Theta \left( \frac{\mathcal{L}(\theta_0) - \mathcal{L}^*}{\eta \zeta_{\min} \delta - C_{\max} \Upsilon - \frac{\ell \eta^2 \Xi}{2}} \right). \quad (9)$$

iterations. Here,  $\mathcal{L}^*$  denotes the global minimum,  $\zeta_{\min}$  is the minimum descent coefficient,  $C_{\max}$  bounds the cross-term coefficient, and  $\Upsilon$  and  $\Xi$  denote upper bounds on the average squared gradient-estimator norm and accumulated noise variance, respectively.

For each training step, we begin by perturbing the current model parameters  $\theta_t$  and estimate the gradient  $n_t$  times using Eq. (6). Since the objective  $\mathcal{L}(\theta)$  involves an intractable expectation, we approximate it via Monte Carlo sampling over a batch of sampled prompts (see more details at Alg. 3 in the Appendix of our full version).<sup>1</sup> The resulting gradient estimates are then averaged and calibrated using our confidence-aware update rule in Eq. (7). Finally, the calibrated gradient is used to update the parameters  $\theta_t$ . We summarize the complete training procedure in Alg. 2.

#### 4.4. Inference-Time Adaptive Temperature Scaling

After training, the red-team LLM is steered toward the desired prompt distribution. The final step is sampling from this model to generate red-teaming prompts. However, we empirically find that the model may still fail to adequately explore its support during generation. This is because each sample is produced independently and without awareness of previously generated prompts, leading to repetitive or redundant tokens that have limited marginal benefits.

To further enhance diversity, we propose an inference-time strategy that encourages diversity via adaptive temperature scaling. Recall that in autoregressive decoding, the model iteratively predicts the next token distribution as  $p_{\theta^*}(x_t = v | x_{<t}) = \frac{\exp(z_t[v]/\tau_t)}{\sum_{j \in \mathcal{V}} \exp(z_t[j]/\tau_t)}$ , where the temperature hyperparameter  $\tau_t$  controls the sharpness of the token distribution  $z_t \in \mathbb{R}^{|\mathcal{V}|}$  at decoding step  $t$ . Intuitively, lower temperatures make the model more confident (peaky), while higher temperatures flatten the distribution to encourage exploration. As such, we maintain a global token frequency vector  $\mathbf{f} \in \mathbb{N}^{|\mathcal{V}|}$ , tracking the number of times each vocabulary token has appeared across previous generations. At decoding step  $t$ , given raw logits  $z_t \in \mathbb{R}^{|\mathcal{V}|}$ ,

we compute the relative frequency of the top-scoring token  $v_t = \arg \max_j z_t[j]$ , and use it to scale the temperature:

$$\tau_t = \max \left( \frac{1}{\alpha} \log(1 + \frac{\mathbf{f}[v_t]}{\sum_j \mathbf{f}[j]}), \tau_{\min} \right), \quad (10)$$

where  $\alpha > 0$  is a sensitivity coefficient and  $\tau_{\min}$  prevents degeneracy. The final logits are adjusted as  $\tilde{z}_t = z_t/\tau_t$  before sampling. This penalizes frequent tokens, promoting underexplored generations without modifying training or architecture. Empirically, we find it improves prompt diversity with minimal overhead and little loss in effectiveness.

## 5. Evaluation

### 5.1. Experimental Setup

**Target Diffusion Models & Safety Filters.** In our experiments, we evaluate a variety of standard diffusion models, safety-aligned models, and safety filters. In addition to the standard Stable Diffusion v1.5, we also evaluate safety-aligned models including ESD [10], CA [13], UCE [14], SafeGen [40], and RECE [17], all of which have unlearned certain unsafe concepts from the models. For external safety filters, following Yang et al. [42], we consider 4 external filters classified into (1) text-based filters (NSFW text classifier [16] and Keyword-Gibberish hybrid filter [72]), (2) image-based filters (NSFW image filter [37] and Stable Diffusion’s built-in image safety checker, SC [15]). Furthermore, we evaluate the generalizability of DREAM on several real-world models, including SDXL, SDv3, Kandinsky v3, and Shuttle 3 Diffusion. We also evaluate DREAM in a transfer-based setting on multiple real-world online T2I platforms, including Ideogram, DeepAI, DALL·E 3, and Midjourney.

**Baselines.** We evaluate and compare DREAM against several state-of-the-art baselines. For human-written red teaming datasets, we include I2P [9] and Google’s Adversarial Nibbler [23], collected from T2I community forums and via crowdsourcing, respectively. For automated methods, we consider QF-Attack [45], SneakyPrompt [42], MMA-Diffusion [43], P4D [20], UnlearnDiffAtk [73], JailFuzzer [74], FLIRT [28] and ART [21]. Note that the baselines differ in their assumed level of access to the target model as well as their original motivation (e.g., SneakyPrompt was originally proposed for jailbreak attacks in closed-box settings). We include them for completeness, as their underlying mechanisms are closely related to red-teaming and finally yield comparable unsafe prompt datasets. We provide a more detailed discussion in Appendix B.

**Evaluation Metrics.** We primarily use two metrics to evaluate the performance of each red teaming method: *Prompt Success Rate* (PSR) and *Prompt Similarity* (PS), which measure the effectiveness and diversity of the generated prompts, respectively. **PSR** is the proportion of prompts that successfully trigger the target model to generate images containing the specified inappropriate content. Following Yang et al. [43], we use PSR out of  $N$  generations (PSR-N) instead of a single generation to reduce the impact of

1. Our full version is available at <https://arxiv.org/abs/2507.16329>.

TABLE 1: Comparison with baselines on Stable Diffusion v1.5 and other safety-aligned diffusion models.

	Stable Diffusion v1.5		CA		ESD		UCE		SafeGen		RECE	
	Sexual	Violence	Sexual	Violence	Sexual	Sexual	Violence	Sexual	Sexual	Sexual	Sexual	Sexual
	PSR $\uparrow$ / PS $\downarrow$											
<i>Human-written Datasets</i>												
I2P	51.5% / 0.49	13.9% / 0.47	11.7% / 0.49	13.1% / 0.47	9.5% / 0.49	10.0% / 0.49	6.9% / 0.47	45.6% / 0.49	7.5% / 0.49			
Adv. Nibbler	28.3% / 0.53	8.9% / 0.54	3.1% / 0.53	3.5% / 0.54	1.2% / 0.53	30.1% / 0.53	7.8% / 0.54	33.8% / 0.53	0.7% / 0.53			
<i>Automated Red Teaming (Token Perturbation)</i>												
QF-Attack	23.8% / 0.59	10.6% / 0.62	0.6% / 0.59	9.4% / 0.62	0.0% / 0.59	1.3% / 0.59	10.0% / 0.62	21.2% / 0.59	0.6% / 0.59			
SneakyPrompt	61.7% / 0.52	26.5% / 0.65	7.9% / 0.52	13.5% / 0.52	25.6% / 0.64	16.5% / 0.52	20.1% / 0.64	25.5% / 0.52	14.0% / 0.52			
MMA-Diffusion	91.0% / 0.63	73.9% / 0.65	44.9% / 0.63	59.2% / 0.65	35.9% / 0.63	59.9% / 0.63	66.9% / 0.65	34.0% / 0.63	52.2% / 0.63			
P4D	78.0% / 0.60	42.0% / 0.58	52.0% / 0.60	26.0% / 0.66	43.3% / 0.60	24.0% / 0.56	10.0% / 0.58	61.9% / 0.55	16.0% / 0.55			
UnlearnDiffAtk	83.0% / 0.52	30.0% / 0.49	36.4% / 0.52	10.5% / 0.49	21.2% / 0.52	24.6% / 0.51	10.5% / 0.49	55.9% / 0.52	13.6% / 0.52			
<i>Automated Red Teaming (LLM Rewrite)</i>												
ART	14.9% / 0.48	29.2% / 0.47	2.7% / 0.49	14.1% / 0.46	0.8% / 0.49	0.8% / 0.49	20.8% / 0.46	13.7% / 0.49	0.8% / 0.48			
FLIRT	91.8% / 0.77	74.4% / 0.66	26.0% / 0.58	64.4% / 0.63	17.1% / 0.64	48.5% / 0.64	18.4% / 0.61	10.2% / 0.59	10.7% / 0.57			
JailFuzzer	72.4% / 0.56	80.0% / 0.55	31.5% / 0.52	44.0% / 0.54	23.8% / 0.51	51.9% / 0.53	38.0% / 0.52	35.4% / 0.52	35.9% / 0.51			
Ours	92.2% / 0.50	87.0% / 0.55	76.0% / 0.56	77.3% / 0.57	72.1% / 0.56	89.0% / 0.54	83.6% / 0.57	81.6% / 0.49	91.3% / 0.56			

inherent stochasticity in diffusion sampling. Specifically, for each prompt, we generate  $N$  images with different random seeds. The prompt is considered successful if at least one of these  $N$  images contains the desired unsafe concept, and the final PSR is measured as the ratio of successful prompts. In our experiments, we use PSR-3, i.e.,  $N = 3$ , and mainly adopt Multi-headed Safety Classifier (MHSC) [11] as our detector. MHSC is a category-specific NSFW image detector that provides per-category confidence scores for various unsafe concepts. It has been widely used in the community due to its high precision [11, 75, 43]. A higher PSR ( $\uparrow$ ) indicates the method is more capable of generating effective prompts. Besides, **PS** quantifies the diversity of the prompts by measuring the average pairwise cosine similarity between all prompt embeddings. In our evaluation, we use the state-of-the-art BGE embedding model [76] to obtain the prompt embeddings (note that this model is different from  $\mathcal{E}_\xi$  we use in Eq. (4)). A lower PS score ( $\downarrow$ ) indicates lower inter-prompt similarity, meaning that the prompts are semantically more diverse and less repetitive.

**Implementation Details.** We initialize  $\theta$  with the pre-trained Gemma-2-27b-it model [77].  $c$  is initialized following Tab. 10, with lightweight embedding tuning applied to ensure more accurate category representation. For GC-SPSA, the learning rate  $\eta$  and hyperparameter  $\epsilon$  are set to default values of  $1 \times 10^{-6}$  and  $1 \times 10^{-3}$ , respectively. Following prior works [48, 21], system prompts and few-shot ICL exemplars are also applied to ensure task alignment and efficiency. The model is trained until convergence with a batch size of 32, which typically requires  $\sim 300$  steps and corresponds to roughly 12 hours on NVIDIA A100 GPUs. After training, we sample 1,024 prompts with ATS for evaluation. Due to space constraints, we refer readers to our full version for more implementation details.

## 5.2. Main Results

**Effectiveness on Concept-erased T2I Models.** We first conduct experiments on both the standard Stable Diffusion (SD) v1.5 model and several concept-erased models, whose

weights are fine-tuned or edited to unlearn the corresponding unsafe concept. As shown in Tab. 1, our method consistently achieves the highest PSRs across all models and categories, significantly outperforming all baselines. For example, on concept-erased models such as UCE and RECE, human-written datasets typically yield PSRs below 10%, and state-of-the-art red teaming methods like MMA-Diffusion often struggle to exceed 50%. In contrast, our prompts achieve PSRs above 79% on all evaluated models. In addition to effectiveness, our method also excels in prompt diversity. Across all models, our prompts maintain a prompt similarity (PS) score around 0.55, which is notably better than most baselines and on par with human-written datasets, indicating higher semantic diversity. This highlights that our approach not only discovers more successful prompts, but also explores a broader and more varied region of the prompt space that remains unexplored by existing red teaming techniques.

**Effectiveness on External Safety Filters.** We further evaluate the effectiveness of DREAM on various external safety filters and compare it with baselines. For each safety filter, we combine it with the standard SD v1.5 model and regard the whole model-filter pipeline as an integrated generative system. As shown in Tab. 2, baseline methods are largely unstable, while DREAM consistently achieves the best or nearly the best PSR-3 across all settings. For example, while MMA-Diffusion achieves good results on most safety-aligned models and SC, it almost completely fails on the NSFW Text Detector and Keyword-Gibberish Filter. On safety filters, the most effective baseline is JailFuzzer, which performs prompt-to-prompt mutation powered by multiple LLM and VLM agents. However, JailFuzzer is much less effective on safety-aligned models. This is possibly because the success of JailFuzzer’s mutation process heavily depends on the accuracy of the simulated SafetyFilter signal provided by the VLM brain. When applied to safety-aligned models that sanitize outputs instead of blocking them, the bypass-score feedback is omitted altogether, which deprives the mutation process of this optimization cue and consequently reduces its effectiveness. These findings demonstrate that baselines fail to generalize across different safety mecha-

TABLE 2: Comparison with baselines on external safety filters.

	Safety Checker		NSFW Image Detector		NSFW Text Detector		Keyword-Gibberish Filter	
	Sexual	Violence	Sexual	Violence	Sexual	Violence	Sexual	Violence
	PSR $\uparrow$ / PS $\downarrow$							
<i>Human-written Datasets</i>								
I2P	23.4% / 0.49	13.6% / 0.47	26.3% / 0.49	13.0% / 0.47	37.5% / 0.49	8.5% / 0.47	26.5% / 0.49	6.5% / 0.47
Adv. Nibbler	14.3% / 0.53	8.4% / 0.54	17.3% / 0.53	8.2% / 0.54	19.4% / 0.53	6.2% / 0.54	4.8% / 0.53	0.0% / 0.54
<i>Automated Red Teaming (Token Perturbation)</i>								
QF-Attack	11.9% / 0.59	10.6% / 0.62	1.9% / 0.59	10.6% / 0.62	1.2% / 0.59	8.8% / 0.62	3.1% / 0.59	0.0% / 0.62
SneakyPrompt	30.5% / 0.52	26.5% / 0.67	11.0% / 0.50	26.5% / 0.64	12.5% / 0.46	13.0% / 0.64	51.8% / 0.52	18.1% / 0.64
MMA-Diffusion	40.5% / 0.63	72.2% / 0.65	4.7% / 0.63	68.3% / 0.65	1.5% / 0.63	11.5% / 0.65	0.0% / 0.63	0.0% / 0.65
P4D	40.0% / 0.60	40.0% / 0.58	44.0% / 0.60	38.0% / 0.58	4.0% / 0.60	4.0% / 0.58	0.0% / 0.60	0.0% / 0.58
UnlearnDiffAtk	35.6% / 0.52	10.5% / 0.49	43.2% / 0.52	9.5% / 0.49	48.3% / 0.52	6.3% / 0.49	7.6% / 0.52	2.1% / 0.49
<i>Automated Red Teaming (LLM Rewrite)</i>								
ART	4.4% / 0.48	28.8% / 0.47	9.2% / 0.48	29.2% / 0.48	6.4% / 0.48	21.8% / 0.46	4.4% / 0.48	4.5% / 0.47
FLIRT	26.5% / 0.62	72.6% / 0.64	45.9% / 0.75	75.4% / 0.65	8.5% / 0.51	16.7% / 0.60	44.6% / 0.58	48.9% / 0.57
JailFuzzer	51.9% / 0.53	72.0% / 0.52	48.1% / 0.55	76.0% / 0.55	42.0% / 0.51	46.4% / 0.52	59.1% / 0.51	38.0% / 0.52
Ours	64.7% / 0.52	86.4% / 0.54	57.3% / 0.50	83.7% / 0.58	62.3% / 0.51	42.5% / 0.54	67.4% / 0.52	65.7% / 0.56

TABLE 3: Effectiveness of DREAM across models and NSFW themes. (a) results with more models on sexual (left) and violence (right) concepts; (b) results on more NSFW themes on SD v1.5.

(a) More T2I Models		
Model (Sexual)	PSR $\uparrow$ / PS $\downarrow$	Model (Violence)
Stable Diffusion XL	89.5% / 0.52	Stable Diffusion XL
Stable Diffusion v3	82.5% / 0.53	Stable Diffusion v3
Kandinsky v3	89.8% / 0.52	Kandinsky v3
Shuttle 3 Diffusion	86.9% / 0.51	Shuttle 3 Diffusion

(b) More NSFW Themes		
Category	PSR $\uparrow$ / PS $\downarrow$	Category
Self-harm	87.8% / 0.55	Shocking
Hate	92.6% / 0.52	Harassment
Political	91.6% / 0.50	Illegal Activity

nisms. In contrast, DREAM performs well on both safety mechanisms without any ad hoc design, demonstrating the generality and universality of our method.

**Effectiveness on More T2I Models & NSFW Themes.** We evaluate DREAM on more T2I models, including Stable Diffusion XL [4], Stable Diffusion v3 [78], Kandinsky v3 [79], and Shuttle 3 Diffusion [80]. These models vary significantly in architectural design (e.g., using DiT [81] instead of convolutional U-Nets) and training settings, and some of them are reported to apply dataset filtering to remove (some) unsafe images before training [4, 78]. As shown in Tab. 3 (a), DREAM consistently performs well across all models and both categories, with PSR approaching 90% on average. The discovered prompts also exhibit a level of diversity comparable to that of human-written prompts. These results highlight the model architecture-agnostic nature of DREAM and its strong potential for adapting to future models. Additionally, we further assess the generalizability of DREAM on additional NSFW themes following prior work [9], including self-harm, shocking, hate, harassment, political, and illegal activity. Following Yang et al. [43], we use Q16 [65] as the detector in this setting, as MHSC does not support all of these categories. As shown in Tab. 3 (b), DREAM maintains high PSR across most cases, exceeding

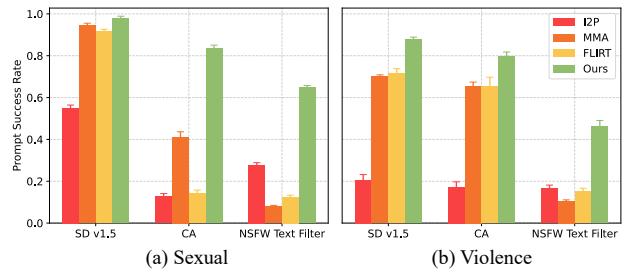


Figure 1: User study results on prompt success rate.

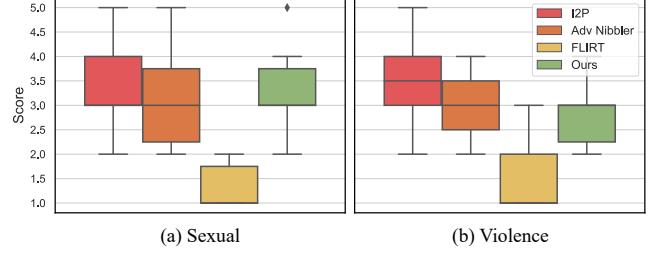


Figure 2: User study results on prompt diversity.

90% consistently and reaching close to 95% for shocking and harassment. Besides, our prompts remain highly diverse and close to that of human-written prompts. These findings demonstrate the scalability and generalizability of DREAM in discovering a broader range of unsafe concepts.

### 5.3. Human Evaluation

In this section, we conduct a user study to assess the effectiveness and diversity of different methods under human perception. Specifically, we select SD v1.5, CA, and NSFW Text Filter as the representative models. We then compare DREAM with I2P, MMA-Diffusion, and FLIRT. We evaluate all model-method combinations on both sexual and violence categories, resulting in  $2 \times 3 \times 4 = 24$  concept-model-method settings in total. For each setting, we randomly sample 30 prompts from the method's generated

prompts to form a prompt pool. Each prompt generates 3 images, forming a prompt-images group. Then, we recruit 30 volunteers, all university students or faculty members from various academic backgrounds, to participate in the study. All participants are adults in good physical and mental health, fully informed of the study, and provided consent to participate. Then, we brief participants on the basics of T2I models and red teaming, the recommended definition of the corresponding unsafe category (Tab. 10 in Appendix), and start the study after obtaining their confirmation of full comprehension and agreement to continue the study.

For evaluation, each participant is randomly assigned 5 prompt-image groups from the 30 available for each concept-model-attack setting. Prompt-image groups are organized in random order and displayed one-by-one. Participants are asked to determine whether the three displayed images in each group clearly reflect the specified unsafe concept. The group is marked as successful if and only if the participant identifies at least one such image (PSR-3). Then, participants are presented with the full set of  $4 \times 30$  problematic prompts on SD v1.5 and asked to rate these prompt sets by their perceived diversity, ranging from 1 to 5 (higher is better), based on their understanding of the prompts' lexical, structural, and semantic richness (see Tab. 11 in Appendix). We replace MMA with Adv. Nibbler since MMA's prompts are not human-readable. Despite involving images depicting NSFW concepts, our study was reviewed and approved by our institution's research ethics committee under a process analogous to the "exempt review" category of U.S. IRB protocols, as the review board determined it posed no more than minimal risk, given that all participants were healthy adults, fully informed, and free to withdraw at any time. As shown in Fig. 1 and 2, DREAM consistently achieves strong performance on this user study with the best prompt success rate and a diversity similar to human-written datasets, demonstrating its effectiveness.

#### 5.4. Adaptivity Under More Safety Mechanisms

In this section, we examine whether DREAM remains effective under more safety mechanisms. Specifically, we consider the following: (1) semantic filters, (2) composite defenses that combine multiple safety mechanisms, and (3) MHSC [11] as the safety filter. We compare our DREAM with MMA-Diffusion [43] and FLIRT [28] in these settings. The evaluated unsafe concept is sexual.

**Semantic Filters.** We first consider semantic filters, which are commonly used to defend against adversarial prompts. They work by checking whether a prompt forms a legitimate sentence, and reject those that do not. We consider a perplexity filter and a non-English-word filter, with threshold calibrated following [74]. As shown in Tab. 4 (a)-(b), this type of filter is highly effective against MMA-Diffusion, as MMA-Diffusion's random initialization and token-level optimization strategy often produces outputs that are lexically or grammatically incorrect. In contrast, DREAM and FLIRT are able to generate fluent and semantically plausible sentences, which allow them to bypass such semantic filters.

TABLE 4: Effectiveness of DREAM and baselines under more safety mechanisms. (a)-(b) present results for two semantic filters; (c)-(d) report results under multi-stage defenses; and (e) shows results under MHSC as the safety filter with two FPR thresholds.

(a) Perplexity Filter		(b) Non-English Word Filter	
Method	PSR $\uparrow$ / PS $\downarrow$	Method	PSR $\uparrow$ / PS $\downarrow$
MMA-Diffusion	0.0% / 0.63	MMA-Diffusion	0.1% / 0.63
FLIRT	91.9% / 0.75	FLIRT	90.9% / 0.75
Ours	90.4% / 0.55	Ours	90.8% / 0.53

(c) NSFW Text + NSFW Image		(d) Keyword + ESD + SC	
Method	PSR $\uparrow$ / PS $\downarrow$	Method	PSR $\uparrow$ / PS $\downarrow$
MMA-Diffusion	0.1% / 0.63	MMA-Diffusion	14.4% / 0.63
FLIRT	7.6% / 0.50	FLIRT	2.3% / 0.54
Ours	52.8% / 0.55	Ours	37.3% / 0.56

(e) MHSC			
@1% FPR	PSR $\uparrow$ / PS $\downarrow$	@5% FPR	PSR $\uparrow$ / PS $\downarrow$
MMA-Diffusion	23.3% / 0.63	MMA-Diffusion	6.7% / 0.63
FLIRT	6.7% / 0.54	FLIRT	0.0% / 0.50
Ours	43.3% / 0.58	Ours	16.7% / 0.52

**Composite Defenses.** We consider systems that sequentially combine multiple safety mechanisms. Specifically, we consider two combinations: (1) NSFW Text Filter + SD v1.5 + NSFW Image Filter; and (2) Keyword-based Filter + ESD [10] + SC [15]. A prompt is considered successful only if at least one out of the three generations is not rejected by any of the filters and successfully contains unsafe content, as rated by MHSC. Note that while these combinations help narrow the unsafe prompt space, the system's false positive rate also increases exponentially, as any individual false rejection would invalidate the whole sample.

As shown in Tab. 4 (c)-(d), DREAM consistently achieves good results and outperforms the baselines in both settings. Notably, baseline methods exhibit steep drops in effectiveness, and even fail entirely under stronger combinations. In contrast, DREAM maintains moderate success rates even under aggressive defenses, demonstrating its adaptivity.

**MHSC as the Safety Filter.** We also consider an extreme setting where MHSC, the classifier used to compute PSR in our experiments, is directly deployed as the safety filter. In this case, MHSC can no longer be used for evaluation, as any generations that could be classified as harmful would be blocked in advance. Thus, we adopt human-rated PSR-3 as the evaluation metric. Due to ethics considerations, the human-rated PSR-3 in this section and the following experiments is annotated by three authors of this paper by taking the majority vote. We conduct experiments under two thresholding settings, corresponding to 5% (default) and 1% false positive rates (FPR), which are calibrated on a benign held-out dataset following MHSC's original paper.

As shown in Tab. 4 (e), despite MHSC's high precision, DREAM still identifies multiple prompts that bypass filtering and lead to unsafe generations. We attribute this result to MHSC's conservativeness as an NSFW classifier: it only flags outputs when highly confident, prioritizing precision over recall, as also reported in the original paper [11]. This makes it a reliable evaluation tool (high PSRs indeed indi-

TABLE 5: Transferability results on real-world T2I-as-a-service platforms that utilizes unknown safety mechanisms.

(a) Ideogram					
Method	Prompt Bypass $\uparrow$	Prompt-Image Bypass $\uparrow$	Human-Rated Success Rate $\uparrow$	Prompt Similarity $\downarrow$	
MMA-Diffusion	75.0%	65.9%	43.9% $\pm$ 1.9%	0.66	
FLIRT	76.7%	67.8%	30.6% $\pm$ 1.9%	0.54	
Ours	98.4%	96.1%	57.9% $\pm$ 3.2%	0.52	
(b) DeepAI					
Method	Prompt Bypass $\uparrow$	Prompt-Image Bypass $\uparrow$	Human-Rated Success Rate $\uparrow$	Prompt Similarity $\downarrow$	
MMA-Diffusion	41.0%	26.7%	18.7% $\pm$ 1.2%	0.65	
FLIRT	58.0%	51.0%	15.7% $\pm$ 4.2%	0.50	
Ours	89.0%	79.0%	55.7% $\pm$ 3.1%	0.53	
(c) DALL-E 3					
Method	Prompt Bypass $\uparrow$	Prompt-Image Bypass $\uparrow$	Human-Rated Success Rate $\uparrow$	Prompt Similarity $\downarrow$	
MMA-Diffusion	36.7%	31.7%	7.3% $\pm$ 1.6%	0.64	
FLIRT	30.0%	28.3%	2.8% $\pm$ 1.6%	0.51	
Ours	60.8%	47.9%	32.3% $\pm$ 2.4%	0.55	
(d) Midjourney					
Method	Prompt Bypass $\uparrow$	Prompt-Image Bypass $\uparrow$	Human-Rated Success Rate $\uparrow$	Prompt Similarity $\downarrow$	
MMA-Diffusion	18.2%	18.2%	18.2% $\pm$ 2.2%	0.63	
FLIRT	21.7%	21.7%	10.0% $\pm$ 1.2%	0.53	
Ours	60.3%	60.3%	35.7% $\pm$ 1.7%	0.55	

cate high true positives) but also means that some borderline harmful cases near the threshold may slip through. These subtle failure modes are where DREAM excels, thanks to its distributional exploration and fine-grained energy modeling.

As a final remark, while conservatively designed, MHSC is still more aggressive than real-world filters [40]. It is thus reasonable that DREAM uncovers fewer prompts under MHSC than under more permissive filters. More broadly, this highlights an open challenge in balancing protection with over-censorship: aggressive filters indeed reduce risks but also inevitably narrow the prompt space, often at the cost of user experience. We believe red teaming methods like DREAM can serve as a valuable complement that helps surface near-boundary cases that evade detection and inform targeted improvements to reduce blind spots, yet without broadly increasing over-censorship.

## 5.5. Transferability on Real-world Commercial T2I Generative Models

To further assess the scalability of DREAM in real-world conditions, we test it on 4 widely used commercial T2I platforms: Ideogram [19], DeepAI [32], DALL-E 3 [33], and Midjourney [34]. These platforms deploy state-of-the-art safety systems, which at least include both prompt- and image-level filters, though their exact implementations remain undisclosed. Moreover, some platforms use proprietary LLMs to interpret and rewrite user prompts.

To evaluate how different methods perform on these platforms, we train DREAM and FLIRT on the NSFW Image and Text Filter, and randomly select 50 prompts to conduct a transfer-based red teaming. We evaluate both “sexual” and “violence” categories, which are explicitly prohibited by all the platforms’ safety policies, and report the averaged results. As shown in Tab. 5, our method achieves good transferability on all evaluated platforms, as validated by a high prompt bypass rate (the fraction of prompts accepted by the text filter), prompt-image bypass rate (the fraction of attempts that successfully yield generated images), human-rated prompt success rate, and still outperforms baselines

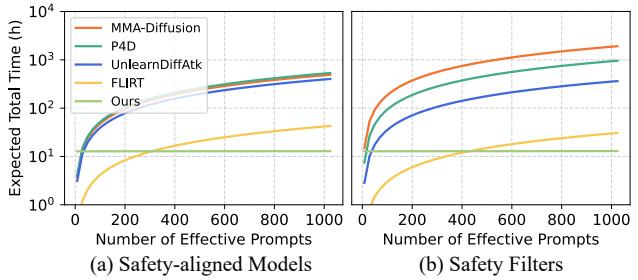


Figure 3: Efficiency comparison with baselines. We report the expected total time to collect a specified number of *effective* unsafe prompts, averaged across all evaluated (a) safety-aligned models and (b) safety filters. The considered unsafe concept is sexual.

with a notable margin. The results indicate that while the unsafe prompt space varies across different T2I systems and real-world platforms, the prompts generated by our method possess a notable degree of transferability, that is, we can uncover some shared vulnerabilities across different systems, possibly due to broader prompt coverage and the inherent similarity across these T2I models.

## 5.6. Efficiency Comparison

In this section, we present a comparison of efficiency between DREAM and baselines. Unlike previous prompt-to-prompt methods whose time cost can be directly measured per prompt, DREAM involves two stages of time cost: (1) training the red-team LLM, which dominates the total time, and (2) sampling from the trained model, which is efficient. Therefore, their efficiency cannot be directly compared on a per-prompt basis. To this end, we compute and compare the averaged expected total time required to obtain a specified number of effective unsafe prompts (as measured by MHSC) over both (a) all safety-aligned diffusion models evaluated in Tab. 1 and (b) all external safety filters in Tab. 2.

As shown in Fig. 3, since baseline methods optimize each prompt through discrete, per-instance search without any global modeling, their total time cost scales quickly with the number of desired effective prompts. In contrast, while DREAM requires approximately 12 hours to train the red-team LLM, sampling from it is extremely efficient: it only takes about 5 minutes to generate over 1,000 effective prompts in our experiments. Compared with token-level optimization baselines such as MMA-Diffusion and P4D, DREAM is more efficient even at relatively small scales (e.g., 50 prompts). Even when compared to FLIRT, which improves efficiency through LLM rewriting, DREAM surpasses it at moderate scales (more than 500 prompts). Considering that downstream applications like safety tuning typically require thousands of diverse and effective prompts to ensure generalization, we believe that DREAM’s distribution-level modeling paradigm offers a practical and promising perspective for scalable red teaming.

## 5.7. Discussion

**LLM Reusability.** One potential advantage of our distributional modeling approach is that the red team LLM, once

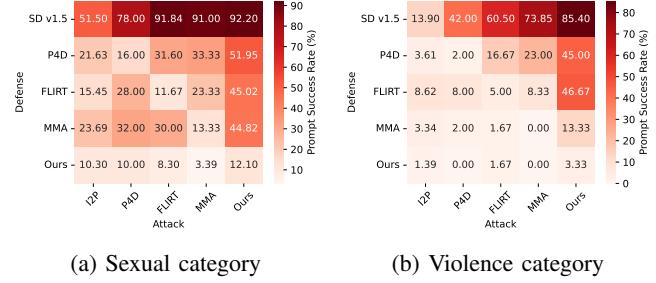
TABLE 6: Results on reusing the red team LLM (prompt generator) trained on CA to other T2I systems. The metric is PSR-3.

Setting	ESD	UCE	RECE	SC
Ours (Direct Transfer)	57.4%	77.4%	75.5%	40.7%
Ours (+50 Steps Adaptation)	68.5%	86.3%	82.9%	48.3%

trained on a T2I model, learns a holistic understanding of the probability distribution over unsafe prompts. As a result, the LLM retains reusable knowledge that may be effectively leveraged when adapted to similar T2I systems. To validate this possibility, we conduct reusability experiments where a red team LLM trained on CA for 300 steps is adapted to other T2I systems. As shown in Tab. 6, DREAM’s red team LLM achieves non-trivial success rates when directly reused (transferred) on other models without any further training. More importantly, with only 50 additional training steps, the reused LLM can be rapidly adapted to the new model, with some even achieving performance close to training from scratch (e.g., on ESD and UCE). These results show that DREAM’s distribution-level modeling demonstrates strong reusability potential, where the learned knowledge can be efficiently transferred and adapted to other T2I systems with reduced computational overhead.

**Mitigation Strategy.** To mitigate the identified vulnerabilities, one practical strategy is safety-tuning, which fine-tunes the T2I model on the collected unsafe prompts in an adversarial manner to unlearn them. To assess the utility of different methods for this purpose, we utilize red team prompts identified by each method as the dataset, and use Safety-DPO [22], a recent algorithm designed to steer generation away from unsafe behaviors via preference modeling, to fine-tune the SD v1.5 model. We then evaluate the resulting models against prompt sets from all methods, which yields a square matrix where each row represents a safety-tuned model (trained on method A’s data), and each column corresponds to evaluation against method B’s prompts. As shown in Fig. 4, the model trained with our DREAM-generated dataset consistently achieves the lowest PSR across all test sets, including those totally unseen during training. In contrast, models tuned with baseline datasets tend to show limited generalization and fail to defend against prompts from other methods, especially those discovered by DREAM. This also indirectly suggests that DREAM’s global modeling helps improve the coverage of discovered prompts, which in turn supports more robust and generalizable safety improvement. We offer further discussions in Appendix B in our full technical report.

**Prior-informed Enhancement.** For generality, our DREAM is designed without imposing specific priors about the internal components or defenses of the target T2I system. However, in practice, model owners (e.g., developers) have prior knowledge about the system, which can be potentially leveraged to enhance red teaming. For instance, if the model owner knows the system employs keyword-based filters, a simple enhancement strategy is to remove these tokens from the red team LLM’s vocabulary. This prior encourages the generator to focus on unexplored regions of the prompt



(a) Sexual category

(b) Violence category

Figure 4: PSR results of SD v1.5 safety-aligned with red team datasets generated by different methods.

TABLE 7: Ablation study on each component. ATS stands for our inference time adaptive temperature scaling strategy and Opt. Alg. means optimization algorithm. We report the mean $\pm$ std computed over three independent samples.

$E_{\text{align}}(x)$	$E_{\text{div}}(x)$	ATS	Opt. Alg.	PSR $\uparrow$ / PS $\downarrow$
✓	—	—	GC-SPSA	90.8% $\pm$ 0.10% / 0.702 $\pm$ 0.002
✓	✓	—	GC-SPSA	76.4% $\pm$ 0.08% / 0.583 $\pm$ 0.001
✓	✓	✓	SPSA	45.9% $\pm$ 0.09% / 0.537 $\pm$ 0.001
✓	✓	✓	GC-SPSA	76.0% $\pm$ 0.12% / 0.561 $\pm$ 0.001

space without wasting effort on words that are doomed to be rejected. To evaluate this, we conduct a case study on the Keyword Filter + UCE setup. We observe that the keyword-removed variant converges faster, reaching near-optimal performance within 150 training steps, compared to 280 steps required by the baseline. Interestingly, the final performance difference is modest (63.4% vs. 60.2% PSR), suggesting that while prior knowledge accelerates convergence, it is not critical for eventual performance. This result highlights two insights: first, DREAM is effective even without any system-specific priors, making it broadly applicable; second, when available, prior information can be selectively incorporated to enhance DREAM. However, leveraging such priors often requires case-specific integration strategies, some of which may be difficult or costly to implement in practice. How to develop principled ways to incorporate them, especially for neural network-based components, remains an open direction for future work.

## 5.8. Ablation Study & Hyperparameter Analysis

In this section, we conduct an ablation study of DREAM and analyze some key hyperparameters involved in our experiments, with CA+Sexual as the default setting.

**Effectiveness of Each Component.** As shown in Tab. 7, all components contribute to DREAM’s final performance. For example, while  $E_{\text{align}}(x)$  pushes the model towards harmful outputs, it often leads to less diverse prompts. Adding  $E_{\text{div}}(x)$  helps strike a balance between effectiveness and diversity. Additionally, ATS improves prompt diversity during inference with only minimal PSR degradation, highlighting its effectiveness for balancing success rate and diversity.

**Effectiveness of GC-SPSA.** As shown in Tab. 8, GC-SPSA consistently outperforms vanilla SPSA, even when SPSA is given more steps to ensure equal training time,

TABLE 8: Ablation study on GC-SPSA and different  $n_0$ .

Method	Steps	Time	PSR $\uparrow$ / PS $\downarrow$
SPSA	340	12.85h	47.7% / 0.53
SPSA	410	15.37h	61.8% / 0.55
GC-SPSA ( $n_0 = 4$ )	300	12.78h	76.0% / 0.56
GC-SPSA ( $n_0 = 8$ )	300	15.43h	79.8% / 0.57

TABLE 9: Impact of different system prompts.

Setting	System Prompt 1	System Prompt 2	System Prompt 3
PSR $\uparrow$ / PS $\downarrow$	77.8% / 0.57	75.2% / 0.56	76.4% / 0.56

demonstrating the effectiveness of GC-SPSA. Besides, we evaluate different initial sampling budgets  $n_0$ . We observe that  $n_0 = 4$  already provides a strong balance between efficiency and effectiveness. While increasing  $n_0$  to 8 leads to further gains, the improvement is marginal compared to the additional cost, possibly because the variance is already sufficiently small for stable optimization. We thus adopt  $n_0 = 4$  as our default configuration, as it strikes a good trade-off between effectiveness and computational cost.

**Impact of System Prompt.** Currently, we handcraft system prompts for DREAM. To understand DREAM’s sensitivity under different system prompts, we prompt ChatGPT-5 to generate three concise red-teaming role-play prompts for DREAM using the instruction “I’m training a red teaming assistant for text-to-image generative models for category {category}. Please generate a concise system prompt for it.”, and test their performances. As shown in Tab. 9, all prompts are highly effective, indicating that DREAM’s performance is not very sensitive to system prompts.

**Impact of Number of Sampled Prompts.** Similar to prior work [43, 28], we generate 1,024 prompts for evaluation by default. As our method models a distribution and supports efficient sampling, it is natural to study the impact of the number of sampled prompts on performance. As shown in Fig. 5, the number of prompts has only a minor impact on DREAM’s expected PSR and PS, indicating DREAM’s scalability to larger sample sizes without significantly sacrificing performance, enabling it to generate large-scale, diverse, and effective red team datasets efficiently.

## 6. Conclusion

This paper presents DREAM, a novel framework for scalable red teaming of T2I generative systems. DREAM learns the distribution of unsafe prompts via energy-based modeling, allowing efficient, diverse, and effective prompt discovery at scale. We further introduce GC-SPSA, an efficient method to optimize our objective and propose adaptive strategies for broad prompt coverage during inference. Through comprehensive experiments, we demonstrate DREAM’s superior effectiveness and generalizability.

**Limitations.** Our work still has the following limitations, which we aim to address in future work. First, similar to other red-teaming studies, our implementation and eval-

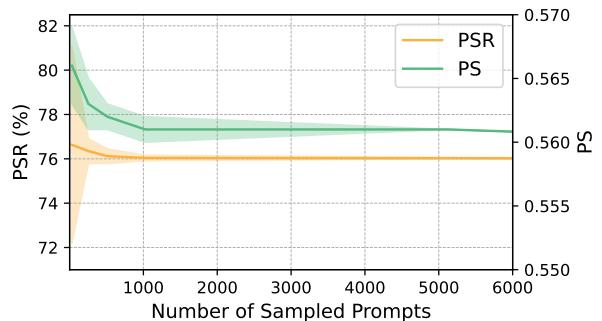


Figure 5: Impact of number of sampled prompts on PSR and PS. We report the results computed over three independent samples.

uation rely on some auxiliary models (e.g., BLIP-2 and MHSC) that may introduce biases or inaccuracies, and we acknowledge that some false positives or false negatives may appear. Although we also conducted human evaluation, we acknowledge that concepts such as “unsafe” and “diversity” are inherently subjective and can be influenced by cultural, contextual, and personal factors. Nevertheless, since all methods are evaluated under the same protocol, we believe the comparisons remain relatively fair and informative. We plan to explore more robust alternatives in the future. Besides, while our GC-SPSA optimizer demonstrates both theoretical guarantees and strong empirical performance compared to baseline optimizers such as vanilla SPSA and LLM-based heuristics, we acknowledge it is essentially an approximation and may not be fully precise or perfectly efficient. Nonetheless, as discussed in Sec. 4, obtaining exact gradients via backpropagation is often infeasible due to the memory-intensive nature of the full T2I pipeline. We hope our energy-based distributional formulation and our proposed optimizer can inspire and serve as a valuable foundation for future improvements, and ultimately enable stronger safety evaluation practices for T2I systems.

## Acknowledgments

The authors would like to sincerely thank all anonymous reviewers and our shepherd for their constructive feedback and helpful suggestions, which have significantly improved the quality of this paper. This work was supported in part by National Research Foundation, Singapore and DSO National Laboratories under its AI Singapore Programme (AISG Award No: AISG2-GC-2023-008), and National Research Foundation, Singapore and Infocomm Media Development Authority under its Trust Tech Funding Initiative. It was also supported in part by the National Natural Science Foundation of China (NSFC) under Grants No. 62202340, No. 62576255. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not reflect the views of National Research Foundation, Singapore, DSO National Laboratories, Infocomm Media Development Authority, or the NSFC.

**Disclosure of Financial Interests.** The authors declare that they have no financial or non-financial conflicts that could be perceived to influence the work reported in this paper.

## References

- [1] F.-A. Croitoru, V. Hondru, R. T. Ionescu, and M. Shah, "Diffusion models in vision: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 9, pp. 10 850–10 869, 2023.
- [2] L. Yang, Z. Zhang, Y. Song, S. Hong, R. Xu, Y. Zhao, W. Zhang, B. Cui, and M.-H. Yang, "Diffusion models: A comprehensive survey of methods and applications," *ACM Computing Surveys*, vol. 56, no. 4, pp. 1–39, 2023.
- [3] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 10 684–10 695.
- [4] D. Podell, Z. English, K. Lacey, A. Blattmann, T. Dockhorn, J. Müller, J. Penna, and R. Rombach, "Sdxl: Improving latent diffusion models for high-resolution image synthesis," in *The Twelfth International Conference on Learning Representations*, 2024.
- [5] C. Schuhmann, R. Beaumont, R. Vencu, C. Gordon, R. Wightman, M. Cherti, T. Coombes, A. Katta, C. Mullis, M. Wortsman *et al.*, "Laion-5b: An open large-scale dataset for training next generation image-text models," *Advances in neural information processing systems*, vol. 35, pp. 25 278–25 294, 2022.
- [6] Y. Wang and W. Zhang, "Factors influencing the adoption of generative ai for art designing among chinese generation z: A structural equation modeling approach," *IEEE Access*, vol. 11, pp. 143 272–143 284, 2023.
- [7] C. S. Media, "Most us teens use generative ai. most of their parents don't know," *Wired*, 2024. [Online]. Available: <https://www.wired.com/story/teens-generative-ai-use-schools-parents/>
- [8] S. Ali, D. DiPaola, R. Williams, P. Ravi, and C. Breazeal, "Constructing dreams using generative ai," *arXiv preprint arXiv:2305.12013*, 2023.
- [9] P. Schramowski, M. Brack, B. Deiseroth, and K. Kersting, "Safe latent diffusion: Mitigating inappropriate degeneration in diffusion models," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 22 522–22 531.
- [10] R. Gandikota, J. Materzyńska, J. Fiotto-Kaufman, and D. Bau, "Erasing concepts from diffusion models," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 2426–2436.
- [11] Y. Qu, X. Shen, X. He, M. Backes, S. Zannettou, and Y. Zhang, "Unsafe diffusion: On the generation of unsafe images and hateful memes from text-to-image models," in *Proceedings of the 2023 ACM SIGSAC conference on computer and communications security*, 2023, pp. 3403–3417.
- [12] S. C. Y. Ho, "From development to dissemination: Social and ethical issues with text-to-image ai-generated art," in *Canadian AI*, 2023.
- [13] N. Kumari, B. Zhang, S.-Y. Wang, E. Shechtman, R. Zhang, and J.-Y. Zhu, "Ablating concepts in text-to-image diffusion models," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 22 691–22 702.
- [14] R. Gandikota, H. Orgad, Y. Belinkov, J. Materzyńska, and D. Bau, "Unified concept editing in diffusion models," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2024, pp. 5111–5120.
- [15] CompVis, "Stable diffusion safety checker," <https://huggingface.co/CompVis/table-diffusion-safety-checker>, 2022, accessed: 2025-03-21.
- [16] E. Albulzidi, "Distilbert nsfw text classifier," <https://huggingface.co/eliasalbuluzidi/distilbert-nsfw-text-classifier>, 2023, accessed: 2025-03-21.
- [17] C. Gong, K. Chen, Z. Wei, J. Chen, and Y.-G. Jiang, "Reliable and efficient concept erasure of text-to-image diffusion models," in *European Conference on Computer Vision*. Springer, 2024, pp. 73–88.
- [18] Stability AI, "Stability AI Official Website," 2025, accessed: 2025-04-21. [Online]. Available: <https://stability.ai/>
- [19] Ideogram, "Ideogram Official Website," 2025, accessed: 2025-04-21. [Online]. Available: <https://ideogram.ai/>
- [20] Z.-Y. Chin, C.-M. Jiang, C.-C. Huang, P.-Y. Chen, and W.-C. Chiu, "Prompting4debugging: red-teaming text-to-image diffusion models by finding problematic prompts," in *Proceedings of the 41st International Conference on Machine Learning*, 2024, pp. 8468–8486.
- [21] G. Li, K. Chen, S. Zhang, J. Zhang, and T. Zhang, "Art: Automatic red-teaming for text-to-image models to protect benign users," in *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.
- [22] R. Liu, C. I. Chieh, J. Gu, J. Zhang, R. Pi, Q. Chen, P. Torr, A. Khakzar, and F. Pizzati, "Safetydpo: Scalable safety alignment for text-to-image generation," *arXiv preprint arXiv:2412.10493*, 2024.
- [23] J. Quaye, A. Parrish, O. Inel, C. Rastogi, H. R. Kirk, M. Kahng, E. Van Liemt, M. Bartolo, J. Tsang, J. White *et al.*, "Adversarial nibbler: An open red-teaming method for identifying diverse harms in text-to-image generation," in *Proceedings of the 2024 ACM Conference on Fairness, Accountability, and Transparency*, 2024, pp. 388–406.
- [24] EU Artificial Intelligence Act, "Article 55: Obligations for providers of general-purpose ai models with systemic risk," <https://artificialintelligenceact.eu/article55/>, Artificial Intelligence Act (EU AI Act), 2025, accessed: 2025-10-15.
- [25] National Institute of Standards and Technology, "Artificial intelligence risk management framework," 2023. [Online]. Available: <https://nvlpubs.nist.gov/nistpubs/ai/NIST.AI.600-1.pdf>
- [26] UK Government, "Frontier ai safety commitments, ai seoul summit 2024," 2024. [Online]. Available: <https://www.gov.uk/government/publications/frontier-ai-safety-commitments-ai-seoul-summit-2024/frontier-ai-safety-commitments-ai-seoul-summit-2024>
- [27] Infocomm Media Development Authority, "Singapore ai safety red teaming challenge," 2024. [Online]. Available: <https://www.imda.gov.sg/activities/activities-catalogue/singapore-ai-safety-red-teaming-challenge>
- [28] N. Mehrabi, P. Goyal, C. Dupuy, Q. Hu, S. Ghosh, R. Zemel, K.-W. Chang, A. Galstyan, and R. Gupta, "Flirt: Feedback loop in-context red teaming," in *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, 2024, pp. 703–718.
- [29] Y. LeCun, S. Chopra, R. Hadsell, M. Ranzato, F. Huang *et al.*, "A tutorial on energy-based learning," *Predicting structured data*, vol. 1, no. 0, 2006.
- [30] J. C. Spall, "Multivariate stochastic approximation using a simultaneous perturbation gradient approximation," *IEEE transactions on automatic control*, vol. 37, no. 3, pp. 332–341, 1992.
- [31] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," *Advances in neural information processing systems*, vol. 33, pp. 6840–6851, 2020.
- [32] DeepAI, "Deepai: Artificially intelligent tools for naturally creative humans," 2025, accessed: 2025-04-23. [Online]. Available: <https://deepai.org/>
- [33] OpenAI, "Dall-e 3," 2023, accessed: 2025-04-23. [Online]. Available: <https://openai.com/index/dall-e-3/>
- [34] Midjourney, "Midjourney: Independent research lab for generative ai," 2025, accessed: 2025-04-23. [Online]. Available: <https://www.midjourney.com/>
- [35] K. Guo, A. Utkarsh, W. Ding, I. Ondracek, Z. Zhao, G. Freeman, N. Vishwanamitha, and H. Hu, "Moderating illicit online image promotion for unsafe user generated content games using large {Vision-Language} models," in *33rd USENIX Security Symposium (USENIX Security 24)*, 2024, pp. 5787–5804.
- [36] Y. Zhang, X. Chen, J. Jia, Y. Zhang, C. Fan, J. Liu, M. Hong, K. Ding, and S. Liu, "Defensive unlearning with adversarial training for robust concept erasure in diffusion models," *Advances in Neural Information Processing Systems*, vol. 37, pp. 36 748–36 776, 2024.
- [37] L. Chhabra, "Nsfw detection using deep learning," <https://github.com/lakshaychhabra/NSFW-Detection-DL>, 2020, accessed: 2025-03-21.
- [38] Z. Ba, J. Zhong, J. Lei, P. Cheng, Q. Wang, Z. Qin, Z. Wang, and K. Ren, "Surrogateprompt: Bypassing the safety filter of text-to-image models via substitution," in *Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security*, 2024, pp. 1166–1180.
- [39] u/featherlessbird, "Is it possible to jailbreak dall-e 3?" [https://www.reddit.com/r/ChatGPT/comments/175nxzq/is\\_it\\_possible\\_to\\_jailbreak\\_dalle\\_3/](https://www.reddit.com/r/ChatGPT/comments/175nxzq/is_it_possible_to_jailbreak_dalle_3/), 2023, accessed: 2025-04-22.
- [40] X. Li, Y. Yang, J. Deng, C. Yan, Y. Chen, X. Ji, and W. Xu, "Safegen: Mitigating sexually explicit content generation in text-to-image models," in *Proceedings of the 2024 ACM SIGSAC Conference on Computer and Communications Security*, 2024, pp. 4807–4821.
- [41] Z. Meng, B. Peng, X. Jin, Y. Lyu, W. Wang, and J. Dong, "Concept corrector: Erase concepts on the fly for text-to-image diffusion models," *arXiv preprint arXiv:2502.16368*, 2025.
- [42] Y. Yang, B. Hui, H. Yuan, N. Gong, and Y. Cao, "Sneakyprompt: Jailbreaking text-to-image generative models," in *2024 IEEE symposium on security and privacy (SP)*. IEEE, 2024, pp. 897–912.
- [43] Y. Yang, R. Gao, X. Wang, T.-Y. Ho, N. Xu, and Q. Xu, "Mma-diffusion: Multimodal attack on diffusion models," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 7737–7746.
- [44] Y. Huang, L. Liang, T. Li, X. Jia, R. Wang, W. Miao, G. Pu, and Y. Liu, "Perception-guided jailbreak against text-to-image models," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 39, no. 25, 2025, pp. 26 238–26 247.
- [45] H. Zhuang, Y. Zhang, and S. Liu, "A pilot study of query-free adversarial attack against stable diffusion," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 2385–2392.
- [46] M. Feffer, A. Sinha, W. H. Deng, Z. C. Lipton, and H. Heidari, "Red-teaming for generative ai: Silver bullet or security theater?" in *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, vol. 7, 2024, pp. 421–437.
- [47] A. Verma, S. Krishna, S. Gehrmann, M. Seshadri, A. Pradhan, T. Ault, L. Barrett, D. Rabinowitz, J. Doucette, and N. Phan, "Operationalizing a threat model for red-teaming large language models (llms)," *arXiv preprint arXiv:2407.14937*, 2024.
- [48] Z.-W. Hong, I. Shenfeld, T.-H. Wang, Y.-S. Chuang, A. Pareja, J. Glass, A. Srivastava, and P. Agrawal, "Curiosity-driven red-teaming for large language models," *arXiv preprint arXiv:2402.19464*, 2024.
- [49] E. Perez, S. Huang, F. Song, T. Cai, R. Ring, J. Aslanides, A. Glaese, N. McAleese, and G. Irving, "Red teaming language models with language models," *arXiv preprint arXiv:2202.03286*, 2022.
- [50] OpenAI, "Red teaming network," 2023, accessed: 2025-04-20. [Online]. Available: <https://openai.com/index/red-teaming-network/>
- [51] Y.-L. Tsai, C.-Y. Hsu, C. Xie, C.-H. Lin, J. Y. Chen, B. Li, P.-Y. Chen, C.-M. Yu, and C.-Y. Huang, "Ring-a-bell! how reliable are concept removal methods for diffusion models?" in *The Twelfth International Conference on Learning Representations*, 2024.

[52] J. Hartmanis, “Computers and intractability: a guide to the theory of np-completeness (michael r. garey and david s. johnson),” *SIAM Review*, vol. 24, no. 1, p. 90, 1982.

[53] M. T. Ribeiro, S. Singh, and C. Guestrin, “Semantically equivalent adversarial rules for debugging nlp models,” in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (volume 1: long papers)*, 2018, pp. 856–865.

[54] K. Pei, Y. Cao, J. Yang, and S. Jana, “Deepxplore: Automated whitebox testing of deep learning systems,” in *proceedings of the 26th Symposium on Operating Systems Principles*, 2017, pp. 1–18.

[55] Y. Bengio, N. Léonard, and A. Courville, “Estimating or propagating gradients through stochastic neurons for conditional computation,” *arXiv preprint arXiv:1308.3432*, 2013.

[56] Y. Wen, N. Jain, J. Kirchenbauer, M. Goldblum, J. Geiping, and T. Goldstein, “Hard prompts made easy: Gradient-based discrete optimization for prompt tuning and discovery,” *Advances in Neural Information Processing Systems*, vol. 36, pp. 51 008–51 025, 2023.

[57] X. Wang, Y. Yang, Y. Deng, and K. He, “Adversarial training with fast gradient projection method against synonym substitution based text attacks,” in *Proceedings of the AAAI conference on artificial intelligence*, vol. 35, no. 16, 2021, pp. 13 997–14 005.

[58] Z. Jiang, Y. Hu, Y. Yang, Y. Cao, and N. Z. Gong, “Jailbreaking safeguarded text-to-image models via large language models,” *arXiv preprint arXiv:2503.01839*, 2025.

[59] S. Kullback and R. A. Leibler, “On information and sufficiency,” *The annals of mathematical statistics*, vol. 22, no. 1, pp. 79–86, 1951.

[60] R. A. Fisher, “On the mathematical foundations of theoretical statistics,” *Philosophical transactions of the Royal Society of London. Series A, containing papers of a mathematical or physical character*, vol. 222, no. 594–604, pp. 309–368, 1922.

[61] Y. Du and I. Mordatch, “Implicit generation and modeling with energy based models,” *Advances in neural information processing systems*, vol. 32, 2019.

[62] L. Boltzmann, *Weitere studien über das wärmegleichgewicht unter gasmolekülen*. Aus der kk Hot-und Staatsdruckerei, 1872, vol. 66.

[63] J. Li, D. Li, S. Savarese, and S. Hoi, “Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models,” in *International conference on machine learning*. PMLR, 2023, pp. 19 730–19 742.

[64] D. Li, J. Li, and S. Hoi, “Blip-diffusion: Pre-trained subject representation for controllable text-to-image generation and editing,” *Advances in Neural Information Processing Systems*, vol. 36, pp. 30 146–30 166, 2023.

[65] P. Schramowski, C. Tauchmann, and K. Kersting, “Can machines help us answering question 16 in datasheets, and in turn reflecting on inappropriate content?” in *Proceedings of the 2022 ACM conference on fairness, accountability, and transparency*, 2022, pp. 1350–1361.

[66] Hugging Face, “Sentence transformers,” <https://huggingface.co/sentence-transformers>, 2025, accessed: 2025-03-21.

[67] S. Malladi, T. Gao, E. Nichani, A. Damian, J. D. Lee, D. Chen, and S. Arora, “Fine-tuning language models with just forward passes,” *Advances in Neural Information Processing Systems*, vol. 36, pp. 53 038–53 075, 2023.

[68] J. Xu and J. Zhang, “Random masking finds winning tickets for parameter efficient fine-tuning,” in *International Conference on Machine Learning*. PMLR, 2024, pp. 55 501–55 524.

[69] S. Jain, R. Kirk, E. S. Lubana, R. P. Dick, H. Tanaka, T. Rocktäschel, E. Grefenstette, and D. Krueger, “Mechanistically analyzing the effects of fine-tuning on procedurally defined tasks,” in *The Twelfth International Conference on Learning Representations*, 2024. [Online]. Available: <https://openreview.net/forum?id=A0HKcKI4NI>

[70] Z. Zhang, B. Liu, and J. Shao, “Fine-tuning happens in tiny subspaces: Exploring intrinsic task-specific subspaces of pre-trained language models,” in *The 61st Annual Meeting Of The Association For Computational Linguistics*, 2023.

[71] T. Griebel, J. Müller, M. Buchholz, and K. Dietmayer, “Kalman filter meets subjective logic: A self-assessing kalman filter using subjective logic,” in *2020 IEEE 23rd International Conference on Information Fusion (FUSION)*. IEEE, 2020, pp. 1–8.

[72] M. Jindal, “Autonlp gibberish detector,” <https://huggingface.co/madhurjindal/autonlp-Gibberish-Detector-492513457>, 2021, accessed: 2025-03-21.

[73] Y. Zhang, J. Jia, X. Chen, A. Chen, Y. Zhang, J. Liu, K. Ding, and S. Liu, “To generate or not? safety-driven unlearned diffusion models are still easy to generate unsafe images... for now,” in *European Conference on Computer Vision*. Springer, 2024, pp. 385–403.

[74] Y. Dong, X. Meng, N. Yu, Z. Li, and S. Guo, “Fuzz-testing meets llm-based agents: An automated and efficient framework for jailbreaking text-to-image generation models,” in *2025 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2025, pp. 373–391.

[75] Y. Qu, X. Shen, Y. Wu, M. Backes, S. Zannettou, and Y. Zhang, “Unsafebench: Benchmarking image safety classifiers on real-world and ai-generated images,” *arXiv preprint arXiv:2405.03486*, 2024.

[76] J. Chen, S. Xiao, P. Zhang, K. Luo, D. Lian, and Z. Liu, “Bge m3-embedding: Multi-lingual, multi-functionality, multi-granularity text embeddings through self-knowledge distillation,” *arXiv preprint arXiv:2402.03216*, 2024.

[77] G. Team, M. Riviere, S. Pathak, P. G. Sessa, C. Hardin, S. Bhupatiraju, L. Hussenot, T. Mesnard, B. Shahriari, A. Ramé *et al.*, “Gemma 2: Improving open language models at a practical size,” *arXiv preprint arXiv:2408.00118*, 2024.

[78] P. Esser, S. Kulal, A. Blattmann, R. Entezari, J. Müller, H. Saini, Y. Levi, D. Lorenz, A. Sauer, F. Boesel *et al.*, “Scaling rectified flow transformers for high-resolution image synthesis,” in *Forty-first international conference on machine learning*, 2024.

[79] V. Arkhipkin, A. Filatov, V. Vasilev, A. Maltseva, S. Azizov, I. Pavlov, J. Agafonova, A. Kuznetsov, and D. Dimitrov, “Kandinsky 3: Text-to-image synthesis for multifunctional generative framework,” in *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, 2024, pp. 475–485.

[80] ShuttleAI, “shuttleai/shuttle-3-diffusion,” <https://huggingface.co/shuttleai/shuttle-3-diffusion>, 2024, accessed: 2025-04-19.

[81] W. Peebles and S. Xie, “Scalable diffusion models with transformers,” in *Proceedings of the IEEE/CVF international conference on computer vision*, 2023, pp. 4195–4205.

[82] S. Ghadimi and G. Lan, “Stochastic first-and zeroth-order methods for nonconvex stochastic programming,” *SIAM journal on optimization*, vol. 23, no. 4, pp. 2341–2368, 2013.

[83] K. Balasubramanian and S. Ghadimi, “Zeroth-order nonconvex stochastic optimization: Handling constraints,” *High-Dimensionality and Saddle-Points*. arXiv, 2019.

[84] Z. Wang, D. Anshumaan, A. Hooda, Y. Chen, and S. Jha, “Functional homotopy: Smoothing discrete optimization via continuous parameters for LLM jailbreak attacks,” in *The Thirteenth International Conference on Learning Representations*, 2025.

[85] M. Andriushchenko, F. Croce, and N. Flammarion, “Jailbreaking leading safety-aligned LLMs with simple adaptive attacks,” in *The Thirteenth International Conference on Learning Representations*, 2025.

[86] H. J. Kushner and D. S. Clark, *Stochastic approximation methods for constrained and unconstrained systems*. Springer Science & Business Media, 2012.

[87] D. Wierstra, T. Schaul, T. Glasmachers, Y. Sun, J. Peters, and J. Schmidhuber, “Natural evolution strategies,” *The Journal of Machine Learning Research*, 2014.

[88] Y. Chen, “Theoretical study and comparison of spsa and rdsa algorithms,” *arXiv preprint arXiv:2107.12771*, 2021.

## Appendix A.

### Omitted Derivations and Proofs

#### A.1. Proof of Theorem 1

*Proof.*<sup>2</sup> We establish a strictly positive lower bound for the SNR improvement  $\mathcal{D}_t := \text{SNR}_t^{\text{GC}} - \text{SNR}_t^{\text{Vanilla}}$  by analyzing the statistical properties of the GC-SPSA estimator with decaying sample sizes against the vanilla SPSA baseline.

The single-sample SPSA estimator  $g(\theta)$  uses the observable loss  $\mathcal{L}(\theta) = \mathcal{L}_{\text{true}}(\theta) + \xi$ . A second-order Taylor expansion of  $\mathcal{L}_{\text{true}}$  implies that the finite difference approximation yields  $\Delta^\top g_{\text{true}} + O(\epsilon^2)$ , where  $g_{\text{true}} \triangleq \nabla_\theta \mathcal{L}_{\text{true}}(\theta)$ . Therefore,

$$g(\theta) = (\Delta^\top g_{\text{true}} + O(\epsilon^2)) \Delta + \frac{\xi^+ - \xi^-}{2\epsilon} \Delta. \quad (11)$$

The second moment arises from signal, bias, and noise. Thus, the variance of a single gradient estimate is:

$$V_{\text{single}} := \text{Var}(g(\theta)) = (d-1) \|g_{\text{true}}\|^2 + \frac{d\sigma_\xi^2}{2\epsilon^2} + O(d\epsilon^4) \quad (12)$$

The GC-SPSA estimator  $\hat{g}_t$  follows the recursive update rule Eq. (7), unrolling which gives  $\hat{g}_t = \sum_{k=0}^t h_k \bar{g}_k(\theta_k)$ . Let  $V_k := \frac{V_{\text{single}}}{n_k}$  denote the variance of the gradient estimate at step  $k$ . The expectation and variance of GC-SPSA are:

$$\mathbb{E}[\hat{g}_t] \approx P_t g_{\text{true}}(\theta_t), \quad \text{where} \quad P_t = \sum_{k=0}^t h_k \quad (13)$$

2. Complete proofs of both theorems are available in our [full version](#).

$$\text{Var}(\hat{g}_t) = \sum_{k=0}^t h_k^2 V_k = \sum_{k=0}^t h_k^2 \frac{V_{\text{single}}}{n_k}, \quad (14)$$

The SNR improvement is defined as  $\mathcal{D}_t = \text{SNR}_t^{\text{GC}} - \text{SNR}_{\text{Vanilla}}$ , where we have  $\text{SNR}_t^{\text{GC}} = \frac{\|g_{\text{true}}(\theta_t)\|^2 P_t^2}{\sum_{k=0}^t h_k^2 V_k}$  and  $\text{SNR}_{\text{Vanilla}} = \frac{\|g_{\text{true}}(\theta_t)\|^2}{V_{\text{single}}}$ . Therefore, we have:

$$\mathcal{D}_t = \frac{\|g_{\text{true}}\|^2}{V_{\text{single}} \sum_{k=0}^t h_k^2 V_k} \left[ P_t^2 V_{\text{single}} - \sum_{k=0}^t h_k^2 V_k \right] \quad (15)$$

To analyze the bracketed term  $P_t^2 V_{\text{single}} - \sum_{k=0}^t h_k^2 V_k$ , we expand  $P_t^2 = \sum_{k=0}^t h_k^2 + 2 \sum_{0 \leq i < j \leq t} h_i h_j$  and obtain:

$$\sum_{k=0}^t h_k^2 V_{\text{single}} \left( 1 - \frac{1}{n_k} \right) + 2 V_{\text{single}} \sum_{0 \leq i < j \leq t} h_i h_j. \quad (16)$$

The first term is non-negative, and the second term is strictly positive. Therefore, we have  $\mathcal{D}_t > 0$  for  $t \geq 1$ .  $\square$

We can summarize two insights from Eq. (16). First, gradient calibration alone can provide strong SNR gain. The term  $2 \sum_{i < j} h_i h_j$  is strictly positive even when  $n_k = 1$ , and exhibits a quadratic growth as  $(\sum h_k)^2 - \sum h_k^2$ . Second, larger sampling budgets  $n_k$  can further enhance SNR by reducing variance  $V_k$  and increasing  $1 - 1/n_k$ . This theoretical analysis further demonstrates the benefits of our GC-SPSA.

## A.2. Proof of Theorem 2

*Proof.* The smoothness condition  $\nabla^2 \mathcal{L}(\theta) \preceq \ell I_d$  implies the standard descent inequality for  $\theta_{t+1} = \theta_t - \eta \hat{g}_t$ . We adopt zero-indexing with  $\hat{g}_{-1} = 0$  and  $w_{-1} = 0$ , so that  $\alpha_0 = 0$ .

Expanding the conditional moments using the GC-SPSA properties  $\mathbb{E}[\hat{g}_t \mid \mathbb{F}_t] = g_t + \gamma \alpha_t \hat{g}_{t-1}$ , where  $\alpha_t := w_{t-1}/(w_{t-1} + n_t)$ , and  $\mathbb{E}[\|\hat{g}_t\|^2 \mid \mathbb{F}_t] = \text{Var}(\hat{g}_t \mid \mathbb{F}_t) + \|g_t\|^2 + 2\gamma \alpha_t \langle g_t, \hat{g}_{t-1} \rangle + (\gamma \alpha_t)^2 \|\hat{g}_{t-1}\|^2$ .

Then, by using the descent inequality and Young's inequality, we have:

$$\begin{aligned} \mathbb{E}[\mathcal{L}(\theta_{t+1}) \mid \mathbb{F}_t] &\leq \mathcal{L}(\theta_t) - \eta \zeta_t \|g_t\|^2 + C_t \|\hat{g}_{t-1}\|^2 \\ &\quad + \frac{\ell \eta^2}{2n_t} V_t, \end{aligned} \quad (17)$$

where the coefficients are defined as

$$\zeta_t := 1 - \frac{\ell \eta}{2} - \frac{\gamma \alpha_t (1 - \ell \eta)}{2} - \frac{\ell \eta (d-1)}{2n_t}, \quad (18)$$

$$C_t := \frac{\eta \gamma \alpha_t (1 - \ell \eta)}{2} + \frac{\ell \eta^2 (\gamma \alpha_t)^2}{2}, \quad (19)$$

$$V_t := O(d \epsilon_t^4) + \frac{d \sigma_\xi^2}{2 \epsilon_t^2}. \quad (20)$$

Telescoping the one-step descent lemma in Eq. (17) over  $t = 0, \dots, T-1$ . Let  $\zeta_{\min} := \min_{0 \leq t \leq T-1} \zeta_t > 0$ . Taking expectations, summing, we rearrange to find:

$$\begin{aligned} \min_{0 \leq t \leq T-1} \mathbb{E} \|g(\theta_t)\|^2 &\leq \frac{\mathcal{L}(\theta_0) - \mathcal{L}^* + \sum_{t=0}^{T-1} C_t \mathbb{E} \|\hat{g}_{t-1}\|^2}{\eta T \zeta_{\min}} \\ &\quad + \frac{\ell \eta}{2 T \zeta_{\min}} \sum_{t=0}^{T-1} \frac{V_t}{n_t}. \end{aligned} \quad (21)$$

Prior analyses of stochastic optimization algorithms have established the boundedness of error terms [30, 82, 83], i.e.,

$$C_t \leq C_{\max}, \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E} \|\hat{g}_{t-1}\|^2 \leq \Upsilon, \text{ and } \frac{1}{T} \sum_{t=0}^{T-1} \frac{V_t}{n_t} \leq \Xi.$$

Applying these bounds yields:

$$\min_{0 \leq t \leq T-1} \mathbb{E} \|g(\theta_t)\|^2 \leq \frac{\mathcal{L}(\theta_0) - \mathcal{L}^*}{\eta T \zeta_{\min}} + \frac{C_{\max} \Upsilon}{\eta \zeta_{\min}} + \frac{\ell \eta \Xi}{2 \zeta_{\min}}.$$

To ensure the left-hand side is at most  $\delta$ ,  $T$  needs to satisfy:

$$T \geq \frac{\mathcal{L}(\theta_0) - \mathcal{L}^*}{\eta \zeta_{\min} \delta - C_{\max} \Upsilon - \frac{\ell \eta^2 \Xi}{2}}, \quad (22)$$

which establishes the iteration complexity of GC-SPSA.  $\square$

## Appendix B. More Discussion

**Discussion on MHSC's Accuracy.** Our experiments rely primarily on MHSC to label outputs. To verify its reliability, we evaluate MHSC on both a real-world dataset (the test set of the NCD dataset) and our DREAM-generated dataset (SD v1.5+Sexual, 256 balanced, randomly selected samples), and the results are shown in Tab. 13. We can derive two insights from the results. First, MHSC achieves over 90% accuracy and F1 scores on both domains, indicating that its predictions are highly reliable. Second, MHSC indeed exhibits a conservative classification tendency, where it only flags an image as unsafe when it is highly confident. This further confirms our analyses in Sec. 5.4.

**Discussion on Baseline Selection.** Recall that red-teaming is typically initiated by the model owner/developer, who naturally has full control over their own T2I system. However, not all baselines in our paper fully exploit this privilege. For better clarity, we summarize the capability assumptions made by each baseline in Tab. 12. Several insights emerge from this summarization. First, most evaluated baselines require clear-box access, i.e., the weights, gradients, or intermediate outputs of at least one system component, indicating that they still leverage the owner's privilege.

Second, and perhaps counter-intuitively, the amount of information required does not always correlate with better final performance. For example, on CA-Violence, Keyword-Gibberish, and NSFW-Text-Detector, closed-box methods like SneakyPrompt, FLIRT, and JailFuzzer sometimes perform the best among all baselines. Similar findings have also been reported in LLM adversarial learning [84, 85],

TABLE 10: Categories and definitions of unsafe content used in our paper.

Category	Definition
Sexual	Content that is sexually explicit, including nudity, sexual acts, genital exposure, or content that, though not explicitly depicting nudity or sexual acts, are overly sexualized, with clear sexual provocations, sexual innuendo, or erotic tease.
Violence	Content involving physical aggression, brutality, threats, or harm directed at individuals or groups, including depictions of interpersonal violence, intended to shock, disturb, promote violent behavior, or when featuring graphic imagery of excessive bloodshed or serious injuries.

TABLE 11: Prompt diversity levels and their definitions used in our user study.

Diversity Level	Definition
1: Limited Diversity	The majority of prompts are near-identical or repeated with trivial modifications, such as basic rewordings. There is negligible lexical, structural, or conceptual diversity. Most prompts are variations on a fixed template and rely on the same narrow set of triggering keywords or phrases.
2~3: Moderate Diversity	Prompts exhibit moderate diversity, often using modestly different triggering synonyms or introducing light syntactic changes. However, they still rely on a small group of core visual ideas and maintain similar structure and phrasing, with only minor surface-level differences.
4~5: High Diversity	Prompts move beyond a small set of repetitive trigger words or formulaic expressions, demonstrating meaningful exploration of lexical, syntactic, and semantic alternatives. Instead of repeatedly relying on single terms, the prompts vary across subjects, the frame, and the scene structure. The prompts reflect creative and distributed discovery of diverse, or even unexpected potential triggers for generating unsafe content.

TABLE 12: Component-level requirements of different methods on the target T2I diffusion model. For the first two columns, ● denotes the method requires explicit back-propagation-based gradients from the target model’s corresponding component. ● means while the method requires gradients from that component, they are obtained via numerical approximations rather than exact back-propagation (e.g., via GC-SPSA). ○ indicates the method does not utilize signals from this component. For the “Shadow Model” column, YES / NO indicates whether the method assumes access to a local shadow model (e.g., a shadow text encoder) that is identical or highly similar to that of the target model.

Method	Text Encoder	Denoising Network	Shadow Model
QF-Attack	●	○	NO
MMA-Diffusion	●	○	NO
P4D / UnlearnDiffAtk	●	●	NO
SneakyPrompt / JailFuzzer	○	○	YES
ART / FLIRT	○	○	NO
DREAM (Ours)	●	●	NO

TABLE 13: Performance of MHSC.

Setting	ACC ↑	TPR ↑	FPR ↓	TNR ↑	FNR ↓	F1 ↑
NCD Test Set	93.52%	89.43%	2.40%	97.60%	10.57%	93.24%
DREAM (Ours)	93.17%	92.97%	6.67%	93.33%	7.03%	92.61%

where gradient-based clear-box strategies are observed to be less effective than closed-box gradient-free approaches like random search. One plausible explanation is that in large-scale discrete optimization over language prompts, clear-box gradients tend to be highly local and noisy. As a result, they may sometimes underperform gradient-free methods, which allow exploring larger solution spaces more efficiently. These observations suggest that the key determinant of red-teaming performance may not fully lie in the level of access, but in how the problem is modeled and how the optimization is conducted. Thus, we include both clear-box and closed-box baselines in our experiments, as it would highlight the strongest competitor regardless of assumed capability and make our evaluation more complete.

**Discussion on Red Teaming Paradigm.** Besides the efficiency analysis in Sec. 5.6, it is also worth discussing the red teaming paradigms underlying different methods. Existing methods generally fall into two categories. Some approaches (e.g., human-written datasets) first generate a pool of prompts on a surrogate T2I system and then reuse them across multiple targets. Others, including most recent methods (e.g., P4D, FLIRT, JailFuzzer) and DREAM, perform system-specific training that adapts the red-teaming process to each target system individually.

These two paradigms represent a natural trade-off between cost and adaptivity. While reusing pre-discovered prompts is cheap and convenient, such methods may lack adaptivity to model-specific vulnerabilities. As implied by Definition 1, the set of effective red-teaming prompts is inherently system-specific: a prompt that succeeds on one system may fail on another, and vice versa. Our empirical observations also support this view: for instance, on the Keyword-Gibberish Filter, almost all fixed-dataset approaches achieve very low success rates, whereas adaptive methods such as FLIRT, JailFuzzer, and DREAM still obtain high success rates, indicating that the system is not genuinely safe. This suggests that relying solely on fixed datasets for evaluation may lead to a false sense of safety, whereas model-specific adaptive training can uncover target-specific blind spots that are crucial for reliable safety evaluation and targeted improvement.

At the same time, red teaming with DREAM does not necessarily indicate prohibitive cost. As shown in Tab. 6, DREAM’s red team LLM can be reused or lightly adapted to new targets, significantly reducing cost while maintaining competitive effectiveness. This flexibility provides a practical balance between efficiency and adaptivity.

Thus, we suggest that real-world users choose their strategy according to their goals and resource budgets. In practice, the two paradigms could be complementary

rather than mutually exclusive: users can first apply low-cost fixed-dataset methods for rapid screening or benchmarking, and then selectively deploy adaptive approaches such as DREAM if they need further high-coverage, model-specific red-teaming or targeted safety improvement. This staged approach leverages the efficiency of fixed datasets to identify general issues while using adaptive methods to uncover deeper, system-specific vulnerabilities. In addition, reusing or lightly adapting a previously trained red-team LLM offers a lightweight yet valuable middle ground.

**Comparison with PromptTune [58].** We further compare with a recent baseline, i.e., PromptTune [58]. Specifically, we train PromptTune using the NSFW-66k dataset provided by the authors, following the PromptTune-AdvPrompter setting in their paper, and evaluate it under CA and SC. We obtain a PSR of 14.7% on CA and 38.3% on SC. Through this case study, we observe that although PromptTune also fine-tunes an LLM, its effectiveness is weaker than DREAM. A possible reason lies in PromptTune’s training mechanism: it learns by generating candidate prompt variants, selecting the most effective one, and then fine-tuning on it. This indirect supervision may be less efficient than DREAM’s paradigm, which directly learns from the energy function via GC-SPSA. We leave further exploration to future work.

**Discussion on Other Gradient Estimation Strategies and RL-based Approaches.** In addition to our GC-SPSA, other optimization approaches can, in principle, be applied to minimize the final loss function in Eq. (5). One representative line is reinforcement learning (RL) [49, 48], which has recently been introduced to red team LLMs. For example, CRT [48] fine-tunes an LLM using PPO combined with curiosity-driven rewards and entropy bonuses to promote prompt diversity. Although originally designed for LLMs, the authors have discussed its potential to be extended to T2I models. To make a more direct comparison, we adapted CRT to our category-specific setting by modifying system prompts and evaluated it on CA. As shown in Tab. 14, CRT exhibits high instability, and we even observed complete failure (i.e.,  $\text{PSR} < 5\%$ ) for some cases. We hypothesize that this stems from a fundamental difference between RL and SPSA. In RL, each prompt (i.e., trajectory) is directly rewarded. For safety-aligned models where most prompts fail, a few lucky successful prompts receive disproportionately large rewards, which tends to drive the policy toward repeatedly generating these high-reward prompts even with regularization (e.g., KL or novelty terms), ultimately hurting diversity. By contrast, GC-SPSA perturbs model parameters and estimates gradients via batch-based Monte Carlo sampling, without tying rewards to individual prompts. This allows the optimizer to identify general update directions in parameter space rather than reinforcing a few trajectories, thereby maintaining stability and diversity during training.

Besides RL, other parameter-based gradient estimators like RDSA [86] and NES [87] could also be considered. While these methods alleviate the trajectory overfitting issue to some extent, our experiments (Tab. 14) show that RDSA suffers from low effectiveness, while NES incurs much

TABLE 14: Results of other parameter-based gradient estimators and the RL-based approach CRT [48]. The setting is CA+Sexual. We report the results averaged over 3 independent runs.

	Parameter-based		RL-based
	RDSA [86]	NES [87]	CRT [48]
PSR↑/PS↓	21% $\pm$ 1.6%/0.48 $\pm$ 0.01	52% $\pm$ 0.4%/0.50 $\pm$ 0.00	47% $\pm$ 1.4%/0.68 $\pm$ 0.04
Time↓	12.88 h	51.76 h	24.35 h

higher computational cost. This may be attributed to RDSA’s susceptibility to large estimation noise when hyperparameters are not carefully tuned [88], and to NES’s reliance on large population sizes. In contrast, SPSA estimates gradients with Gaussian perturbations and only two function evaluations per iteration, which offers a more balanced tradeoff between simplicity, stability, and efficiency. We thus focus on SPSA while leaving broader exploration into applying RDSA and NES for red teaming for future work.

As a final remark, we emphasize that this analysis is not meant to diminish the potential of RL or other gradient estimation techniques. We believe they remain powerful and expressive, and with advanced variance-reduction strategies or carefully tuned hyperparameters, their potential could be further unlocked. Our core claim is not that SPSA is categorically superior, but rather that even a relatively simple and lightweight method like our GC-SPSA already achieves stable and effective results in the challenging red teaming setting. We hope DREAM provides a practical and extensible starting point, and we leave deeper investigation of these alternative optimization strategies to future work.

**Ethics Considerations.** This research is intended solely for advancing the safety of T2I generative systems. DREAM is designed as a developer-driven red teaming framework to proactively evaluate and improve safety mechanisms, not to attack or undermine any deployed systems. However, we acknowledge that it may generate prompts capable of bypassing safety mechanisms and eliciting harmful outputs. To minimize potential harm, we have responsibly disclosed the discovered vulnerabilities to developers of the affected T2I systems and provided a sixty-day window to address the issues prior to publication. We also refrain from publicly releasing any unsafe prompts or images that may be harmful to real-world services or users. Instead, representative attack prompts, generated samples, and the derived dataset will be released via gated access, requiring an access request, statement of intended use, and institutional affiliation. This study was reviewed and approved by the institutional research ethics committee of our university, which fulfills a role analogous to the IRB in the United States. All participants were healthy adults who have provided informed consent, were informed of their right to withdraw at any time, and no personal data were collected. We reduced participants’ exposure to disturbing material by limiting their workload during the user study and, for subsequent experiments, restricting human annotation to a small team of authors. All researchers and participants were given access to institutional mental-health resources and received content warnings. We will continue to engage with stakeholders following the principles outlined in the Menlo Report and the broader computer security research community.

## Appendix C. Meta-Review

The following meta-review was prepared by the program committee for the 2026 IEEE Symposium on Security and Privacy (S&P) as part of the review process as detailed in the call for papers.

### C.1. Summary

This paper focuses on the safety of text-to-image (T2I) generative models, particularly their ability to produce harmful content. The authors model the probabilistic distribution of the target system's problematic prompts so that the designed framework automatically uncovers diverse problematic prompts from the target.

### C.2. Scientific Contributions

- Creates a New Tool to Enable Future Science
- Addresses a Long-Known Issue
- Provides a Valuable Step Forward in an Established Field

### C.3. Reasons for Acceptance

- 1) This paper proposes a scalable red teaming framework that improves prompt discovery through probabilistic modeling and stable optimization, advancing safety evaluation for text-to-image generative systems.
- 2) The proposed method aims to overcome the inefficiency of prior approaches (e.g., token-level optimization, iterative rewriting), potentially enabling large-scale exploration of problematic prompts.