CLOSING THE DISTRIBUTION GAP IN ADVERSARIAL TRAINING FOR LLMS

Anonymous authors
Paper under double-blind review

ABSTRACT

Adversarial training has been one of the most promising methods to reliably improve robustness against adversaries and has been recently extended to LLMs. However, despite significant progress, models remain vulnerable to trivial exploits, such as rephrasing the same request or repeated sampling. We argue that this persistent fragility stems from a fundamental limitation in current adversarial training algorithms: they minimize adversarial loss on their training set but inadequately cover the data distribution, resulting in vulnerability to seemingly simple attacks. To bridge this gap, we propose Distributional Adversarial Training, **DAT**. We leverage Diffusion LLMs to approximate the true joint distribution of prompts and responses, enabling generation of diverse, high-likelihood samples that address generalization failures. By combining optimization over the data distribution provided by the diffusion model with continuous adversarial training, DAT achieves substantially higher adversarial robustness than previous methods.

1 INTRODUCTION

Ensuring the safety of Large Language Models (LLMs) is a prerequisite for their reliable deployment. While alignment techniques such as RLHF (Ouyang et al., 2022) successfully reduce the likelihood of harmful outputs in normal use, models remain highly vulnerable to adversarial prompts (Zou et al., 2023; Zhu et al., 2023; Li et al., 2024). Adversarial Training (AT) has emerged as one of the most promising defenses, aiming to improve robustness by augmenting training data with worst-case adversarial inputs (Goodfellow et al., 2015; Madry et al., 2018). Recently, AT has been successfully scaled to LLMs (Xhonneux et al., 2024; Casper et al., 2024). Yet, despite these advances, models remain surprisingly brittle, exhibiting simple *generalization failures*. A model might refuse a complex, optimized attack string, yet its safety training can be bypassed simply by trivially augmented variants of the same request (Hughes et al., 2024) or even repeated sampling with the same attack (Beyer et al., 2025b). These are not adversarial examples in the traditional sense of optimized *model-specific* perturbations. Rather, they are valid, natural inputs that lie just outside the distribution observed during training.

We argue that these failures stem from the decomposition of the robust risk (Madry et al., 2018). Minimizing the robust risk over the true underlying data distribution $q(x, y)$ inherently involves minimizing two distinct sources of error: the *data distribution approximation error* and the *adversarial optimization error*. The former relates to how well the training data covers $q(x, y)$, while the latter relates to finding the worst-case perturbation within a local region. Standard AT algorithms primarily address the optimization error by training against *model-specific* perturbations in a local ball. However, limited data coverage can leave *data-specific* vulnerabilities that naturally occur in $q(x, y)$ but lie outside the fixed training set (Figure 1), such as augmented variations of known prompts. To address the approximation error, we propose optimizing over a generative surrogate, where we can conditionally generate prompts x for harmful responses y. Diffusion LLMs (Zhu et al., 2025a; Lüdke et al., 2025) are well suited for this purpose. Diffusion models learn a distribution over prompts and responses and support inpainting. This enables a tractable approach to the inverse problem: sampling diverse prompts x conditioned on a harmful response y, thereby discovering data-specific vulnerabilities missing in the original data.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

In this work, we propose the Distributional Adversarial Training (**DAT**) framework that simultaneously tackles both sources of error. We leverage pretrained Diffusion LLMs to sample *data-specific* adversarial examples that explore a broader range of prompts and uncover vulnerabilities missed by fixed datasets. We then apply continuous adversarial training to these samples to minimize worst-case error using an approximation of the local adversarial loss. This approach bridges the gap between data-specific and model-specific adversarial vulnerabilities. Our contributions are:

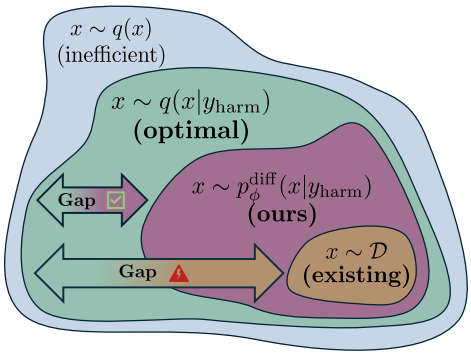

Figure 1: Standard AT minimizes empirical risk over a fixed dataset $\mathcal{D}$ (brown), which can poorly approximate the population robust risk. This leaves a distribution gap where the model misses prompts associated with harmful responses $q(x \mid y_{\text{harm}})$ (green) within the manifold of natural prompts $q(x)$ (blue). DAT targets this gap using a surrogate distribution $p_\phi^{\text{diff}}(x \mid y_{\text{harm}})$ (purple) to broaden training coverage.

Decomposition of the Robustness Gap: We frame the failure of current defenses as a gap between minimizing empirical and population-robust risk, distinguishing between model-specific perturbations and data-specific generalization failures.

Distributional Adversarial Training: We introduce DAT, which addresses both sources of error. Conditional sampling from Diffusion LLMs broadens coverage of the data distribution, and continuous adversarial optimization targets worst-case perturbations for each sample. Furthermore, we provide a theoretical motivation for our method, showing that conditional surrogate fidelity bounds the gap between surrogate and population robust risk for a bounded loss.

Single Data Source, Broad Robustness: Our experiments indicate that DAT successfully narrows this gap and robustness improves as the training data covers more of the distribution. Unlike prior methods that combine several attack types for data augmentation (Dékány et al., 2025) yet remain vulnerable, DAT uses a single diffusion-generated data source and substantially improves worst-case robustness against diverse attacks not seen during training.

2 Related Work

Adversarial Training in LLMs. Adversarial training has been one of the most effective methods to improve robustness in deep learning (Szegedy et al., 2014; Goodfellow et al., 2015; Madry et al., 2018; Shafahi et al., 2019; Wong et al., 2019; Kireev et al., 2022). However, its application to LLMs faces significant computational challenges due to the discrete nature of text. Early approaches often relied on computationally expensive discrete optimization methods to generate adversarial examples during training (Mazeika et al., 2024). To address this, more efficient methods operating in the continuous embedding space or latent representations have been proposed Casper et al. (2024); Xhonneux et al. (2024). Other works focus on the model's internal representations: Zou et al. (2024b) proposed short-circuiting to remap harmful representations to refusal states and Yu et al. (2025) proposed Refusal Feature Adversarial Training (ReFAT), which ablates specific refusal features in the residual stream. Additionally, Liu et al. (2024) proposed a two-stage adversarial tuning framework involving token-level and out-of-distribution adversarial prompt generation. See also Section C.8.

Data Augmentation for Safety Training. Related to our motivation of better approximating the population risk, early works in image classification use generative models to generate synthetic data (Wang et al., 2023) or established data augmentation techniques (Li & Spratling, 2023). In the LLM domain, Rainbow Teaming constructs a diverse but static dataset of adversarial prompts for fine-tuning (Samvelyan et al., 2024).[1] MART instead iteratively generates adversarial prompts against the current model checkpoint (Ge et al., 2024). Yet, neither Rainbow Teaming nor MART decomposes the robustification problem. They either do not optimize for strong worst-case examples in the inner loop or do not address broad coverage of the outer data distribution. MixAT is most

[1]Rainbow Teaming was used in the safety post-training of Llama 3 (Grattafiori et al., 2024), which nevertheless remains vulnerable to a broad range of attacks in our evaluation; see Table 1.

closely related to our work (Dékány et al., 2025). It augments the training data with PAP prompts generated by an autoregressive model and applies continuous adversarial training in the inner loop. MixAT+GCG additionally uses online GCG attacks, at substantially higher training cost. Both variants remain vulnerable to several attacks in our evaluation. We attribute this to limited coverage of the data distribution. Adding specific attack types improves robustness against those attacks but does not cover the broader range of prompts that elicit harmful responses. Supporting this argument, we find that replacing our diffusion surrogate with an autoregressive attacker makes the model robust mainly against that attacker but leaves it vulnerable to others (see Appendix C.2). In the following, we propose DAT to avoid this problem.

3 Background

Adversarial Training (AT) is formulated as a min-max optimization problem. Let x be some general input and y the output of a probabilistic neural network $p_\theta(y \mid x)$ parametrized by θ. Further, let $\ell_{rob}(x, y; \theta) = \max_{\delta \in \Delta} \mathcal{L}(p_\theta(y \mid x + \delta))$ denote the robust loss, which represents the worst-case loss within a local perturbation set Δ (e.g., an ϵ-ball in the embedding space). For a data distribution q on input-output pairs, the goal is to minimize the *population robust risk* (Madry et al., 2018)

$$\mathcal{R}_{pop}(\theta) = \mathbb{E}_{(x,y)\sim q}[\ell_{rob}(x, y; \theta)].$$

In practice, we draw a finite dataset $\mathcal{D} \sim q^n$ and minimize the *empirical robust risk*:

$$\mathcal{R}_{emp}(\theta) = \mathbb{E}_{(x,y)\sim \mathcal{D}}[\ell_{rob}(x, y; \theta)] \approx \mathcal{R}_{pop}(\theta).$$

Here, adversarial training algorithms solve an inner maximization to approximate the maximum of the loss within Δ, followed by an outer minimization to update θ. While previous methods in the image domain implicitly addressed this through data augmentation (Wang et al., 2023; Li & Spratling, 2023), LLM safety research has mostly focused on the inner loop and the gap between empirical and population robust risk remains under-addressed in current methods.

3.1 Large Language Models

Let $\mathcal{V}$ be a vocabulary and let $\mathcal{Z} = \mathcal{V}^{\leq m}$ denote the set of token sequences of length at most m. We denote the true distribution of language data as q over $z \in \mathcal{Z}$. Whenever convenient, we identify a sequence z with a prompt–response pair (x, y) via $z = x \oplus y$ where $x, y \in \mathcal{Z}$, and (by a slight abuse of notation) write $q(z)$ and $q(x, y)$ interchangeably for the corresponding probability mass function. Autoregressive large language models (LLMs) aim to approximate the data-generating conditional $q(y \mid x)$ by modeling

$$p_\theta^{\mathrm{AR}}(y \mid x) \;=\; \prod_t p_\theta^{\mathrm{AR}}\big(y_t \mid x, y_{<t}\big),$$

for tokens $y_t \in \mathcal{V}$. While such models are highly effective at conditional modeling, this conditional formulation does not explicitly parameterize the input marginal $q(x)$.

3.2 Adversarial Training for LLMs

For classification problems, adversarial training is typically phrased as enforcing *label invariance* under small perturbations of the input: for a fixed ground-truth label y, the learner should maintain its prediction when x is replaced by $x + \delta$. In safety-oriented robustness for large language models, another definition is used (Xhonneux et al., 2024; Sheshadri et al., 2024). In the presence of jailbreak-style attacks, the central requirement is *output safety*: the model should avoid emitting harmful content, regardless of which prompt happens to elicit it. Accordingly, it is natural to view prompts x primarily as *triggers* for undesirable responses, rather than as inputs whose semantics must be preserved under perturbations.

We model harmfulness through an indicator function $h : \mathcal{Z} \to \{0, 1\}$, where $h(y) = 1$ denotes that a sequence y contains harmful content.

To make this perspective explicit, we define the adversarial-training distribution as the natural language distribution restricted to harmful outputs assuming $q(h(y) = 1) > 0$

$$\tilde{q}(x, y) \;:=\; q\big(x, y \mid h(y) = 1\big) \;=\; \frac{q(x, y)\,\mathbf{1}\{h(y) = 1\}}{q\big(h(y) = 1\big)}.$$

By construction, this restriction preserves the relative frequencies of harmful responses and their associated prompts as they occur under q (in particular, $\tilde{q}(y) = q(y \mid h(y) = 1)$).

Moreover, for any harmful response y with $\tilde{q}(y) > 0$, the induced prompt conditional is unchanged,

$$\tilde{q}(x \mid y) \;=\; \frac{\tilde{q}(x, y)}{\tilde{q}(y)} \;=\; \frac{q(x, y)}{q(y)} \;=\; q(x \mid y). \tag{1}$$

Consequently, sampling $(x, y) \sim \tilde{q}$ admits a two-stage interpretation: first draw a harmful response $y \sim q(\cdot \mid h(y) = 1)$, and then draw a prompt $x \sim q(\cdot \mid y)$.

In practice, the first stage is approximated using a dataset of harmful responses, while the second stage is approximated by a conditional generator for $x \mid y$. Note that below, $\mathcal{R}_{pop}$ denotes risk under $\tilde{q}$.

Crucially, this choice prioritizes prompts that are *naturally compatible* with a harmful response under q. With limited adversarial-training budget, focusing on such prompt–response pairs is desirable, whereas spending capacity on prompts that are exceedingly unlikely to elicit harmful outputs may be less informative for improving real-world jailbreak robustness.

3.3 Model- and Data-Specific Prompts

We call a prompt x *data-specific* if it has high likelihood under the harmful prompt marginal $\tilde{q}(x)$, i.e., if it is likely to elicit a harmful response under the underlying data distribution. Since different models are generally trained on large fractions of the same data, such prompts tend to transfer across models. We hypothesize that these marginals approximately agree across models:

$$\tilde{q}(x) \approx \tilde{p}_{\theta_1}(x) \approx \cdots \approx \tilde{p}_{\theta_N}(x), \tag{2}$$

with the normalized marginal $\tilde{p}_\theta(x) \coloneqq \frac{\sum_{y:\, h(y)=1} p_\theta(x,y)}{p_\theta(h(Y)=1)}$. Here $p_\theta(x, y) := q_X(x) p_\theta^{\mathrm{AR}}(y \mid x)$ uses the common prompt marginal $q_X(x) := \sum_y q(x, y)$, and the denominator is assumed positive.

We contrast this to model-specific prompts, which we define as prompts that have a low probability of triggering harmful responses under the underlying data distribution but a high probability under an individual model, $q(h(Y) = 1 \mid x) \ll p_{\theta_1}(h(Y) = 1 \mid x)$. We expect such prompts to transfer less when the exploited weakness is not shared across models. We conduct an analysis related to this concept in the later part of the manuscript (see Figure 2).

4 Method

We introduce Distributional Adversarial Training (DAT), a framework that leverages generative surrogate models to address a generalization gap in previous adversarial training methods for LLMs by better approximating the population risk.

4.1 Adversarial Training Suffers from a Generalization Gap

We argue that current AT approaches for LLMs fail to generalize because the empirical risk $\mathcal{R}_{emp}$ insufficiently approximates the population risk $\mathcal{R}_{pop}$ for two main reasons. First, the diversity of open-ended natural language is difficult to cover with small, finite datasets $\mathcal{D}$. Secondly, robust learning requires high sample complexity (Schmidt et al., 2018). Consequently, models remain vulnerable to simple exploits like trivially augmented variants of the same request (Hughes et al., 2024) or repeated sampling with the same attack (Beyer et al., 2025b).

4.2 Generative Surrogates

To bridge the generalization gap, we propose to expand the static dataset $\mathcal{D}$ using a conditional generative surrogate $p_\phi(x \mid y)$, aiming to better approximate the population risk over harmful pairs $\tilde{q}(x, y)$. A suitable surrogate must therefore satisfy three key desiderata.

Conditional Sampling. First, the surrogate must be able to effectively invert the generation process, to sample data x conditioned on a harmful response y, as in Equation 1.

Data Specific. Second, sampled prompts should be *data specific* as described in Equation 2.

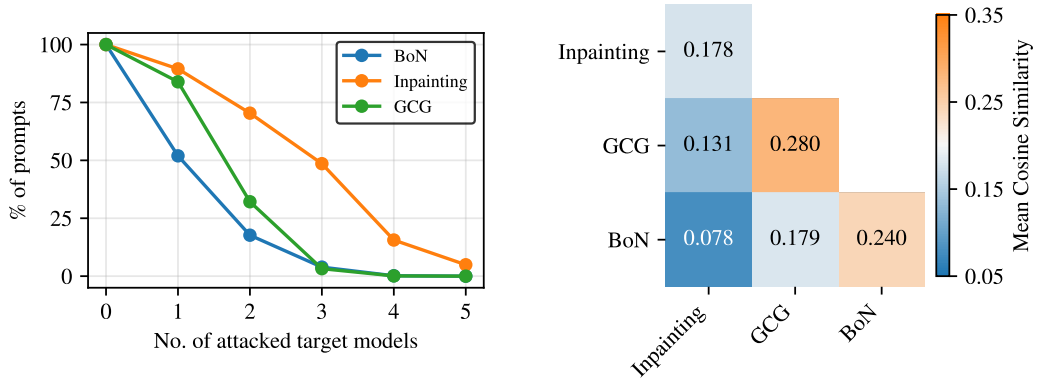

Figure 2: **Diversity and Transferability. Left:** Cumulative transfer ASR from Llama3-8B to five target models: Gemma3, Qwen2.5, Zephyr, Llama3-LAT, Llama3-CB (see Section B.1 for more details). Diffusion-based Inpainting exhibits significantly higher transferability than GCG or BoN. **Right:** Diversity of generated attack strings measured using SBERT embeddings (all-MiniLM-L6-v2; Reimers & Gurevych, 2019). Inpainting attacks exhibit the lowest intra-method similarity (0.178), indicating substantially greater sample diversity. These results support using diffusion prompts to expose vulnerabilities shared across architectures.

Diversity. Third, it should cover diverse prompts to limit overfitting to a narrow set of attack patterns.

4.3 Diffusion LLMs as Surrogate

We employ Diffusion LLMs as our generative surrogate to meet these technical desiderata. Diffusion models have recently emerged as a powerful alternative for text generation, learning a distribution over prompt–response pairs (Nie et al., 2025). Crucially, they support conditioning on a fixed response y. Recently, Lüdke et al. (2025) demonstrated that fixing a target response y and performing inpainting-like conditioning, diffusion LLMs can effectively sample adversarial prompts $x \sim p_\phi^{\text{diff}}(x|y)$, fulfilling our first requirement.

Moreover, as shown in Figure 2, prompts sampled via diffusion exhibit significantly higher transferability than GCG or BoN, supporting their ability to capture data-specific vulnerabilities shared across models. Additionally, diffusion models address our diversity criterion. While training prompts could alternatively be generated with discrete adversarial attacks (e.g., GCG), Figure 2 shows that diffusion-based sampling yields substantially more diverse samples, supporting our choice of surrogate. Lastly, we evaluated an autoregressive version of DAT using the PAIR attack. However, it achieved considerably worse robustness (see Appendix C.2).

4.4 Tractability via Monte Carlo Sampling

Our idealized surrogate risk averages over harmful responses $y \sim \tilde{q}(y)$ and prompts $x \sim p_\phi^{\text{diff}}(x \mid y)$. We approximate this expectation using a finite response dataset and Monte Carlo prompt sampling. In practice, we leverage an initial dataset of harmful pairs (x, y). We fix its affirmative harmful responses y, creating $\mathcal{D}_{\text{harm}}$, and use the diffusion model to *conditionally generate* diverse, new data samples $\tilde{x} \sim p_\phi^{\text{diff}}(x \mid y)$ following Lüdke et al. (2025). This expands the training set with prompt variations for each harmful target. We additionally filter the generated prompts by the StrongREJECT score of completions $\hat{y}$ from the frozen, pre-AT target model, retaining the highest-scoring candidates.

4.5 Distributional Adversarial Training

Our proposed DAT unifies these concepts. Before filtering, the objective is:

$$\mathcal{L}_{DAT} = \underbrace{\mathbb{E}_{y\sim\mathcal{D}_{\text{harm}},x\sim p_\phi^{\text{diff}}(x|y)}}_{\text{Monte Carlo Sampling}} \left[\underbrace{\max_{\delta\in\Delta} \mathcal{L}_{adv}(x+\delta, y, y_{refusal})}_{\text{Adversarial Perturbation}} \right] + \lambda_{\text{ret}}\mathcal{L}_{KL} \tag{3}$$

After filtering, we average over the retained pairs. We use continuous adversarial training (CAT) to compute the adversarial loss $\mathcal{L}_{adv}$ (Xhonneux et al., 2024). Specifically, we approximate the **inner loop** maximum using finite-step continuous perturbations δ to token embeddings:

$$\mathcal{L}_{adv} = \lambda_A \underbrace{\log p_\theta(y \mid x+\delta)}_{\text{Away Loss}} - \lambda_T \underbrace{\log p_\theta(y_{refusal} \mid x+\delta)}_{\text{Toward Loss}}. \tag{4}$$

Here, y is the desired affirmative harmful response and $y_{refusal}$ is the model's safe refusal, and $\lambda_A, \lambda_T \in \mathbb{R}^+$. The **outer loop** minimization then penalizes the model for these discovered vulnerabilities. To prevent the model from collapsing to degenerate solutions and to maintain general performance, we regularize the outer minimization with a KL divergence term on a retain set $\mathcal{D}_{\text{ret}}$ weighted by a scalar $\lambda_{\text{ret}} \in \mathbb{R}^+$:

$$\mathcal{L}_{KL} = \mathbb{E}_{x \sim \mathcal{D}_{\text{ret}}}[D_{\text{KL}}(p_{\theta_0}(\cdot \mid x) \parallel p_\theta(\cdot \mid x))], \tag{5}$$

where p_{θ_0} is the model before AT. This unified objective trains the model on diverse diffusion-generated samples while targeting adversarial perturbations in the inner loop.

4.6 Theoretical Motivation

The diffusion surrogate can be formally motivated for a bounded robust loss ℓ_{rob}. Assuming $q(h(y) = 1) > 0$, let $\mathcal{R}_{pop}(\theta) = \mathbb{E}_{\tilde{q}}[\ell_{rob}]$ be the harmful-population risk and $\mathcal{R}_{diff}(\theta) = \mathbb{E}_{y \sim \tilde{q}(y)} \mathbb{E}_{x \sim p_\phi^{\text{diff}}(x|y)}[\ell_{rob}]$ the diffusion surrogate risk. We bound their discrepancy.

Theorem 4.1 (Fidelity Bound)**.** *For a fixed θ, assume the robust loss is bounded, i.e., $|\ell_{rob}(x, y; \theta)| \leq M$ for all $(x, y) \in \mathcal{Z}$, and the diffusion surrogate satisfies the conditional fidelity assumption*

$$\mathbb{E}_{y \sim \tilde{q}(y)} \left[\text{TV}\big(q(\cdot \mid y), p_\phi^{\text{diff}}(\cdot \mid y)\big) \right] \leq \varepsilon.$$

Then

$$|\mathcal{R}_{pop}(\theta) - \mathcal{R}_{diff}(\theta)| \leq 2M\varepsilon.$$

Under the fidelity assumption, a closer conditional approximation tightens the bound. Additional unfiltered samples can reduce sampling error, not surrogate bias. This motivates conditional generation, without guaranteeing our filtered, finite-data training with an unclipped loss. Transferability and diversity do not establish fidelity. The proof and optimization consequence are in Section A.

5 Experiment Setup

Models. We study three instruction-tuned LLMs: Llama3-8B-Instruct (Grattafiori et al., 2024), Qwen2.5-14B-Instruct (Qwen et al., 2025), and Gemma3-12B-Instruct (Gemma Team et al., 2025). [2] We run most ablations on Llama3-8B and report multi-model results in Table 1. For more information, see Appendix B.1.

Training Data. We use the HarmBench training split (Mazeika et al., 2024) as the source of harmful behaviors. For each behavior, we generate candidate harmful prompts via diffusion Inpainting following Lüdke et al. (2025): we condition LLaDA-8B-Base (Nie et al., 2025) on an affirmative harmful target and sample 1000 prompt variants. Because conditional samples can vary in linguistic quality and effectiveness, we filter these candidates by evaluating each prompt on the target model and scoring the resulting completion with the StrongREJECT rubric judge (Souly et al., 2024). We find that this data generation and filtering step is inexpensive in practice (see also Appendix B.4). For the main experiments, we keep the top 12 prompts per behavior, yielding 1200 diffusion-generated harmful prompts in total, which matches the training data size of CAT and MixAT. We study the effect of dataset size and filtering in Section 6.4 and Section 6.5. As in prior work (Xhonneux et al., 2024), we use UltraChat200k as a retain set to preserve utility.

DAT Training. We perform DAT by applying CAT (Xhonneux et al., 2024) to the diffusion-generated prompts. Based on preliminary findings in our experiments that they improve the utility-robustness trade-off, we slightly adapt the original CAT algorithm. We replace the cross-entropy retain loss with a KL divergence term, which was found to improve utility in Sheshadri et al. (2024). Additionally, we

[2] For brevity, we refer to the models as Llama3, Qwen2.5, and Gemma3.

Table 1: **Main Results** on Llama3-8B, Qwen2.5-14B, and Gemma3-12B across robustness and utility benchmarks. ASR is measured using the StrongREJECT judge with a harmfulness threshold of 0.5. Diff-only models are trained without adversarial optimization; see Section 6.3. Worst Case Attack (WCA) is the per-behavior union over all available attacks. To demonstrate improved robustness beyond Inpainting, WCA/Inp. is computed over all other attacks except for Inpainting.

Model	Method	ASR ↓									Utility ↑			
		GCG	PAIR	BoN	Direct	GCG+	Inpaint	Claudini	WCA	WCA/Inp.	XSTest	MMLU	ARCe	ARCc
Llama3	No Defense	0.39	0.43	0.99	0.33	0.98	0.98	0.88	1.00	1.00	0.660	0.638	0.815	0.530
	CAT	0.04	0.21	0.00	0.00	0.05	0.88	0.09	0.88	0.24	0.464	0.637	0.810	0.520
	LAT	0.09	0.18	0.00	0.01	0.83	0.94	0.51	1.00	1.00	0.444	0.619	0.784	0.485
	CB	0.03	0.10	0.56	0.13	0.18	0.92	0.10	1.00	1.00	0.672	0.636	0.815	0.533
	MixAT	0.31	0.03	0.00	0.00	0.78	0.90	0.61	0.94	0.92	0.426	0.618	0.830	0.527
	Diff-only	0.20	0.15	0.88	0.04	0.30	0.40	0.25	0.92	0.88	0.536	0.644	0.823	0.536
	DAT	0.04	0.13	0.00	0.00	0.06	0.24	0.09	0.34	0.17	0.424	0.643	0.816	0.530
Qwen2.5	No Defense	0.92	0.83	0.98	0.27	1.00	0.99	0.96	1.00	1.00	0.748	0.789	0.862	0.619
	CAT	0.09	0.25	0.11	0.00	0.18	0.97	0.33	0.98	0.51	0.544	0.787	0.856	0.607
	MixAT	0.06	0.04	0.00	0.00	0.43	0.79	0.21	0.83	0.62	0.358	0.776	0.882	0.632
	Diff-only	0.27	0.18	0.51	0.02	0.50	0.20	0.38	0.65	0.61	0.620	0.782	0.851	0.596
	DAT	0.00	0.09	0.00	0.00	0.04	0.08	0.00	0.18	0.11	0.436	0.772	0.810	0.584
Gemma3	No Defense	0.59	0.78	0.94	0.29	0.99	0.99	0.90	1.00	1.00	0.696	0.715	0.767	0.612
	CAT	0.20	0.36	0.35	0.00	0.58	0.98	0.46	1.00	0.86	0.488	0.715	0.769	0.617
	DAT	0.09	0.08	0.01	0.00	0.12	0.00	0.03	0.19	0.19	0.443	0.711	0.780	0.624

remove loss thresholds, which we found to improve robustness. Regarding the adversarial constraints, we set the ϵ-ball radius to infinity, limiting attack strength solely by the number of iterations to reduce the number of tunable hyperparameters. These changes are applied to both DAT and the CAT baseline and ablated in Section C.4. We tuned hyperparameters independently for all trained models to enable a fair comparison. For a detailed list of hyperparameters, see Section B.2.

Baselines. We compare DAT against the original models, standard AT methods (CAT (Xhonneux et al., 2024), LAT (Sheshadri et al., 2024)), hybrid approaches that combine discrete and continuous attacks (MixAT (Dékány et al., 2025)), and Circuit Breakers (CB; (Zou et al., 2024b)). We further consider more expensive methods like MixAT+GCG, an extension with online GCG augmentation that improves robustness at the cost of $\sim 8\times$ higher training compute in Section C.3.

Evaluation. We evaluate robustness on 100 malicious behaviors from JAILBREAKBENCH (Chao et al., 2024) using a suite of model-specific and data-specific attacks: GCG (Zou et al., 2023), PAIR (Chao et al., 2023), BoN (Hughes et al., 2024), and diffusion Inpainting (Lüdke et al., 2025). To evaluate out-of-distribution generalization, we use LLADA-MOE-7B-A1B-BASE (Zhu et al., 2025b) as the attack model, which differs from the diffusion model used to generate the training data.[3] We further evaluate robustness against sampling-based attacks, including Direct Attack (with original prompts) (Scholten et al., 2025) and GCG+ (Beyer et al., 2025b). We also evaluate Claudini (Panfilov et al., 2026), an autoresearch-based framework that discovers state-of-the-art adversarial attacks. For each behavior–attack pair we generate multiple completions and score every output with the StrongREJECT judge (Souly et al., 2024), which assigns a harmfulness score $\mathcal{H} \in [0, 1]$ to the prompt–response pair. We classify an output as harmful if $\mathcal{H}$ exceeds a fixed threshold of 0.5 and mark the attack as successful. To capture worst-case robustness across attacks, we also report a Worst-Case Attack (WCA) ASR, analogous to the ALO-ASR metric of MixAT (Dékány et al., 2025), where a behavior is considered broken if any available attack succeeds. WCA thus measures maximum capability over the evaluated attacks. We provide a list of attack hyperparameters in Appendix B.3. Finally, we assess utility using XSTest (Röttger et al., 2024) for helpfulness, and MMLU (Hendrycks et al., 2021), ARC-E, and ARC-C (Clark et al., 2018) for general capabilities, following their standard evaluation protocols.

[3] Additional Inpainting evaluations using the Dream and RND1 diffusion models are reported in Section C.5.

6 RESULTS

In the following, we evaluate the effectiveness of DAT and ablate different design choices.

6.1 DAT CONSIDERABLY IMPROVES WORST-CASE ROBUSTNESS

Table 1 summarizes the robustness and utility of all methods. While baseline defenses achieve low ASR on gradient-based and optimization attacks, they exhibit considerable generalization failures. Across all three models, every evaluated baseline has at least one substantial failure beyond Inpainting. Notably, CB remains vulnerable to BoN, LAT and MixAT to GCG+, and CAT to PAIR. Moreover, the Inpainting attack breaks all prior defenses except the diffusion-trained variants. Diffusion-only reduces Inpainting ASR, but remains vulnerable under the aggregate WCA metric. In contrast, DAT achieves the lowest WCA on every model: 34% on Llama3-8B, compared with at least 88% for the baselines; 18% on Qwen2.5-14B, compared with at least 65%; and 19% on Gemma3-12B, compared with 100%. Crucially, this advantage persists after excluding Inpainting: DAT obtains WCA/Inp. rates of 17%, 11%, and 19%, compared with 24%, 51%, and 86% for the strongest baseline on each model. Thus, the robustness gains extend beyond the attack family used to generate the training data. DAT also obtains XSTest scores of $0.424/0.436/0.443$ while maintaining competitive general capabilities on MMLU and ARC. While CB achieves a higher XSTest score (0.672) on Llama3-8B, we note that its training process explicitly incorporates the XSTest dataset, rendering a direct comparison with the other methods for this specific metric non-meaningful. Together, these results support our hypothesis that robustness requires both data-specific diffusion samples and continuous adversarial optimization.

6.2 DAT EXHIBITS PARETO-OPTIMAL ROBUST–UTILITY TRADE-OFFS

For a defense to be practical, it must preserve the model's helpfulness and compliance on benign requests. We evaluate this trade-off on Llama3-8B by varying the regularization strength λ_{ret} (balancing the robust loss and the KL-divergence on the retain set), and the number of performed attack iterations in the inner loop of adversarial training. This allows us to trace the Pareto frontier between *Inpainting Robustness* ($1 -$ ASR) and *XSTest Compliance*. As illustrated in Figure 3 (left), DAT consistently expands the Pareto-front across all hyperparameter configurations. Notably, as we tune for higher compliance and utility, DAT maintains substantially greater robustness than baseline AT methods. For instance, even for compliance levels close to that of the original model, DAT remains as robust as previous AT approaches. These results show that DAT maintains a higher level of robustness across various compliance targets compared to baseline approaches, which can be effectively controlled with interpretable hyperparameters.

6.3 APPROXIMATING THE ADVERSARIAL LOSS IMPROVES ROBUSTNESS

Next, we analyze a Diffusion-only ablation, using diffusion-generated training prompts without continuous adversarial optimization on Llama3-8B and Qwen2.5-14B. Results are shown in Table 1 under the method name Diff-only. The trained models achieve remarkable robustness against Inpainting (40% and 20% ASR on Llama3-8B and Qwen2.5-14B, respectively) compared to previous approaches. However, the Diffusion-only model remains vulnerable to other adversarial attacks (e.g., GCG+, BoN, Claudini). In contrast, DAT achieves high robustness against both attack types. These results support a dual approach: broadening the training data in the outer loop and approximating the worst-case adversarial loss for each data point in the inner loop.

6.4 BETTER APPROXIMATING THE DATA DISTRIBUTION IMPROVES ROBUSTNESS

A core motivation of DAT is to reduce approximation error in robust learning by expanding the training distribution via a generative surrogate. To investigate this, we study how the number of unique diffusion-generated prompts per behavior affects the final model robustness. As shown in Figure 3 (right), increasing the number of samples M used in our filtered Monte Carlo approximation results in a consistent reduction in the Inpainting ASR, which is the strongest individual attack against the prior defenses in our evaluation.

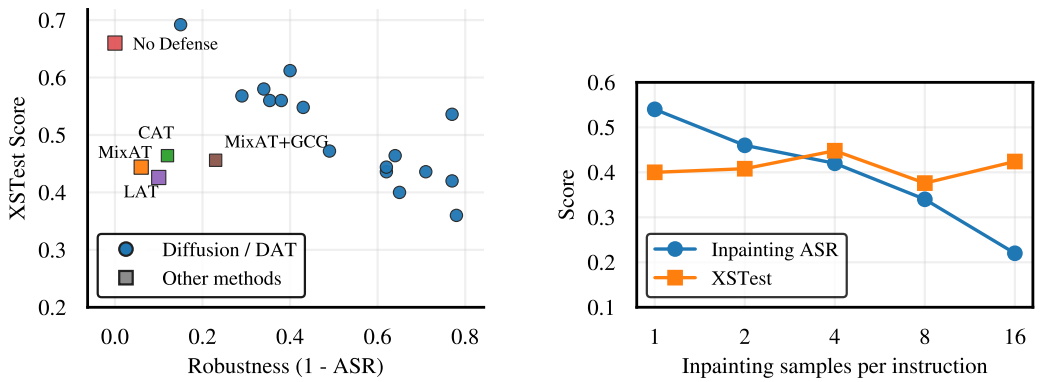

Figure 3: **Performance and Ablation Analysis. Left:** Pareto frontier for Llama3-8B showing the trade-off between Inpainting Robustness ($1 - \text{ASR}$) and XSTest compliance. DAT (blue dots) achieves superior trade-offs compared to baselines like LAT and MixAT+GCG. **Right:** Impact of dataset size on performance. ASR drops as the number of unique diffusion samples M per behavior increases, while helpfulness remains consistent.

Specifically, as the number of unique prompts per harmful behavior in training data increases from 1 to 16, the ASR on Llama3-8B decreases from 54% to 22%. Consistent with the distributional motivation of Theorem 4.1, increasing the number of diffusion-generated training samples empirically improves robustness and supports the hypothesis that broader coverage can narrow the finite-data–population robust-risk gap. Notably, this scaling of harmful data does not compromise the model's helpfulness. Helpfulness scores on XSTest remain stable around 0.4 across all sample sizes.

6.5 Ablation of the Filtering Strategy

The number of training samples for a given harmful behavior is limited to $M = 12$ in our experiments to keep training cost moderate and the training data size consistent with prior work. The diffusion surrogate samples a broad range of prompts, of which only a small percentage trigger harmful completions in the target model. With a limited number of training samples, selecting effective prompts seems promising for achieving high robustness. To test this design choice, we removed this filtering step and chose random surrogate samples instead. We observed a 25-percentage-point reduction in worst-case robustness (details in Section C.7). While random sampling does not obtain the best robustness, it still achieves superior performance to baseline methods, making it a viable candidate for training at scale where the increased cost of filtering is of relevance.

7 Conclusion

Our results suggest that the choice of training data in the outer loop of adversarial training is essential. Understanding which properties of this data, such as diversity, transferability, or harmfulness, lead to the largest robustness gains is a promising direction for future work.

Limitations. While not a major overhead in our experiments, the computational cost of generating large-scale synthetic datasets could become non-trivial in real-world scale training runs, and further research into more efficient sampling could potentially further increase robustness. Moreover, we excluded recent reasoning models because existing adversarial training frameworks cannot be directly applied to these architectures. In preliminary experiments, established baselines failed to yield robust models, even when reasoning capabilities were disabled. Adapting adversarial training to this setting requires designing specialized algorithmic variants, such as objectives defined over multi-step reasoning trajectories, which represents an independent research objective beyond the scope of this work. Consequently, evaluating DAT alone would prevent a controlled baseline comparison. We therefore restrict our scope to standard autoregressive model families where all methods can be evaluated under uniform experimental conditions. We believe adapting adversarial training to reasoning models is an interesting gap for future work.

Reproducibility statement

We provide all relevant method details and hyperparameters required to reproduce our results in the main paper and appendices. Moreover, we provide data used to train the DAT models in the supplemental material. We have also provided a simple version of our code to reproduce training. Together with the continuous adversarial training code (Xhonneux et al., 2024) available online and the hyperparameters given in the appendix, they directly enable reproducing our training runs. Evaluation runs can be directly reproduced with the framework from Beyer et al. (2025a). Trained models will be uploaded to Hugging Face after acceptance to enable evaluations for future work.

AI use statement

We used GPT-5.6 as a writing assistant to improve the clarity, grammar, and presentation of the manuscript. We also used it for retrieval and discovery by checking whether relevant related work was missing. The authors reviewed and verified all resulting edits and references. Further, the authors used Opus coding agents during software development and execution.

References

Maksym Andriushchenko, Francesco Croce, and Nicolas Flammarion. Jailbreaking leading safety-aligned llms with simple adaptive attacks. *arXiv preprint arXiv:2404.02151*, 2024.

Tim Beyer, Jonas Dornbusch, Jakob Steimle, Moritz Ladenburger, Leo Schwinn, and Stephan Günnemann. Adversariallm: A unified and modular toolbox for llm robustness research. *arXiv preprint arXiv:2511.04316*, 2025a.

Tim Beyer, Yan Scholten, Leo Schwinn, and Stephan Günnemann. Sampling-aware adversarial attacks against large language models. *arXiv preprint arXiv:2507.04446*, 2025b.

Stephen Casper, Lennart Schulze, Oam Patel, and Dylan Hadfield-Menell. Defending against unforeseen failure modes with latent adversarial training. *arXiv preprint arXiv:2403.05030*, 2024.

Patrick Chao, Alexander Robey, Edgar Dobriban, Hamed Hassani, George J Pappas, and Eric Wong. Jailbreaking black box large language models in twenty queries. *arXiv preprint arXiv:2310.08419*, 2023.

Patrick Chao, Edoardo Debenedetti, Alexander Robey, Maksym Andriushchenko, Francesco Croce, Vikash Sehwag, Edgar Dobriban, Nicolas Flammarion, George J Pappas, Florian Tramer, et al. Jailbreakbench: An open robustness benchmark for jailbreaking large language models. *arXiv preprint arXiv:2404.01318*, 2024.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv:1803.05457v1*, 2018.

Csaba Dékány, Stefan Balauca, Robin Staab, Dimitar I Dimitrov, and Martin Vechev. Mixat: Combining continuous and discrete adversarial training for llms. *arXiv preprint arXiv:2505.16947*, 2025.

Suyu Ge, Chunting Zhou, Rui Hou, Madian Khabsa, Yi-Chia Wang, Qifan Wang, Jiawei Han, and Yuning Mao. MART: Improving LLM safety with multi-round automatic red-teaming. In Kevin Duh, Helena Gomez, and Steven Bethard (eds.), *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 1927–1937, Mexico City, Mexico, June 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.naacl-long.107. URL https://aclanthology.org/2024.naacl-long.107/.

Simon Geisler, Tom Wollschläger, MHI Abdalla, Johannes Gasteiger, and Stephan Günnemann. Attacking large language models with projected gradient descent. *arXiv preprint arXiv:2402.09154*, 2024.

Simon Geisler, Tom Wollschläger, MHI Abdalla, Vincent Cohen-Addad, Johannes Gasteiger, and Stephan Günnemann. REINFORCE adversarial attacks on large language models: An adaptive, distributional, and semantic objective. *arXiv preprint arXiv:2502.17254*, 2025.

Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, et al. Gemma 3 technical report. *arXiv preprint arXiv:2503.19786*, 2025.

Alison L. Gibbs and Francis Edward Su. On choosing and bounding probability metrics. *arXiv preprint arXiv:math/0209021*, 2002. URL https://arxiv.org/abs/math/0209021.

Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *ICLR*, 2015.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding, 2021. URL https://arxiv.org/abs/2009.03300.

John Hughes, Sara Price, Aengus Lynch, Rylan Schaeffer, Fazl Barez, Sanmi Koyejo, Henry Sleight, Erik Jones, Ethan Perez, and Mrinank Sharma. Best-of-n jailbreaking. *arXiv preprint arXiv:2412.03556*, 2024.

Klim Kireev, Maksym Andriushchenko, and Nicolas Flammarion. On the effectiveness of adversarial training against common corruptions. In *UAI*, pp. 1012–1021, 2022.

Raz Lapid, Ron Langberg, and Moshe Sipper. Open sesame! universal black box jailbreaking of large language models. *arXiv preprint arXiv:2309.01446*, 2023.

Lin Li and Michael Spratling. Data augmentation alone can improve adversarial training. *arXiv preprint arXiv:2301.09879*, 2023.

Nathaniel Li, Ziwen Han, Ian Steneker, Willow Primack, Riley Goodside, Hugh Zhang, Zifan Wang, Cristina Menghini, and Summer Yue. Llm defenses are not robust to multi-turn human jailbreaks yet. *arXiv preprint arXiv:2408.15221*, 2024.

Fan Liu, Zhao Xu, and Hao Liu. Adversarial tuning: Defending against jailbreak attacks for llms. *arXiv preprint arXiv:2406.06622*, 2024.

David Lüdke, Tom Wollschläger, Paul Ungermann, Stephan Günnemann, and Leo Schwinn. Diffusion llms are natural adversaries for any llm. *arXiv preprint arXiv:2511.00203*, 2025.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards Deep Learning Models Resistant to Adversarial Attacks. In *ICLR*, 2018.

Mantas Mazeika, Long Phan, Xuwang Yin, Andy Zou, Zifan Wang, Norman Mu, Elham Sakhaee, Nathaniel Li, Steven Basart, Bo Li, et al. Harmbench: A standardized evaluation framework for automated red teaming and robust refusal. *arXiv preprint arXiv:2402.04249*, 2024.

Shen Nie, Fengqi Zhu, Zebin You, Xiaolu Zhang, Jingyang Ou, Jun Hu, Jun Zhou, Yankai Lin, Ji-Rong Wen, and Chongxuan Li. Large language diffusion models, 2025. URL https://arxiv.org/abs/2502.09992.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *NeurIPS*, 2022.

Alexander Panfilov, Peter Romov, Igor Shilov, Yves-Alexandre de Montjoye, Jonas Geiping, and Maksym Andriushchenko. Claudini: Autoresearch discovers state-of-the-art adversarial attack algorithms for llms. 2026. URL https://api.semanticscholar.org/CorpusID:286775666.

Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report, 2025. URL https://arxiv.org/abs/2412.15115.

Radical Numerics Inc. Training diffusion language models at scale using autoregressive models. Technical report, Radical Numerics Inc., October 2025. URL https://www.radicalnumerics.ai/assets/rnd1_report.pdf.

Nils Reimers and Iryna Gurevych. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan (eds.), *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 3982–3992, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1410. URL https://aclanthology.org/D19-1410/.

Paul Röttger, Hannah Kirk, Bertie Vidgen, Giuseppe Attanasio, Federico Bianchi, and Dirk Hovy. XSTest: A test suite for identifying exaggerated safety behaviours in large language models. In Kevin Duh, Helena Gomez, and Steven Bethard (eds.), *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 5377–5400, Mexico City, Mexico, June 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.naacl-long.301. URL https://aclanthology.org/2024.naacl-long.301/.

Mikayel Samvelyan, Sharath Chandra Raparthy, Andrei Lupu, Eric Hambro, Aram H. Markosyan, Manish Bhatt, Yuning Mao, Minqi Jiang, Jack Parker-Holder, Jakob Foerster, Tim Rocktäschel, and Roberta Raileanu. Rainbow teaming: Open-ended generation of diverse adversarial prompts, 2024. URL https://arxiv.org/abs/2402.16822.

Ludwig Schmidt, Shibani Santurkar, Dimitris Tsipras, Kunal Talwar, and Aleksander Madry. Adversarially robust generalization requires more data. In *NeurIPS*, 2018.

Yan Scholten, Stephan Günnemann, and Leo Schwinn. A probabilistic perspective on unlearning and alignment for large language models. In *ICLR*, 2025.

Leo Schwinn, David Dobre, Sophie Xhonneux, Gauthier Gidel, and Stephan Günnemann. Soft prompt threats: Attacking safety alignment and unlearning in open-source LLMs through the embedding space. In *NeurIPS*, 2024.

Ali Shafahi, Mahyar Najibi, Mohammad Amin Ghiasi, Zheng Xu, John Dickerson, Christoph Studer, Larry S Davis, Gavin Taylor, and Tom Goldstein. Adversarial training for free! In *NeurIPS*, 2019.

Abhay Sheshadri, Aidan Ewart, Phillip Guo, Aengus Lynch, Cindy Wu, Vivek Hebbar, Henry Sleight, Asa Cooper Stickland, Ethan Perez, Dylan Hadfield-Menell, et al. Latent adversarial training improves robustness to persistent harmful behaviors in llms. *arXiv preprint arXiv:2407.15549*, 2024.

Alexandra Souly, Qingyuan Lu, Dillon Bowen, Tu Trinh, Elvis Hsieh, Sana Pandey, Pieter Abbeel, Justin Svegliato, Scott Emmons, Olivia Watkins, et al. A strongreject for empty jailbreaks. *arXiv preprint arXiv:2402.10260*, 2024.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian J. Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *ICLR*, 2014.

Lewis Tunstall, Edward Beeching, Nathan Lambert, Nazneen Rajani, Kashif Rasul, Younes Belkada, Shengyi Huang, Leandro Von Werra, Clémentine Fourrier, Nathan Habib, et al. Zephyr: Direct distillation of lm alignment. *arXiv preprint arXiv:2310.16944*, 2023.

Zekai Wang, Tianyu Pang, Chao Du, Min Lin, Weiwei Liu, and Shuicheng Yan. Better diffusion models further improve adversarial training. In *ICML*, 2023.

Eric Wong, Leslie Rice, and J. Zico Kolter. Fast is better than free: Revisiting adversarial training. In *ICLR*, 2019.

Sophie Xhonneux, Alessandro Sordoni, Stephan Günnemann, Gauthier Gidel, and Leo Schwinn. Efficient adversarial training in llms with continuous attacks. *NeurIPS*, 2024.

Jiacheng Ye, Zhihui Xie, Lin Zheng, Jiahui Gao, Zirui Wu, Xin Jiang, Zhenguo Li, and Lingpeng Kong. Dream 7b: Diffusion large language models, 2025. URL https://arxiv.org/abs/2508.15487.

Lei Yu, Virginie Do, Karen Hambardzumyan, and Nicola Cancedda. Robust LLM safeguarding via refusal feature adversarial training. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=s5orchdb33.

Fengqi Zhu, Rongzhen Wang, Shen Nie, Xiaolu Zhang, Chunwei Wu, Jun Hu, Jun Zhou, Jianfei Chen, Yankai Lin, Ji-Rong Wen, and Chongxuan Li. Llada 1.5: Variance-reduced preference optimization for large language diffusion models, 2025a. URL https://arxiv.org/abs/2505.19223.

Fengqi Zhu, Zebin You, Yipeng Xing, Zenan Huang, Lin Liu, Yihong Zhuang, Guoshan Lu, Kangyu Wang, Xudong Wang, Lanning Wei, Hongrui Guo, Jiaqi Hu, Wentao Ye, Tieyuan Chen, Chenchen Li, Chengfu Tang, Haibo Feng, Jun Hu, Jun Zhou, Xiaolu Zhang, Zhenzhong Lan, Junbo Zhao, Da Zheng, Chongxuan Li, Jianguo Li, and Ji-Rong Wen. Llada-moe: A sparse moe diffusion language model, 2025b. URL https://arxiv.org/abs/2509.24389.

Sicheng Zhu, Ruiyi Zhang, Bang An, Gang Wu, Joe Barrow, Zichao Wang, Furong Huang, Ani Nenkova, and Tong Sun. Autodan: Automatic and interpretable adversarial attacks on large language models. *arXiv preprint arXiv:2310.15140*, 2023.

Andy Zou, Zifan Wang, J Zico Kolter, and Matt Fredrikson. Universal and transferable adversarial attacks on aligned language models. *arXiv preprint arXiv:2307.15043*, 2023.

Andy Zou, Long Phan, Justin Wang, Derek Duenas, Maxwell Lin, Maksym Andriushchenko, Rowan Wang, Zico Kolter, Matt Fredrikson, and Dan Hendrycks. Improving alignment and robustness with circuit breakers, 2024a.

Andy Zou, Long Phan, Justin Wang, Derek Duenas, Maxwell Lin, Maksym Andriushchenko, Rowan Wang, Zico Kolter, Matt Fredrikson, and Dan Hendrycks. Improving alignment and robustness with short circuiting. *arXiv preprint arXiv:2406.04313*, 2024b.

A Proof of Theorem 4.1

A.1 Setup

Let q denote a natural language distribution over $(x, y) \in \mathcal{Z}$ for a set of token sequences $\mathcal{Z}$. Further, let $h : \mathcal{Z} \to \{0, 1\}$ be a harmfulness indicator, where $h(y) = 1$ denotes a harmful response. Assuming $q(h(y) = 1) > 0$, we define the distribution of prompts and their harmful responses as

$$\tilde{q}(x, y) \;:=\; q\big(x, y \mid h(y) = 1\big) \;=\; \frac{q(x, y)\,\mathbf{1}\{h(y) = 1\}}{q(h(y) = 1)}.$$

We let $p_\theta^{\mathrm{AR}}(y|x)$ denote the target autoregressive model. We define the robust loss

$$\ell_{rob}(x, y; \theta) \;:=\; \sup_{\delta \in \Delta} \mathcal{L}\big(p_\theta^{\mathrm{AR}}(y \mid x + \delta)\big),$$

where Δ is a perturbation set (e.g. in embedding space) and use the standard total variation distance

$$\mathrm{TV}(P, Q) \;:=\; \sup_A \big|P(A) - Q(A)\big| \;=\; \frac{1}{2}\,\|P - Q\|_1.$$

Assumptions For a fixed θ, we assume this loss is bounded, i.e.,

$$|\ell_{rob}(x, y; \theta)| \le M \qquad \text{for all } (x, y) \in \mathcal{Z}. \tag{6}$$

This assumption permits a TV-based bound. It can be satisfied by a bounded robust-loss functional, for example by clipping ℓ_{rob} to $[-M, M]$. The experimental log-ratio objective is not clipped, and no such bound is established for it; the theorem provides motivation through a bounded-loss idealization.

Further, we assume that the diffusion surrogate satisfies the conditional fidelity bound

$$\mathbb{E}_{y \sim \tilde{q}(y)} \left[\mathrm{TV}\big(q(\cdot \mid y), p_\phi^{\mathrm{diff}}(\cdot \mid y)\big)\right] \le \varepsilon.$$

True population robust risk.

$$\mathcal{R}_{pop}(\theta) = \mathbb{E}_{(x,y)\sim\tilde{q}}[\ell_{rob}(x, y; \theta)]. \tag{7}$$

Diffusion surrogate risk. Using a fixed diffusion model as a surrogate for the prompt conditional $q(x \mid y)$, while retaining the harmful-response marginal $\tilde{q}(y)$, gives the risk

$$\mathcal{R}_{diff}(\theta) = \mathbb{E}_{y\sim\tilde{q}(y)}\mathbb{E}_{x\sim p_\phi^{\mathrm{diff}}(x|y)}[\ell_{rob}(x, y; \theta)]. \tag{8}$$

Restriction to harmful y. Since $\tilde{q}$ is the restriction of q to harmful responses, its marginal satisfies

$$\tilde{q}(y) \;=\; q\big(y \mid h(y) = 1\big).$$

Moreover, for any y with $\tilde{q}(y) > 0$, the conditional over prompts is unchanged:

$$\tilde{q}(x \mid y) \;=\; \frac{\tilde{q}(x, y)}{\tilde{q}(y)} \;=\; \frac{q(x, y)}{q(y)} \;=\; q(x \mid y). \tag{9}$$

A.2 Proof

We wish to bound the absolute difference $|\mathcal{R}_{pop}(\theta) - \mathcal{R}_{diff}(\theta)|$.

Proof. By the law of iterated expectations under $\tilde{q}$ (equivalently, rearranging sums since $\mathcal{Z}$ is discrete) and using (9), for y with $\tilde{q}(y) > 0$ we may equivalently view the inner conditional $x \sim \tilde{q}(x \mid y)$ as $x \sim q(x \mid y)$,

$$\begin{aligned}
\mathcal{R}_{pop}(\theta) &= \mathbb{E}_{(x,y)\sim\tilde{q}}\big[\ell_{rob}(x, y; \theta)\big] \\
&= \mathbb{E}_{y\sim\tilde{q}(y)}\;\mathbb{E}_{x\sim\tilde{q}(x|y)}\big[\ell_{rob}(x, y; \theta)\big] \\
&= \mathbb{E}_{y\sim\tilde{q}(y)}\;\mathbb{E}_{x\sim q(x|y)}\big[\ell_{rob}(x, y; \theta)\big].
\end{aligned} \tag{10}$$

Combining (10) with the definition of $\mathcal{R}_{diff}$ gives

$$|\mathcal{R}_{pop}(\theta) - \mathcal{R}_{diff}(\theta)| = \left|\mathbb{E}_{y\sim\tilde{q}(y)}\left[\mathbb{E}_{x\sim q(x|y)}[\ell_{rob}] - \mathbb{E}_{x\sim p_\phi^{\text{diff}}(x|y)}[\ell_{rob}]\right]\right| \tag{11}$$

$$\leq \mathbb{E}_{y\sim\tilde{q}(y)}\left[\left|\mathbb{E}_{x\sim q(x|y)}[\ell_{rob}] - \mathbb{E}_{x\sim p_\phi^{\text{diff}}(x|y)}[\ell_{rob}]\right|\right], \tag{12}$$

where the triangle inequality is applied.

We use the following standard inequality for any function $f : \mathcal{Z} \to [-M, M]$ and distributions P, Q on $\mathcal{Z}$,

$$|\mathbb{E}_{x\sim P}[f(x)] - \mathbb{E}_{x\sim Q}[f(x)]| \leq 2M \cdot \text{TV}(P, Q), \tag{13}$$

where TV is total variation (Gibbs & Su, 2002). Applying (13) to (12) with

$$P = q(\cdot \mid y), \qquad Q = p_\phi^{\text{diff}}(\cdot \mid y), \qquad f(\cdot) = \ell_{rob}(\cdot, y; \theta),$$

and using the boundedness assumption (6), we obtain

$$\left|\mathbb{E}_{x\sim q(x|y)}[\ell_{rob}] - \mathbb{E}_{x\sim p_\phi^{\text{diff}}(x|y)}[\ell_{rob}]\right| \leq 2M \cdot \text{TV}\big(q(\cdot \mid y),\ p_\phi^{\text{diff}}(\cdot \mid y)\big). \tag{14}$$

Substituting (14) into (12) yields

$$|\mathcal{R}_{pop}(\theta) - \mathcal{R}_{diff}(\theta)| \leq \mathbb{E}_{y\sim\tilde{q}(y)}\left[2M \cdot \text{TV}\big(q(\cdot \mid y),\ p_\phi^{\text{diff}}(\cdot \mid y)\big)\right]$$

$$= 2M \cdot \mathbb{E}_{y\sim\tilde{q}(y)}\left[\text{TV}\big(q(\cdot \mid y),\ p_\phi^{\text{diff}}(\cdot \mid y)\big)\right]. \tag{15}$$

Finally, using the fidelity assumption we obtain

$$|\mathcal{R}_{pop}(\theta) - \mathcal{R}_{diff}(\theta)| \leq 2M \cdot \varepsilon.$$

□

This concludes the proof. The bound shows that the approximation error of our surrogate objective is linear in the *fidelity* of the diffusion surrogate, measured as an expected conditional TV distance under the harmful-response marginal $y \sim \tilde{q}(y)$.

A.3 Optimization Consequence and Practical Scope

If the same bound M holds for every θ in a parameter class Θ, the fidelity bound holds uniformly over Θ. Thus, for any η-approximate minimizer $\hat{\theta}$ of the surrogate risk, meaning $\mathcal{R}_{diff}(\hat{\theta}) \leq \inf_{\theta\in\Theta} \mathcal{R}_{diff}(\theta) + \eta$,

$$\mathcal{R}_{pop}(\hat{\theta}) \leq \inf_{\theta\in\Theta} \mathcal{R}_{pop}(\theta) + 4M\varepsilon + \eta.$$

Indeed, applying the fidelity bound twice gives

$$\mathcal{R}_{pop}(\hat{\theta}) \leq \mathcal{R}_{diff}(\hat{\theta}) + 2M\varepsilon \leq \inf_{\theta\in\Theta} \mathcal{R}_{diff}(\theta) + \eta + 2M\varepsilon \leq \inf_{\theta\in\Theta} \mathcal{R}_{pop}(\theta) + \eta + 4M\varepsilon.$$

The same argument applies when both risks include the same retain-set regularizer.

This analysis motivates conditional prompt generation with diffusion models: under the stated assumptions, optimizing the surrogate controls population robust risk. The argument applies to faithful conditional generators generally, motivating surrogates that better approximate the relevant prompt conditionals; diffusion models offer one practical realization through inverse prompt generation.

Our implementation additionally uses a curated response dataset, filters generated prompts, and optimizes an unclipped loss with finite-step adversarial attacks. These choices are not covered by the stated assumptions. The result therefore serves as theoretical motivation, rather than a guarantee for the implemented algorithm or its jailbreak ASR.

B Experiment Details

B.1 Models and Adapters

We train and evaluate defenses on Llama3-8B-Instruct (Grattafiori et al., 2024), Qwen2.5-14B-Instruct (Qwen et al., 2025), and Gemma3-12B-Instruct (Gemma Team et al., 2025). We additionally use Qwen2.5-7B-Instruct and Zephyr-7B (Tunstall et al., 2023) in transfer evaluations.

For baselines and transfer attacks, we utilize public adversarially trained adapters including Circuit Breaker (CB) (Zou et al., 2024a) and LAT (Sheshadri et al., 2024). The attack model for our inpainting is the LLaDA-MoE-7B diffusion model Zhu et al. (2025b). The generative surrogate used for our DAT framework is the LLaDA-8B model (Nie et al., 2025). Table 2 provides the exact Hugging Face identifiers for all checkpoints used in our experiments.

Table 2: Sources of Hugging Face models and adapters.

Base Model	Adapter	HF Source
Llama3-8B	–	`meta-llama/Meta-Llama-3-8B-Instruct`
	CB	`GraySwanAI/Llama-3-8B-Instruct-RR`
	LAT	`LLM-LAT/robust-llama3-8b-instruct`
	MixAT	`INSAIT-Institute/Llama3-8B-MixAT`
	MixAT+GCG	`INSAIT-Institute/Llama3-8B-MixAT+GCG`
Qwen2.5-14B	–	`Qwen/Qwen2.5-14B-Instruct`
	MixAT	`INSAIT-Institute/Qwen-14B-MixAT`
	MixAT+GCG	`INSAIT-Institute/Qwen-14B-MixAT+GCG`
Qwen2.5-7B	–	`Qwen/Qwen2.5-7B-Instruct`
Zephyr-7B	–	`HuggingFaceH4/zephyr-7b-beta`
Gemma3-12B	–	`google/gemma-3-12b-it`
LLaDA-8B	–	`GSAI-ML/LLaDA-8B-Base`
LLaDA-MoE-7B	–	`inclusionAI/LLaDA-MoE-7B-A1B-Base`
Dream-7B	–	`Dream-org/Dream-v0-Base-7B`
RND1-30B-A3B	–	`radicalnumerics/RND1-Base-0910`
Vicuna-13B	–	`lmsys/vicuna-13b-v1.5`

B.2 Training

Table 3 and Table 4 report the hyperparameter search spaces and selected configurations for our DAT method and the CAT baseline, respectively. We tuned each method independently and selected configurations based on their robustness–utility trade-off. We performed more extensive preliminary checks for CAT during the modification of the original algorithm. We performed all training experiments on a cluster of NVIDIA H100 and H200 GPUs.

Table 3: Hyperparameter search spaces and selected configurations for models trained with DAT.

Hyperparameter	Search space	Llama3-8B	Qwen2.5-14B	Gemma3-12B
Learning rate	$[1 \times 10^{-5}, 2 \times 10^{-4}]$	1×10^{-4}	1×10^{-4}	1×10^{-4}
Batch size	$[4, 16]$	8	8	8
Number of epochs	$[1, 6]$	6	6	6
Optimizer	–	AdamW	AdamW	AdamW
Adv. learning rate	–	1×10^{-1}	1×10^{-1}	1×10^{-1}
Attack steps	$[10, 40]$	20	20	20
Attack radius	–	1×10^{7}	1×10^{7}	1×10^{7}
Utility data ratio	–	0.5	0.5	0.5
Max seq. length	–	128	128	128
Away/toward weights	see text below	$(0.5, 0.5)$	$(0.3, 0.7)$	$(0.35, 0.18)$
Utility weight	$[0.04, 1.2]$	0.45	0.25	0.15
Quantization	–	none (bf16)	none (bf16)	none (bf16)

Table 4: Hyperparameter search spaces and selected configurations for models trained with CAT.

Hyperparameter	Search space	Llama3-8B	Qwen2.5-14B	Gemma3-12B
Learning rate	$[1 \times 10^{-5}, 2 \times 10^{-4}]$	2×10^{-4}	1×10^{-4}	2×10^{-4}
Batch size	–	16	16	16
Number of epochs	–	6	6	6
Optimizer	–	AdamW	AdamW	AdamW
Attack radius	–	1×10^{7}	1×10^{7}	1×10^{7}
Adv. learning rate	–	1×10^{-1}	1×10^{-1}	1×10^{-1}
Attack steps	$[10, 100]$	20	20	20
Utility data ratio	–	0.5	0.5	0.5
Max seq. length	–	128	128	128
Away/toward weights	see text below	$(0.5, 0.5)$	$(0.3, 0.7)$	$(0.03, 0.5)$
Utility weight	$[0.04, 1.2]$	0.25	0.25	0.04
Quantization	–	none (fp16)	none (bf16)	none (bf16)

Away/toward loss weights. We tune the away and toward loss weights jointly, using the same model-specific paired configurations for DAT and CAT. We evaluate $(0.3, 0.7)$, $(0.5, 0.5)$, $(0.6, 0.4)$, $(0.7, 0.3)$, $(0.8, 0.2)$, and $(0.9, 0.1)$ for Llama3-8B; $(0.3, 0.7)$ and $(0.5, 0.5)$ for Qwen2.5-14B; and $(0.03, 0.50)$, $(0.10, 0.43)$, $(0.20, 0.33)$, and $(0.35, 0.18)$ for Gemma3-12B.

Table 5: Training time. *For MixAT and MixAT+GCG, we report costs from Dékány et al. (2025).

Method	Llama3-8B	Qwen2.5-14B
CAT	1h30	6h00
MixAT*	1h40	6h00
DAT	1h17	6h48
MixAT+GCG*	12h50	48h30

B.3 Attacks

We list the detailed attack hyperparameters in Table 6. We used the attack code base AdversariaLLM (Beyer et al., 2025a). If an attack uses multiple steps or produces different versions of attacks, we prompt the target model on each candidate and report success if at least one candidate achieves a successful judge score. Unless stated otherwise, we use greedy decoding when sampling the target model.

B.4 Training Cost

The cost of DAT has two components: generating the diffusion samples and training the defended model on those samples. The first component is an offline preprocessing step that can be amortized across all downstream training runs. As a calibration point, our largest inpainting generation for evaluation produced 100,000 prompts (1000 per behavior) in roughly 5h on a single H100 GPU. Smaller training sets are cheaper: in the no-filtering setting, where we directly sample the prompts used for training, generation takes approximately 5 minutes for 12 samples per behavior.

The second component is the adversarial training run itself. We report training runtime on one H200 cluster. As shown in Table 5, DAT, CAT, and MixAT have similar training costs. MixAT+GCG instead performs online GCG optimization during training, making it 8 times more expensive.

Table 6: Detailed attack hyperparameters.

GCG (Zou et al., 2023) hyperparameters	**Value**
Num steps	250
Adv string init	`"x x x x x x x x x x x x x x x x x x x x"` (20 suffix tokens)
Search width	512
Batch size	512
Top-k	256
Tokens replaced per step	1
Loss	`ce`
Early stopping	`False`
Target generations per step	1
Target temperature	0

GCG+ (Beyer et al., 2025b) hyperparameters	**Value**
Num steps	500
Target generations per step	50
Target temperature	1.0
All other fields identical to GCG.	

PAIR (Chao et al., 2023) hyperparameters	**Value**
Streams	20
Steps	3
Keep last N	3
Max retries	10
Attack max tokens	512
Attack temperature	1.0
Attack top-p	0.7
Attack model	`lmsys/vicuna-13b-v1.5`
Target max tokens	256
Target temperature	0

BoN (Hughes et al., 2024) hyperparameters	**Value**
Num steps	1000
σ	0.4
Perturbations	word scrambling; random capitalization; ASCII
Target temperature	1.0

Direct (Scholten et al., 2025) hyperparameters	**Value**
Target temperature	1.0
Target max tokens	256
Target generations per behavior	1000

Inpainting (Lüdke et al., 2025) hyperparameters	**Value**
Samples per behavior	1000
Target temperature	0
Target max tokens	256
Target generations per prompt	1

Claudini (Panfilov et al., 2026) hyperparameters	**Value**
Variant	`claude-oss-v53`
Num steps	1000
Num adv tokens	20
Final attack target generations	1000
Target temperature	1.0
Num starts	6

C Additional Analyses and Related Work

This section provides additional discussion and analyses supporting the main text, including comparisons with related data-augmentation approaches, further baseline and robustness evaluations, and implementation ablations.

C.1 Relation to Data Augmentation Approaches

Related approaches use automated red teaming to generate data for safety fine-tuning. Rainbow Teaming generates a diverse, static dataset of adversarial prompts and fine-tunes the target model on the resulting data (Samvelyan et al., 2024). However, it does not explicitly solve the inner adversarial optimization problem against the trained model and is therefore not designed to provide worst-case robustness. MART instead iteratively finds discrete adversarial examples for the current target-model checkpoint and uses them for safety fine-tuning (Ge et al., 2024). It does not explicitly optimize for worst-case examples in a continuous inner loop or address broad coverage of the outer data distribution, including contexts that do not themselves elicit harmful outputs.

MixAT (Dékány et al., 2025) combines model-specific discrete and continuous attacks to improve robustness. It augments the original data with perturbed examples generated by the PAP attack algorithm.[4] Because MixAT uses an autoregressive model to generate these prompts and subsequently applies continuous adversarial training, it provides a strong published autoregressive-generator comparison to DAT. Specifically, its MixAT, DualAT, and PAP-AT variants differ in how the PAP prompts are utilized for training. MixAT applies continuous perturbations on top of PAP prompts, DualAT trains on PAP prompts and continuously perturbed original prompts side by side, and PAP-AT trains on PAP prompts alone. Dékány et al. (2025) ablate these combinations together with the proportion of PAP data and find MixAT to perform best, making it a representative autoregressive-generator baseline. We further evaluate a PAIR-based autoregressive surrogate under DAT's exact training setup in Section C.2. MixAT+GCG additionally performs online GCG optimization as a third data source, improving robustness at substantially higher training cost. In contrast, DAT uses diffusion-generated adversarial training prompts without requiring additional discrete attack sources and achieves a lower worst-case ASR, including relative to MixAT+GCG (see Section C.3).

These approaches establish the value of generated adversarial data for defense. DAT differs by using a diffusion model to conditionally sample prompts given harmful responses, thereby broadening the distribution represented in the outer expectation, and by combining these samples with continuous adversarial optimization in the inner loop. Under the assumptions of Theorem 4.1, improving the surrogate distribution tightens its approximation to the population robust risk.

C.2 Autoregressive Surrogate Control

To test whether an autoregressive attack generator can replace the diffusion surrogate under an identical training setup, we construct PAIR-DAT, which substitutes PAIR (Chao et al., 2023) for the diffusion model. PAIR conditions on both the harmful behavior and the affirmative target response. We retain the same filtering rule, 1200 training prompts, and CAT training budget as DAT, and optimize the hyperparameters for a favorable safety–utility trade-off. We additionally report the public MixAT checkpoint, which augments training with autoregressively generated PAP prompts. We evaluate robustness on the first 40 JailbreakBench behaviors.

Table 7: Matched comparison of diffusion and autoregressive surrogates on Llama3-8B.

Method	ASR ↓				Utility ↑			
	GCG	GCG+	Inpainting	PAIR	XSTest	MMLU	ARC-E	ARC-C
No Defense	0.400	1.000	0.975	0.425	0.660	0.638	0.815	0.530
MixAT	0.350	0.750	0.900	0.025	0.426	0.618	0.830	0.527
PAIR-DAT	0.350	0.650	0.875	0.000	0.392	0.647	0.765	0.535
DAT	0.050	0.075	0.225	0.175	0.424	0.643	0.816	0.530

[4]The public MixAT configuration uses `mistralai/Mixtral-8x7B-Instruct-v0.1` over 40 persuasion strategies to generate jailbreaks autoregressively with the PAP algorithm; see the config file.

Both autoregressive variants are robust mainly to attacks resembling their training generator. PAIR-DAT reaches 0.000 ASR against PAIR, and MixAT, whose PAP prompts are likewise LLM-generated semantic rewrites, reaches 0.025. Against GCG and GCG+, which fall outside both generators' families, PAIR-DAT remains at 0.350 and 0.650 and MixAT at 0.350 and 0.750, whereas DAT reaches 0.050 and 0.075. Conversely, DAT is less robust against PAIR (0.175), the attack PAIR-DAT specializes in. On general capabilities, DAT matches the undefended model, whereas PAIR-DAT loses 5 points on ARC-E. These findings mirror those of Dékány et al. (2025), who observe that only by adding online GCG as a third data source can it improve overall robustness and conclude that the diversity of training attacks is crucial for broad robustness.

In contrast, DAT never observes PAIR-like prompts during training yet generalizes across all four attacks while maintaining comparable utility. Rather than combining several attack-specific data sources, DAT obtains broad coverage from a single diffusion surrogate that samples training prompts from a broader conditional distribution. This result supports using the diffusion surrogate for broad attack coverage rather than specializing to a single generator.

C.3 Comparison with MixAT+GCG

Table 8: Comparison of DAT with MixAT and MixAT+GCG on Llama3-8B and Qwen2.5-14B.

Model	Method	ASR ↓								Utility ↑			
		GCG	Pair	BoN	Direct	GCG+	Inpaint	Claudini	WCA	XSTest	MMLU	ARCe	ARCc
Llama	MixAT	0.31	0.03	0.00	0.00	0.78	0.90	0.61	0.94	0.426	0.618	0.830	0.527
	MixAT+GCG	0.04	0.02	0.02	0.01	0.08	0.77	0.05	0.77	0.456	0.612	0.827	0.513
	DAT	0.04	0.13	0.00	0.00	0.06	0.24	0.09	0.34	0.424	0.643	0.816	0.530
Qwen	MixAT	0.06	0.04	0.00	0.00	0.43	0.79	0.21	0.83	0.358	0.776	0.882	0.632
	MixAT+GCG	0.05	0.03	0.00	0.00	0.05	0.75	0.05	0.75	0.388	0.778	0.862	0.619
	DAT	0.00	0.09	0.00	0.00	0.04	0.08	0.00	0.18	0.436	0.772	0.810	0.584

MixAT+GCG augments MixAT with online GCG optimization during training, making it a useful high-compute reference point. As shown in Table 8, DAT remains competitive across different attacks and utility, and achieves the lowest WCA rate while using only a single data source for training and avoiding costly online discrete search, as shown in Section B.4.

C.4 Ablation of CAT Implementation Choices

We isolate the effects of three changes to the original CAT implementation: replacing the cross-entropy retain loss with KL divergence, removing the loss cutoffs, and removing the finite ϵ-ball constraint. Starting from the CAT configuration used in this paper, each ablation changes one component while retaining the remaining training settings. We evaluate all variants on Llama3-8B using the first 40 JailbreakBench behaviors.

Table 9: Ablation of the CAT implementation choices on Llama3-8B. Our configuration achieves a favorable helpfulness–robustness trade-off while preserving general model capabilities. Results that are substantially worse than the other configurations are marked in red.

	ASR ↓		Utility ↑			
Configuration	GCG	Inpainting	XSTest	ARC-E	ARC-C	MMLU
CE retain loss	0.050	1.000	0.624	0.754	0.454	0.545
Loss cutoffs	0.850	0.950	0.656	0.815	0.529	0.638
$\epsilon = 0.2$	0.050	0.850	0.440	0.812	0.515	0.639
$\epsilon = 0.1$	0.075	0.850	0.468	0.814	0.524	0.638
$\epsilon = 0.05$	0.075	0.850	0.508	0.816	0.521	0.639
CAT (ours)	0.025	0.800	0.464	0.810	0.520	0.637

The ablations reveal a clear robustness–utility trade-off. The red entries highlight the principal failure modes: replacing KL divergence with cross-entropy increases XSTest compliance but substantially reduces ARC and MMLU utility and weakens Inpainting robustness, whereas restoring the loss

cutoffs largely preserves utility but sharply increases ASR. Finite ϵ-ball constraints maintain similar general-capability scores, yet provide no robustness benefit over the unconstrained configuration, which obtains the lowest ASR on both attacks. These results support applying the same modified CAT implementation to both CAT and DAT in the main comparison.

C.5 Robustness Across Diffusion Attack Models

To test whether robustness transfers beyond the LLaDA-MoE, we construct additional Inpainting attacks with Dream-7B-Base (Ye et al., 2025) and RND1-30BA3B-Base (Radical Numerics Inc., 2025). We evaluate the first 40 JailbreakBench behaviors. Dream and the original LLaDA-MoE attack use 1000 samples per query. Because of the computational cost of RND1, we use 128 samples per query and include an LLaDA-MoE result at the same sampling budget for comparison.

Table 10: Inpainting ASR on Llama3-8B under different diffusion attack models and sampling budgets. The best result in each column is shown in bold.

	LLaDA-MoE-128	RND1-128	Dream-1000	LLaDA-MoE-1000
No Defense	0.925	0.800	0.400	0.975
CAT	0.500	0.325	0.150	0.800
MixAT	0.550	0.250	0.100	0.900
DAT	**0.100**	**0.050**	**0.000**	**0.225**

DAT achieves the lowest ASR for every attack model and sampling budget. In particular, its advantage persists for Dream and RND1, where baselines show vulnerabilities. This indicates that the robustness against Inpainting is not specific to attacks generated by one model.

C.6 Training–Evaluation Prompt Overlap

We additionally test whether the Inpainting results can be explained by overlap between the diffusion-generated training and evaluation prompts. The training set contains 1200 prompts, whereas the evaluation set contains 100,000 prompts generated from a separate set of behaviors. An exhaustive comparison between the two sets finds only one exact duplicate across all train–evaluation prompt comparisons.

To measure semantic similarity, we compute TF–IDF cosine similarity between each evaluation prompt and its nearest training prompt. The median nearest-neighbor similarity is 0.104. As a within-evaluation reference, the median similarity to the nearest evaluation prompt associated with a different behavior is 0.149. Evaluation prompts are therefore no more similar to the training set than they are to prompts from unrelated evaluation behaviors. Together, the lexical and semantic comparisons provide no evidence that train–evaluation overlap explains DAT's Inpainting robustness.

C.7 Effect of Prompt Selection

Our main data-selection procedure ranks diffusion-generated prompts according to the StrongREJECT score of the target model's response. To assess the value of this filtering step, we compare models trained on either the 12 highest-scoring or 12 randomly selected prompts per behavior, while holding all other training settings fixed.

Table 11: Effect of filtering diffusion-generated training prompts on Llama3-8B over the first 40 JailbreakBench behaviors.

	ASR ↓					**Utility ↑**			
Method	GCG	PAIR	BoN	Inpainting	WCA	XSTest	MMLU	ARC-E	ARC-C
DAT-top12	0.050	0.175	0.000	0.225	0.350	0.424	0.643	0.816	0.530
DAT-random12	0.225	0.050	0.025	0.375	0.600	0.412	0.638	0.751	0.531

Top-score filtering substantially improves aggregate robustness, reducing WCA from 0.60 to 0.35 and lowering GCG and Inpainting ASR from $0.23/0.38$ to $0.05/0.23$. It also improves ARC-E by 6.5 percentage points, while the remaining utility scores are similar. These results demonstrate the benefit of filtering when robustness is the primary objective. Random selection nevertheless remains competitive against PAIR and BoN and avoids the cost of generating and scoring a larger candidate pool, providing a lower-compute alternative at the cost of weaker overall robustness.

C.8 Adversarial Attacks in LLMs

Adversarial attacks on LLMs have evolved rapidly, ranging from optimization-based methods to automated generation techniques. Optimization-based attacks like GCG (Zou et al., 2023), soft prompts (Schwinn et al., 2024) and PGD (Geisler et al., 2024) use gradient information to find adversarial suffixes or continuous perturbations. Automated methods such as AutoDAN (Zhu et al., 2023), Open Sesame (Lapid et al., 2023), and PAIR (Chao et al., 2023) leverage attacker LLMs or genetic algorithms to generate interpretable jailbreaks. Other notable approaches include purely random and model-unspecific attacks, such as Best-of-N jailbreaking (Hughes et al., 2024), or simple adaptive attacks (Andriushchenko et al., 2024) which combine human manual tuning with optimization. Recently, Geisler et al. (2025) proposed using REINFORCE to optimize for expected harmfulness, addressing limitations in fixed-target objectives. Some of these attack methods have been integrated into standardized evaluation frameworks like HarmBench (Mazeika et al., 2024) and used for adversarial training in works like MixAT (Dékány et al., 2025). Finally, Lüdke et al. (2025) demonstrates that diffusion LLMs can conditionally generate diverse adversarial prompts targeting a harmful response y without requiring gradient-based optimization. We use this approach to broaden the training data for adversarial training.

C.9 Diffusion LLMs as Natural Adversaries

Lüdke et al. (2025) demonstrated that Diffusion LLMs can act as natural adversaries. Because these models learn the joint distribution of prompts and responses, they can effectively invert the conditional distribution. By fixing a harmful target response y, one can sample from the diffusion model to generate diverse, high-likelihood adversarial prompts that are naturally effective against a wide range of LLMs without requiring gradient-based optimization. This enables us to sample training data from a learned approximation to the true conditional distribution, with the aim of approximating the population risk during training. Importantly, this approach focuses on sampling from the specific regions of the manifold that are highly likely to elicit the harmful response y. Thus, instead of drawing samples from the entire distribution q, where the majority are not relevant for the task, we can efficiently target the specific vulnerabilities relevant for adversarial training.