Certifying Trustworthy Machine Learning

Hanbin Hong, Ph.D.

University of Connecticut, 2025

Ensuring the robustness of machine learning classifiers against adversarial perturbations is paramount for their reliable deployment in security-sensitive applications. This dissertation introduces a series of novel frameworks to enhance certified robustness and uncover fundamental vulnerabilities in black-box settings. First, we present UniCR (Universally Approximated Certified Robustness), a groundbreaking framework that provides a universal robustness certification for any classifier against any ℓ_p perturbations using noise from any continuous probability distribution. UniCR automates robustness certification, offering both theoretical guarantees and empirical validation on optimal noise distributions used in randomized smoothing. Building upon this foundation, we introduce UCAN (Universally Amplifying Randomized Smoothing for Certified Robustness with Anisotropic Noise), which extends randomized smoothing techniques by leveraging anisotropic noise to amplify certified robustness. This approach generalizes existing smoothing techniques across different threat models, demonstrating substantial improvements in certified accuracy across benchmark datasets. Next,

we propose Certifiable Black-Box Attacks with Randomized Adversarial Examples, introducing a new paradigm of black-box attacks with provable guarantees. Unlike traditional empirical black-box attacks, certifiable black-box attacks guarantee the attack success probability (ASP) of adversarial examples before querying the target model. This attack paradigm reveals critical vulnerabilities in state-of-the-art defenses, providing a framework to construct an infinite space of adversarial examples with high ASP, bypassing detection mechanisms with provable confidence. We establish a theoretical foundation for ensuring ASP using randomized adversarial examples, propose novel optimization techniques to generate imperceptible attacks, and comprehensively evaluate their effectiveness on multiple datasets, benchmarking against 16 SOTA black-box attacks and various defenses in vision and speech domains. Finally, we systematically address the growing security challenges in large language models (LLMs), which have become foundational in real-world AI applications. We present a holistic systematization of knowledge (SoK) on prompt security, encompassing a unified multi-level taxonomy of attacks, defenses, and vulnerabilities, along with formalized threat models and open-source tools for standardized, reproducible evaluation. We introduce JAILBREAKDB—the largest curated corpus of jailbreak and benign prompts—and provide a comprehensive benchmark and leaderboard for state-of-the-art LLM security research. Our findings unify previously fragmented research efforts, lay the foundation for rigorous evaluation of LLM robustness, and facilitate the development of robust, trustworthy language models for deployment in high-stakes environments.

Certifying Trustworthy Machine Learning

Hanbin Hong

B.S., Xi'an Jiaotong University, 2018

A Dissertation

Submitted in Partial Fulfillment of the

Requirements for the Degree of

Doctor of Philosophy

at the

University of Connecticut

2025

APPROVAL PAGE

Doctor of Philosophy Dissertation

Certifying Trustworthy Machine Learning

Presented by

Hanbin Hong, B.S.

Major Advisor

Yuan Hong

Associate Advisor

Benjamin Fuller

Associate Advisor

Derek Aguiar

University of Connecticut

2025

ACKNOWLEDGEMENTS

It is with a profound sense of gratitude and humility that I reflect upon the manifold kindnesses and encouragements bestowed upon me throughout the course of my scholarly pursuits, the culmination of which is herein presented. Foremost, I am deeply indebted to my esteemed advisor, Professor Yuan Hong, whose sagacious counsel, unfailing patience, and inspiring example have served as a beacon, guiding me through the labyrinthine paths of academic inquiry and personal growth. I likewise extend my heartfelt appreciation to the distinguished members of my dissertation committee—Professors Benjamin Fuller, Derek Aguiar, Dongjin Song, and Fei Miao—whose discerning judgment and thoughtful guidance have, at every turn, enriched and elevated this work beyond measure. To my learned collaborators, Professor Binghui Wang, Dr. Xinyu Zhang, and Dr. Wentao Bao, I owe a debt of gratitude for the intellectual companionship, rigorous discourse, and generous sharing of ideas that have so greatly enlivened my research journey. I am no less grateful to my colleagues within the laboratory—Han Wang, Shuya Feng, Shenao Yan, Ali Arastehfard, Nima Naderloui, and Qin Yang—together with the many earnest students with whom I have labored side by side; the spirit of camaraderie and mutual striving which pervaded our daily endeavors has rendered the arduous passage lighter and more rewarding. I must also gratefully acknowledge the University of Connecticut and the Department of Computer Science &

Engineering for affording me an academic home both inspiring and sustaining, where the pursuit of knowledge is held in the highest esteem. Above all, my heart overflows with gratitude towards my family, whose steadfast love, fortitude, and faith have been my constant refuge and source of strength. In particular, I am forever indebted to my beloved wife, Biying Liu, whose steadfast companionship and unwavering support have been my surest harbor through every storm. To all who have touched this journey with their kindness and wisdom, I render my sincerest thanks.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1

Introduction

The rapid advancement of machine learning (ML) has fundamentally transformed numerous domains, ranging from computer vision and natural language processing to healthcare and autonomous systems. As ML models, particularly deep neural networks, are increasingly deployed in real-world, safety-critical applications, their robustness and trustworthiness have become paramount concerns. Despite achieving remarkable predictive performance, these models remain alarmingly vulnerable to adversarial perturbations—malicious, often imperceptible modifications to input data that can cause drastic changes in model predictions. Such vulnerabilities have raised critical questions about the reliability and safety of deploying ML models in high-stakes scenarios.

To address these challenges, the research community has invested substantial effort into developing *certified defenses*, which seek to provide formal, provable guarantees on the robustness of machine learning models against carefully bounded adversarial attacks. Among these, *randomized smoothing* has emerged as a leading approach, owing to its strong theoretical foundation and its applicability to a wide range of models. Randomized smoothing constructs a smoothed classifier by adding noise to the

input and defining the output as the most probable class under the induced distribution. This method enables one to derive a *certified radius*—the largest perturbation (measured in an ℓ_p norm) within which the model's prediction is provably invariant. However, despite their theoretical appeal, existing certified defenses are fundamentally limited by their lack of universality and flexibility. Most state-of-the-art methods are tied to specific combinations of noise distributions and norm constraints, such as Gaussian noise for ℓ_2 perturbations or Laplace noise for ℓ_1 perturbations. These approaches require case-by-case theoretical analyses and fail to generalize across models, input types, and attack scenarios. For instance, the certified radius derived by Cohen et al. [1] is restricted to Gaussian noise and ℓ_2-norm perturbations, while others [2, 3] have attempted to extend the theory to a limited set of other norms and distributions. Nevertheless, these methods still fall short of providing a truly universal and automated certification procedure applicable to arbitrary classifiers, continuous noise distributions, and perturbation norms.

Moreover, as ML systems are deployed at scale, the limitations of existing frameworks become even more pronounced. The requirement for manual, case-specific analysis and the lack of optimizable, analysis-free certification pipelines hinder the adoption of certified robustness in practice. Furthermore, while randomized smoothing theoretically enables robust classification, the tightness and optimality of the certified radius depend critically on the choice of noise distribution. Yet, the identification or validation of optimal noise distributions for given models and threat models remains largely

unexplored, leaving practitioners with suboptimal robustness guarantees in real-world settings.

In parallel, the emergence of large language models (LLMs) has introduced a new frontier of security concerns. LLMs have rapidly become foundational components in various AI-driven systems, but their deployment has been marred by unique vulnerabilities such as prompt injection, jailbreak attacks, and model misuse. Despite the severity of these risks, there is a lack of systematic understanding and standardized benchmarks for evaluating LLM security, leaving both researchers and practitioners ill-equipped to measure and defend against these threats.

Motivated by these challenges, this thesis seeks to establish a principled and comprehensive foundation for *certifying trustworthy machine learning* in both vision and language domains. First, we present UniCR (Universally Approximated Certified Robustness) [4], a groundbreaking framework that, for the first time, provides universal robustness certification for any classifier against any ℓ_p perturbations using noise from any continuous probability distribution. UniCR automates the certification process, delivering both theoretical guarantees and empirical validation on the optimality of noise distributions used in randomized smoothing. Building upon this foundation, we introduce UCAN (Universally Amplifying Randomized Smoothing for Certified Robustness with Anisotropic Noise) [5], which generalizes randomized smoothing techniques by leveraging anisotropic noise to amplify certified robustness. This approach unifies and extends existing smoothing techniques across different threat models, and demonstrates

substantial improvements in certified accuracy across benchmark datasets.

In addition to advancing the frontier of certified defense, we propose a new paradigm of *certifiable black-box attacks* with randomized adversarial examples [6]. Unlike traditional empirical black-box attacks that require iterative querying or rely on transferability from surrogate models, our certifiable black-box attack paradigm provides provable guarantees on attack success probability before querying the target model. This fundamentally challenges the conventional understanding of black-box security, revealing critical vulnerabilities in current state-of-the-art defenses, and providing a framework to construct an infinite space of adversarial examples with high success probability that can bypass detection mechanisms with provable confidence. Our work establishes the theoretical foundation for attack success probability under randomized adversarial examples, introduces novel optimization techniques for generating imperceptible attacks, and thoroughly evaluates their effectiveness against a wide array of models and defenses, spanning CIFAR10/100, ImageNet, and LibriSpeech benchmarks.

Finally, we systematically address the growing security challenges in large language models. Recognizing the urgent need for standardized, reproducible, and comprehensive evaluation, we present a holistic systematization of knowledge on LLM prompt security. This includes a unified multi-level taxonomy of attacks, defenses, and vulnerabilities, the formalization of threat models, and the development of open-source tools and resources for benchmarking. Notably, we introduce *JAILBREAKDB*,

the largest curated corpus of jailbreak and benign prompts, and establish the first comprehensive benchmark and leaderboard for LLM security research. Our systematization not only unifies previously fragmented research efforts but also lays the groundwork for rigorous, community-driven evaluation and robust model development.

Collectively, the contributions of this thesis illuminate fundamental limitations in existing adversarial defenses, demonstrate the necessity and feasibility of robust certification frameworks, and uncover new attack strategies that challenge the prevailing assumptions of both black-box and LLM security. Through a combination of theoretical innovation, algorithmic development, and empirical validation, this work aims to lay a unified and practical foundation for certifying and evaluating trustworthy machine learning systems, ultimately supporting their safe and reliable deployment in high-stakes real-world applications.

Chapter 2

UniCR: Universally Approximated Certified Robustness via Randomized Smoothing

We study certified robustness of machine learning classifiers against adversarial perturbations. In particular, we propose the first universally approximated certified robustness (UniCR) framework, which can approximate the robustness certification of *any* input on *any* classifier against *any* ℓ_p perturbations with noise generated by *any* continuous probability distribution. Compared with the state-of-the-art certified defenses, UniCR provides many significant benefits: (1) the first universal robustness certification framework for the above 4 "any"s; (2) automatic robustness certification that avoids case-by-case analysis, (3) tightness validation of certified robustness, and (4) optimality validation of noise distributions used by randomized smoothing. We conduct extensive experiments to validate the above benefits of UniCR and the advantages of UniCR over state-of-the-art certified defenses against ℓ_p perturbations.

2.1 Introduction

Machine learning (ML) classifiers are vulnerable to adversarial perturbations [7, 8, 9, 10, 11]). Certified defenses [4, 12, 13, 14, 15, 16, 17, 18, 19, 20] were recently proposed to ensure provable robustness against adversarial perturbations. Typically, certified defenses aim to derive a certified radius such that an arbitrary ℓ_p (e.g., ℓ_1, ℓ_2 or ℓ_∞) perturbation, when added to a testing input, cannot fool the classifier, if the ℓ_p-norm value of the perturbation is within the radius. Among all certified defenses, randomized smoothing [1, 21, 22] based certified defense has achieved the state-of-the-art certified radius and can be applied to *any* classifier. Specifically, given a testing input and any classifier, randomized smoothing first defines a noise distribution and adds sampled noises to the testing input; then builds a smoothed classifier based on the noisy inputs, and finally derives certified radius for the smoothed classified, e.g., using the Neyman-Pearson Lemma [1].

However, existing randomized smoothing based (and actually all) certified defenses only focus on specific settings and cannot universally certify a classifier against *any* ℓ_p perturbation or *any* noise distribution. For example, the certified radius derived by Cohen et al. [1] is tied to the Gaussian noise and ℓ_2 perturbation. Recent works [3, 18, 23] propose methods to certify the robustness for multiple norms/noises, e.g., Yang et al. [3] propose the level set and differential method to derive the certified radii for multiple noise distributions. However, the certified radius derivation for different norms is still *subject to case-by-case theoretical analyses*. These methods, although

Table 2.1: Comparison with highly-related works.

	Classifier	Smoothing Noise	Perturbations	Tightness	Optimizable	Analysis-free
Lecuyer et al. [22]	Any	Gaussian/Laplace	Any $\ell_p, p \in \mathbb{R}^+$	Loose	No	No
Cohen et al. [1]	Any	Gaussian	ℓ_2	Strictly Tight	No	No
Teng et al. [24]	Any	Laplace	ℓ_1	Strictly Tight	No	No
Dvijotham et al. [25]	Any	f-divergence-constrained	Any $\ell_p, p \in \mathbb{R}^+$	Loose	No	No
Croce et al. [18]	ReLU-based	No	Any ℓ_p for $p >= 1$	Loose	No	No
Yang et al. [3]	Any	Multiple types	Any $\ell_p, p \in \mathbb{R}^+$	Strictly Tight	No	No
Zhang et al. [23]	Any	ℓ_p-term-constrained	$\ell_1, \ell_2, \ell_\infty$	Strictly Tight	No	Yes
Ours (UniCR)	Any	Any continuous PDF	Any $\ell_p, p \in \mathbb{R}^+$	Approx. Tight	Yes	Yes

achieving somewhat generalized certified robustness, are still lack of universality (See Table 2.1 for the summary).

In this paper, we develop the first Universally Approximated Certified Robustness (UniCR) framework based on *randomized smoothing*. Our framework can automate the robustness certification for any input on any classifier against any ℓ_p perturbation with noises generated by any *continuous probability density function (PDF)*. As shown in Figure 3.1, our UniCR framework provides four unique significant benefits to make certified robustness more universal, practical and easy-to-use with the above four "any"s. Our key contributions are as follows:

1. **Universal Certification**. UniCR is the first universal robustness certification framework for the 4 "any"s.

2. **Automatic Certification**. UniCR provides an automatic robustness certification for all cases. It is easy-to-launch and avoids case-by-case analysis.

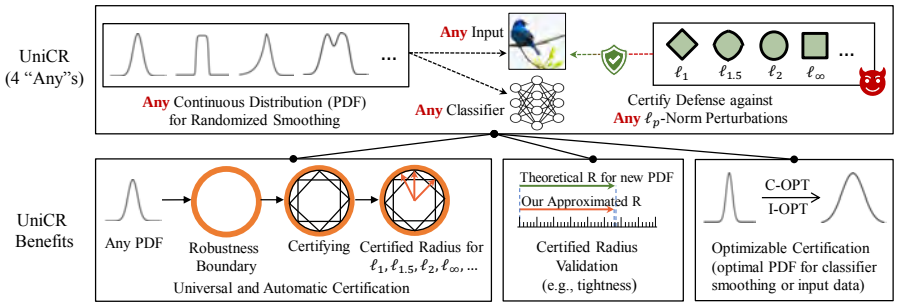


Fig. 2.1: Our Universally Approximated Certified Robustness (UniCR) framework.

3. **Tightness Validation of Certified Radius**. It is also the first framework that can validate the *tightness* of the derived certified radius in existing certification methods [1, 21, 22] or future methods based on any continuous noise PDF. In Section 2.2.3, we validate the tightness of the state-of-the-art certification methods (e.g., see Figure 2.4).

4. **Optimality Validation of Noise PDFs**. UniCR can also automatically tune the parameters in noise PDFs to strengthen the robustness certification against any ℓ_p perturbations. For instance, On CIFAR10 and ImageNet datasets, UniCR improves as high as 38.78% overall performance over the state-of-the-art certified defenses against all ℓ_p perturbations. In Section 2.3, we show that Gaussian noise and Laplace noise are not the optimal randomization distribution against the ℓ_2 and ℓ_1 perturbation, respectively.

2.2 Universally Approximated Certified Robustness

In this section, we propose the theoretical foundation for universally certifying a testing input against any ℓ_p perturbations with noise from any continuous PDF.

2.2.1 Universal Certified Robustness

Consider a general classification problem that classifies input data in $\mathbb{R}^d$ to a class belonging to a set of classes $\mathcal{Y}$. Given an input $x \in \mathbb{R}^d$, an *any* (base) classifier f that maps x to a class in $\mathcal{Y}$, and a random noise ϵ from *any* continuous PDF μ_x. We define a smoothed classifier g as the most probable class over the noise-perturbed input:

$$g(x) = \arg\max_{c \in \mathcal{Y}} \mathbb{P}(f(x + \epsilon) = c) \tag{2.1}$$

Then, we show that the input has a certified accurate prediction against any l_p perturbation and its certified radius is given by the following theorem.

Theorem 1. ***(Universal Certified Robustness)*** *Let $f : \mathbb{R}^d \to \mathcal{Y}$ be any deterministic or random classifier, and let ϵ be drawn from an arbitrary continuous PDF μ_x. Denote g as the smoothed classifier in Equation (2.1), the most probable and second probable classes for predicting a testing input x via g as $c_A, c_B \in \mathcal{Y}$, respectively. If the lower bound of the class c_A's prediction probability $\underline{p_A} \in [0, 1]$, and the upper bound of the class c_B's prediction probability $\overline{p_B} \in [0, 1]$ satisfy:*

$$\mathbb{P}(f(x + \epsilon) = c_A) \geq \underline{p_A} \geq \overline{p_B} \geq \max_{c \neq c_A} \mathbb{P}(f(x + \epsilon) = c) \tag{2.2}$$

Then, we guarantee that $g(x+\delta) = c_A$ for all $\|\delta\|_p \leq R$, where R is called the **certified radius** *and it is the minimum ℓ_p-norm of all the adversarial perturbations δ that satisfies the* **robustness boundary conditions** *as below:*

$$\mathbb{P}(\frac{\mu_x(x-\delta)}{\mu_x(x)} \leq t_A) = \underline{p_A}, \quad \mathbb{P}(\frac{\mu_x(x-\delta)}{\mu_x(x)} \geq t_B) = \overline{p_B},$$
$$\mathbb{P}(\frac{\mu_x(x)}{\mu_x(x+\delta)} \leq t_A) = \mathbb{P}(\frac{\mu_x(x)}{\mu_x(x+\delta)} \geq t_B) \tag{2.3}$$

where t_A and t_B are auxiliary parameters to satisfy the above conditions.

Proof. See the detailed proof in Appendix B.1. □

Robustness Boundary. Theorem 11.1 provides a novel insight that meeting certain conditions is equivalent to deriving the certified robustness. The conditions in Equation (2.3) construct a boundary in the perturbation δ space, which is defined as the "*robustness boundary*". Within this robustness boundary, the prediction outputted by the smoothed classifier g is certified to be consistent and correct. The robustness boundary, rather than the certified radius, is actually more general to measure the certified robustness since the space constructed by each certified radius (against any specific ℓ_p perturbation) is only a subset of the space inside the robustness boundary. Figure 2.2 illustrates the relationship between certified radius and the robustness boundary against ℓ_1, ℓ_2 and ℓ_∞ perturbations.

Notice that, given any continuous noise PDF, the corresponding robustness boundary for all the ℓ_p-norms would naturally exist. Each maximum ℓ_p ball is a subspace of the robustness boundary, and gives the certified radius for that specific ℓ_p-norm. Thus,

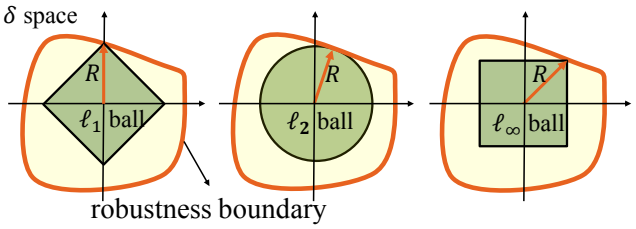


Fig. 2.2: An illustration to Theorem 11.1. The conditions in Theorem 11.1 construct a "**Robustness Boundary**" in δ space. In case of a perturbation inside the robustness boundary, the smoothed prediction can be certifiably correct. From left to right, the figures show that the minimum $\|\delta\|_1$, $\|\delta\|_2$ and $\|\delta\|_\infty$ on the robustness boundary are exactly the certified radius R in ℓ_1, ℓ_2 and ℓ_∞-norm, respectively.

all the certified radii can be universally derived, and Theorem 11.1 provides a theoretical foundation to certify any input against any ℓ_p perturbations with any continuous noise PDF.

All ℓ_p Perturbations. Although we mainly introduce UniCR against ℓ_1, ℓ_2 and ℓ_∞ perturbations, our UniCR is not limited to these three norms. We emphasize that any $p \in \mathbb{R}^+$ (See Appendix D.5) can be used and our UniCR can derive the corresponding certified radius since our robustness boundary gives a general boundary in the δ perturbation space.

2.2.2 Approximating Tight Certified Robustness

The tight certified radius can be derived by finding a perturbation δ on the robustness boundary that has a minimum $\|\delta\|_p$ (for any $p \in \mathbb{R}^+$). However, it is challenging to either find a perturbation δ that is exactly on the robustness boundary, or find the minimum $\|\delta\|_p$. Here, we design an alternative two-phase optimization scheme to accurately approximate the tight certification in practice. In particular, Phase I is to suffice the conditions such that δ is on the robustness boundary, and Phase II is to minimize the ℓ_p-norm.

We perform Phase I by the "scalar optimization", where any perturbation δ will be λ-scaled to the robustness boundary (see ① in Figure 2.3). We perform Phase II by the "direction optimization", where the direction of δ will be optimized towards a minimum $\|\lambda\delta\|_p$ (see ② in Figure 2.3). In the two-phase optimization, the direction optimization will be iteratively executed until finding the minimum $\|\lambda\delta\|_p$, where the perturbation δ will be scaled to the robustness boundary beforehand in every iteration. Thus, the intractable optimization problem in Equation 2.3 can be converted to:

$$
\begin{aligned}
& R = \|\lambda\delta\|_p, \\
& s.t. \quad \delta \in \arg\min_{\delta} \|\lambda\delta\|_p, \quad \lambda = \arg\min_{\lambda} |K|, \\
& \qquad \mathbb{P}(\frac{\mu_x(x-\lambda\delta)}{\mu_x(x)} \leq t_A) = \underline{p_A}, \quad \mathbb{P}(\frac{\mu_x(x-\lambda\delta)}{\mu_x(x)} \geq t_B) = \overline{p_B}, \\
& \qquad K = \mathbb{P}(\frac{\mu_x(x)}{\mu_x(x+\lambda\delta)} \leq t_A) - \mathbb{P}(\frac{\mu_x(x)}{\mu_x(x+\lambda\delta)} \geq t_B).
\end{aligned} \tag{2.4}
$$

The scalar optimization in Equation (2.4) aims to find the scale factor λ that scales a perturbation δ to the boundary so that $|K|$ approaches 0. With the scalar λ for ensuring

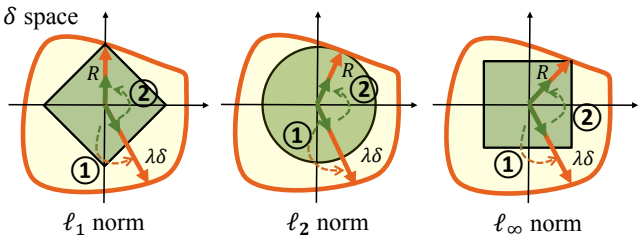


Fig. 2.3: An illustration to estimating the certified radius. The scalar optimization (①) and direction optimization (②) effectively find the minimum $\|\delta\|_p$ within the robustness boundary, which is the certified radius R.

that the scaled δ is nearly on the boundary, the direction optimization optimizes the perturbation δ's direction to find the certified radius $R = \|\lambda\delta\|_p$. We also present the theoretical analysis on the certification confidence and the optimization convergence in Appendix B.4 and B.5, respectively.

2.2.3 Deriving Certified Radius within Robustness Boundary

In this section, we will introduce how to universally and automatically derive the certified radius against any ℓ_p perturbations within the robustness boundary constructed by any noise PDF. In particular, we will present practical algorithms for solving the two-phase optimization problem to approximate the certified radius, empirically validate that our UniCR approximates the tight certified radius derived by recent works [1, 3, 24], and finally discuss how to apply UniCR to validate the radius of existing certified defenses.

2.2.4 Calculating Certified Radius in Practice

Following the existing randomized smoothing based defenses [1, 24], we first use the Monte Carlo method to estimate the probability bounds ($\underline{p_A}$ and $\overline{p_B}$). Then, we use them in our two-phase optimization scheme to derive the certified radius.

Estimating Probability Bounds. The two-phase optimization needs to estimate the probabilities bounds $\underline{p_A}$ and $\overline{p_B}$ and compute two auxiliary parameters t_A and t_B (required by the certified robustness based on the Neyman-Pearson Lemma in Appendix A). Identical to existing works [1, 24], the probabilities bounds $\underline{p_A}$ and $\overline{p_B}$ are commonly estimated by the Monte Carlo method [1]. Given the estimated $\underline{p_A}$ and $\overline{p_B}$ as well as any given noise PDF and a perturbation δ, we also use the Monte Carlo method to estimate the cumulative density function (CDF) of fraction $\mu_x(x - \lambda\delta)/\mu_x(x)$. Then, we can compute the auxiliary parameters t_A and t_B. Specifically, the auxiliary parameters t_A and t_B can be computed by $t_A = \Phi^{-1}(\underline{p_A})$ and $t_B = \Phi^{-1}(\overline{p_B})$, where Φ^{-1} is the inverse CDF of the fraction $\mu_x(x - \lambda\delta)/\mu_x(x)$. The procedures for computing t_A and t_B are detailed in Algorithm 1 in Appendix C.

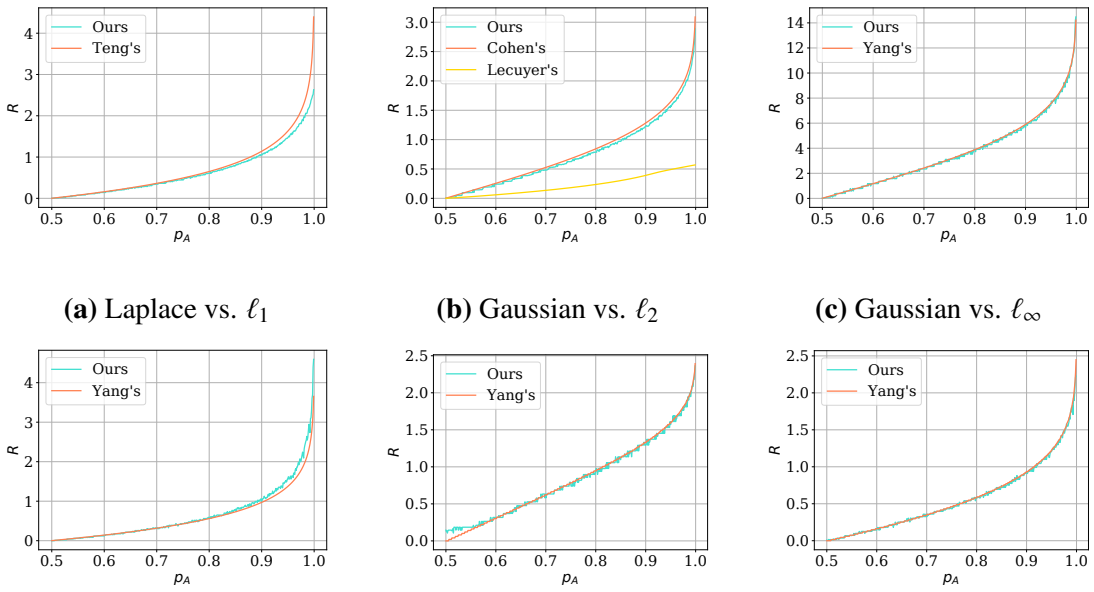
Scalar Optimization. Finding a perturbation δ that is exactly on the robustness boundary is computationally challenging. Thus, we alternatively scale the δ to approach the boundary. We use the binary search algorithm to find a scale factor that minimizes $|K|$ (*the distance between δ and the robustness boundary*). The algorithm and detailed description are presented in Appendix C.2.

Direction Optimization. We use the Particle Swarm Optimization (PSO) method [26]

to find δ that minimizes the ℓ_p-norm after scaling to the robustness boundary. In each iteration of PSO, the particle's position represents δ, and the cost function is $f_{PSO}(\delta) = \|\lambda\delta\|_p$, where the scalar λ is found by the scalar optimization. The PSO aims to find the position δ that can minimize the cost function. To pursue convergence, we choose some initial positions in symmetry for different ℓ_p-norms. Empirical results show that the radius obtained by PSO with these initial positions can accurately approximate the tight certified radius. We show how to set the initial positions in Appendix C.3.

In our experiments, the certification (deriving the certified radius) can be efficiently completed on MNIST [27], CIFAR10 [28] and ImageNet [29] datasets (less than 10 seconds per image), as shown in Appendix D.4.

Certified Radius Comparison with State-of-The-Arts. We compare the certified radius obtained by our two-phase optimization method and that by the state-of-the-arts [1, 3, 24] and the comparison results are shown in Figure 2.4. Note that the certified radius is a function of p_A (the prediction probability of the top-1 class). The p_A-R curve can well depict the certified radius R w.r.t. p_A. We observe that our p_A-R curve highly approximates the tight theoretical curves in existing works, e.g., the Gaussian noise against ℓ_2 and ℓ_∞ perturbations [1, 3], Laplace noise against ℓ_1 perturbations [24], as well as General Normal noise and General Exponential noise derived by Yang et al. [3]'s method.



(a) Laplace vs. ℓ_1 **(b)** Gaussian vs. ℓ_2 **(c)** Gaussian vs. ℓ_∞

(d) Pareto vs. ℓ_1 **(e)** General Normal vs. ℓ_1 **(f)** Exponential Mix. vs. ℓ_1

Fig. 2.4: p_A-R curve comparison of our method and state-of-the-art methods (i.e., Teng et al. [24], Cohen et al. [1], Lecuyer et al. [22], and Yang et al. [3]). We observe that the certified radius obtained by our UniCR is close to that obtained by the state-of-the-art. These results demonstrate that our UniCR can approximate the tight certification for any input in any ℓ_p norm with any continuous noise distribution. We also evaluate our UniCR's defense accuracy against a diverse set of attacks, including universal attacks [30], white-box attacks [31, 32], and black-box attacks [10, 33], and against ℓ_1, ℓ_2, and ℓ_∞ perturbations. The experimental results show that UniCR is as robust as the state-of-the-art methods (100% defense accuracy) against all the types of real attacks. The detailed experimental settings and results are presented in Appendix D.2.

Tightness Validation of Certified Radius. Since our UniCR accurately approximates the tight certified radius, it can be used as an auxiliary tool to validate whether an obtained certified radius is tight or not. For example, the certified radius derived by PixelDP [22][1] is loose, because [22]'s p_A-R curve in Figure 2.4(b) is far below ours. Also, Yang et al. [3] derives a low bound certified radius for Pareto Noise (Figure 2.4(d))— It shows that this certified radius is not tight either since it is below ours. For those theoretical radii that are slightly above our radii, they are likely to be tight.

Moreover, due to the high university, our UniCR can even derive the certified radii for complicated noise PDFs, e.g., mixture distribution in which the certified radii are difficult to be theoretically derived. In Section 2.3.2, we show some examples of deriving radii using UniCR on a wide variety of noise distributions in Figure 2.6-2.8. In most examples, the certified radii have not been studied before.

2.2.5 Optimizing Noise PDF for Certified Robustness

UniCR can derive the certified radius using any continuous noise PDF for randomized smoothing. This provides the flexibility to optimize a noise PDF for enlarging the certified radius. In this section, we will optimize the noisy PDF in our UniCR framework for obtaining better certified robustness.

[1] PixelDP [22] adopts differential privacy [34], e.g., Gaussian mechanism to generate noises for each pixel such that certified robustness can be achieved for images.

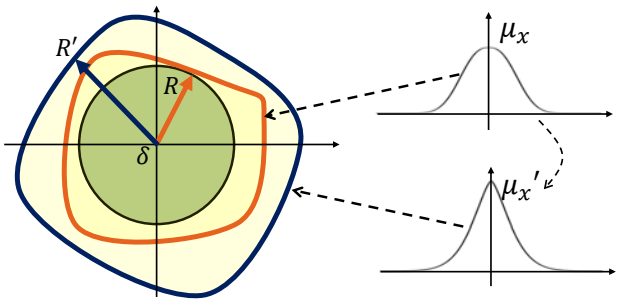


Fig. 2.5: An illustration to noise PDF optimization (take ℓ_2-norm perturbation as an example). The noise distribution is tuned from μ_x to μ_x', which enlarges the robustness boundary. Thus, UniCR can find a larger certified radius R'.

2.2.6 Noise PDF Optimization

All the existing randomized smoothing methods [1, 3, 23, 24] use the same noise for training the smoothed classifier and certifying the robustness of testing inputs. The motivation is that: the training can improve the lower bound of the prediction probability over the the same noise as the certification. Here, the question we ask is: Must we necessarily use the same noise PDF to train the smoothed classifier and derive the certified robustness? Our answer is No.

We study the master optimization problem that uses UniCR as a function to maximize the certified radius by tuning the noise PDF (different randomization), as shown in Figure 2.5. To defend against certain ℓ_p perturbations for a classifier, we consider the noise PDF as a variable (Remember that UniCR can provide a certified radius for each noise PDF), and study the following two master optimization problems with two

different strategies:

1. *Classifier-Input Noise Optimization* ("**C-OPT**"): finding the optimal noise PDF and injecting *the same noise* from this noise PDF into both the training data to train a classifier and testing input to build a smoothed classifier.

2. *Input Noise Optimization* ("**I-OPT**"): Training a classifier with the standard noise (e.g., Gaussian noise), while finding the optimal noise PDF for the testing input and injecting noise from this PDF into the testing input only.

2.2.7 C-OPT and I-OPT

Before optimizing the certified robustness, we need to define metrics for them. First, since I-OPT only optimizes the noise PDF when certifying each testing input, a "better" randomization in I-OPT can be directly indicated by a larger certified radius for a specific input. Second, since C-OPT optimizes the noise PDF for the entire dataset in both training and robustness certification, a new metric for the performance on the entire dataset need to be defined.

Existing works [3, 23] draw several certified accuracy vs. certified radius curves computed by noise with different variances (See Figure 10 in Appendix D.1). These curves represent the certified accuracy at a range of certified radii, where the certified accuracy at radius R is defined as the percent of the testing samples with a derived certified radius larger than R (and correctly predicted by the smoothed classifier). To simply measure the overall performance, we use the area under the curve as an overall

metric to the certified robustness, namely "robustness score". Then, we design the C-OPT method based on this metric. Specifically, the robustness score R_{score} is formally defined as below:

$$R_{score} = \int_0^{+\infty} \max_{\sigma}(Acc_{\sigma}(R))dR, \sigma \in \Sigma, \tag{2.5}$$

where $Acc_{\sigma}(R)$ is the certified accuracy at radius R computed by the noises with variance σ, and Σ is a set of candidate σ.

Notice that our UniCR can automatically approximate the certified radius and compute the robustness score w.r.t. different noise PDFs, thus we can tune the noise PDF towards a better robustness score. From the perspective of optimization, denoting the noise PDF as μ, the C-OPT and the I-OPT problems are defined as $\max_{\mu} R_{score}$ for a classifier and $\max_{\mu} R$ for an input, respectively.

Algorithms for Noise PDF Optimization. We use grid-search in C-OPT to search the best parameters of the noise PDF. We use Hill-Climbing algorithms in I-OPT to find the best parameters of the noise PDF around the noise distribution used in training while maintaining the certified accuracy.

Optimality Validation of Noise PDF. Finding an optimal noise PDF against a specific ℓ_p perturbation is important. Although Gaussian distribution can be used for defending against ℓ_2 perturbations with tight certified radius, there is no evidence showing that Gaussian distribution is the optimal distribution against ℓ_2 perturbations. Our UniCR can also somewhat validate the optimality of using different noise PDFs against different ℓ_p perturbations. For instance, Cohen et al. [1]'s certified radius is tight for

Gaussian noise against ℓ_2 perturbations (see Figure 2.4(b)). However, it is validated as not-optimal distribution against ℓ_2 perturbations in our experiments (see Table 2.2).

2.3 Experiments

In this section, we thoroughly evaluate our UniCR framework, and benchmark with state-of-the-art certified defenses. First, we evaluate the *universality* of UniCR by approximating the certified radii w.r.t. the probability p_A using a variety of noise PDFs against ℓ_1, ℓ_2 and ℓ_∞ perturbations. Second, we validate the certified radii in existing works (results have been discussed and shown in Section 2.2.3). Third, we evaluate our noise PDF optimization on three real-world datasets. Finally, we compare our best certified accuracy on CIFAR10 [28] and ImageNet [29] with the state-of-the-art methods.

2.3.1 Experimental Setting

Datasets. We evaluate our performance on MNIST [27], CIFAR10 [28] and ImageNet [29], which are common datasets to evaluate the certified robustness.

Metrics. We use certified accuracy[1] and the robustness score (Equation (5)) to evaluate the performance of proposed methods.

Experimental Environment. All the experiments were performed on the NSF Chameleon Cluster [35] with Intel(R) Xeon(R) Gold 6126 2.60GHz CPUs, 192G RAM, and NVIDIA Quadro RTX 6000 GPUs.

2.3.2 Universality Evaluation

As randomized smoothing derives certified robustness for any input and any classifier, our evaluation targets "any noise PDF" and "any ℓ_p perubations".

The certified radii of some noise PDFs, e.g., Gaussian noise against ℓ_2 perturbations [1], Laplace noise against ℓ_1 perturbations [24], Pareto noise against ℓ_1 perturbations [3], have been derived. These distributions have been verified by our UniCR framework in Figure 2.4, where our certified radii highly approximate these theoretical radii. However, there are numerous noise PDFs of which the certified radii have not been theoretically studied, or they are difficult to derive. It is important to derive the certified radii of these distributions in order to find the optimal PDF against each of the ℓ_p perturbations. Therefore, we use our UniCR to approximately compute the certified radii of numerous distributions (including some mixture distributions, see Table 7 in Appendix D.3), some of which have not been studied before. Specifically, we evaluate different noise PDFs with the same variance, i.e., $\sigma = \mathbb{E}_{\epsilon\sim\mu}[\sqrt{\frac{1}{d}\|\epsilon\|_2^2}] = 1$. For those PDFs with multiple parameters, we set β as 1.5, 1.0 and 0.5 for General Normal, Pareto, and mixture distributions, respectively. Following Cohen et al. [1], and Yang et al. [3], we consider the binary case (Theorem 3) and only compute the certified radius when $p_A \in (0.5, 1.0]$.

In Figure 2.6-2.8, we plot the R-p_A curves for the noise distributions listed in Table 7 in Appendix D.3 against ℓ_1, ℓ_2 and ℓ_∞ perturbations. Specifically, we present the ℓ_∞ radius scaled by $\times 255$ to be consistent with the existing works [23]. We observe that

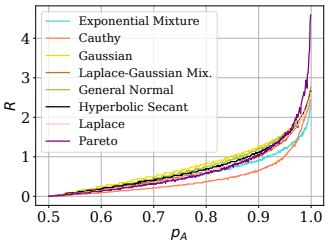


Fig. 2.6: R-p_A curve vs. ℓ_1

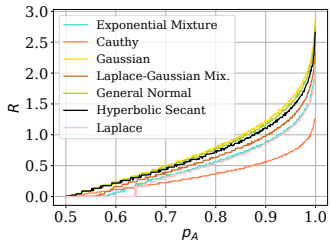


Fig. 2.7: R-p_A curve vs. ℓ_2

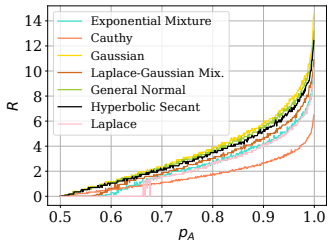


Fig. 2.8: R-p_A curve vs. ℓ_∞

for all ℓ_p perturbations, the Gaussian noise generates the largest certified radius for most of the p_A values. All the noise distribution has very close R-p_A curves except the Cauthy distribution. We also notice that when p_A is low against ℓ_2 and ℓ_∞ perturbations, our UniCR cannot find the certified radius for the Laplace-based distributions, e.g., Laplace distribution, and Gaussian-Laplace mixture distribution. This matches the findings on injecting Laplace noises for certified robustness in Yang et al. [3]—The certified radii for Laplace noise against ℓ_2 and ℓ_∞ perturbations are difficult to derive.

We also conduct experiments to illustrate UniCR's universality in deriving ℓ_p norm certified radius for any real number $p > 0$ in Appendix D.5. Besides, we also conduct fine-grained evaluations on General Normal, Laplace-Gaussian Mixture, and Exponential Mixture noises with various β parameters (See Figure 13 in Appendix D.6), and we can draw similar observations from such results.

2.3.3 Optimizing Certified Radius with C-OPT

We next show how C-OPT uses UniCR to improve the certification against any ℓ_p perturbations. Recall that tight certified radii against ℓ_1 and ℓ_2 perturbations can be derived

Table 2.2: Classifier-input noise optimization (C-OPT). We show the Robustness Score w.r.t. different β settings of General Normal distribution ($\propto e^{-|x/\alpha|^\beta}$). The σ is set to 1.0 for all distributions by adjusting the α parameter in General Normal.

β	0.25	0.50	0.75	1.00	1.25	1.50	1.75	2.00	2.25	2.50	2.75	3	4.00	5.00
vs. ℓ_1	1.8999	2.6136	**2.8354**	2.7448	2.5461	2.4254	2.3434	2.2615	2.2211	2.1730	2.1081	2.0679	1.9610	1.8925
vs. ℓ_2	0.0000	0.0003	1.0373	1.5954	1.9255	2.0882	2.1746	2.1983	**2.2081**	2.1771	2.1184	2.0655	1.8857	1.7296
vs. ℓ_∞	0.0000	0.0109	0.0420	0.0641	0.0771	0.0839	0.0871	0.0879	**0.0880**	0.0870	0.0847	0.0825	0.0758	0.0693

by the Laplace [24] and Gaussian [1] noises, respectively. However, there does not exist any theoretical study showing that Laplace and Gaussian noises are the optimal noises against ℓ_1 and ℓ_2 perturbations, respectively. [3, 23] have identified that there exists other better noise for ℓ_1 and ℓ_2 perturbations. Therefore, we use our C-OPT to explore the optimal distribution for each ℓ_p perturbation. Since the commonly used noise, e.g., Laplace and Gaussian noises, are only special cases of the General Normal Distribution ($\propto e^{-|x/\alpha|^\beta}$), we will find the optimal parameters α and β that generate the best noises for maximizing certified radius against each ℓ_p perturbation.

In the experiments, we use the grid search method to search the best parameters. We choose β as the main parameter, and α will be set to satisfy $\sigma = 1$. Specifically, we evaluate C-OPT on the MNIST dataset, where we train a model on the training set for each round of the grid search and certify $1,000$ images in the test set. Specifically, for each pair of parameters α and β in the grid search, we train a Multiple Layer Perception on MNIST with the smoothing noise. Then, we compute the robustness score over a set

of $\sigma = [0.12, 0.25, 0.50, 1.00]$. When approximating the certified radius with UniCR, we set the sampling number as $1,000$ in the Monte Carlo method. The results are shown in Table 2.2.

We observe that the best β for ℓ_1-norm is 0.75 in the grid search. It indicates that the Laplace noise ($\beta = 1$) is not the optimal noise against ℓ_1 perturbations. A slightly smaller β can provide a better trade-off between the certified radius and accuracy (measured by the robustness score). When $\beta < 1.0$, the radius is observed to be larger than the radius derived with Laplace noise at $p_A \approx 1$ (see Figure 13(a)). Since p_A on MNIST is always high, the noise distribution with $\beta = 0.75$ will give a larger radius at most cases. Furthermore, we observe that the best performance against ℓ_2 and ℓ_∞ are given by $\beta = 2.25$, showing that the Gaussian noise is not the optimal noise against ℓ_2 and ℓ_∞ perturbations, either.

2.3.4 Optimizing Certified Radius with I-OPT

The optimal noises for different inputs are different. We customize the noise for each input using the I-OPT. Specifically, we adapt the hyper-parameters in the noise PDF to find the optimal noise distribution for each input (the classifier is smoothed by a standard method such as Cohen's [1]).

We perform I-OPT for noise PDF optimization with a Gaussian-trained ResNet50 classifier ($\sigma = 1$) on ImageNet. We compare our derived radius with the theoretical radius in [1, 3]. We use the General Normal distribution to generate the noise for

Table 2.3: Average Certified Radius with Input Noise Optimization (I-OPT) against ℓ_1, ℓ_2 and ℓ_∞ perturbations on ImageNet.

Top ℓ_1 radius	20%	40%	60%	80%	100%
Yang's Gaussian [3]	**2.44**	2.10	1.59	1.19	0.95
Ours with I-OPT	2.36	**2.11**	**1.64**	**1.23**	**0.98**
Top ℓ_2 radius	20%	40%	60%	80%	100%
Cohen's Gaussian [1]	**2.43**	2.10	1.58	1.19	0.95
Ours with I-OPT	2.36	**2.11**	**1.64**	**1.23**	**0.98**
Top ℓ_∞ radius $\times 255$	20%	40%	60%	80%	100%
Yang's Gaussian [3]	1.60	1.38	1.04	0.78	0.63
Ours with I-OPT	**1.75**	**1.54**	**1.20**	**0.90**	**0.72**

input certification since it provides a new parameter dimension for tuning, and tune the parameters α and β in $e^{-|x/\alpha|^\beta}$. The Gaussian distribution is only a specific case of the General Normal distribution with $\beta = 2$. In the two baselines [1, 3], they set $\sigma = 1$ and $\beta = 2$, respectively. In the I-OPT, we initialize the noise with the same setting, but optimize the noise for each input. The Monte Carlo sample is set to $1,000$ for ImageNet.

Table 2.3 presents the average values of the top 20%-100% certified radius (the higher the better). It shows that our I-OPT significantly improves the certified radius over the tight certified radius since it provides a personalized noise optimization to each input.

2.3.5 Best Performance Comparison

In this section, we compare our best performance with the state-of-the-art certified defense methods on the CIFAR10 and ImageNet datasets. Following the setting in [1], we use a ResNet110 [36] classifier for the CIFAR10 dataset and a ResNet50 [36] classifier for the ImageNet dataset. We evaluate the certification performance with the noise PDF of a range of variances σ. The σ is set to vary in $[0.12, 0.25, 0.5, 1.0]$ for CIFAR10 and $[0.25, 0.5, 1.0]$ for ImageNet. We also present the Robustness Score based on this set of variances. We use the General Normal distribution and perform the I-OPT. The distribution is initialized with the same setting in the baselines, e.g., $\beta = 1$ (or 2) for Laplace (Gaussian) baseline. We benchmark it with the Laplace noise [24] on CIFAR10 when against ℓ_1 perturbations; and the Gaussian noise [1, 3] on both CIFAR10 and ImageNet against all ℓ_p perturbations. For both our method and baselines, we use $1,000$ and $4,000$ Monte Carlo samples on ImageNet and CIFAR10, respectively, due to different scales, and the certified accuracy is computed over the certified radius of 500 images randomly chosen in the test set for both CIFAR10 and ImageNet.

The results are shown in Table 2.4 and 2.5. Both on CIFAR10 and ImageNet, we observe a significant improvement on the certified accuracy and robustness score. Specifically, on CIFAR10, our robustness score outperforms the state-of-the-arts by 25.34%, 32.44% and 38.78% against ℓ_1, ℓ_2 and ℓ_∞ perturbations, respectively. On ImageNet, our robustness score outperforms the state-of-the-arts by 8.75%, 4.55% and 14.81% against ℓ_1, ℓ_2 and ℓ_∞ perturbations, respectively.

Table 2.4: Certified accuracy and robustness score against ℓ_1, ℓ_2 and ℓ_∞ perturbations on CIFAR10. Ours: General Normal with I-OPT.

ℓ_1 radius	0.50	1.00	1.50	2.00	2.50	R_{score}
Teng's Laplace [24]	39.2	17.2	10.0	6.0	2.8	0.5606
Ours	**45.8**	**22.4**	**14.8**	**8.2**	**3.6**	**0.7027**
ℓ_2 radius	0.50	1.00	1.50	2.00	2.50	R_{score}
Cohen's Gaussian [1]	38.6	17.4	8.6	3.4	1.6	0.5392
Ours	**48.4**	**26.8**	**16.6**	**6.8**	**2.0**	**0.7141**
ℓ_∞ radius	$\frac{2}{255}$	$\frac{4}{255}$	$\frac{6}{255}$	$\frac{8}{255}$	$\frac{10}{255}$	R_{score}
Yang's Gaussian [3]	43.6	21.8	10.8	5.6	2.6	0.0098
Ours	**53.4**	**30.4**	**21.2**	**13.2**	**5.6**	**0.0136**

Table 2.5: Certified accuracy and robustness score against ℓ_1, ℓ_2 and ℓ_∞ perturbations on ImageNet (Teng's Laplace [24] is not available). Ours: General Normal with I-OPT.

ℓ_1 radius	0.50	1.00	1.50	2.00	2.50	R_{score}
Yang's Gaussian [3]	58.8	45.6	34.6	27.0	0.0	1.0469
Ours	**63.4**	**49.6**	**36.8**	**29.6**	**6.6**	**1.1385**
ℓ_2 radius	0.50	1.00	1.50	2.00	2.50	R_{score}
Cohen's Gaussian [1]	58.8	44.2	34.0	27.0	0.0	1.0463
Ours	**62.6**	**49.0**	**36.6**	**28.6**	**2.0**	**1.0939**
ℓ_∞ radius	$\frac{0.25}{255}$	$\frac{0.50}{255}$	$\frac{0.75}{255}$	$\frac{1.00}{255}$	$\frac{1.25}{255}$	R_{score}
Yang's Gaussian [3]	63.6	52.4	39.8	34.2	28.0	0.0027
Ours	**69.2**	**57.4**	**47.2**	**38.2**	**33.0**	**0.0031**

2.4 Related Work

Certified Defenses. Existing certified defenses methods can be classified into leveraging Satisfiability Modulo Theories [13, 14, 37, 38], mixed integer-linear programming [15, 39, 40], linear programming [12, 41], semidefinite programming [17, 42], dual optimization [43, 44], global/local Lipschitz constant methods [16, 45, 46, 47, 48], abstract interpretation [19, 49, 50], and layer-wise certification [19, 50, 51, 52, 53], etc. However, none of these methods is able to scale to large models (e.g., deep neural networks) or is limited to specific type of network architecture, e.g., ReLU based networks. Randomized smoothing was recently proposed certified defenses [1, 21, 22] that is scalable to large models and applicable to arbitrary classifiers. Lecuyer et al. [22] proposed the first randomized smoothing-based certified defense via differential privacy [34]. Li et al. [21] proposed a stronger guarantee for Gaussian noise using information theory. The first tight robustness guarantee against l_2-norm perturbation for Gaussian noise was developed by Cohen et al. [1]. After that, a series follow-up works have been proposed for other ℓ_p-norms, e.g., ℓ_1-norm [24], ℓ_0-norm [54], etc. However, all these methods are limited to guarantee the robustness against only a specific ℓ_p-norm perturbation.

Universal Certified Defenses. More recently, several works [3, 23] aim to provide more universal certified robustness schemes for all ℓ_p-norms. Yang et al. [3] proposed a level set method and a differential method to derive the upper bound and lower bound of the certified radius, while the derivation is relying on the case-by-case theoretical analysis. Zhang et al. [23] proposed a black-box optimization scheme that automati-

cally computes the certified radius, but the solvable distribution is limited to ℓ_p-norm-constrained. Our UniCR framework can automate the robustness certification for any classifier against any l_p-norm perturbation with any noise PDF.

Certified Defenses with Optimized Noise PDFs/Distributions. Yang et al. [3] proposed to use the Wulff Crystal theory [55] to find optimal noise distributions. Zhang et al. [23] claimed that the optimal noise should have a more central-concentrated distribution from the optimization perspective. However, no existing works provide quantitative solutions to find optimal noise distributions. We propose the **C-Opt** and **I-Opt** schemes to quantitatively optimize the noisy PDF in our UniCR framework and provide better certified robustness. Table 2.1 summarizes the differences in all the closely-related works.

2.5 Conclusion

Randomize smoothing has achieved great success in certifying the adversarial robustness. However, the state-of-the-art methods lack universality to certify robustness. We propose the first randomized smoothing-based universal certified robustness approximation framework against any ℓ_p perturbations with any continuous noise PDF. Extensive evaluations on multiple image datasets demonstrate the effectiveness of our UniCR framework and its advantages over the state-of-the-art certified defenses against any ℓ_p perturbations.

Chapter 3

Towards Strong Certified Defense with Universal Asymmetric Randomization

Randomized smoothing has become essential for achieving certified adversarial robustness in machine learning models. However, current methods primarily use isotropic noise distributions that are uniform across all data dimensions, such as image pixels, limiting the effectiveness of robustness certification by ignoring the heterogeneity of inputs and data dimensions. To address this limitation, we propose UCAN: a novel technique that Universally Certifies adversarial robustness with Anisotropic Noise. UCAN is designed to enhance any existing randomized smoothing method, transforming it from symmetric (isotropic) to asymmetric (anisotropic) noise distributions, thereby offering a more tailored defense against adversarial attacks. Our theoretical framework is versatile, supporting a wide array of noise distributions for certified robustness in different ℓ_p-norms and applicable to any arbitrary classifier by guaranteeing the classifier's prediction over perturbed inputs with provable robustness bounds through tailored noise injection. Additionally, we develop a novel framework equipped with three exemplary

noise parameter generators (NPGs) to optimally fine-tune the anisotropic noise parameters for different data dimensions, allowing for pursuing different levels of robustness enhancements in practical settings. Furthermore, we introduce a generalized metric to accurately measure the certified robustness of both isotropic and anisotropic randomized smoothing, addressing the inadequacy of the traditional "certified radius" in the anisotropic context. Empirical evaluations underscore the significant leap in UCAN's performance over existing state-of-the-art methods, demonstrating up to 182.6% improvement in certified accuracy at large certified radii on MNIST, CIFAR10, and ImageNet datasets.

3.1 Introduction

Deep learning models have demonstrated remarkable performance across a wide range of applications. However, they are notoriously vulnerable to adversarial perturbations—carefully crafted minor modifications to inputs can lead to severe misclassification or misrecognition [8, 56]. These adversarial examples pose significant threats in real-world scenarios, such as autonomous driving [57], medical diagnosis [58], and face recognition systems [59], where even small prediction errors can result in catastrophic consequences.

To mitigate these vulnerabilities, robust defense mechanisms with certified guarantees are highly desirable. While empirical defense methods, including adversarial training [7, 60], perturbation destruction [61, 62], and feature regularization [63, 64],

have shown promise, they often fall short against adaptive and stronger adversaries [31, 65, 66]. These methods typically cannot provide formal guarantees of robustness, as their defenses can be broken by more sophisticated attacks.

In contrast, certified robustness methods [1, 12, 22] offer provable guarantees by ensuring that no adversarial perturbation within a specified boundary—typically defined by an ℓ_p-norm ball (e.g., ℓ_1, ℓ_2, or ℓ_∞)—can alter the model's prediction. Among these, randomized smoothing has emerged as a state-of-the-art (SOTA) technique due to its universal applicability to any arbitrary classifier [1, 22, 24]. By injecting noise to data during both the training and inference phases, randomized smoothing transforms any classifier into a smoothed classifier with certified robustness guarantees.

Despite its success, existing randomized smoothing methods predominantly rely on isotropic noise distributions, where the same noise parameters are uniformly applied across all data dimensions (e.g., all pixels in an image). This uniform approach overlooks the inherent heterogeneity of different data dimensions, potentially limiting the effectiveness of robustness certification. For instance, using identical noise for all pixels might not be optimal, as certain pixels may be more critical to the model's decision-making process than others. Consequently, the *accuracy* of the smoothed classifier on both perturbed and clean inputs may be compromised, and the *certified radius* based on the ℓ_p-norm ball might not fully capture the true robustness potential. In reality, the ℓ_p-norm ball serves as a *sufficient but not necessary* condition for certification, as some regions outside the ℓ_p-norm ball can still maintain robustness.

To address these limitations, we introduce UCAN: a novel technique that Universally Certifies adversarial robustness with Anisotropic Noise. UCAN significantly enhances any existing randomized smoothing method by transitioning from symmetric (isotropic) to asymmetric (anisotropic) noise distributions. This tailored noise injection allows for more effective and adaptive defenses against adversarial attacks by assigning different noise parameters to various data dimensions based on their specific characteristics, importance, and vulnerability.

Developing UCAN involves overcoming two primary and complex challenges:

1) **Universal Certification Guarantee**: Establishing a robust theoretical foundation that universally supports various anisotropic noise distributions. This framework must ensure strict and sound certified robustness guarantees when different means and variances are assigned to different data dimensions.

2) **Optimal Noise Parameterization**: Designing mechanisms to derive appropriate means and variances for generating anisotropic noise tailored to various data dimensions. This ensures maximal certification performance without compromising robustness.

To tackle these challenges, UCAN offers the following key contributions:

1) **Universal Certification Theory for Anisotropic Noise**. We present a universal theory for certifying the robustness of randomized smoothing with any anisotropic noise distribution. This theory seamlessly transforms existing certifications based

on isotropic noise into those with anisotropic noise, supporting various ℓ_p-norm perturbations (e.g., ℓ_1, ℓ_2, ℓ_∞) and ensuring sound certified robustness across different noise distributions.

2) **Customizable Anisotropic Noise Parameter Generators (NPGs)**. We design three distinct NPGs, including two novel neural network-based methods, to efficiently generate the element-wise hyper-parameters (mean and variance) of the anisotropic noise distributions for all data dimensions (e.g., image pixels). These NPGs provide different efficiency-optimality trade-offs, significantly amplifying the certification performance while ensuring certified robustness.

3) **Significantly Boosted Certification Performance**. Empirical evaluations on benchmark datasets (MNIST, CIFAR10, and ImageNet) demonstrate that UCAN drastically outperforms SOTA randomized smoothing-based certified robustness methods. Specifically, UCAN achieves up to 182.6% improvement in certified accuracy at large certified radii compared to existing methods.

In summary, UCAN represents a significant advancement in certified robustness by introducing anisotropic noise into randomized smoothing. This approach not only enhances theoretical robustness guarantees but also provides practical mechanisms for optimizing noise parameters, leading to substantial improvements in certification across various datasets.

The remainder of this paper is organized as follows. Section 3.2 introduces

the preliminaries, including the threat model and the isotropic randomized smoothing method for anisotropic certified robustness. Section 3.3 presents the proposed universal theory and metrics for robustness region. Section 3.4 details the three different methods for customizing anisotropic noise to enhance certification performance. Section 3.5 discusses and proves the soundness of the certification-wise anisotropic noise. Section 3.6 provides experimental results demonstrating UCAN's superior performance. Finally, Sections 3.7 and 3.8 discuss related work and conclude the paper, respectively.

3.2 Preliminary

In this section, we provide a brief overview of randomized smoothing with isotropic noise for certified robustness, which forms the foundation for our proposed UCAN framework.

3.2.1 Randomized Smoothing and Certified Robustness

Randomized smoothing is a technique that constructs a smoothed classifier from an arbitrary base classifier by adding random noise to the inputs. The smoothed classifier inherits certain robustness properties, allowing for certified guarantees against adversarial perturbations within a specific norm ball.

Consider a classification task where inputs $x \in \mathbb{R}^d$ are mapped to labels in a finite set of classes C. Given any base classifier $f : \mathbb{R}^d \to C$, randomized smoothing defines a smoothed classifier g that, for any input x, outputs the class most likely to be

predicted by f when noise is added to x. Specifically, let $\epsilon \in \mathbb{R}^d$ be noise drawn from an arbitrary isotropic probability distribution ϕ (e.g., $\mathcal{N}(0, 1)$), and define the random variable $X = x + \epsilon$. The smoothed classifier g is then defined as:

$$g(x) = \arg\max_{c \in \mathcal{C}} \mathbb{P}(f(X) = c) \tag{3.1}$$

3.2.2 Certified Robustness via Isotropic Randomized Smoothing

Randomized smoothing provides a way to certify the robustness of the smoothed classifier g against adversarial perturbations measured in terms of the ℓ_p-norm. The core idea is that if the smoothed classifier predicts a class c_A with high probability, and all other classes have significantly lower probabilities, then small perturbations to the input will not change the predicted class. We summarize existing results on certified robustness via randomized smoothing with isotropic noise in the following unified theorem.

Theorem 2 (**Certified Robustness via Randomized Smoothing with Isotropic Noise**). *Let $f : \mathbb{R}^d \to \mathcal{C}$ be any (possibly randomized) base classifier, and let ϕ be an isotropic probability distribution used to generate noise $\epsilon \in \mathbb{R}^d$. Define the smoothed classifier g as in Equation* (3.1)*. For a specific input $x \in \mathbb{R}^d$, let $X = x + \epsilon$, and suppose that there exists a class $c_A \in \mathcal{C}$ and bounds $\underline{p_A}, \overline{p_B} \in [0, 1]$ such that:*

$$\mathbb{P}(f(X) = c_A) \geq \underline{p_A} \geq \overline{p_B} \geq \max_{c \neq c_A} \mathbb{P}(f(X) = c) \tag{3.2}$$

Then, for any perturbation $\delta \in \mathbb{R}^d$ where $\|\delta\|_p < R(\underline{p_A}, \overline{p_B})$, the smoothed classi-

fier g will consistently predict class c_A at $x + \delta$, i.e., $g(x + \delta) = c_A$. Here, $\|\cdot\|_p$ denotes the ℓ_p-norm (with $p = 1, 2, \infty$), and $R(\underline{p_A}, \overline{p_B})$ is the certified radius, which depends on the noise distribution ϕ and the norm ℓ_p.

The certified radius $R(\underline{p_A}, \overline{p_B})$ quantifies the robustness of the smoothed classifier g around the input x. It ensures that any adversarial perturbation δ with $\|\delta\|_p < R(\underline{p_A}, \overline{p_B})$ cannot change the predicted class. The exact form of R varies depending on the noise distribution ϕ and the norm ℓ_p. For example, when ϕ is an isotropic Gaussian distribution with standard deviation σ, and the ℓ_2 norm is considered, the certified radius is given by [1]:

$$R = \frac{\sigma}{2}\left(\Phi^{-1}(\underline{p_A}) - \Phi^{-1}(\overline{p_B})\right) \tag{3.3}$$

where Φ^{-1} is the inverse cumulative distribution function (CDF) of the standard normal distribution.

3.2.3 Limitations of Isotropic Noise in Randomized Smoothing

While randomized smoothing with isotropic noise provides a powerful tool for certified robustness, it applies the same noise distribution uniformly across all data dimensions. This isotropic approach may not fully exploit the potential robustness, as it ignores the heterogeneity and varying importance of different input features. Certain dimensions (e.g., pixels in an image) may be more sensitive or critical to the classification task, and treating them uniformly can limit the defense performance.

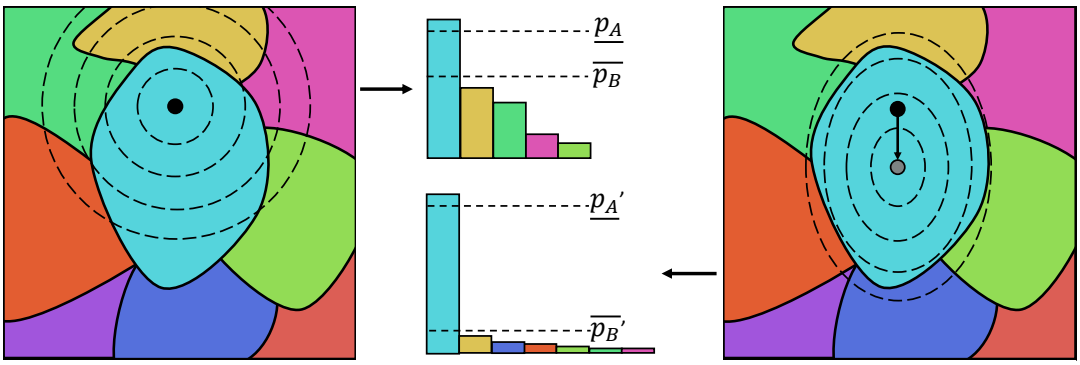


Fig. 3.1: Anisotropic noise vs isotropic noise. The decision regions of f are denoted in different colors. The dashed lines are the level sets of the noise distribution. The left figure shows the RS with isotropic Gaussian noise $\mathcal{N}(0, \lambda^2\mathbf{I})$ in [1] whereas the right figure shows the RS with anisotropic Gaussian noise $\mathcal{N}(\mu, \Sigma)$, where $\Sigma = \lambda^2 diag(\sigma_1^2, \sigma_2^2, ..., \sigma_d^2)$. Certification can be improved by enlarging the gap between $\underline{p_A}$ and $\overline{p_B}$.

To address these limitations, our work extends randomized smoothing to anisotropic noise distributions, where different noise parameters can be assigned to different data dimensions. This extension poses significant challenges in developing universal theoretical guarantees and in designing efficient methods to optimize the noise parameters. In the following sections, we present our proposed UCAN framework, which overcomes these challenges to enhance certified robustness.

3.3 UCAN: Theorem and Metrics for Certified Region

In this section, we establish a universal theory for the certification via randomized smoothing with *anisotropic* noise. Given any isotropic randomized smoothing methods, our method can universally transform them to anisotropic randomized smoothing for enhanced certified robustness.

3.3.1 General Anisotropic Noise

Given an arbitrary isotropic noise ϵ, we define the corresponding anisotropic noise ϵ' as:

$$\epsilon' = \epsilon^\top \Sigma + \mu \tag{3.4}$$

where $\Sigma \in \mathbb{R}^{d \times d}$ is an *invertible* covariance matrix, and $\mu \in \mathbb{R}^d$ is the mean offset vector. The covariance matrix Σ introduces dependencies between different dimensions of the noise, capturing potential correlations, while μ allows for mean shifts in the noise distribution. See Figure 3.8 for examples.

Then, certified robustness with anisotropic noise can be ensured per Theorem 3.

Theorem 3 (**Asymmetric Randomized Smoothing via Universal Transformation**)**.** *Let $f : \mathbb{R}^d \to C$ be any deterministic or randomized function. Suppose that for the multivariate random variable with isotropic noise $X = x + \epsilon$ in Theorem 2, the certified radius function is $R(\cdot)$. Then, for the corresponding anisotropic input $Y = x + \epsilon^\top \Sigma + \mu$,*

if there exist $c'_A \in C$ *and* $\underline{p_A}', \overline{p_B}' \in [0, 1]$ *such that:*

$$\mathbb{P}\left(f(Y) = c'_A\right) \geq \underline{p_A}' \geq \overline{p_B}' \geq \max_{c \neq c'_A} \mathbb{P}\left(f(Y) = c\right) \tag{3.5}$$

then for the anisotropic smoothed classifier $g'(x + \delta') \equiv \arg\max_{c \in C} \mathbb{P}\left(f(Y + \delta') = c\right)$*, we can guarantee* $g'(x + \delta') = c'_A$ *for all perturbations* $\delta' \in \mathbb{R}^d$ *such that:*

$$\left\|\Sigma^{-1}\delta'\right\|_p \leq R\left(\underline{p_A}', \overline{p_B}'\right) \tag{3.6}$$

provided that Σ *is invertible.*

Proof. See the detailed proof in Appendix 7.7. □

The general anisotropic noise with covariance allows for capturing correlations between different dimensions of the input data. However, adopting this generalized anisotropic noise introduces certain practical limitations: 1) **Invertibility of** Σ: Our theory requires the covariance matrix Σ to be invertible. If Σ is not invertible, a small regularization term can be added to its diagonal to ensure invertibility and retain the validity of our certification guarantees. 2) **Computational Complexity**: Optimizing or learning a full covariance matrix $\Sigma \in \mathbb{R}^{d \times d}$ is computationally intensive, especially for high-dimensional data (e.g., images with $d = 150{,}528$ for ImageNet). The memory and computational requirements scale quadratically with the input dimension, making it impractical for large-scale applications.

Given these limitations, in practice, one might consider structured covariance matrices that balance expressiveness with computational efficiency, such as low-rank

Table 3.1: Certified radii (binary-case) for randomized smoothing with independent isotropic and anisotropic noise. d is the dimension size. Φ^{-1} is the inverse CDF of Gaussian distribution. λ is the scalar parameter of the isotropic noise. σ is the anisotropic scale multiplier.

Distribution	PDF	Adv.	Isotropic Guarantee	ℓ_p Radius for Iso. RS	Anisotropic Guarantee	ℓ_p Radius for Ani. RS
Gaussian [1]	$\propto e^{-\|\frac{z}{2}\|_2^2}$	ℓ_2	$\|\delta\|_2 \leq \lambda(\Phi^{-1}(\underline{p_A}))$	$\lambda(\Phi^{-1}(\underline{p_A}))$	$\|\delta \oslash \sigma\|_2 \leq \lambda(\Phi^{-1}(\underline{p_A}'))$	$\min\{\sigma\}\lambda(\Phi^{-1}(\underline{p_A}))$
Gaussian [3]	$\propto e^{-\|\frac{z}{2}\|_2^2}$	ℓ_1	$\|\delta\|_1 \leq \lambda(\Phi^{-1}(\underline{p_A}))$	$\lambda(\Phi^{-1}(\underline{p_A}))$	$\|\delta \oslash \sigma\|_1 \leq \lambda(\Phi^{-1}(\underline{p_A}'))$	$\min\{\sigma\}\lambda(\Phi^{-1}(\underline{p_A}))$
		ℓ_∞	$\|\delta\|_\infty \leq \lambda(\Phi^{-1}(\underline{p_A}))/\sqrt{d}$	$\lambda(\Phi^{-1}(\underline{p_A}))/\sqrt{d}$	$\|\delta \oslash \sigma\|_\infty \leq \lambda(\Phi^{-1}(\underline{p_A}'))/\sqrt{d}$	$\min\{\sigma\}\lambda(\Phi^{-1}(\underline{p_A}))/\sqrt{d}$
Laplace [24]	$\propto e^{-\|\frac{z}{2}\|_1}$	ℓ_1	$\|\delta\|_1 \leq -\lambda\log(2(1-\underline{p_A}))$	$-\lambda\log(2(1-\underline{p_A}))$	$\|\delta \oslash \sigma\|_1 \leq -\lambda\log(2(1-\underline{p_A}'))$	$-\min\{\sigma\}\lambda\log(2(1-\underline{p_A}))$
Exp. ℓ_∞ [3]	$\propto e^{-\|\frac{z}{2}\|_\infty}$	ℓ_1	$\|\delta\|_1 \leq 2d\lambda(\underline{p_A}-\frac{1}{2})$	$2d\lambda(\underline{p_A}-\frac{1}{2})$	$\|\delta \oslash \sigma\|_1 \leq 2d\lambda(\underline{p_A}'-\frac{1}{2})$	$2\min\{\sigma\}d\lambda(\underline{p_A}-\frac{1}{2})$
		ℓ_∞	$\|\delta\|_\infty \leq \lambda\log(\frac{1}{2(1-\underline{p_A})})$	$\lambda\log(\frac{1}{2(1-\underline{p_A})})$	$\|\delta \oslash \sigma\|_\infty \leq \lambda\log(\frac{1}{2(1-\underline{p_A}')})$	$\min\{\sigma\}\lambda\log(\frac{1}{2(1-\underline{p_A})})$
Uniform ℓ_∞ [67]	$\propto \mathbb{I}(\|z\|_\infty \leq \lambda)$	ℓ_1	$\|\delta\|_1 \leq 2\lambda(\underline{p_A}-\frac{1}{2})$	$2\lambda(\underline{p_A}-\frac{1}{2})$	$\|\delta \oslash \sigma\|_1 \leq 2\lambda(\underline{p_A}'-\frac{1}{2})$	$2\min\{\sigma\}\lambda(\underline{p_A}-\frac{1}{2})$
		ℓ_∞	$\|\delta\|_\infty \leq 2\lambda(1-\sqrt[d]{\frac{3}{2}-\underline{p_A}})$	$2\lambda(1-\sqrt[d]{\frac{3}{2}-\underline{p_A}})$	$\|\delta \oslash \sigma\|_\infty \leq 2\lambda(1-\sqrt[d]{\frac{3}{2}-\underline{p_A}'})$	$2\min\{\sigma\}\lambda(1-\sqrt[d]{\frac{3}{2}-\underline{p_A}})$
Power Law ℓ_∞ [3]	$\propto \frac{1}{(1+\|\frac{z}{2}\|_\infty)^a}$	ℓ_1	$\|\delta\|_1 \leq \frac{2d\lambda}{a-d}(\underline{p_A}-\frac{1}{2})$	$\frac{2d\lambda}{a-d}(\underline{p_A}-\frac{1}{2})$	$\|\delta \oslash \sigma\|_1 \leq \frac{2d\lambda}{a-d}(\underline{p_A}-\frac{1}{2})$	$\min\{\sigma\}\frac{2d\lambda}{a-d}(\underline{p_A}-\frac{1}{2})$

approximations, block-diagonal matrices, or sparse covariance matrices. In this paper, we focus on the independent anisotropic noise where noise parameters are independent along dimensions.

Special Case: Diagonal Covariance Matrix. The case where Σ is a diagonal matrix corresponds to anisotropic noise with independent dimensions (i.e., no covariance between dimensions). Let $\Sigma = \text{diag}(\sigma_1, \sigma_2, \ldots, \sigma_d)$, where $\sigma_i > 0$ for all i. In this case, $\Sigma^{-1} = \text{diag}\left(\frac{1}{\sigma_1}, \frac{1}{\sigma_2}, \ldots, \frac{1}{\sigma_d}\right)$.

Corollary 3.1 (**Asymmetric Randomized Smoothing with Independent Anisotropic Noise**). *Under the same conditions as Theorem 3 if* $\Sigma = \text{diag}(\sigma_1, \sigma_2, \ldots, \sigma_d)$*, then the*

certified robustness guarantee becomes:

$$\|\delta' \oslash \sigma\|_p \leq R\left(\underline{p_A}', \overline{p_B}'\right) \tag{3.7}$$

where $\sigma = [\sigma_1, \sigma_2, \ldots, \sigma_d]^\top$, $\oslash$ *denotes element-wise division, and* $\delta' \in \mathbb{R}^d$ *is the perturbation in the anisotropic space.*

Proof. Since Σ is diagonal, its inverse is also diagonal with entries $1/\sigma_i$. Therefore, we have:

$$\left\|\Sigma^{-1}\delta'\right\|_p = \|\delta' \oslash \sigma\|_p \tag{3.8}$$

Substituting this into Equation (3.6) yields the desired result. □

Theorem 3 provides a robustness guarantee with anisotropic noise, building on traditional isotropic RS theories. Equation (3.6) from this theorem is both explicit and widely applicable, significantly enhancing existing RS frameworks' performance. In Table 3.1, we outline some representative isotropic RS methods and their extension to anisotropic noise via Theorem 3. Our method can also be readily adapted to other RS methods like those in [4, 23] that lack explicit radius certifications, applying it directly to their numerical results. Details on the binary classifier version are discussed in Appendix 7.8.

In Figure 3.1, we clearly illustrate the significant benefits that the anisotropic noise can bring to randomized smoothing. In Theorem 3, we observe that the mean offset μ does not affect the derivation of the certified robustness with anisotropic noise (Equation (3.6)). Thus, it is likely that the gap between probabilities $\underline{p_A}'$ and $\overline{p_B}'$ can

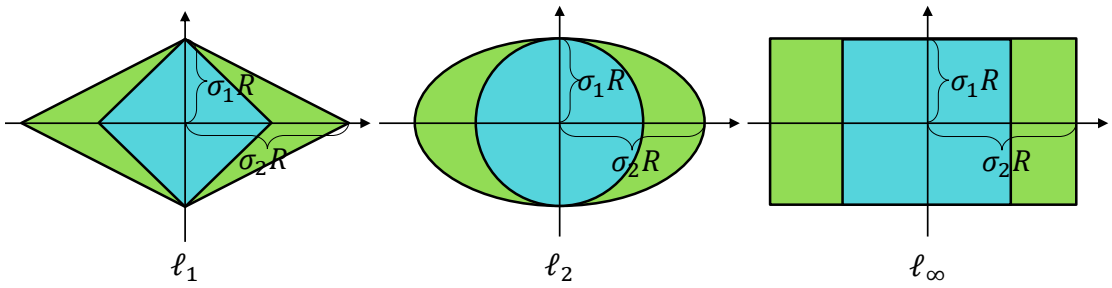


Fig. 3.2: An example illustration of the full certified region (green) and the certified radius (blue), where $\sigma_1 < \sigma_2$.

be improved by a properly chosen mean offset of the anisotropic noise (without affecting the robustness guarantee). Additionally, with the heterogeneous variance, the anisotropic noise can better fit the different dimensions of the input without causing over-distortion.

Corollary 3.2. *For the anisotropic input Y in Theorem 3, if Equation (3.5) is satisfied, then* $g'(x+\delta') \equiv \arg\max_{c\in C} \mathbb{P}(f(Y+\delta')=c) = c'_A$ *for all* $\|\delta'\|_p \leq R'$ *such that*

$$R' = \min\{\sigma\}R \tag{3.9}$$

where R is the certified radius of randomized smoothing via isotropic noise, and $\min\{\cdot\}$ *denotes the minimum entry.*

Proof. The guarantee in Theorem 3 holds for $\|\delta' \oslash \sigma\|_p \leq R$. Since $\|\delta' \oslash \sigma\|_p \leq \|\frac{\delta'}{\min\{\sigma\}}\|_p$, if $\|\frac{\delta'}{\min\{\sigma\}}\|_p \leq R$, the guarantee still holds. This requires $\|\delta'\|_p \leq \min\{\sigma\}R$. □

3.3.2 Certified Region and Two Metrics

In Corollary 3.2, we derive the certified radii R' for asymmetric randomized smoothing in the formation of ℓ_p-ball, i.e., $\|\delta'\|_p \leq R'$. While the certified radius reflects the maximum size of the tolerated perturbation within the ℓ_p-ball. However, especially under asymmetric circumstances, the certified region can be highly asymmetric, and the ℓ_p-ball (which is symmetric) may only represent a subset of the full robustness region, which is given by $(\sum_i^d (\frac{\delta_i'}{\sigma_i})^p)^{\frac{1}{p}} \leq R(\underline{p_A}', \overline{p_B}')$, as illustrated in Figure 3.2. Therefore, to provide a more comprehensive and complementary assessment, we adopt an additional metric to measure the overall size of the certified region.

Radius and ALM for Certified Region. In addition to the ℓ_p radius, we also adopt the Alternative Lebesgue Measure (ALM[1]) for measuring the certified region. Specifically, we consider the certified region under the anisotropic guarantee as a d-dimensional super-ellipsoid, as defined in Definition 1. It is worth noting that the ℓ_p-norm ball is a sufficient but not necessary condition for certified robustness: while robustness within the ℓ_p ball is guaranteed, certain points outside this ball may also remain robust due to the true shape of the certified region. The super-ellipsoid formulation can represent this broader space.

Definition 1 (d-dimensional Generalized Super-ellipsoid of δ). *The d-dimensional generalized super-ellipsoid ball of δ is defined as*

$$S(d,p) = \{(\delta_1, \delta_2, ..., \delta_d) : \sum_{i=1}^{d} |\frac{\delta_i}{\sigma_i R}|^p \leq 1, p > 0\} \tag{3.10}$$

[1] This metric is mathematically equivalent to the "proxy radius" in [68].

Theorem 4 (**Lebesgue Measure of the Robust Perturbation Set** $S(d,p)$)**.** *Define* $S(d,p)$ *per Definition 1, then the Lebesgue measure of the robust perturbation set is given by*

$$V_S(d,p) = \frac{(2R\Gamma(1+\frac{1}{p}))^d \prod_{i=1}^{d} \sigma_i}{\Gamma(1+\frac{d}{p})} \tag{3.11}$$

where Γ *is the Euler gamma function defined in Definition 4 in Appendix 7.9.*

Proof. See the detail proof in Appendix 7.10. □

Recall that we aim to find an auxiliary metric that can measure the volume of this super-ellipsoid, so we derive the Lebesgue Measure of this super-ellipsoid as in Theorem 4 since the Lebesgue Measure quantifies the "volume" of high-dimensional space, and then we simplify it by removing the constants w.r.t. the dimension and p, as well as transforming it to the radius scale. To this end, the ALM can be simplified as: $ALM = \sqrt[d]{\prod_{i=1}^{d} \sigma_i} R$.

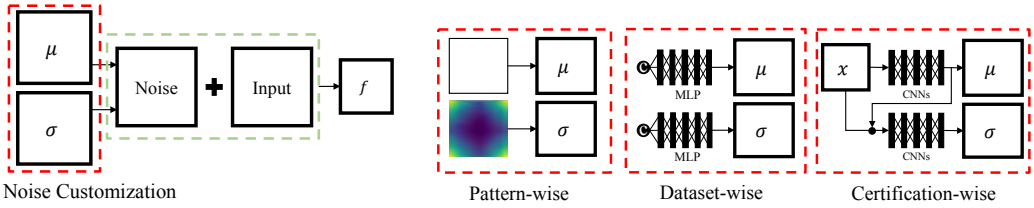


Fig. 3.3: Left: the framework for customizing anisotropic noise. Right: three example noise parameter generators (NPGs).

It serves as an additional measure for evaluating the robustness region. This is particularly useful because the ℓ_p radius (*as a more strict metric*) cannot fully capture the robustness region in certain dimensions due to its inherent symmetry, as illus-

trated in Figure 3.2. In summary, the ℓ_p radius serves as a conservative, worst-direction certificate—being controlled by the smallest anisotropic scale $\min_i \sigma_i$—whereas the ALM provides an auxiliary, volume-oriented characterization of the full certified region through the geometric mean $(\prod_{i=1}^{d} \sigma_i)^{1/d}$. It is worth noting that ALM subsumes the certified radius: in any dimension, the certified radius corresponds to the smallest semi-axis length of the certified region, while ALM (as the geometric mean of all semi-axes times R) is always greater than or equal to this value, and coincides with it in the isotropic case. Therefore, ALM serves as an auxiliary metric for the size of certified region instead of a formal guarantee. See Appendix 7.9 for detailed analysis and discussions for ALM.

3.4 Customizing Anisotropic Noise

Theorem 3 formally guarantees robustness when heterogeneous noise parameters are assigned across different data dimensions. However, finding more optimal heterogeneous noise parameters rather than randomly assigning them remains a challenge. To this end, in UCAN, we design a unified framework to customize anisotropic noise for randomized smoothing (left in Figure 3.3), which includes three *noise parameter generators* (NPGs) with different scales of trainable parameters (right in Figure 3.3) and optimality levels.

Here, the "optimality" refers to the degree to which each NPG can optimize the noise parameters to maximize certified robustness (measured by either certified radius

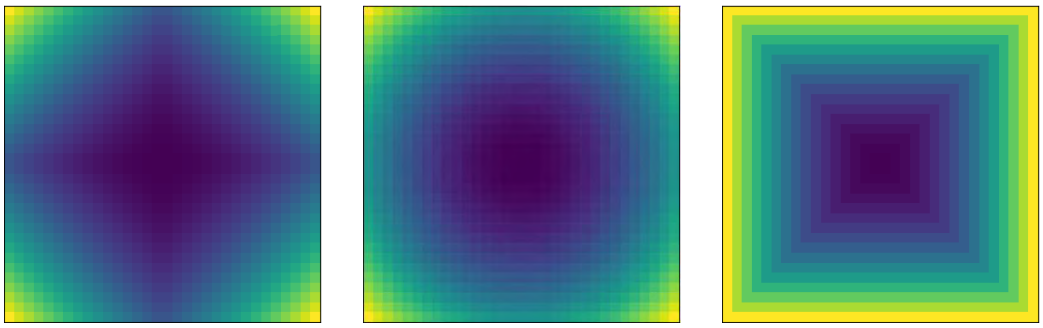
or ALM), while maintaining prediction accuracy.

- **Pattern-wise Anisotropic Noise** (Low optimality): Uses pre-defined patterns for noise variances, offering basic but non-adaptive and relatively lower optimal robustness.

- **Dataset-wise Anisotropic Noise** (Moderate optimality): Learns a global set of noise parameters optimized for the entire dataset, enabling some adaptation for different input data but still derived at the dataset level.

- **Certification-wise Anisotropic Noise** (High optimality): Generates noise parameters specifically tailored to each individual input at the certification time, achieving the most fine-grained and input-specific robustness optimization.

The optimality levels reflect a fundamental trade-off: higher-optimality approaches allow more precise and effective maximization of certified robustness, but require increased computational resources for training and/or inference. These three NPGs are representative examples—other designs are possible depending on application needs. In any case, the covariance matrix should be kept invertible (e.g., by adding a small positive constant to the diagonal if necessary). Implementation details and specific algorithms can be found in Appendix 3.4.4.



$\sigma(a,b) = \|(a,b)\|_1^2 + 1$ $\sigma(a,b) = \|(a,b)\|_2^2 + 1$ $\sigma(a,b) = \|(a,b)\|_\infty^2 + 1$

Fig. 3.4: Spatial distributions for noise variances (pattern-wise).

3.4.1 Pattern-wise Anisotropic Noise

The motivation for pattern-wise anisotropic noise in NPG is based on the understanding that different data regions affect predictions differently [69]. Typically, an image's center contains more critical visual information, requiring lower variance to preserve clarity, whereas the borders may accommodate higher variance without significantly impacting predictions. Thus, this NPG utilizes fixed spatial patterns for anisotropic noise.

Consider a function $\sigma(a,b)$ representing the variance distribution across an image, where (a,b) are the coordinates with the center at $(0,0)$. The variance is intuitively smaller at the center than at the borders due to varying feature importance. We propose three distinct spatial distribution types:

$$\sigma(a,b) = \kappa\|(a,b)\|_p^2 + \iota, \quad p = 1, 2, \infty \tag{3.12}$$

where $\|(a,b)\|_p$ denotes the ℓ_p-norm distance between (a,b) and $(0,0)$, κ is a

constant parameter tuning the overall magnitude of the variance, ι denotes the variance of a center pixel since $\sigma(0,0) = \iota$, and $\iota > 0$ such that $\sigma(a,b) > 0$. Figure 3.4 shows three example spatial distributions when $p = 1, 2, \infty$, where $\kappa = 1$ and $\iota = 1$. The noise mean is set as 0 to avoid unnecessary deviation in the images.

Pattern-wise anisotropic noise modifies predictions based on spatial contributions (e.g., pixels). Yet, without fine-tuning, this approach may impair performance, particularly with datasets' diverse characteristics. Then we propose an automated method to fine-tune the variance spatial distribution of anisotropic noise for each dataset.

3.4.2 Dataset-wise Anisotropic Noise

The NPG employs a constant-input neural network generator [70] to learn asymmetric variances during noise-based robust training (see Figure 3.3 right). The generator outputs tensors for anisotropic σ and μ, where μ_i and σ_i represent the mean offset and variance multiplier per pixel, respectively. Anisotropic noise $\epsilon' = \epsilon^\top \sigma + \mu$ is will be generated from base isotropic noise ϵ and added to the input for randomized smoothing.

Architecture. We adopt the generator architecture (a multi-layer perception) in Generative Adversarial Network (GAN) [71] to design a novel neural network generator. Different from GAN, this NPG does not depend on the input data but depends on the entire dataset. Therefore, we fixed the input as constants. Following [71], the NPG consists of 5 linear layers, and the first 4 of them are followed by activation layers. The output will be then transformed by a hyperbolic tangent function with an amplification

factor γ, i.e., $\gamma tanh(\cdot)$. This amplified hyperbolic tangent layer limits the value of the variances since an infinite value in the noise parameters will crash the training. It is worth noting that for other tasks in Natural Language Processing or Audio Processing, other NPG structures, e.g., transformer, NNs, or even heuristic algorithms can be also designed to fit the targeted tasks.

Loss Function. The NPG and classifier can be trained together for optimal synergy. Designing an appropriate loss function is crucial for guiding NPG to desired convergence, targeting enhanced certified robustness via the ALM or certified radius. We aim to maximize either $\sqrt[d]{\prod \sigma}R$ or $\min\{\sigma\}R$, leading to two loss function variants that focus on increasing $\sqrt[d]{\prod \sigma}$ (equivalent to $mean\{\sigma\}$ in log scale) or $\min\{\sigma\}$, respectively. As the certified radius R is influenced by prediction accuracy against noise, enhancing this accuracy also boosts R, aligning with the smoothed classifier's training objectives.

$$\mathcal{L}(\theta_f, \theta_g) = \underbrace{-\,mean/min\{\sigma(\theta_g)\}}_{Variance\ Loss} + \underbrace{\sum_{k=1}^{N} y_k \log \hat{y}_k(x + \epsilon^\top \sigma(\theta_g) + \mu(\theta_g), \theta_f, \theta_g)}_{Smoothing\ Loss} \tag{3.13}$$

where the variance loss can be $mean\{\sigma(\theta_g)\}$ or $min\{\sigma(\theta_g)\}$ for improving ALM or certified radius, respectively. θ_f and θ_g denote the model parameters of the classifier and parameter generator, respectively, k denotes the prediction class, N represents the total number of classes, y_k denotes the label of input x, and $\hat{y}_k$ is the prediction of y_k. The training of the NPG for μ is also guided by the smoothing loss to improve the prediction over the dataset.

3.4.3 Certification-wise Anisotropic Noise

While dataset-wise anisotropic noise fine-tunes parameters during training for improved robustness, it doesn't fully account for the heterogeneity among different input samples. The certification also varies by input, being valid only for the certified input and the corresponding radius. Thus, we propose a certification-wise NPG that generates tailored anisotropic noise for each sample, considering heterogeneity across both inputs and data dimensions. Unlike dataset-wise noise, the parameter generators for μ and σ in certification-wise NPG are cascaded, not parallel (see right in Figure 3.3). The mean parameter generator first processes the input x to produce a μ map, followed by the variance generator using $x + \mu$ to generate a σ map. Then the smoothed classifier is based on the generated μ and σ through the classification.

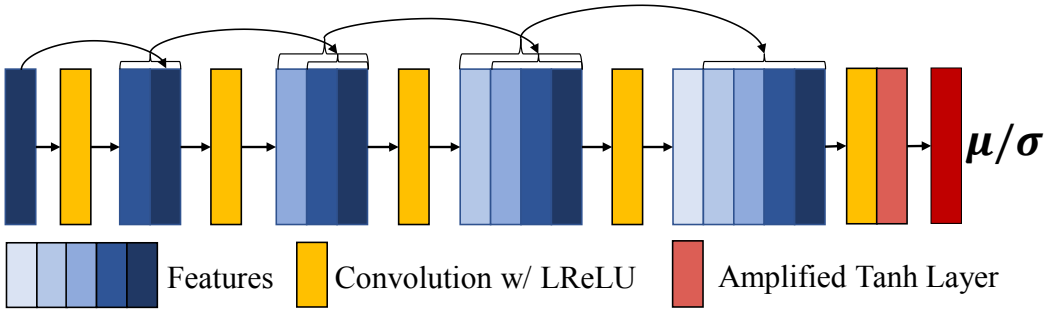


Fig. 3.5: Architecture of parameter generator for certification-wise anisotropic noise.

Architecture and Loss Function. This NPG learns the mapping from the image to the μ and σ maps, which is similar to the function of neural networks in image transformation tasks. Hence, inspired by the techniques used in image super-resolution [72], we also adopt the "dense blocks" [73] as the main architecture to design the NPG (see

Figure 3.5). It consists of 4 convolutional layers followed by leaky-ReLU [74]. Similar to the generator in dataset-wise anisotropic noise, the output is rectified by the amplified hyperbolic tangent function to stabilize the overall training process. Note that our parameter generator is a relatively small network (5 layers), thus it can be plugged in before any classifier for generating the certification-wise anisotropic noise without consuming too many computing resources (see Section 3.6.5 for a detailed discussion on running time). For both μ and σ, we train a corresponding parameter generator for each using the same architecture. The loss function is similar to that used for the dataset-wise anisotropic noise, but the NPG takes x as input and outputs μ and σ.

3.4.4 Practical Algorithms

Following Cohen et al. [1], we also use the Monte Carlo algorithm to bound the prediction probabilities of smoothed classifier and compute the ALM (certified region). Different from Cohen et al. [1], our noise distributions are either pre-assigned (as pattern-wise) or produced by the parameter generator (either dataset-wise or certification-wise). Our algorithms for certification and prediction using different noise generation methods are summarized in Algorithm 1 and 2 (w.l.o.g., taking the binary classifier as an example).

For simplicity of notations, the generation of anisotropic μ and σ are summarized by the noise generation method(s) M. In the case of pattern-wise anisotropic noise, M outputs pre-assigned fixed variance and zero-means; in case of dataset-wise and

Algorithm 1 UCAN-Certification

Given: Base classifier f, anisotropic noise generation method M, input (e.g., image) x, number of Monte Carlo samples n_0 and n, confidence $1 - \alpha$

$\mu, \sigma \leftarrow M$

$counts\_select \leftarrow$ ClassifySamples(f, x, μ, σ, n_0)

$\hat{c}_A \leftarrow$ **top index in** $counts\_select$

$counts \leftarrow$ ClassifySamples(f, x, μ, σ, n)

$\underline{p_A}' \leftarrow$ LowerConfBound$(counts[\hat{c}_A], n, 1 - \alpha)$

if $\underline{p_A}' > \frac{1}{2}$ **then**

    return prediction $\hat{c}_A$ and certified radius min$\{\sigma\}$R/ALM $\sqrt[d]{\prod_i \sigma_i} R$

else

    return ABSTAIN

certification-wise anisotropic noise, M adopts the parameter generators to generate the mean and variance maps. In the certification (Algorithm 1), we select the top-1 class $\hat{c_A}$ by the ClassifySamples function, in which the base classifier outputs the prediction on the noisy input sampled from the noise distribution. Once the top-1 class is determined, classification will be executed on more samples and the LowerConfBound function will output the lower bound of the probability $\underline{p_A}'$ computed by the Binomial test. If $\underline{p_A}' > \frac{1}{2}$, we output the prediction class and the ALM (measuring the certified region). Otherwise, it outputs ABSTAIN. In the prediction (Algorithm 2), we also generate the noise and then compute the prediction counts over the noisy inputs. If the Binomial test succeeds, then it outputs the prediction class. Otherwise, it returns ABSTAIN.

Algorithm 2 UCAN-Prediction

Given: Base classifier f, anisotropic noise generation method M, input (e.g., image) x, number of Monte Carlo samples n, confidence $1 - \alpha$

$\mu, \sigma \leftarrow M$

$counts \leftarrow \textsc{ClassifySamples}(f, x, \mu, \sigma, n)$

$\hat{c}_A \leftarrow$ **top index in** $counts$

$n_A \leftarrow counts[\hat{c}_A]$

if $\textsc{BinomialPValue}(n_A, n, 0.5) \leq \alpha$ **then**

    return prediction $\hat{c}_A$

else

    return ABSTAIN

3.5 Soundness Analysis of Certification-Wise Anisotropic Randomized Smoothing

In this section, we address the potential soundness pitfall in existing input-dependent randomized smoothing methods and demonstrate how our certification-wise approach provides enhanced robustness guarantees. Specifically, we provide formal definitions, theorems, and detailed proofs to establish the soundness of our method. Additionally, we carefully explain why randomized smoothing inherently depends on the clean input for reliable certification.

3.5.1 Potential Soundness Pitfall in Existing Input-Dependent Randomized Smoothing Methods

Recent attempts to enhance randomized smoothing have introduced input-dependent noise parameters that vary with both the input x and potential adversarial perturbations δ [68, 75]. In these methods, the noise parameters $\mu(x, \delta)$ and $\sigma(x, \delta)$ are functions of both the clean input and the perturbation, leading to noise distributions that change based on the adversary's actions.

While such approaches aim to tailor the noise to each input and perturbation, it introduce potential concerns in the certified robustness guarantee. Specifically, when certifying a sample x, the noise is generated as $\epsilon(x)$, depending on x. However, when applying the guarantee to a potentially perturbed sample $x + \delta$, the noise changes to $\epsilon(x+\delta)$, which likely follows a different distribution from $\epsilon(x)$. Randomized smoothing requires that the prediction on the certified input x and the perturbed input $x+\delta$ be based on the same smoothed classifier with the same noise distribution [1]. Therefore, this input-dependent noise may affect the soundness of randomized smoothing. As noted in [68, 75], an additional component (e.g., fixing the distribution or adopting memory-based certification) has been utilized to maintain valid guarantees by sacrificing the system performance, as the "price paid for soundness".

3.5.2 Certification-wise Anisotropic Randomized Smoothing

Different from [68, 75], in our *certification-wise* anisotropic randomized smoothing method, the noise parameters $\mu(x)$ and $\sigma(x)$ depend solely on the clean input x and remain the same when applied to any potential perturbed samples during certification. This design ensures that the smoothed classifier remains consistent across all perturbations applied to x, inherently preserving the soundness of the certified robustness guarantee.

After generating the fine-tuned noise on the clean input x (to be certified), our method constructs a robustness region that conceptually ensures robustness for x against any perturbation δ within the certified radius, rather than actually injecting noise into the perturbed inputs. It is worth noting that the theorems in this paper universally work for isotropic and anisotropic noise independent of the noise generation method, which is sound for any randomized smoothing method. Furthermore, the "pattern-wise" and "dataset-wise" noise in Section 3.4 have no potential concerns regarding this soundness problem.

3.5.3 Certified Robustness Guarantee on Certification-wise Noise

We now formally establish the soundness of our method by proving that the certified robustness guarantee holds under our certification-wise anisotropic noise.

Theorem 5 (Certified Robustness of Certification-Wise Anisotropic Randomized Smoothing)**.** *Let $f : \mathbb{R}^d \to \mathcal{C}$ be any deterministic or randomized base classifier. Suppose*

that for the random variable $X = x + \epsilon$, where ϵ is drawn from an isotropic distribution, the certified radius function is $R(\cdot)$. Define the anisotropic random variable $Y = x + \epsilon^\top \sigma(x) + \mu(x)$, where $\sigma(x) \in \mathbb{R}^d$ and $\mu(x) \in \mathbb{R}^d$ are functions of x only. If there exist $c_A \in C$ and bounds $\underline{p_A}, \overline{p_B} \in [0, 1]$ such that:

$$\mathbb{P}_\epsilon\left(f(Y) = c_A\right) \geq \underline{p_A} \geq \overline{p_B} \geq \max_{c \neq c_A} \mathbb{P}_\epsilon\left(f(Y) = c\right) \tag{3.14}$$

then, for any perturbation $\delta \in \mathbb{R}^d$ satisfying:

$$\|\delta \oslash \sigma(x)\|_p \leq R\left(\underline{p_A}, \overline{p_B}\right) \tag{3.15}$$

the smoothed classifier g will consistently predict class c_A at $x + \delta$, i.e., $g(x + \delta) = c_A$. Here, $\oslash$ denotes element-wise division, and $\|\cdot\|_p$ denotes the ℓ_p norm.

3.5.4 Proof of Theorem 5

Proof. Let $X = x + \epsilon$, where $\epsilon \in \mathbb{R}^d$ follows an isotropic noise distribution. Define the anisotropic random variable $Y = x + \epsilon^\top \sigma(x) + \mu(x)$.

Define a transformation $h_x : \mathbb{R}^d \to \mathbb{R}^d$ specific to input x:

$$h_x(z) = z^\top \sigma(x) + \mu(x) \tag{3.16}$$

where the multiplication and addition are element-wise.

Given any deterministic or randomized function $f : \mathbb{R}^d \to C$, consider the composed classifier $f' = f \circ h_x$, mapping $z \mapsto f(h_x(z))$.

Under the transformation, the condition on the class probabilities becomes:

$$\mathbb{P}_\epsilon\left(f'(X) = c_A\right) = \mathbb{P}_\epsilon\left(f(h_x(X)) = c_A\right) \tag{3.17}$$

$$= \mathbb{P}_\epsilon\left(f(Y) = c_A\right) \geq \underline{p_A} \tag{3.18}$$

$$\max_{c \neq c_A} \mathbb{P}_\epsilon\left(f'(X) = c\right) = \max_{c \neq c_A} \mathbb{P}_\epsilon\left(f(Y) = c\right) \leq \overline{p_B} \tag{3.19}$$

Thus, the probability bounds required for certification are satisfied by f' under isotropic noise ϵ.

From the standard randomized smoothing theory (e.g., Theorem 2), since the transformed classifier f' satisfies the probability bounds with respect to the isotropic noise ϵ, we have that for any perturbation $\delta' \in \mathbb{R}^d$ satisfying $\|\delta'\|_p \leq R(\underline{p_A}, \overline{p_B})$, the prediction of f' remains constant:

$$\arg\max_{c \in C} \mathbb{P}_\epsilon\left(f'(x + \delta' + \epsilon) = c\right) = c_A \tag{3.20}$$

Now, consider a perturbation $\delta \in \mathbb{R}^d$ in the original input space such that $\delta' = \delta \oslash \sigma(x)$. Then, the perturbed input after transformation is:

$$h_x(x + \delta) = (x + \delta)^\top \sigma(x) + \mu(x) \tag{3.21}$$

$$= x^\top \sigma(x) + \delta^\top \sigma(x) + \mu(x) \tag{3.22}$$

$$= h_x(x) + \delta^\top \sigma(x) \tag{3.23}$$

Substituting back, we have:

$$g(x + \delta) = \arg\max_{c \in C} \mathbb{P}_\epsilon\left(f\left(x + \delta + \epsilon^\top \sigma(x) + \mu(x)\right) = c\right) \tag{3.24}$$

$$= \arg\max_{c \in C} \mathbb{P}_\epsilon\left(f\left(h_x(x) + \delta^\top \sigma(x) + \epsilon^\top \sigma(x)\right) = c\right) \tag{3.25}$$

Since $\delta^\top \sigma(x) + \epsilon^\top \sigma(x) = (\delta + \epsilon)^\top \sigma(x)$, we have:

$$g(x+\delta) = \arg\max_{c \in C} \mathbb{P}_\epsilon \left(f\left(h_x(x) + (\delta+\epsilon)^\top \sigma(x) \right) = c \right) \tag{3.26}$$

$$= \arg\max_{c \in C} \mathbb{P}_\epsilon \left(f'\left(x + \delta' + \epsilon \right) = c \right) \tag{3.27}$$

By the robustness of f' under isotropic noise (Equation (3.20)), we have $g(x+\delta) = c_A$ whenever $\|\delta'\|_p = \|\delta \oslash \sigma(x)\|_p \leq R(\underline{p_A}, \overline{p_B})$.

Therefore, the smoothed classifier g maintains its prediction c_A within the certified region defined by the anisotropic noise parameters, establishing the soundness of our method. □

3.5.5 Why Randomized Smoothing Depends on the Clean Input

Randomized smoothing inherently depends on the clean input x because the certification process aims to guarantee the classifier's robustness for that specific input. The certified radius $R(\underline{p_A}, \overline{p_B})$ is computed based on the class probabilities at x, which are estimated using noise added to x. Consequently, the smoothed classifier g and the corresponding robustness guarantee are tied to the clean input.

In our method, the noise parameters $\sigma(x)$ and $\mu(x)$ are functions of x, further emphasizing this dependency. By designing the noise to be input-specific but independent of perturbations, we ensure that the certification process accurately reflects the classifier's behavior around the clean input, providing a meaningful and sound robustness guarantee.

3.6 Experiments

We present a comprehensive evaluation of UCAN in this section. Specifically, in Section 3.6.1, UCAN's performance with three anisotropic noise types is tested against isotropic noise baselines. Section 3.6.2 assesses UCAN's universality regarding noise distributions and resistance to various ℓ_p perturbations. In Section 3.6.3, we compare UCAN's top performance with state-of-the-art randomized smoothing methods. Section 3.6.4 and 3.6.5 present the visualization and the efficiency. Additional experiments are detailed in Appendix 7.11.

Metrics. Existing randomized smoothing methods often adopt the *approximate certified test set accuracy* from [1], defined as the proportion of the test set correctly certified above a radius R. Besides, we also report certified accuracy based on the ALM, representing the fraction of the test set certified correctly with *at least ALM*. Formally, for asymmetric RS, we define $Acc(\min\{\sigma\}R) = \frac{1}{N}\sum_{j=1}^{N} \mathbf{1}_{[g'(x^j+\delta)=y^j]}$, $\forall\ \|\delta\|_p \leq \min\{\sigma\}R$ and $Acc(ALM) = \frac{1}{N}\sum_{j=1}^{N} \mathbf{1}_{[g'(x^j+\delta)=y^j]}$, $\forall\ \|\delta\|_p \leq \sqrt[d]{\prod \sigma_i}R$. In isotropic RS, these metrics converge to $Acc(R)$.

To fairly position our methods, when compared to the SOTA methods, we present the certified accuracy w.r.t. the certified radius and optionally w.r.t. ALM.

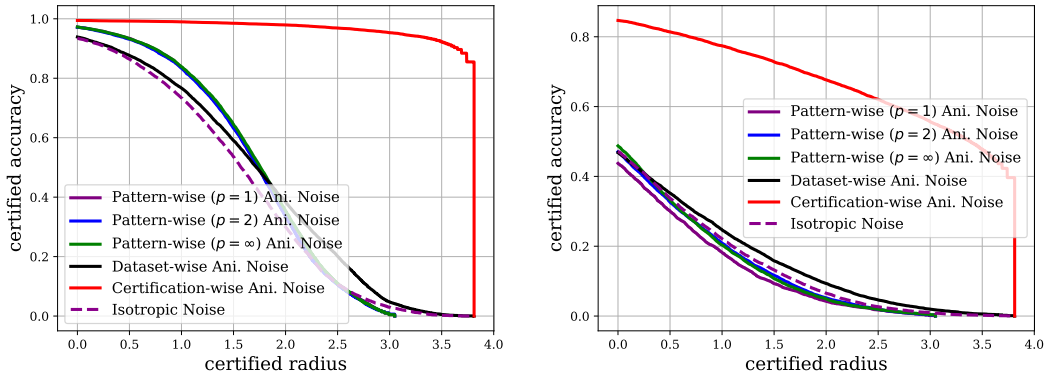
Experimental Settings. All the experiments are performed on three datasets: MNIST [27], CIFAR10, [28] and ImageNet [29]. Following [1], we obtain the certified accuracy on the entire test set in CIFAR10 and MNIST while randomly picking 500 samples in

the test set of ImageNet; we set $\alpha = 0.001$ and the numbers of Monte Carlo samples $n_0 = 100$ and $n = 100,000$. We use the original size of the images in MNIST and CIFAR10, i.e., 28×28 and $3 \times 32 \times 32$, respectively. For the ImageNet dataset, we resize the images to $3 \times 224 \times 224$. In the training, we train the base classifier and the parameter generator (if needed) with all the training set in three datasets. For the MNIST dataset, we use a simple two-layer CNN as the base classifier. For the CIFAR10 and ImageNet datasets, we use the ResNet110 and ResNet50 [36] as the base classifier, respectively. Dataset-wise NPG uses a 5-layer MLP, and certification-wise NPG uses a 4-layer CNN, both trained with Adam optimizer (learning rate 1×10^{-2}, batch size 128, 200 epochs). Noise parameters are constrained to be positive.

Experimental Environment. All the experiments were performed on the NSF Chameleon Cluster [35] with Intel(R) Xeon(R) Gold 6230 CPU @ 2.10GHz, 128G RAM, and Tesla V100 SXM2 32GB.

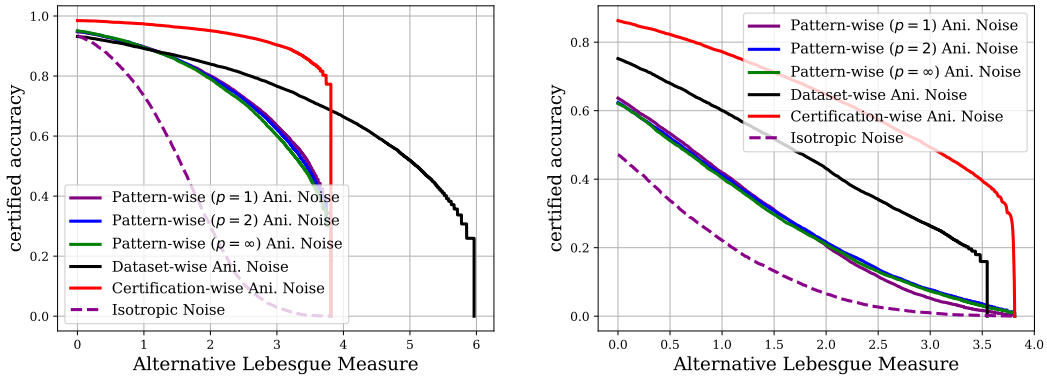
3.6.1 Anisotropic vs Isotropic Noise in Randomized Smoothing

We first evaluate randomized smoothing with anisotropic noise generated by the three example NPGs. W.l.o.g., we adopt the most common setting as default: Gaussian distribution against ℓ_2 perturbations, compare with the isotropic Gaussian baseline [1] (with zero-mean), which derives the tight certified radius (under multi-class setting) against ℓ_2 perturbations. Other distributions against different ℓ_p perturbations (*universality* of UCAN) are evaluated in Section 3.6.2 and 3.6.3.



(a) Acc vs. Radius (MNIST)

(b) Acc vs. Radius (CIFAR10)



(c) Acc vs. ALM (MNIST)

(d) Acc vs. ALM (CIFAR10)

Fig. 3.6: RS with anisotropic noise vs. baseline [1] with isotropic noise (Gaussian for ℓ_2) – UCAN gives significantly better certified accuracy and larger certified radius/ALM.

Parameter Setting. For a fair comparison, we follow [1] to set the variance $\lambda = 1$ for isotropic Gaussian noise to benchmark with our methods. For our pattern-wise method, since the variance varies in different dimensions, we re-scale $\sigma(a, b)$ such that $mean\{\sigma\} \approx 1.0$. For the dataset-wise noise, we empirically select the γ to achieve the

best trade-off on each dataset. For the certification-wise noise, the amplified factor γ in the parameter generator is set as 1.0 for all datasets.

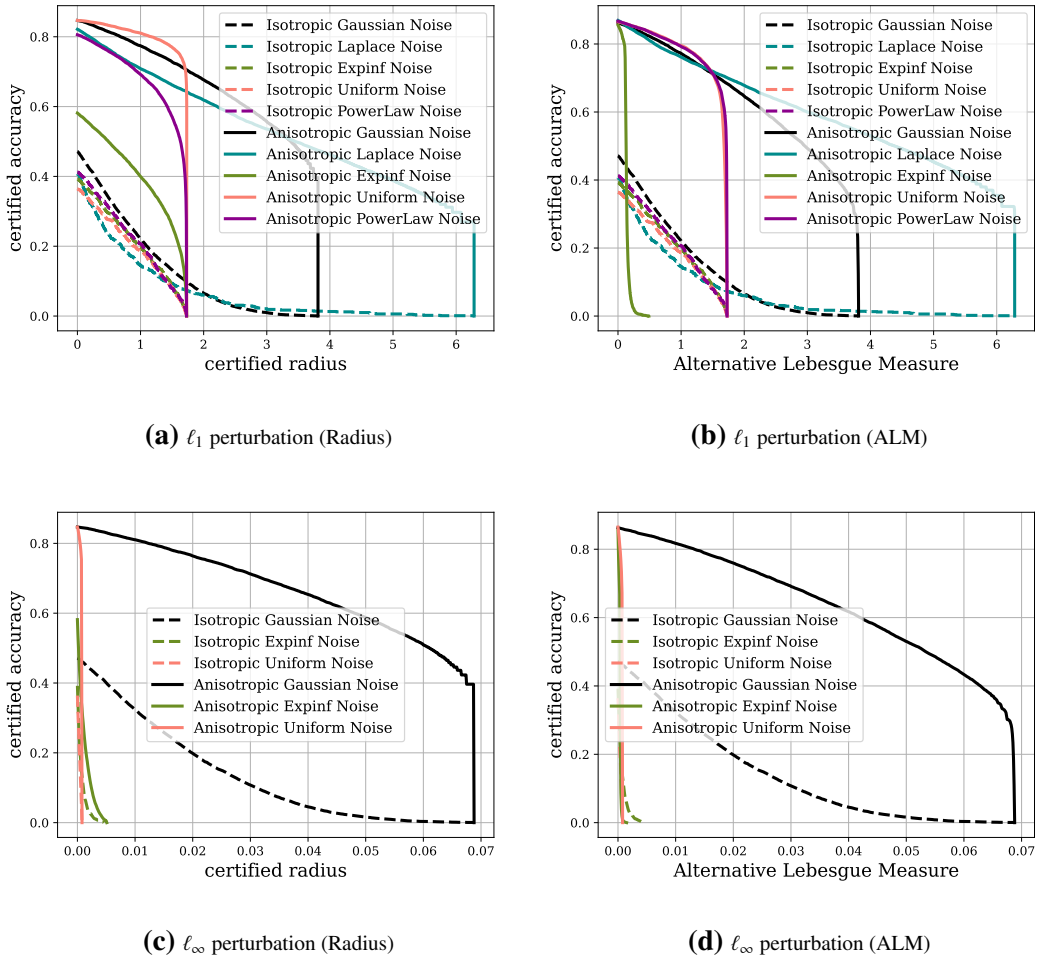
Experimental Results. The results on MNIST and CIFAR10 are presented in Figure 3.6, and other results on ImageNet datasets are deferred to Appendix 7.11. First, as shown in Figure 3.4, the certified accuracy of certification-wise anisotropic noise dominates all the noise customization methods across various settings. This indicates that optimizing anisotropic noise for each certification (of a specific input) can universally boost the performance w.r.t. the certified radius and the ALM since it achieves the best optimality on each certification (both the mean and variance can be optimized according to the input that will be certified). Second, dataset-wise anisotropic noise offers a modest improvement in certified accuracy w.r.t. the certified radius but significantly boosts certified accuracy w.r.t. the ALM. The reason is that the training of dataset-wise NPG can achieve a better trade-off between the prediction accuracy and the $mean\{\sigma\}$ since NPG can learn to assign small variance to the key data dimensions to improve the prediction while maintaining the $mean\{\sigma\}$ by increasing the variance in other data dimensions. However, it is hard to improve the trade-off between the prediction accuracy and the $\min\{\sigma\}$ since decreasing $\min\{\sigma\}$ drops the certified radius while improving the prediction (see some examples for the dataset-wise anisotropic noise in Appendix 3.6.4). Similarly, the pattern-wise anisotropic noise only improves the certified accuracy w.r.t. certified radius slightly (even reduces the performance on CIFAR10), but we also observe a better trade-off between the certified accuracy w.r.t. ALM. Finally,

we also observe that the dataset-wise anisotropic noise can significantly improve the ALM on MNIST (as much as $ALM = 6$). These improvements stem from our method's ability to optimize noise parameters at different levels of optimality. Unlike fixed noise patterns, certification-wise anisotropic noise adapts both mean and variance to each input's characteristics, leading to better prediction accuracy while maintaining robustness guarantees. The significant ALM improvements (up to 6x on MNIST) result from our optimization targeting the overall certified volume rather than just the worst-case radius.

3.6.2 Universality (Noise Distributions for ℓ_p Perturbations)

In this section, we evaluate the universality of UCAN with the certification-wise anisotropic noise over different noise distributions against different ℓ_p perturbations. Specifically, we evaluate the randomized smoothing methods with noise listed in Table 3.1, and follow [3] to set the scalar parameter λ of different noise distributions with variance 1. For the anisotropic noise, we follow the aforementioned parameter settings for pattern-wise, dataset-wise, and certification-wise anisotropic noise. We present the results against ℓ_1 and ℓ_∞ perturbations due to the lack of existing RS theories for non-Gaussian noise against ℓ_2 perturbations.

In Figure 3.7, for all settings, UCAN can universally amplify the certified robustness of isotropic RS. It also shows that the anisotropic Laplace noise and the anisotropic Gaussian noise achieve the best trade-offs between certified accuracy and certified ra-



(a) ℓ_1 perturbation (Radius)

(b) ℓ_1 perturbation (ALM)

(c) ℓ_∞ perturbation (Radius)

(d) ℓ_∞ perturbation (ALM)

Fig. 3.7: UCAN (certification-wise anisotropic) vs. isotropic randomized smoothing: different noise PDFs against various ℓ_p perturbations on CIFAR10.

dius/ALM against ℓ_1 perturbation and ℓ_∞ perturbation, respectively. This universal improvement across different ℓ_p norms and noise distributions validates our linear transformation theory, which can seamlessly convert any isotropic randomized smoothing method to its anisotropic counterpart. The consistent performance gains demonstrate that our approach captures fundamental properties of optimal noise distribution regard-

less of ℓ_p norms and noise types. More experiments with various NPG can be found in Appendix 7.11.

3.6.3 Best Performance Comparison vs. SOTA Methods

We also compare our best performance (certification with certification-wise anisotropic noise) with the best performance of 15 SOTA methods. Here we present the certified accuracy w.r.t. both *ALM* and ℓ_p *radius*. *Note that the ALM and radius are equivalent for SOTA methods with isotropic noise.*

Following the same settings in such existing randomized smoothing methods [1, 76, 77, 78], we focus on the Gaussian noise against ℓ_2 perturbations to benchmark with them. Results are shown in Table 3.3, 3.4, and 3.5. We also present the improvement of our method over the best baseline in percentage.

On all three datasets, UCAN significantly boosts the certified accuracy. For instance, it achieves the improvement of 142.5%, 182.6%, and 121.1% over the best baseline on MNIST, CIFAR10, and ImageNet, respectively. UCAN also achieves the best trade-off between certified accuracy and radius/ALM (*two complementary metrics*): 1) UCAN presents both larger radius/ALM and higher certified accuracy in general, and 2) On large radius/ALM, UCAN can still achieve high certified accuracy. We also observe that our certified accuracy is smaller than the SOTA performances at some low radius/ALM on MNIST and ImageNet, this is because of the lower training performance of the classifier. The substantial improvements (up to 182.6%) at larger radii

demonstrate our method's strength in maintaining high certified accuracy where traditional methods degrade rapidly. This advantage stems from the combination of the linear transformation approach and the certification-wise noise, which preserves the statistical properties of the original smoothing while enabling dimension-specific noise adaptation. The larger the certified region required, the more pronounced our method's benefits become, as anisotropic noise can better accommodate the heterogeneity across input dimensions.

Beyond ℓ_2 Norm. We also compare our certification-wise method's performance against ℓ_1 and ℓ_{inf} perturbation with existing baselines [2] (see Table 3.2). Across both norms, UCAN consistently improves certified accuracy (CA) over isotropic RS. For ℓ_1, anisotropic Gaussian yields the highest CA at small–medium radii (e.g., 72% at R=1.5 vs. 55% in [3]), while anisotropic Laplace dominates at larger radii (e.g., 61% at R=2.0 vs. $\leq 25\%$ in baselines), whereas anisotropic Uniform peaks early then collapses (0% beyond R=2.0). For ℓ_∞, anisotropic Gaussian is uniformly best (e.g., 73% at 6/255 vs. 31–38% in [1, 23]), maintaining sizable margins up to 16/255. These trends confirm our linear-transformation theory's universality and show that choosing the noise family to match the threat norm (Laplace for ℓ_1, Gaussian for ℓ_∞) and employing certification-wise NPGs yields the highest practical "optimality" under fixed budgets.

[2] Due to the limited number of prior studies on the ℓ_1 and ℓ_∞ norms, we compare our method with two available baselines for each case.

Table 3.2: Certified accuracy vs. ℓ_1 and ℓ_{inf} perturbations (CIFAR10). Best and ≥ SOTA

Radius (ℓ_1 norm)	0.50	1.00	1.50	2.00	2.5	3.0	3.5	4.0
Teng et al.'s [24]	61%	39%	24%	16%	11%	7%	4%	3%
Yang et al.'s [3]	74%	62%	55%	48%	43%	40%	37%	33%
Ours (Ani. Uniform)	**84%**	**82%**	**76%**	0%	0%	0%	0%	0%
Ours (Ani. Gaussian)	**81%**	**77%**	**72%**	**66%**	**61%**	**56%**	**47%**	0%
Ours (Ani. Laplace)	**75%**	**70%**	**66%**	**61%**	**57%**	**52%**	**50%**	**46%**
Radius (ℓ_{inf} norm)	2/255	4/255	6/255	8/255	10/255	12/255	14/255	16/255
Cohen et al.'s [1]	58%	42%	31%	25%	18%	13%	–	–
Zhang et al.'s [23]	60%	47%	38%	32%	23%	17%	–	–
Ours (Ani. Gaussian)	**81%**	**78%**	**73%**	**70%**	**65%**	**60%**	**54%**	**48%**

Table 3.3: Certified accuracy vs. ℓ_2 perturbations (MNIST). Best and ≥ SOTA

Radius and ALM (equivalent for isotropic)	0.0	0.25	0.50	0.75	1.00	1.25	1.50	1.75	2.00	2.25
Cohen's [1]	83%	61%	43%	32%	22%	17%	14%	9%	7%	4%
Sample-wise [78]	98%	97%	96%	93%	88%	81%	73%	57%	41%	25%
Input-depend [77]	99%	98%	97%	94%	88%	79%	58%	27%	0%	0%
MACER [79]	99%	99%	96%	95%	90%	83%	73%	50%	36%	28%
SmoothMix [80]	99%	99%	98%	97%	93%	89%	82%	71%	45%	37%
DRT [81]	99%	98%	98%	97%	93%	89%	83%	70%	48%	40%
Ours (certified accuracy w.r.t. ALM)	98%	98%	98%	98%	97%	97%	96%	96%	95%	94%
Improvement over the Best Baseline (%)	-1.0%	-1.0%	+0.0%	+1.0%	+4.3%	+9.0%	+15.7%	+35.2%	+98.0%	+135.0%
Ours (certified accuracy w.r.t. radius)	**99%**	**99%**	**99%**	**99%**	**99%**	**99%**	**98%**	**98%**	**98%**	**97%**
Improvement over the Best Baseline (%)	+0.0%	+0.0%	+1.0%	+2.1%	+6.5%	+11.2%	+18.1%	+38.0%	+104.2%	+142.5%

Table 3.4: Certified accuracy vs. ℓ_2 perturbations (CIFAR10). Best and $\geq$ SOTA

Radius and ALM (equivalent for isotropic)	0.0	0.25	0.50	0.75	1.00	1.25	1.50	1.75	2.00	2.25
Cohen's [1]	83%	61%	43%	32%	22%	17%	14%	9%	7%	4%
SmoothAdv[82]	–	81%	63%	52%	37%	33%	29%	25%	18%	16%
MACER [79]	81%	71%	59%	47%	39%	33%	29%	23%	19%	17%
Consistency [83]	78 %	69%	58%	49%	38%	34%	30%	25%	20%	17%
SmooothMix [80]	77%	68%	58%	48%	37%	32%	26%	20%	17%	15%
Boosting [84]	83%	71%	60%	52%	39%	34%	30%	25%	20%	17%
DRT [81]	73 %	67%	60%	51%	40%	36%	30%	24%	20%	–
Black-box [23]	–	61%	46%	37%	25%	19%	16%	14%	11%	9%
Data-depend [76]	82%	68%	53%	44%	32%	21%	14%	8%	4%	1%
Sample-wise [78]	84%	74%	61%	52%	45%	41%	36%	32%	27%	23%
Input-depend [77]	83%	62%	43%	27%	18%	11%	5%	2%	0%	0%
Denoise 1 [85]	80%	70%	55%	48%	37%	32%	29%	25%	15%	14%
Denoise 2 [86]	85%	76%	66%	57%	44%	37%	31%	25%	22%	20%
ANCER [68]	84%	80%	67%	–	34%	–	15%	–	11%	–
RANCER [75]	–	81%	48%	28%	11%	1%	–	–	–	–
Ours (certified accuracy w.r.t. ALM)	**86%**	**84%**	**82%**	**80%**	74%	71%	68%	65%	61%	57%
Improvement over the Best Baseline (%)	+1.2%	+3.7%	+6.5%	+40.4%	+39.6%	+73.2%	+88.9%	+103.1%	+125.9%	+147.8%
Ours (certified accuracy w.r.t. radius)	85%	83%	81%	80%	**77%**	**75%**	**73%**	**70%**	**68%**	**65%**
Improvement over the Best Baseline (%)	+0.0%	+2.5%	+5.2%	+40.4%	+45.3%	+82.9%	+102.8%	+118.8%	+151.9%	+182.6%

3.6.4 Visualization

We present several examples of anisotropic (generated with ALM loss) and isotropic noise in Figure 3.8. All the proposed anisotropic noise generation methods find better spatial distribution to generate the anisotropic noise. Both the pattern-wise and dataset-wise anisotropic reduce the variance on the key area, except the dataset-wise anisotropic noise on ImageNet (it seems the parameter generator does not find a constant key area on ImageNet due to the complicated data distribution in ImageNet). We also observe that the parameter generator for certification-wise anisotropic noise gener-

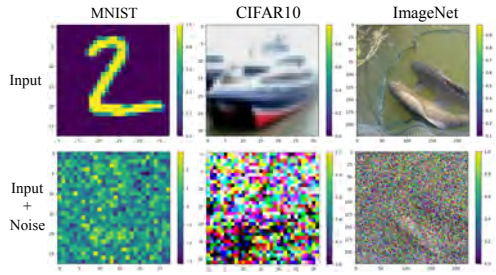
Table 3.5: Certified accuracy vs. ℓ_2 perturbations (ImageNet). Best and $\geq$ SOTA

Radius and ALM (equivalent for isotropic)	0.00	0.50	1.00	1.50	2.00	2.50	3.00	3.50
Cohen's [1]	67%	49%	37%	28%	19%	15%	12%	9%
SmoothAdv [82]	67%	56%	45%	38%	28%	26%	20%	17%
MACER [79]	68%	57%	43%	37%	27%	25%	20%	–
Consistency [83]	57%	50%	44%	34%	24%	21%	17%	–
SmoothMix [80]	55%	50%	43%	38%	26%	24%	20%	–
Boosting [84]	68%	57%	45%	38%	29%	25%	21%	19%
DRT[81]	50%	47%	44%	39%	30%	29%	23%	–
Black-box [23]	–	50%	39%	31%	21%	17%	13%	10%
Data-depend [76]	62%	59%	48%	43%	31%	25%	22%	19%
Denoise 1 [85]	48%	41%	30%	24%	19%	16%	13%	–
Denoise 2 [86]	66%	59%	48%	40%	31%	25%	22%	–
ANCER [68]	66%	66%	62%	58%	44%	37%	32%	–
Ours (certified accuracy w.r.t. ALM)	**71%**	**66%**	**62%**	**58%**	**54%**	**51%**	**47%**	**42%**
Improvement over the Best Baseline (%)	+4.4%	+0.0%	+0.0%	+0.0%	+22.7%	+37.8%	+46.9%	+121.1%
Ours (certified accuracy w.r.t. radius)	65%	62%	58%	53%	50%	46%	43%	38%
Improvement over the Best Baseline (%)	-4.4%	-6.1%	-6.5%	-8.6%	+13.6%	+24.3%	+34.4%	+100.0%

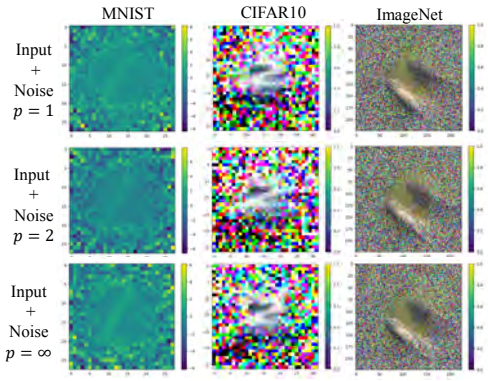
ates large mean offsets to compensate for the high σ values. It turns out that with high variance ($\sigma_i \approx 1$), the object is still recognizable in the certification-wise anisotropic noise.

3.6.5 Efficiency

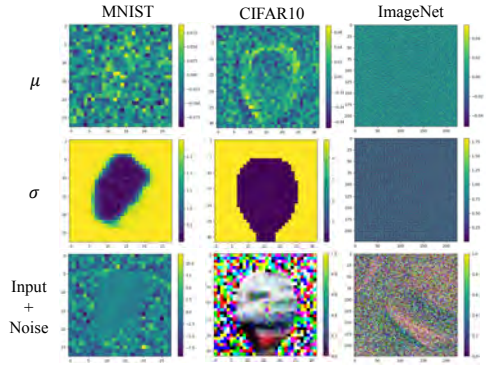
UCAN is a universal framework that can be readily integrated into existing randomized smoothing to boost performance. Whether the extra neural network components (parameter generator) in UCAN will degrade the efficiency of existing randomized



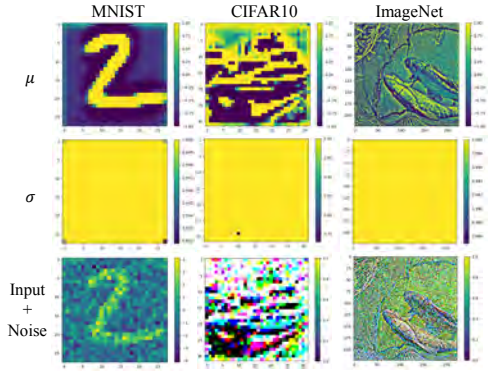
(a) Example of Isotropic Noise



(b) Examples of pattern-wise anisotropic noise



(c) Examples of dataset-wise anisotropic noise



(d) Examples of certification-wise anisotropic noise

Fig. 3.8: Visualization of anisotropic vs. isotropic noise, based on the input in (a).

smoothing is an important question. We show that the running time overhead resulting from the parameter generator is negligible compared to the running time of the certification, since for each input, the classifier needs to evaluate N noise samples while μ and σ are generated once. Typically, $N = 100,000$. UCAN can be trained offline and tested online to boost the performance of randomized smoothing. We evaluate the online certi-

fication running time for certification-wise anisotropic noise generation and traditional randomized smoothing [1] on ImageNet with four Tesla V100 GPUs and $2,000$ batch size, the average runtimes over 500 samples are 27.43s and 27.09s per sample for our method and [1]'s method, respectively. Thus, the NPG will only slightly increase the overall runtime. Also, Training a certification-wise NPG requires 200 epochs, taking 31.67 minutes on an H100 GPU.

3.7 Related Work

In this section, we review the related works for certified defenses against evasion attacks on machine learning models.

Certified Defenses. Certified defenses guarantee robustness against adversarial perturbations within a specified boundary (e.g., ℓ_1, ℓ_2, or ℓ_∞ ball of radius R). Existing approaches fall into two categories: exact and conservative certified defenses. Exact methods leverage satisfiability modulo theories [13, 14, 38, 87] or mixed-integer linear programming [15, 39, 40, 88] to precisely determine the existence of adversarial examples within R, but are not scalable to large networks. Conservative methods use global/local Lipschitz constants [16, 45, 46, 47, 48], optimization-based certification [12, 17, 41, 44], or layer-by-layer approaches [19, 50, 51, 52, 53], which scale better but typically yield conservative or architecture-specific guarantees. Neither can certify arbitrary classifiers, a gap addressed by randomized smoothing.

Randomized Smoothing. Randomized smoothing was first explored by Lecuyer et

al. [22], who derived a loose theoretical robustness bound via Differential Privacy [89, 90]. Cohen et al. [1] later provided the first tight guarantee, showing that adding Gaussian noise transforms any classifier into a smoothed classifier with certified ℓ_2 robustness. Subsequent work extended smoothing to other ℓ_p norms: Teng et al. [24] analyzed ℓ_1 robustness with Laplace noise, and Lee et al. [54] studied ℓ_0 robustness with uniform noise. Unified frameworks have also been proposed: Zhang et al. [23] presented an optimization-based approach certifying ℓ_1, ℓ_2, and ℓ_∞ robustness; Yang et al. [3] introduced methods based on level sets and differentials to bound certified radii for various distributions and norms; Hong et al. [4] proposed a framework to approximately certify any ℓ_p-norm adversary with any noise. However, existing randomized smoothing methods use fixed noise distributions (e.g., Gaussian, Laplace) applied uniformly to all input dimensions, overlooking data heterogeneity and limiting optimal protection for different inputs.

Data-Dependent Randomized Smoothing. Data-dependent randomized smoothing improves certified robustness by optimizing noise distributions for individual inputs. Most existing approaches focus on isotropic noise, typically optimizing the variance to enlarge the certified radius. For instance, Alfarra et al. [76] optimize the Gaussian variance via gradient methods; Sukenik et al. [77] model variance as an input-dependent function; Wang et al. [78] use grid search for variance selection. However, these methods still inject noise with identical variance across all dimensions, due to the absence of anisotropic theory.

Asymmetric Randomized Smoothing. Prior work has already shown that randomized smoothing can certify *anisotropic* regions: ANCER [68] introduces axis-aligned ellipsoidal certificates by scaling per-dimension noise, and RANCER [75] generalizes this to full (rotated) covariance structures. Both frameworks derive guarantees by enforcing Lipschitz constraints on the smoothed classifier and by (optionally) adapting noise parameters per input to enlarge the certified set.

Our approach departs from the Lipschitz-based route and instead relies on an explicit *linear transformation* between isotropic and anisotropic noise distributions. Concretely, let the isotropic guarantee certify x within radius R under noise ϵ. For any invertible covariance matrix Σ (cf. our notation in Eq. (4)), the same argument transfers to the anisotropic setting by the change of variables $\delta' = \Sigma\delta$: $\|\delta\|_p \leq R \iff \|\Sigma^{-1}\delta'\|_p \leq R$, so the certified region becomes an ellipsoid (or generalized ellipsoid) parameterized by Σ. This algebraic reduction preserves the original Neyman–Pearson optimality of isotropic smoothing and avoids the typically looser (or at least not tighter) bounds introduced by gradient/Lipschitz upper bounds used in [68, 75].

[68, 75] allow input-dependent noise but must cache those parameters to apply the same distribution for all evaluations during certification; otherwise the proof assumptions change. Our "certification-wise" design fixes the noise after observing the clean input and reuses it throughout the procedure, eliminating the memory-based workaround while retaining input adaptivity at x. By replacing Lipschitz bounds with a linear change of variables, we (i) simplify the theoretical pipeline, (ii) obtain tighter

certificates (no gradient over-approximation), and (iii) remove the need for parameter memorization. Empirically (Table 3.4), at radius 1.25, our method certifies 75% accuracy, far exceeding ANCER (31%) and RANCER (1%). Across all large radii, we achieve up to 182.6% improvement over these baselines.

3.8 Conclusion

In this paper, we introduce UCAN (Universally Certifies adversarial robustness with Anisotropic Noise), a novel and flexible randomized smoothing framework that transforms any isotropic noise-based scheme into schemes utilizing anisotropic noise, providing strict and theoretically grounded robustness guarantees. We also propose a unified and customizable noise customization framework with three methods for fine-tuning anisotropic noise in classifier smoothing. Extensive and comprehensive evaluation of the proposed method on MNIST, CIFAR10, and ImageNet confirms that UCAN substantially enhances the certified robustness of existing randomized smoothing methods. By enabling flexible and dimension-aware noise design, UCAN opens new possibilities for certified defense in more complex and heterogeneous data domains.

Chapter 4

Certifiable Black-Box Attacks with Randomized Adversarial Examples: Breaking Defenses with Provable Confidence

Black-box adversarial attacks have demonstrated strong potential to compromise machine learning models by iteratively querying the target model or leveraging transferability from a local surrogate model. Recently, such attacks can be effectively mitigated by state-of-the-art (SOTA) defenses, e.g., detection via the pattern of sequential queries, or injecting noise into the model. To our best knowledge, we take the first step to study a new paradigm of black-box attacks with provable guarantees – certifiable black-box attacks that can guarantee the attack success probability (ASP) of adversarial examples before querying over the target model. This new black-box attack unveils significant vulnerabilities of machine learning models, compared to traditional empirical black-box attacks, e.g., breaking strong SOTA defenses with provable confidence, constructing a space of (infinite) adversarial examples with high ASP, and the ASP of the generated adversarial examples is theoretically guaranteed without verification/queries over the target model. Specifically, we establish a novel theoretical foundation for ensuring

the ASP of the black-box attack with randomized adversarial examples (AEs). Then, we propose several novel techniques to craft the randomized AEs while reducing the perturbation size for better imperceptibility. Finally, we have comprehensively evaluated the certifiable black-box attacks on the CIFAR10/100, ImageNet, and LibriSpeech datasets, while benchmarking with 16 SOTA black-box attacks, against various SOTA defenses in the domains of computer vision and speech recognition. Both theoretical and experimental results have validated the significance of the proposed attack.

4.1 Introduction

Machine learning (ML) models have achieved unprecedented success and have been widely integrated into many practical applications. However, it is well known that minor perturbations injected into the input data are sufficient to induce model misclassification [91]. Many state-of-the-art (SOTA) adversarial attacks [91, 92, 93, 94, 95, 96, 97, 98, 99, 100, 101, 102] have been proposed to explore the vulnerabilities of a variety of ML models. Wherein, the stringent *black-box* attack is believed to be closer to real-world security practice [93, 103].

In black-box attacks, the adversary only has access to the target ML model's outputs (either prediction scores or hard labels). Through iteratively querying the target model, the adversary progressively updates the perturbation until convergence. Existing black-box attack methods primarily utilize gradient estimation [97, 98, 104, 105, 106], surrogate models [103, 107, 108, 109], or heuristic algorithms [99, 101, 110, 111, 112]

to generate adversarial perturbations. Although these attack algorithms can empirically achieve relatively high attack success rates (e.g., on CIFAR-10 [28]), their query process is shown to be easy to detect or interrupt due to the minor perturbation changes and high reliance on the previous perturbation [113, 114, 115, 116]. For example, "Blacklight" [113] can achieve 100% detection rate on most of the existing black-box attacks by checking the similarity of queries; some "randomized defense" methods [114, 115, 116, 117, 118] inject random noise to the inputs, outputs, intermediate features or model parameters such that the performance of existing black-box attacks can be significantly degraded (since the query results are obfuscated to be unpredictable).

To break such types of SOTA defenses [113, 114, 116, 117, 118], it is challenging to design an effective attack equipped with both *high degree of randomness to bypass the strong detection* (e.g., Blacklight [113]) and *high robustness to resist randomized defense*. A feasible solution is to add random noise to the adversarial example by the adversary, but it will make the query intractable. Therefore, an innovative method is desirable to carefully craft the adversarial example based on feedback from queries using randomly generated inputs.

To this end, we propose a novel attack paradigm, termed *Certifiable Attack*, that ensures a provable attack success probability (ASP) on the randomized adversarial examples against the equipped defenses (or no defense). Specifically, our attack strategy integrates random noise into the queries while preserving the adversarial efficacy of

these queries. In particular, we model the adversarial examples as a random variable in the input space following an underlying noise distribution φ, namely "*Adversarial Distribution*". Then, we design a novel query strategy and establish the theoretical foundation to guarantee the ASP of the distribution throughout the crafting process. A novel framework is also developed to find the initial *Adversarial Distribution*, optimize it, and use it to sample the adversarial examples.

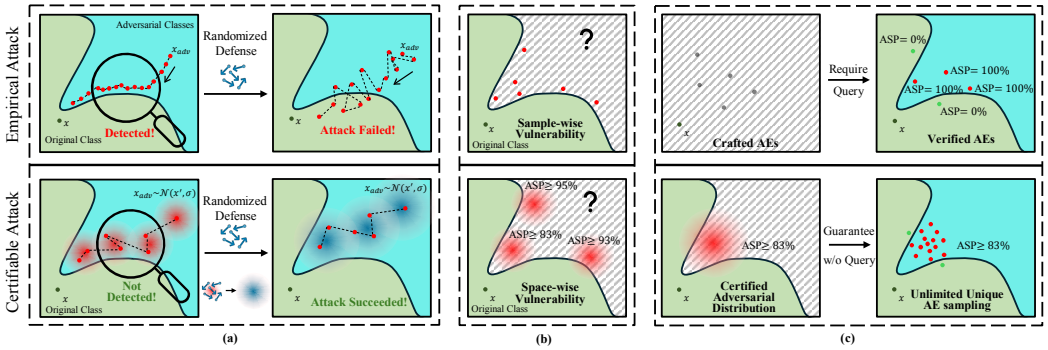


Fig. 4.1: Empirical attacks vs. Certifiable attacks. (a) Certifiable attack can break the SOTA AE detection and randomized defenses. (b) Certifiable attack uncovers space-wise vulnerability rather than sample-wise vulnerability. (c) Once certified, Certifiable Attack can generate unlimited unique AEs with a guaranteed minimum ASP without querying the model for verification, while the empirical attack requires verifying the attack result of crafted AE by query.

4.1.1 Certifiable Attacks vs. Empirical Attacks

Compared with existing empirical black-box attacks, the Certifiable Attack demonstrates multi-faceted advantages (also see Figure 4.1):

(a) **Strong attack to break SOTA defenses.** The randomness in the certifiable attack allows it to effectively bypass detection methods that rely on the similarity between the attacker's sequence of queries (e.g., Blacklight [113]), while traditional empirical attacks often create a suspicious trajectory of highly similar perturbations. The certifiable attack also provides a provable guarantee of success for attacks using randomized inputs, by taking into account the equipped defense and target model, enhancing its resistance to randomized defense [114, 115].

(b) **Adversarial space vs. Adversarial example (AE).** Distinct from traditional empirical adversarial attacks, which uncover model vulnerabilities with sample-wise inputs, the Certifiable Attack seeks to explore an adversarial input space constructed by an *Adversarial Distribution*. This continuous space facilitates the generation of numerous (potentially infinite) adversarial examples with a high ASP, thus revealing a more consistent and severe vulnerability of the target model.

(c) **Adversarial Examples (AEs) sampled from the adversarial distribution are verification-free.** Empirical attacks search AEs by iteratively querying the target model and verifying the query outputs (*the final successful AE is also used to query over the target model; then it will be verified and recorded by the defender/target*

Table 4.1: Comparison of state-of-the-art empirical black-box attacks with certifiable attack

Black-box Attacks	Query Type	Perturbation Type	ASP Guarantee	vs. Detection on Attacker's Queries	vs. Randomized Pre-process. Defense	vs. Randomized Post-process. Defense
Bandit [119], NES [98], Parsimonious [120], Sign [121], Square [99], ZOSignSGD [122]	Score-based	ℓ_∞-bounded	×	×	×	✓
GeoDA [123], HSJ [93], Opt [104], RayS [124], SignFlip [125], SignOPT [126]	Label-based	ℓ_∞-bounded	×	×	×	×
Bandit [119], NES [98], Simple [112], Square [99], ZOSignSGD [122]	Score-based	ℓ_2-bounded	×	×	×	✓
Boundary [101], GeoDA [123], HSJ [93], Opt [104], SignOPT [126]	Label-based	ℓ_2-bounded	×	×	×	×
PointWise [127], SparseEvo [128]	Label-based	Optimized	×	×	×	×
Certifiable Attack (ours)	Label-based	Optimized	✓	✓	✓	✓

model). Instead, the certifiable attack crafts the *adversarial distribution* with a guaranteed lower bound of the ASP. Due to the highly dimensional and continuous input space, AEs sampled from the adversarial distribution can be considered unique (with noise in all the dimensions) and have a negligible probability of being recorded by the defender/target model after verification. The ASP of such AEs are theoretically guaranteed (verification-free), and they are new to the defender, posing more challenges for mitigation.

4.1.2 Randomization for Certifiable Attacks

To pursue certifiable attacks, we theoretically bound the ASP of *Adversarial Distribution* based on a novel way of utilizing randomized smoothing [129], a technique achieving great success in the certified defenses with probabilistic guarantees.

The design for the randomization-based certifiable attack follows an intuitive goal, i.e., *ensuring that the classification results are consistently wrong over the distribution*. However, many new significant challenges should be addressed. First, existing

theories (randomization for certified defenses, e.g., [129]) cannot be directly adapted to certifiable attacks since they have completely different goals and settings. Second, how to efficiently craft the *Adversarial Distribution* that can ensure the ASP is challenging since it requires maintaining the wrong prediction over a large number of randomized samples drawing from the distribution. Third, how to make the *Adversarial Distribution* as imperceptible as possible is also challenging due to their randomness. By addressing these new challenges, in this paper, we make the following significant contributions:

1) To our best knowledge, we introduce the first *certifiable attack theory* based on randomization for the black-box setting, which universally guarantees the attack success probability of AEs drawn from different noise distributions, e.g., Gaussian, Laplace, and Cauthy distributions, enabling a novel transition from deterministic to probabilistic adversarial attacks.

2) We propose a novel *certifiable attack framework* that can efficiently craft certifiable *Adversarial Distribution* with provable ASP and imperceptibility. Specifically, we design a novel *randomized parallel query method* to efficiently collect probabilistic query results from any target model, which supports the certifiable attack theory. We propose a novel *self-supervised localization* method as well as a binary-search localization method to efficiently generate certifiable *Adversarial Distribution*. We design a novel *geometric shifting* method to reduce the perturbation size for better imperceptibility while ensuring the ASP. Finally, we have validated that diffusion

models [130] can be used to further denoise the randomized AEs with guaranteed ASP.

3) We comprehensively evaluate the performance of the certifiable attack with different settings on 4 datasets, while benchmarking with 16 SOTA empirical black-box attacks, against various defenses. Experimental results consistently demonstrate that our certifiable attack effectively breaks the SOTA defenses, including adversarial detection, randomized pre-processing and post-processing defenses, as well as adversarial training defenses (Also, Table 4.1 shows a summary of the certifiable attack vs. SOTA black-box attacks).

4.2 Problem Definition

Threat Model: We consider designing a certifiable attack where the target model may or may not be protected by a defense mechanism.

- **Adversary**: We focus on the hard-label black-box attack, where the adversary only knows the predicted label by querying the target ML model. The adversary's objective is to craft adversarial examples to fool the model based on the query results.

- **Model Owner**: The model owner pursues the model utility. We consider three different levels of the model owner's knowledge and capability: 1) The model owner has no awareness of the adversarial attacks and is not equipped with any defense; 2) The model owner is aware of the adversarial attack but has no knowledge of the

attack method. The model owner can deploy general defense methods such as adversarial training [91]; 3) The model owner is aware of the adversarial attack and has knowledge about the attack method. The model owner can deploy adaptive defenses that are specifically designed for the attack.

Problem Formulation: We first briefly review adversarial examples, and then formally define our problem. Given an ML classifier f and a testing data $x \in \mathbb{R}^d$ with label y from a label set $\mathcal{Y} = [1, \cdots C]$ (where C is the number of classes). An adversary carefully crafts a perturbation on the data x such that the classifier f misclassifies the perturbed data x_{adv}, i.e., $f(x_{adv}) \neq y$ under $x_{adv} \in [\Pi_a, \Pi_b]^d$, where $[\Pi_a, \Pi_b]^d$ is the valid input space. The perturbed data x_{adv} is called *adversarial example*. Imperceptibility is usually achieved by restricting the ℓ_2 or ℓ_∞ norm of the perturbation $x_{adv} - x$, or by minimizing the magnitude of this perturbation.

In the black-box setting, an adversary can use *empirical* black-box attack techniques (details in Section 4.5) to iteratively query the classifier f and progressively update the perturbation until finding a successful adversarial example for a testing example. However, such attack strategies have key limitations: 1) query inefficient, usually $>$ 100 queries per adversarial example; 2) easy to be detected by observing the query trajectory [113, 114, 115, 116]; and 3) lack of guaranteed attack performance, i.e., cannot provably guarantee a (un)successful adversarial example under a given budget.

We aim to address all these limitations and design an efficient and effective certifiable black-box attack in the paper. Particularly, instead of inefficiently searching

adversarial *examples* one-by-one, we want to certifiably find the underlying adversarial *distribution* that the adversarial examples lie on.

Definition 2 (Certifiable black-box attack)**.** *Given a classifier $f : \mathbb{R}^d \to \mathcal{Y}$, a clean input $x \in \mathbb{R}^d$ with label $y \in \mathcal{Y}$, and an Attack Success Probability Threshold p, the certifiable attack is to find an* Adversarial Distribution *$\varphi(x', \boldsymbol{\kappa})$ with mean x' and parameters $\boldsymbol{\kappa}$[1], such that data sampled from φ have at least p probability of being misclassified (i.e., adversarial examples). That is,*

$$\mathbb{P}_{x_{adv} \sim \varphi(x', \boldsymbol{\kappa})}[f(x_{adv}) \neq y] \geq p \tag{4.1}$$

$$s.t.\ x_{adv} \in [\Pi_a, \Pi_b]^d. \tag{4.2}$$

Design Goals: We expect our attack to achieve the below goals.

1) **Certifiable:** It can provide provable guarantees on the minimum attack success probability of the crafted adversarial examples.

2) **Verification free:**

   It can not only verify examples to be adversarial *after* querying the model, but also verify examples *before* the query by giving its ASP. This significantly boosts the effectiveness of adversarial examples generation.

[1] If φ is a Gaussian distribution, κ is the standard deviation of φ. If φ is a Generalized normal distribution, $\kappa = (a, b)$, with a and b the scale and shape parameters of φ, respectively. Notice that, the distribution will be applied to all the dimensions in the input, and *Adversarial Distribution* is a noise distribution over the input space.

3) **Query efficient:** It needs as few number of queries as possible. Fewer queries can definitely save the adversary's cost.

4) **Bypass defenses:** It can generate imperceptible adversarial perturbations that can bypass the existing detection and pre/post-processing based defenses [113, 114, 115, 116].

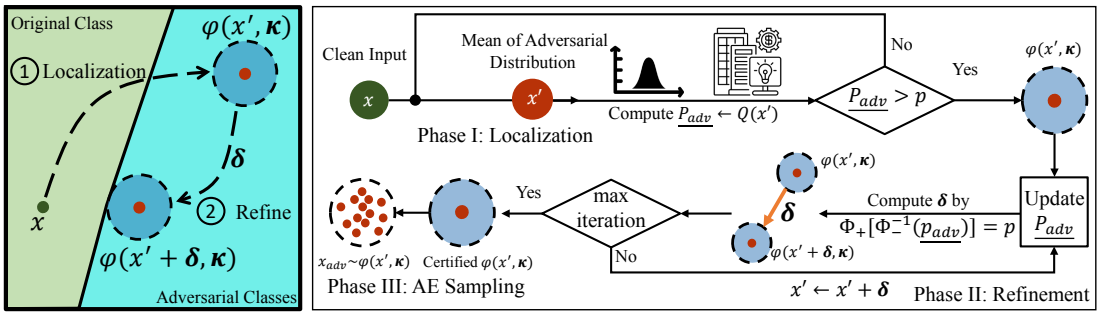


Fig. 4.2: Overview of our certifiable black-box attack to generate certified adversarial distribution.

4.2.1 Attack Overview

At a high level, our certifiable black-box attack can be divided into three phases. The overview of our attack is depicted in Figure 4.2.

Phase I: Adversarial Distribution Localization. This phrase initially locates a feasible *Adversarial Distribution* φ with guarantees on the lower bound of attack success probabilities (i.e., satisfying Eq. (4.1)). There are a few challenges. First, computing the exact probability $\mathbb{P}[f(x_{adv}) \neq y]$ is intractable due to the high-dimensional contin-

uous input space. Second, due to the black-box nature, there exists no gradient information that can be used. To address the first challenge and ensure query efficiency, we propose a Randomized Parallel Query (RPQ) strategy that can approximate the probability and ensure multiple queries are implemented in parallel. To address the second challenge, we design two localization strategies to enable learning a feasible adversarial distribution. The first strategy adapts the existing self-supervised perturbation (SSP) technique [109], which facilitates designing a classifier-unknown loss on a pretrained feature extractor such that the adversarial examples/perturbations can be optimized. The second one is based on binary search. It first randomly initializes a qualified *Adversarial Distribution* , and then reduces the perturbation size using the binary search algorithm. See Section 4.3.1 for more details.

Phase II: Adversarial Distribution Refinement. While successfully generating the adversarial distribution, the adversarial examples from it often induce relatively large perturbation sizes. This phrase further refines the adversarial distribution by reducing the perturbation size and maintains the guarantee of attack success probability as well. Particularly, we propose to shift the adversarial distribution close to the decision boundary of the classifier. This problem can be solved by two steps: *the first step finds the shifting direction, and the second step derives the shifting distance and maintains the guarantee*. We design a novel shifting method to find the local-optimal *Adversarial Distribution* by considering the geometric relationship between the decision boundary and *Adversarial Distribution*. Deciding the shifting distance can then be converted to

an optimization problem. We then propose a binary search algorithm to achieve the goal. See Section 4.3.2 for more details.

Phase III: Adversarial Example Sampling. Phases I and II craft an *Adversarial Distribution* with guaranteed attack success probability, called "certifiable attack". To transform the *Adversarial Distribution* into concrete AEs, we need to sample the AE from the *Adversarial Distribution*. The sampled AEs naturally maintain the certified ASP without the need for additional model queries. Optionally, the adversary can verify the success of these sampled AEs to ensure a successful attack, turning the certifiable attack into an empirical attack. Specifically, the adversary can sequentially sample the adversarial examples from *Adversarial Distribution* and query the target model until finding the successful adversarial example(s).

4.3 Certifiable Black-box Attack

In this section, we present our certifiable black-box attack in detail. We first introduce the Randomized Parallel Query strategy that estimates the lower bound probability of being the adversarial example (Section 4.3.1). We then develop two algorithms to locate the feasible *Adversarial Distribution* (Section 4.3.1). Next, we propose our refinement method to reduce the perturbation size, while maintaining the guarantees of attack success probability (Section 4.3.2). We also provide the theoretical analysis of the convergence and confidence bound of the Shifting method.

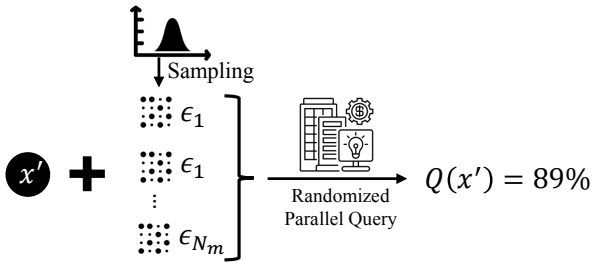


Fig. 4.3: Illustration of randomized parallel query (returning the probability $Q(x')$ that $x' + \epsilon$ is an adversarial example).

4.3.1 Adversarial Distribution Localization

Randomized Parallel Query

As stated, computing the exact probability $\mathbb{P}[f(x_{adv}) \neq y]$ with $x_{adv} \sim \varphi(x', \boldsymbol{\kappa})$ is intractable. Here, we propose to estimate its low bound probability by the Monte Carlo method. This requires the adversary to query the classifier with random instances sampled from an *Adversarial Distribution*. By noting that random instances can be queried efficiently in parallel, we propose the Randomized Parallel Query (RPQ) to compute the lower bound of the attack success probability as below:

$$\begin{aligned} Q(x') = \underline{p_{adv}} &\leq \mathbb{P}_{x_{adv} \sim \varphi(x', \boldsymbol{\kappa})}[f(x_{adv}) \neq y] \\ &= \mathbb{P}_{\epsilon \sim \varphi(0, \boldsymbol{\kappa})}[f(x' + \epsilon) \neq y]. \end{aligned} \tag{4.3}$$

With a given x', the lower bound probability $\underline{p_{adv}}$ can be estimated via the Binomial testing on a zero-mean distribution $\varphi(0, \boldsymbol{\kappa})$ using Clopper-Pearson confidence interval [22] following the Algorithm 3, where the LowerConfBound$(k, N_m, 1 - \alpha)$ returns the

Algorithm 3 Lower Bound of Attack Success Probability

Input: Mean x' of the *Adversarial Distribution* φ, classifier f, confidence level α, Monte Carlo samples N_m, ground truth label y.

Output: The lower bound of attack success probability $\underline{p_{adv}}$

1: $\epsilon_1, \epsilon_2, ..., \epsilon_{N_m} \sim \varphi(0, \kappa)$

2: Incorrect prediction count $k \leftarrow \sum_{N_m}^{i=1} \mathbf{1}[f(x' + \epsilon_i) \neq y]$

3: **return** $\underline{p_{adv}} \leftarrow \textsc{LowerConfBound}(k, N_m, 1 - \alpha)$

one-sided $(1 - \alpha)$ lower confidence interval.

Now we can estimate $\underline{p_{adv}}$ given an *Adversarial Distribution* with known mean/location x'. The next question is how to decide x' to satisfy Eq. (4.1), i.e., locating the adversarial distribution that includes certifiable adversarial examples (with probability at least p).

The simplest way is random localization, where the input x is uniformly sampled from the input space $[\Pi_a, \Pi_b]^d$, e.g., $[0,\ 1]^d$, followed by the RPQ to check if $\underline{p_{adv}}$ is larger than p. However, random localization could not generate a good initial adversarial distribution due to the high-dimensional input space. Below we propose two practical localization methods to mitigate the issue.

Proposed Localization Algorithms

We notice the adversarial distribution localization is similar to empirical black-box attacks on generating adversarial examples. Here, we propose to adapt these empirical attack algorithms and design two localization algorithms.

Algorithm 4 Smoothed Self-Supervised Perturbation (SSSP)

Input: Clean input x, feature extractor $\mathcal{F}$, noise distribution $\varphi(0, \boldsymbol{\kappa})$, maximum iterations n_{max}, perturbation budget π, step size η, and noise sampling number N_s.

Output: Updated mean x' of *Adversarial Distribution*

$x' = x$

for $n = 1$ to n_{max} **do**

  $\mathcal{L}(x') \leftarrow \frac{1}{N_s} \sum_i^{N_s} [\|\mathcal{F}(x' + \epsilon_i) - \mathcal{F}(x + \epsilon_i)\|_2],\ \epsilon_i \sim \varphi$

  $x' \leftarrow x' + \eta\ sgn(\nabla_{x'} \mathcal{L})$

  $x' \leftarrow Clip(x',\ x - \pi,\ x + \pi)$

  $x' \leftarrow Clip(x',\ 0.0,\ 1.0)$ (if x is an image)

return x'

Smoothed Self-Supervised Localization: To better locate the *Adversarial Distribution*, we propose to adapt the self-supervised perturbation (SSP) technique [109]. Specifically, SSP generates generic adversarial examples by distorting the features extracted by a pre-trained feature extractor on a large-scale dataset in a self-supervised manner. The rationale is that the extracted (adversarial) features can be transferred to other classifiers as well.

As our attack uses RPQ, we compute the feature distortion over *a set of random samples* from the *Adversarial Distribution*. Formally,

$$x' = \arg\max_{x'}\ \mathbb{E}_{\epsilon \sim \varphi(0,\boldsymbol{\kappa})}[\|\mathcal{F}(x' + \epsilon) - \mathcal{F}(x + \epsilon)\|_2]$$
$$\text{s.t. } \|x' - x\|_\infty \leq \pi \tag{4.4}$$

Algorithm 5 Smoothed SSP for Certifiable Attack Localization

Input: Clean input x, feature extractor $\mathcal{F}(\cdot)$, RPQ function $Q(\cdot)$, smoothed SSP algorithm $SSSP(\cdot)$ (Algorithm 4), initial perturbation budget π_{init}, step size γ, ASP Threshold p, maximum iterations N_{max}.

Output: Mean x' of *Adversarial Distribution* φ, number of RPQs q.

$x' = x, \pi = \pi_{init}, N = 0, q = 0$

while $Q(x') < p$ and $N < N_{max}$ **do**

    $N \leftarrow N + 1, q \leftarrow q + 1, \pi \leftarrow \pi + \gamma$

    $x' \leftarrow SSSP(x', \mathcal{F}, \pi)$

if $Q(x') < p$ **then**

    return Abstain

else

    return x' and q

where $\mathcal{F}$ is a pre-trained feature extractor. The perturbation budget π is initially set to a small value and later increased in multiple attempts of localization, ensuring that smaller perturbations are identified first. This optimization problem can be solved via the Projected Gradient Ascent method [91]. Let the adversarial loss be $\mathcal{L}(x') \equiv \mathbb{E}_{\epsilon\sim\varphi}[\|\mathcal{F}(x' + \epsilon) - \mathcal{F}(x + \epsilon)\|_2]$. Then we can locate the *Adversarial Distribution* via iteratively update x' with $x' = x' + \eta\ sgn(\nabla_{x'}\mathcal{L})$, where $sgn(\cdot)$ is the sign function, and η denotes the step size. The details for localizing the *Adversarial Distribution* are summarized in Algorithms 4 and 5.

Binary Search Localization: Another method is to randomly initialize the location

Algorithm 6 Binary Search for Certifiable Attack Localization

Input: Clean input x, RPQ function $Q(\cdot)$, ASP Threshold p, random search iterations N_r, and binary search iteration N_b, error tolerance Ω.

Output: Mean of initial *Adversarial Distribution* x', number of RPQs q.

$n = 0, m = 0, q = 0, x^* = x$

while $Q(x') < p$ and $n \leq N_r$ **do**

  $x' \sim \text{Uniform}([0,1]^d)$

  $q \leftarrow q + 1,$

if $n > N_r$ **then return Abstain**

while $m < N_b$ and $\|x' - x^*\|_2 \leq \Omega$ **do**

  if $Q(\frac{x^*+x'}{2}) \geq p$ **then**

    $x' = \frac{x^*+x'}{2}$

  else

    $x^* = \frac{x^*+x'}{2}$

return x'

of *Adversarial Distribution* such that $\underline{p_{adv}} \geq p$, and then reduce the gap between $\underline{p_{adv}}$ and p, as well as the perturbation via binary search. The algorithm is presented in Algorithm 6. This method is efficient in reducing the perturbation size once the feasible *Adversarial Distribution* is found by random search. Figure 4.6 and 4.7 visualize some x_{adv} during the crafting process for both Binary Search Localization and SSSP Localization.

4.3.2 Adversarial Distribution Refinement

Though our localization algorithms can find an effective *Adversarial Distribution*, our empirical results found the perturbation size can be large (See Table 4.4.4). This occurs possibly because the pretrained feature extractor is too generic and the generated adversarial perturbation is suboptimal for our target classifier. To mitigate the issue, we propose to reduce the perturbation by refining the *Adversarial Distribution* while still maintaining the condition Eq. (4.1).

Our key observation is that the optimal perturbation is achieved when the adversarial example is close to the decision boundary of the target classifier. Hence, we propose to shift the *Adversarial Distribution* until intersecting the decision boundary, thereby locating the locally optimal point on that boundary.

Certification for *Adversarial Distribution* Shifting

We propose a theory on shifting the *Adversarial Distribution* while maintaining the attack success probability. We denote $\varphi(x' + \delta, \boldsymbol{\kappa})$ as a shifted distribution for the *Adversarial Distribution* $\varphi(x', \boldsymbol{\kappa})$ by a shifting vector δ. Then, the shifted *Adversarial Distribution* ensures the ASP if δ satisfies the condition presented in Theorem 6.

Theorem 6. *(**Certifiable Adversarial Distribution Shifting**) Let f be a classifier, ϵ be the noise drawn from any continuous probability density function $\varphi(0, \boldsymbol{\kappa})$. Let p be the predefined attack success possibility threshold. Denote $\underline{p_{adv}}$ as the lower bound of the attack success probability. For any x' satisfies*

$$\mathbb{P}[f(x' + \epsilon) \neq y] \geq \underline{p_{adv}} = Q(x') \geq p, \tag{4.5}$$

$\mathbb{P}[f(x' + \delta + \epsilon) \neq y] \geq p$ *is guaranteed for any shifting vector* δ *when*

$$\Phi_+[\Phi_-^{-1}(\underline{p_{adv}})] \geq p \tag{4.6}$$

where Φ_-^{-1} *is the inverse cumulative density function (CDF) of the random variable* $\frac{\varphi(\epsilon-\delta,\boldsymbol{\kappa})}{\varphi(\epsilon,\boldsymbol{\kappa})}$*, and* Φ_+ *the CDF of random variable* $\frac{\varphi(\epsilon,\boldsymbol{\kappa})}{\varphi(\epsilon+\delta,\boldsymbol{\kappa})}$*.*

Proof. See detailed proof in Appendix 7.12.1. □

Theorem 6 ensures the minimum attack success probability if Eq. (4.5) and Eq. (4.6) hold while without querying $\varphi(x' + \delta, \boldsymbol{\kappa})$. Eq. (4.5) requires finding a x' such that the RPQ on samples of $\varphi(x', \boldsymbol{\kappa})$ returns a $\underline{p_{adv}} \geq p$, and Eq. (4.6) ensures any δ meeting this condition will not reduce the attack success probability of the shifted *Adversarial Distribution* below p. Further, Theorem 6 works for any continuous noise distributions, e.g., Gaussian, Laplace, Exponential, and mixture PDFs. We also present the case when the noise is Gaussian in Corollary 11.1 in Appendix 7.12.2. It shows the shifting perturbation δ should satisfy $\|\delta\|_2 \leq \sigma[\Phi^{-1}(\underline{p_{adv}}) - \Phi^{-1}(p)]$, where Φ^{-1} is the inverse of Gaussian CDF.

Obtaining Refined *Adversarial Distribution*

Since the *Adversarial Distribution* can be shifted by any δ satisfying Eq. (4.6), we propose to shift it toward the clean input with the maximum δ that does not break the

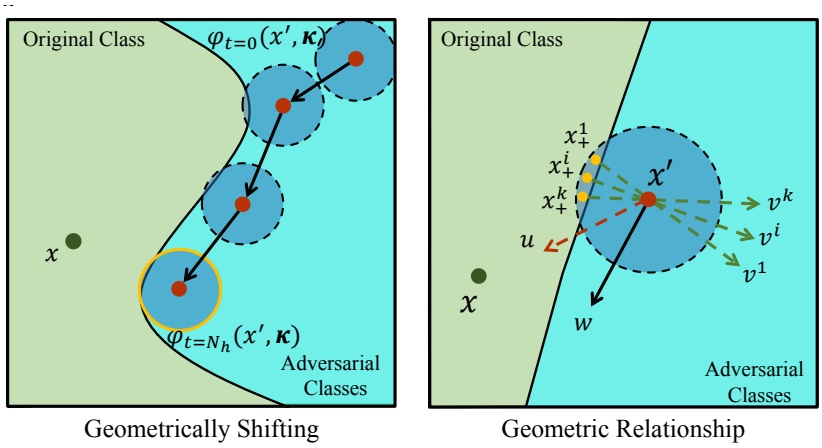


Fig. 4.4: Illustration of geometrically shifting.

guarantee. By iteratively executing the RPQ and applying the Theorem 6, the *Adversarial Distribution* can be repeatedly shifted with a guarantee until approaching the decision boundary (where $\underline{p_{adv}} = p$).

The problem of shifting the *Adversarial Distribution* to reduce the perturbation can be solved by two steps: *first finding the shifting direction, and then deriving the shifting distance while maintaining the guarantee*. Here, we design a novel shifting method to find the locally optimal *Adversarial Distribution* by considering the geometric relationship between the decision boundary and the *Adversarial Distribution*, which is called "Geometrical Shifting". Specifically, through using the noisy samples of *Adversarial Distribution* to "probe" the decision boundary, we shift the RandAE along the decision boundary and towards the clean input until finding the local optimal point on the decision boundary (see Figure 4.4 for the illustration). If none of the noisy samples can approach the decision boundary, we simply shift the *Adversarial Distribution* directly toward the clean input without considering the decision boundary.

Algorithm 7 Shifting Direction

Input: Mean of the *Adversarial Distribution* x', clean input x, vectors $\{v^i\}$, a vector u, maximum iteration M, updating step size η'.

Output: The shifting direction w

Initialize w with random noise

if $\{v^i\}$ is empty **then**

    $w = x - x'$

else

    for $j = 1$ to M **do**

        $w \leftarrow w + \eta' \ sgn[\nabla_w(\sum_{i=1}^{k} \sin(v^i, w) + \cos(u, w))]$

        $w \leftarrow \frac{w}{\|w\|_2}$

return w

Finding the Shifting Direction: The geometrical relationship is presented on the right-hand side of Figure 4.4. Denote x' as the mean of the current *Adversarial Distribution*. When sampling the adversarial examples from the *Adversarial Distribution*, we mark the failed adversarial examples as $x_+^1, x_+^2, ..., x_+^i, ..., x_+^k$, aka., "samples fell into the original class". The normalized vector from x_+^i to x' is denoted as v^i. The normalized vector from x' to x is denoted as u. If the *Adversarial Distribution* has no samples crossing the decision boundary, then we can shift the *Adversarial Distribution* straight toward the clean input (along the direction of u) until it intersects the decision boundary, otherwise, the shifting should be along the decision boundary but not cross it (without changing the certifiable attack guarantee). Note that the input space is high-dimensional, thus

Algorithm 8 Shifting Distance

Input: Mean of *Adversarial Distribution* x', noise distribution φ, randomized query function $Q(\cdot)$, the shifting direction algorithm $SD(\cdot)$ (Algorithm 7), error threshold e, ASP Threshold p.

Output: The shifting perturbation δ

$w \leftarrow SD(x')$, $\underline{p_{adv}} \leftarrow Q(x')$

find a scalar a such that $\delta = aw$ and $\Phi_+[\Phi_-^{-1}(\underline{p_{adv}})] > p$

find a scalar b such that $\delta = bw$ and $\Phi_+[\Phi_-^{-1}(\underline{p_{adv}})] < p$

while $\Phi_+[\Phi_-^{-1}(\underline{p_{adv}})] < p$ or $> p + e$ and $n \leq N_k$ **do**

    if $\Phi_+[\Phi_-^{-1}(\underline{p_{adv}})] > p$ **then**

        $a \leftarrow \frac{(a+b)}{2}$

    else

        $b \leftarrow \frac{(a+b)}{2}$

    $\delta \leftarrow \frac{(a+b)}{2} w$, $n \leftarrow n + 1$

return δ

there could be many directions along the decision boundary. To reduce the perturbation, the direction should be similar to the vector u as much as possible. Based on these geometric analyses, the goals of the geometrical shifting can be summarized as: *The shifting direction should lie relatively parallel to the direction of u; and be relatively vertical to the vectors* v^i.

Formally, denoting the shifting direction as w, then the goal of finding the shifting

Algorithm 9 Certifiable Attack Shifting

Input: Mean of *Adversarial Distribution* x', noise distribution φ, randomized query function $Q(\cdot)$, shifting distance algorithm SHIFT$(\cdot)$ (Algorithm 8), distance threshold e_s, ASP Threshold p, max iteration N_h.

Output: The shifted mean x'

1: $\underline{p_{adv}} \leftarrow Q(x')$, $\delta \leftarrow$ SHIFT(x'), $n = 0$

2: **while** $\underline{p_{adv}} > p$ and $\|\delta\|_2 \geq e_s$ and $n \leq N_h$ **do**

3: $\quad x' \leftarrow x' + \delta$, $\underline{p_{adv}} \leftarrow Q(x')$, $\delta \leftarrow$ SHIFT(x'), $n \leftarrow n + 1$

4: $\quad$ **if** $\|x' - x\|_2 \leq \|\delta\|_2$ **then return** x

5: **return** x'

direction can be formulated as:

$$w = \arg\max \sum_{i=1}^{k} \sin(v^i, w) + \cos(u, w) \tag{4.7}$$

where $\sin(\cdot)$ and $\cos(\cdot)$ denote the sine and cosine function, and Eq. (4.7) can be solved via the gradient ascent algorithm.

Calculating the Shifting Distance: The shifting distance can be determined by maximizing $\|\delta\|_2$ that satisfies the constraint of Eq. (4.6), i.e., when the equality holds. We use binary search to approach the equality and the Monte Carlo method to estimate the CDF of random variable $\frac{\varphi(\epsilon-\delta,\kappa)}{\varphi(\epsilon,\kappa)}$ and $\frac{\varphi(\epsilon,\kappa)}{\varphi(\epsilon+\delta,\kappa)}$, similar to [4]. Algorithm 7, 8, and 9 show the details of finding the shifting direction, shifting distance, and the shifting process, respectively.

Convergence Guarantee and Confidence Bound: Any δ computed by Algorithm 8 will satisfy the certifiable attack guarantee since it strictly ensures $\Phi_+[\Phi_-^{-1}(\underline{p_{adv}})] \geq p$.

Further, with a centralized noise distribution, the shifting algorithm is guaranteed to converge once the located *Adversarial Distribution* is feasible.

Theorem 7. *If the PDF of noise distribution $\varphi(x)$ decreases as the $|x|$ increases, with the satisfaction of Eq. (4.5), given any direction vector w, the Shifting Distance algorithm guarantees to find δ such that $\Phi_+[\Phi_-^{-1}(\underline{p_{adv}})] = p$ with confidence $(1-\alpha)(1-2e^{-2N_m\Delta^2})^2$, where $(1-\alpha)$ is the confidence for of estimating $\underline{p_{adv}}$, N_m is the Monte Carlo samples, and Δ is the error bound for the CDF estimation.*

Proof. See detailed proof in Appendix 7.12.3. □

4.3.3 Discussions on Our Attack

Realizing Our Certifiable Attack: Our certifiable attack does not have extra requirements on realization compared to empirical black-box attacks. To implement our attack, we only need to predefine a continuous noise distribution and a threshold of certified attack success probability. The adversary then adds the noise sampled from the distribution to the inputs and queries the target model. Then, *Adversarial Distribution* can be crafted by RPQ and our theory.

Randomized Query vs. Deterministic Query: The proposed randomized query returns a *probability* over a batch of inputs with injected random noises, while the traditional query returns a *deterministic* output (score or hard label) from the target model. This probability return may provide more information that better guides the attack. In addition, the randomized queries can be executed in parallel for query acceleration. See

results in Section 4.4.3.

Imperceptibility with Diffusion Denoiser: The certifiable adversarial examples sampled from *Adversarial Distribution* are noise-injected inputs that still might be perceptible when the noise is large. We can further leverage the recent innovation for image synthesis, i.e., diffusion model [130], to denoise the adversarial examples for better imperceptibility. The key idea is to consider the noise-perturbed adversarial examples as the middle sample in the forward process of the diffusion model [85, 86]. This is shown to improve the imperceptibility and the diversity of the adversarial examples. More technical details are shown in Appendix 7.13 and results in Table 7.7 in Appendix 7.14.4.

Extension to Certifiable White-Box Attack: Our certifiable attack can be readily extended to the white-box setting by adapting/designing a white-box localization method. Specifically, the Smoothed SSP localization method can directly compute the gradients of the noise-perturbed examples rather than leveraging the feature extractor, which may significantly improve the certified accuracy of the certifiable attack. In our experiments, when leveraging the PGD-like white-box attacks as the localization method, the certified accuracy can be increased to 100% for ResNet and CIFAR10, compared to the 92.54% certified accuracy in the black-box setting.

Extension to Targeted Certifiable Attack: Our attack design focuses on the untargeted certifiable attack. It can also be generalized to the targeted attack setting, where we require the majority of the noise-perturbed inputs to be *certifiably* misclassified to

a specific *target* label. However, we admit it would be more challenging to find a successful *Adversarial Distribution* in this scenario.

Attacks under Adaptive Blacklight: The defender might design an adaptive countermeasure, such as an adaptive blacklight defense, to mitigate certified attacks. For instance, the defender could attempt to eliminate randomness by assuming the noise distribution is known. However, this approach presents several challenges: 1) The defender would need detailed knowledge about the attack's design, including the noise distribution, which is often an impractical assumption. 2) Even if the noise distribution were known, the sampled adversarial examples would remain random, making it difficult to accurately estimate the center of the noise distribution.

4.4 Evaluations

We comprehensively evaluate our certifiable black-box attack in various experimental settings. Particularly, we would like to study the following research questions:

- **RQ1:** How effective is the learnt *Adversarial Distribution*? Particularly, how large is the probability of samples from it being successful adversarial examples?

- **RQ2:** Can our certifiable attack outperform empirical attacks in terms of attack effectiveness and query efficiency?

- **RQ3:** How effective is our attack to break SOTA defenses?

Table 4.2: Summary of Experiments

Experiments	Dataset	Model	Reference
Comparison with empirical attacks against Blacklight detection	CIFAR10	VGG16	Table 7.8
	CIFAR10	ResNet110	Table 7.9
	CIFAR10	ResNext29	Table 7.10
	CIFAR10	WRN28	Table 7.11
	CIFAR100	VGG16	Table 7.12
	CIFAR100	ResNet110	Table 7.13
	CIFAR100	ResNext29	Table 7.14
	CIFAR100	WRN28	Table 7.15
	ImageNet	ResNet18	Table 4.3
Comparison with empirical attacks against RAND pre-processing defense	CIFAR10	VGG16	Table 7.16
	CIFAR10	ResNet110	Table 7.17
	CIFAR10	ResNext29	Table 7.18
	CIFAR10	WRN28	Table 7.19
	CIFAR100	VGG16	Table 7.20
	CIFAR100	ResNet110	Table 7.21
	CIFAR100	ResNext29	Table 7.22
	CIFAR100	WRN28	Table 7.23
	ImageNet	ResNet18	Table 4.4
Comparison with empirical attacks against RAND post-processing defense	CIFAR10	VGG16	Table 7.24
	CIFAR10	ResNet110	Table 7.25
	CIFAR10	ResNext29	Table 7.26
	CIFAR10	WRN28	Table 7.27
	CIFAR100	VGG16	Table 7.28
	CIFAR100	ResNet110	Table 7.29
	CIFAR100	ResNext29	Table 7.30
	CIFAR100	WRN28	Table 7.31
	ImageNet	ResNet18	Table 4.5
Comparison with empirical attack against adversarial training	CIFAR10	ResNet110 (ℓ_2)	Table 4.6
	CIFAR10	ResNet110 (ℓ_∞)	
Ablation: CA vs. different noise variance	CIFAR10	ResNet110	Table 4.7
	ImageNet	ResNet50	
	LibriSpeech	ECAPA-TDNN	Table 7.32
Ablation: CA vs. different p	CIFAR10	ResNet110	Table 4.8
	ImageNet	ResNet50	
	LibriSpeech	ECAPA-TDNN	Table 7.33
Ablation: CA vs. different Localization/Shifting	CIFAR10	ResNet110	Table 4.9
	ImageNet	ResNet50	
	LibriSpeech	ECAPA-TDNN	Table 7.34
Ablation: CA vs. different noise PDF	CIFAR10	ResNet110	Table 4.10
	ImageNet	ResNet50	
Ablation: CA w/ and w/o Diffusion Denoise	CIFAR10	ResNet110	Table 7.7
	ImageNet	ResNet50	
CA vs. Feature Squeezing	CIFAR10	ResNet110	Figure 7.9
CA vs. Adaptive Denoiser	CIFAR10	ResNet110	Table 4.11
CA vs. Rand. Smoothing	CIFAR10	ResNet110	Table 7.6

- **RQ4:** What is the impact of the design components and their hyperparameters on our attack?

Accordingly, we first assess the empirical attack success possibility of the *Adversarial Distribution* in Section 4.4.2. Then, we evaluate our certifiable attack on various models with defenses while benchmarking with empirical black-box attacks in Section 4.4.3. In Section 4.4.4, we conduct ablation studies to explore in-depth our certifiable attack. *All sets of experiments are summarized in Table 4.2 for reference.*

4.4.1 Experimental Setup

Datasets and Models. We use three benchmark datasets for image classification: CIFAR10/CIFAR100 [28] and ImageNet [29]. CIFAR10 and CIFAR100, both consisting of $60,000$ $32x32$ color images split into 10 and 100 classes, respectively. ImageNet is a large-scale dataset with $1,000$ classes. The training set contains $1,281,167$ images and the validation set contains $50,000$ images (resized to $3\times224\times224$). we use VGG [131], ResNet [36], ResNext [132], and WRN [133] as the target model. We use a pre-trained ResNet34 on ImageNet as the feature extractor (in the Smoothed SSP localization). We also test our attacks on the audio dataset LibriSpeech [134] for the speaker verification task, and results are shown in Appendix 7.14.5.

Baseline Attacks. We compare our certifiable (hard label-based) black-box attack with SOTA black-box attacks including 7 *hard label-based* black-box attacks: GeoDA [123], HSJ [93], Opt [104], RayS [124], SignFlip [125], SignOPT [126], and Boundary [101];

and 7 *score-based* black-box attacks: Bandit [119], NES [98], Parsimonious [120], Sign [121], Square [99], ZOSignSGD [122], Simple attack [112]. As our method does not constrain the perturbation budget but minimizing the perturbation, we also compare with two similar attacks: SparseEvo [128] and PointWise [127]. We evaluate our attack with both SSSP localization and binary-search localization. For optimized-based attacks, we limit the AEs in the valid image space. For a fair comparison with optimization-based attacks, the perturbation budget for ℓ_p-bounded attacks are set to 0.1 for ℓ_∞ and 5 for ℓ_2 on CIFAR10 and CIFAR100, while on ImageNet, they are $\ell_\infty = 0.1$ and $\ell_2 = 40$. The maximum query limits are $10,000$ for CIFAR10 and CIFAR100 and $1,000$ for ImageNet. We evaluate $1,000$ randomly selected images for each dataset.

Defenses. We select 4 SOTA defenses against black-box attacks for evaluation: Blacklight detection [113], Randomized pre-processing defense (RAND-Pre) [114], Randomized post-processing defense (RAND-Post) [115], and Adversarial Training based TRADES [135]. Blacklight has recently proposed to mitigate query-based black-box attacks by utilizing the similarity among queries. It has been shown to detect 100% adversarial examples generated in multiple attacks. RAND-Pre and RAND-Post respectively add noise to the inputs and prediction logits to obfuscate the gradient estimation or local search. TRADES has demonstrated SOTA robustness performance against adversarial attacks by training on adversarial examples.

Metrics. We use the below metrics to evaluate all compared attacks.

- **Model Accuracy:** the model accuracy under attack and defense.

- **Number of RPQ (# RPQ):** the number of the randomized parallel query for certifiable attack.

- **Number of Query (# Q):** the total number of queries for empirical attack. For our method, it is equal to Monte Carlo Sampling Number $\times$ # RPQ + additional queries for sampling from the *Adversarial Distribution*.

- **Certified Accuracy@p:** the certified accuracy at the ASP Threshold p. It is the percentage of the testing samples that have the certified ASP at least p, e.g., a 95% certified accuracy with ASP Threshold $p = 90\%$ means the adversary can guarantee to have 90% probability to attack successfully for 95% testing samples.

- ℓ_2 **Perturbation Size (Dist. ℓ_2):** ℓ_2 distance between the adversarial example x_{adv} and the clean input x, i.e., $\|x_{adv} - x\|_2$.

- ℓ_2 **Mean Distance (Mean Dist. ℓ_2):** ℓ_2 distance between the mean x' of *Adversarial Distribution* and clean input x, i.e., $\|x' - x\|_2$.

- **Detection Success Rate (Det. Rate):** the detection success rate of Blacklight detection.

- **Average # Queries for Detection (# Q to Det.):** the average number of queries before Blacklight detects an AE.

- **Detection Coverage (Det. Cov.):** the percent of queries in an attack's query sequence that Blacklight identified as attack queries.

Parameters Settings. There exist many parameters that may affect the performance of our certifiable attack. For instance, the Monte Carlo sampling number, the attack success probability p, and the family of the adversarial distribution and its parameters. If not specified, we set Monte Carlo sampling number to be 50, $p = 10\%$, and use Gaussian distribution with variance $\sigma = 0.025$. We will also study the impact of these parameters in Section 4.4.3. All the parameter details are summarized in Table 7.5 in Appendix 7.14.1.

Experimental Environment. We implemented a PyTorch library[2] including 16 black-box attacks, 4 defenses, 6 datasets, and 9 models by integrating several open-source libraries[3]. The experiments were run on a server with AMD EPYC Genoa 9354 CPUs (32 Core, 3.3GHz), and NVIDIA H100 Hopper GPUs (80GB each).

4.4.2 Verifying the Adversarial Distribution

We first assess the ASP of the crafted *Adversarial Distribution*. Specifically, given an ASP Threshold p and the certified *Adversarial Distribution* $\varphi(x', \boldsymbol{\kappa})$, we randomly sample $1,000$ examples $x_{adv} \sim \varphi(x', \boldsymbol{\kappa})$, and query the model. We visualize the query results for 4 certified *Adversarial Distribution*s with different p using 2D t-SNE[4] [136]. We also report the provable lower bound of ASP $\underline{p_{adv}}$ and the empirical ASP in Figure 4.5. It validates that the sampled AEs ensure the minimum ASP via the *Adversarial*

[2] The codes are available at `https://github.com/datasec-lab/CertifiedAttack`

[3] BlackboxBench, pytorch image classification, Blacklight, SparseEvo, and TRADES

[4] t-SNE reduces the prediction logits of the random samples to 2-dimension.

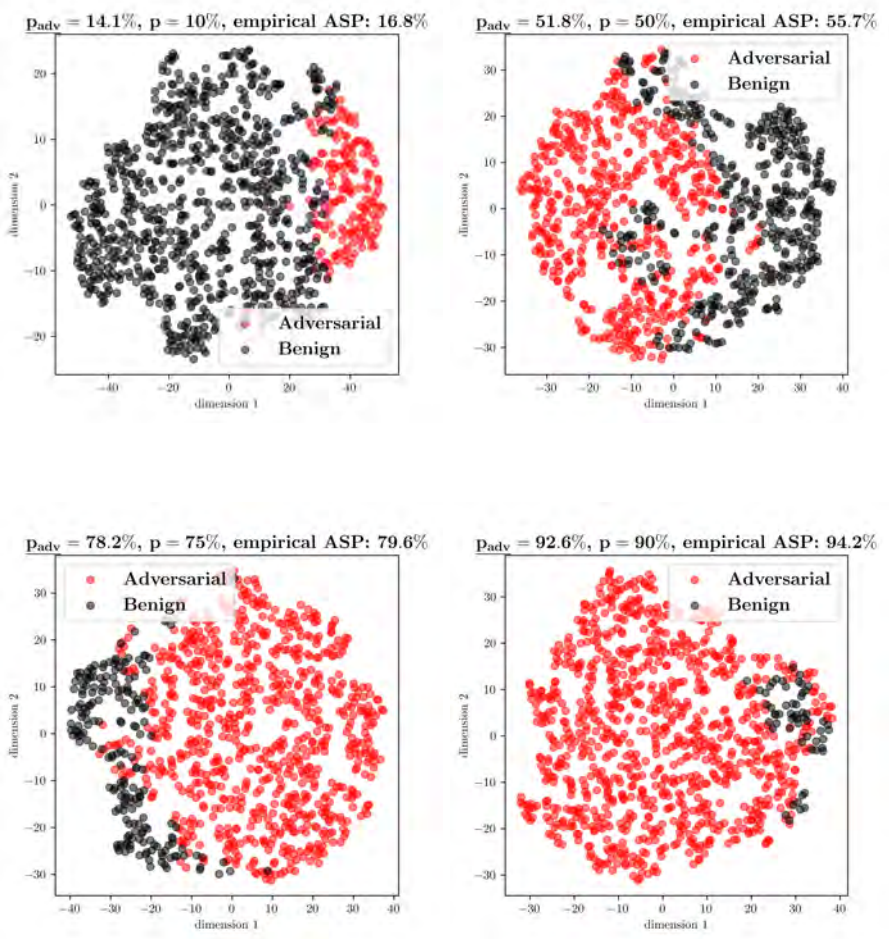


Fig. 4.5: t-SNE visualization of adversarial example sampling from the adversarial distribution.

Distribution, and the *Adversarial Distribution* lies on the decision boundary.

4.4.3 Attack Performance against SOTA Defenses

In this section, we evaluate our certifiable attack and empirical attacks against the 4 studied SOTA defenses.

Attack Performance under Blacklight Detection [113]

We use the default setting from [113] with a threshold of 25. The results are presented in Table 4.3 and Tables 7.8-7.15 in Appendix 7.14. We have the following key observations: 1) Our certifiable attack consistently circumvents Blacklight with 0% detection success rate, and 0% detection coverage on all settings. This indicates that none of the queries from our attack are detected. In contrast, existing black-box attacks are highly susceptible to Blacklight, with most achieving a 100% detection success rate on various datasets and models. Even the most resilient attack, as shown in Appendix 7.14, Table 7.12, attains an 86.5% detection success rate on the CIFAR100 dataset using the VGG16 model. 2) With the strong ability to bypass the detection, our certifiable attack still maintains top attack performance on all the datasets and models such that the model accuracy can be attacked to 0% with moderate ℓ_2 perturbation size and few queries. The high attack accuracy and low detection rate of certifiable attacks stem from the randomness of *Adversarial Distribution* and the guarantee of the attack success probability.

Attack Performance under RAND-Pre [114]

We follow [114] to inject the Gaussian noise with standard deviation 0.02 to the query (in the input space). The experimental results are presented in Table 4.4 and Tables 7.16-7.23 in Appendix 7.14. Based on a comprehensive analysis of all results, it is evident that the RAND-Pre consistently reduces the attack success rate of existing black-

Table 4.3: Attack performance under Blacklight detection on ResNet and ImageNet (Clean Accuracy: 67.9%)

Attack	Query Type	Pert. Type	Det. Rate %	# Q to Det.	Det. Cov. %	Model Acc.	# Q	Dist. ℓ_2
Bandit	Score	ℓ_∞	100.0	1.0	64.2	1.9	25	25.42
NES	Score	ℓ_∞	100.0	10.3	17.3	7.0	337	8.28
Parsimonious	Score	ℓ_∞	100.0	2.0	96.7	3.8	282	25.24
Sign	Score	ℓ_∞	100.0	2.0	91.5	0.5	126	25.50
Square	Score	ℓ_∞	100.0	2.0	66.9	0.0	14	25.54
ZOSignSGD	Score	ℓ_∞	100.0	2.0	50.2	12.5	322	8.53
GeoDA	Label	ℓ_∞	100.0	1.0	88.9	5.1	151	17.99
HSJ	Label	ℓ_∞	100.0	7.3	94.9	35.6	212	9.82
Opt	Label	ℓ_∞	99.9	8.4	81.4	61.2	646	0.98
RayS	Label	ℓ_∞	100.0	4.4	83.5	4.2	260	29.63
SignFlip	Label	ℓ_∞	100.0	8.5	70.3	4.4	148	27.64
SignOPT	Label	ℓ_∞	99.9	8.4	69.8	55.9	570	1.32
Bandit	Score	ℓ_2	100.0	1.0	99.5	1.7	431	9.60
NES	Score	ℓ_2	100.0	10.2	32.8	61.2	571	0.45
Simple	Score	ℓ_2	100.0	1.0	99.9	53.6	883	0.88
Square	Score	ℓ_2	100.0	2.0	68.8	0.0	16	26.30
ZOSignSGD	Score	ℓ_2	100.0	2.0	52.4	65.1	531	0.30
Boundary	Label	ℓ_2	100.0	7.2	76.3	37.9	60	11.63
GeoDA	Label	ℓ_2	100.0	1.0	89.3	3.9	181	19.14
HSJ	Label	ℓ_2	100.0	7.3	93.4	11.4	255	22.21
Opt	Label	ℓ_2	100.0	8.5	67.9	41.2	610	16.71
SignOPT	Label	ℓ_2	99.9	8.4	62.9	36.7	485	17.54
PointWise	Label	Opt.	100.0	1.0	99.8	0.0	920	13.53
SparseEvo	Label	Opt.	100.0	1.0	99.9	0.0	1000	7.68
CA (sssp)	Label	Opt.	0.0	∞	0.0	1.4	148	13.74
CA (bin search)	Label	Opt.	0.0	∞	0.0	0.0	603	33.14

Table 4.4: Attack performance under RAND Pre-processing Defense on ResNet and ImageNet (Clean Accuracy: 67.0%)

Attack	Query Type	Perturbation Type	# Query	Model Acc.	Dist. ℓ_2
Bandit	Score	ℓ_∞	10	6.7	25.26
NES	Score	ℓ_∞	428	49.8	10.26
Parsimonious	Score	ℓ_∞	243	62.7	25.12
Sign	Score	ℓ_∞	116	40.6	25.20
Square	Score	ℓ_∞	27	10.4	24.96
ZOSignSGD	Score	ℓ_∞	428	49.4	10.36
GeoDA	Label	ℓ_∞	150	40.0	18.08
HSJ	Label	ℓ_∞	232	58.5	8.76
Opt	Label	ℓ_∞	905	69.4	0.44
RayS	Label	ℓ_∞	235	47.9	28.14
SignFlip	Label	ℓ_∞	46	52.8	13.06
SignOPT	Label	ℓ_∞	394	59.1	0.39
Bandit	Score	ℓ_2	583	58.2	12.99
NES	Score	ℓ_2	341	66.8	0.43
Simple	Score	ℓ_2	258	67.2	0.10
Square	Score	ℓ_2	18	13.6	25.96
ZOSignSGD	Score	ℓ_2	249	67.3	0.28
Boundary	Label	ℓ_2	38	49.2	15.12
GeoDA	Label	ℓ_2	149	47.4	17.70
HSJ	Label	ℓ_2	225	55.7	14.30
Opt	Label	ℓ_2	1000	58.2	12.28
SignOPT	Label	ℓ_2	406	52.2	15.41
PointWise	Label	Optimized	942	54.6	16.90
SparseEvo	Label	Optimized	1000	61.7	11.33
CA (sssp)	Label	Optimized	154	1.7	13.98
CA (bin search)	Label	Optimized	603	0.0	32.16

box attacks. Specifically, the defense reduces the average attack success rate of empirical black-box attacks from 92% to 30% on CIFAR10, from 95% to 29% on CIFAR100, and from 69% to 25% on ImageNet, respectively. However, our attack still achieves the average attack success rate of 93%, 99%, and 99% respectively on the three datasets under RAND-Pre. Further, we highlight that, with RAND-Pre applied across all datasets and models, the average ℓ_2 perturbation size and number of queries in our certifiable attack *decrease by* 4.2% *and* 2.1%, respectively. This intriguing observation matches our findings in Section 4.4.4 where a larger variance leads to smaller ℓ_2 mean distance and # RPQ in the certifiable attack. This is because the Gaussian noise injected by the defense (e.g., $\epsilon_1 \sim \mathcal{N}(0, v)$) is added to the adversary's noise (e.g., $\epsilon_2 \sim \mathcal{N}(0, u)$), leading to a larger variance $v + u$ and hence further enhancing our attack.

Attack Performance under RAND-Post [115]

We follow [115] to inject the Gaussian noise with standard deviation 0.2 to the output logits of each query (applied to both hard label-based and score-based attacks). The experimental results are presented in Table 4.5, and Tables 7.24-7.31. Similarly, we find that RAND-Post can strongly degrade the average attack success rate of hard label-based empirical attacks from 84% to 41% on CIFAR10, from 89% to 45% on CIFAR100, and from 60% to 24% on ImageNet, respectively. On the other hand, it moderately degrades the average attack success rate of score-based empirical attacks from 100% to 91% on CIFAR10, from 100% to 95% on CIFAR100, and from 72%

Table 4.5: Attack performance under RAND Post-processing Defense on ResNet and ImageNet (Clean Accuracy: 68.0%)

Attack	Query Type	Perturbation Type	# Query	Model Acc.	Dist. ℓ_2
Bandit	Score	ℓ_∞	17	2.7	25.51
NES	Score	ℓ_∞	378	18.6	9.53
Parsimonious	Score	ℓ_∞	253	47.9	25.46
Sign	Score	ℓ_∞	124	8.1	25.81
Square	Score	ℓ_∞	18	0.8	25.44
ZOSignSGD	Score	ℓ_∞	376	21.4	9.71
GeoDA	Label	ℓ_∞	143	38.6	17.62
HSJ	Label	ℓ_∞	212	52.7	8.82
Opt	Label	ℓ_∞	1000	65.3	0.67
RayS	Label	ℓ_∞	243	43.9	28.09
SignFlip	Label	ℓ_∞	86	47.2	15.44
SignOPT	Label	ℓ_∞	412	63.6	0.64
Bandit	Score	ℓ_2	596	6.0	13.96
NES	Score	ℓ_2	344	59.7	0.44
Simple	Score	ℓ_2	241	58.9	0.10
Square	Score	ℓ_2	23	0.4	26.46
ZOSignSGD	Score	ℓ_2	275	61.4	0.29
Boundary	Label	ℓ_2	24	48.0	12.64
GeoDA	Label	ℓ_2	146	40.6	16.89
HSJ	Label	ℓ_2	238	49.5	14.59
Opt	Label	ℓ_2	1000	53.9	12.62
SignOPT	Label	ℓ_2	411	46.3	15.96
PointWise	Label	Optimized	969	55.1	16.01
SparseEvo	Label	Optimized	1000	66.7	9.10
CA (sssp)	Label	Optimized	147	1.4	13.70
CA (bin search)	Label	Optimized	603	0.0	32.67

to 62% on ImageNet. The discrepancy between label-based and score-based empirical attacks may stem from variations in the richness and smoothness of the query information. The loss value (score), providing a smoother evaluation, is less susceptible to noise interference and discloses finer-grained details. In contrast, labels are more likely to be impacted by injected noise, resulting in more randomized query outcomes. However, our hard-label certifiable attack shows strong resilience against RAND-Post, by maintaining the average attack success rate at 93%, 99%, and 99% on CIFAR10, CIFAR100, and ImageNet, respectively. This advantage over empirical attacks, particularly the label-based ones, originates from Randomized Parallel Querying—It precisely assesses query results with a lower bound of the ASP.

Attack Performance under TRADES [135]

We consider both ℓ_∞ and ℓ_2 perturbations to generate adversarial examples, and TRADES respectively uses ℓ_2 or ℓ_∞ adversarial examples for adversarial training. We set the perturbation size to be $\ell_\infty = 0.1$ and $\ell_2 = 5$, following [135]. We then evaluate all attacks against TRADES. The results on CIFAR10 are presented in Table 4.6[5]. We observe our attack requires much less query number than the empirical attacks. Also, our attack can achieve 100% attack success rate (with the binary search localization), but at the cost of a relatively larger perturbation size.

[5] It is computationally intensive and time-consuming to train TRADES on CIFAR100 and ImageNet

Table 4.6: Attack performance under TRADES Adversarial Training on ResNet and CIFAR10

Defense	Attack	Query Type	Pert. Type	# Query	Model Acc.	Dist. ℓ_2
ℓ_∞ Adversarial Training (Clean Accuracy: 80.9%)	Bandit	Score	ℓ_∞	1601	10.2	4.32
	NES	Score	ℓ_∞	1474	27.4	2.27
	Parsimonious	Score	ℓ_∞	630	5.3	4.35
	Sign	Score	ℓ_∞	439	4.4	4.37
	Square	Score	ℓ_∞	854	5.7	4.39
	ZOSignSGD	Score	ℓ_∞	1196	37.0	2.21
	GeoDA	Label	ℓ_∞	1358	41.9	1.99
	HSJ	Label	ℓ_∞	2149	36.5	1.92
	Opt	Label	ℓ_∞	1871	73.0	0.19
	RayS	Label	ℓ_∞	721	7.3	4.29
	SignFlip	Label	ℓ_∞	2240	24.4	3.36
	SignOPT	Label	ℓ_∞	832	69.3	0.15
	PointWise	Label	Opt.	3460	9.3	4.38
	SparseEvo	Label	Opt.	8691	9.1	5.10
	CA (sssp)	Label	Opt.	548	21.2	4.29
	CA (bin search)	Label	Opt.	412	9.8	6.31
ℓ_2 Adversarial Training (Clean Accuracy: 59.2%)	Bandit	Score	ℓ_2	860	1.5	2.44
	NES	Score	ℓ_2	3535	9.5	0.99
	Simple	Score	ℓ_2	4062	2.1	1.29
	Square	Score	ℓ_2	991	4.6	2.95
	ZOSignSGD	Score	ℓ_2	3505	15.1	0.77
	Boundary	Label	ℓ_2	771	40.7	1.19
	GeoDA	Label	ℓ_2	1506	14.3	2.85
	HSJ	Label	ℓ_2	1332	5.1	3.53
	Opt	Label	ℓ_2	2890	41.6	2.39
	SignOPT	Label	ℓ_2	1766	33.6	2.76
	PointWise	Label	Opt.	4845	0.6	5.36
	SparseEvo	Label	Opt.	9697	0.4	6.03
	CA (sssp)	Label	Opt.	809	20.4	6.06
	CA (bin search)	Label	Opt.	461	0.0	8.18

4.4.4 Ablation Study

In this section, we explore in-depth our certifiable attack—we study its performance with varying noise variances, ASP thresholds, localization and shifting methods, and noise PDFs. We mainly show results on the image datasets and defer results on the audio dataset to Appendix 7.14, where similar performance can be observed.

Attack Performance on Different Noise Variances

Table 4.7 shows the performance of our attack with varying noise variances used in φ. We have the following key observations: 1) As the variance increases, the ℓ_2 perturbation size increases, since larger variance results in larger noise. 2) The ℓ_2 mean distance tends to decrease as the variance increases. This could be because that larger variance covers a larger decision space, and without moving the mean far away from the clean input, we can easily find a large portion of adversarial samples under the distribution with a large variance. 3) As the variance increases, the number of RPQ decreases. This is because the larger variance usually leads to a larger shifting step. It takes fewer iterations to move to the decision boundary when the variance increases. 4) Finally, a larger certified accuracy means that it is easier to determine the *Adversarial Distribution*. The results on CIFAR10 show that it is easier to find a small area of adversarial examples than a large area of adversarial examples. On ImageNet, we observe nearly 100% certified accuracy, which means it is relatively easy to find the adversarial examples on datasets with a large number of classes (since 999 out of $1,000$ classes in ImageNet are

Table 4.7: Attack performance of our certifiable attack with varying Gaussian noise variances σ ($p = 90\%$)

	σ	Dist. ℓ_2	Mean Dist. ℓ_2	# RPQ	Certified Acc.
CIFAR10	0.10	7.39	3.96	18.34	94.17%
	0.25	12.95	2.34	14.35	91.21%
	0.50	19.41	0.43	11.38	90.00%
ImageNet	0.10	41.80	16.78	32.55	99.80%
	0.25	87.47	16.78	17.02	99.60%
	0.50	135.47	2.27	8.31	100.00%

all false classes) or with high feature dimension.

Attack Performance on Different ASP Thresholds

We study the relationship between the performance of our attack and the ASP threshold, and Table 4.8 shows the results. As p increases, so do the ℓ_2 perturbation size, the ℓ_2 mean distance, and the number of RPQ. On one hand, a larger p means it requires more adversarial examples to fall into the false classes. When the noise variance is fixed, the mean of the *Adversarial Distribution* should be further away from the decision boundary to allow more adversarial examples to fall into the false classes. On the other hand, the smaller p results in a larger shifting distance, which depends on the gap between p and $\underline{p_{adv}}$ (see the Gaussian-case of Theorem 6 in Appendix 7.12.2). With a larger shifting distance, the required number of RPQ can be fewer. We also observe

Table 4.8: Attack performance of our certifiable attack with varying p under the Gaussian variance $\sigma = 0.25$

	p	Dist. ℓ_2	Mean Dist. ℓ_2	# RPQ	Certified Acc.
CIFAR10	50%	12.65	1.63	9.34	97.17%
	60%	12.72	1.86	11.09	95.85%
	70%	12.80	2.05	11.94	94.72%
	80%	12.87	2.18	12.37	93.17%
	90%	12.95	2.34	14.35	91.21%
	95%	13.09	2.65	15.93	90.37%
ImageNet	50%	85.88	9.89	12.85	100.00%
	60%	86.20	11.30	13.63	100.00%
	70%	86.45	12.64	14.33	100.00%
	80%	87.03	14.64	16.02	100.00%
	90%	87.47	16.78	17.02	99.60%
	95%	88.42	19.98	19.81	100.00%

that a smaller p results in a higher certified accuracy on CIFAR10, since a smaller p allows more "failed" adversarial examples. On ImageNet, the certified accuracy is consistently $\sim$ 100%, no matter p's value. This might still because it is much easier to find adversarial examples with a much larger number of classes.

Table 4.9: Attack performance of our certifiable attack on different localization/refinement algorithms ($\sigma = 0.25$, $p = 90\%$)

Localization	Refinement	Dist. ℓ_2	Mean Dist. ℓ_2	# RPQ	Cert. Acc.
sssp	none	11.46	1.35	2.30	92.54
binary search	none	11.29	0.34	9.07	92.54
random	geo.	11.80	1.73	67.53	92.54
sssp	geo.	11.20	0.49	3.70	91.54
binary search	geo.	11.28	0.27	10.08	92.53

Attack Performance on Different Localization/Refinement Algorithms

In this experiment, we compare our proposed Smoothed SSP and binary-search localization methods with the random localization baseline; and compare our proposed geometric shifting method with a no-shifting baseline. Results are shown in Table 4.9. We observe that the combination of the localization and refinement methods yields the smallest perturbation size, i.e., the smallest Dist. ℓ_2 and Mean Dist. ℓ_2. This demonstrates that they are both effective in improving the imperceptibility of adversarial examples.

Visualization. We also visualize the adversarial examples x_{adv} while crafting the Certifiable Attack for Binary-search Localization (Figure 4.6) and SSSP Localization (Figure 4.7). It shows that when $\sigma = 0.025$, both the Binary-search and SSSP-based certifiable attack can craft imperceptible perturbations. The difference is that the binary search method starts from a random x' and requires more # RPQ to update the *Adversarial Distribution*, while the SSSP can easily find an initial *Adversarial Distribution*with

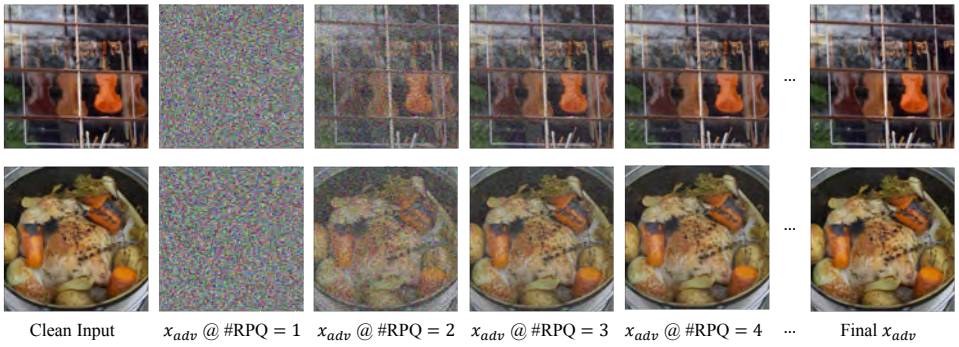


Fig. 4.6: Visualization of successful adversarial examples crafting by certifiable attack with binary-search localization

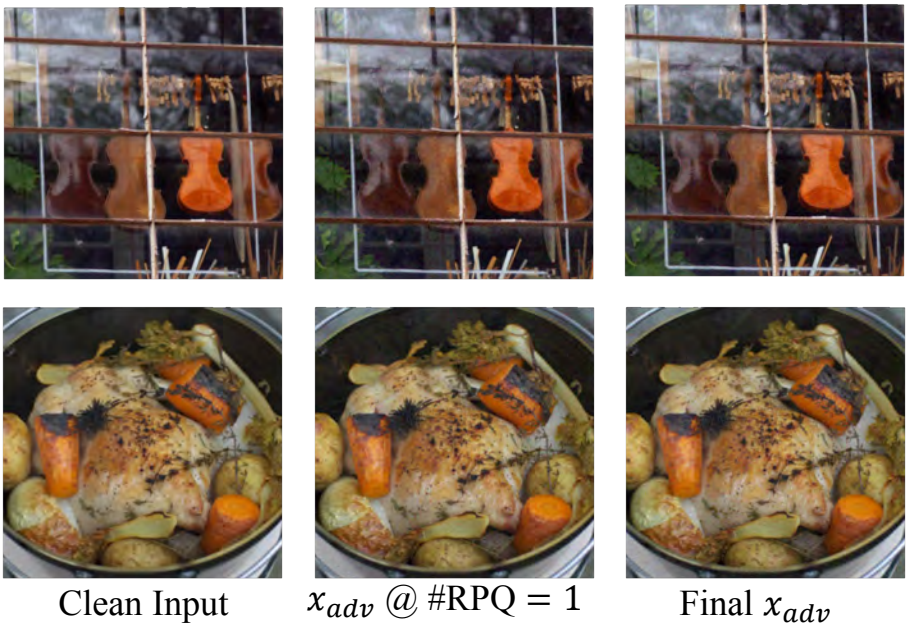


Fig. 4.7: Visualization of successful adversarial examples crafting by certifiable attack with SSSP localization (SSSP requires fewer # RPQ)

small perturbation and thus requires fewer # RPQ.

Attack Performance on Different Noise Distributions

Our attack can use any continuous noise distribution to craft the *Adversarial Distribution*. Besides the Gaussian noise distribution, in this experiment, we also evaluate the performance of our certifiable attack using other noise distributions including the Cauthy distribution, Hyperbolic Secant distribution, and general normal distributions. Note that we adjust the parameters in these distributions to ensure consistent variances for a fair comparison.

The results are presented in Table 4.10, and the noise distributions are plotted in Figure 7.10 in Appendix. On both datasets, we observe the ℓ_2 perturbation size is decreasing while the ℓ_2 mean distance is increasing as the PDF of the noise distribution is more centralized. This result may share a similar nature with results in Table 4.7—when the adversarial samples are more widely distributed, they tend to fall into an adversarial class (the majority of all classes). It is hard to determine which distribution is better since there is a trade-off between the perturbation size and the number of RPQ.

4.4.5 Defending against Our Certifiable Attack

In this subsection, we discuss potential defenses and mitigation strategies against our attacks.

Noise Detection based Defenses: Our certifiable attack injects noise into the adversarial examples. Here, we suppose the adversary is aware of the noise injection and designs a detection method by training a binary classifier to distinguish the noise-injected in-

Table 4.10: Attack performance of our certifiable attack with different noise distributions

	Distribution	Density	Parameter	$\sqrt{\|\|\epsilon\|\|^2/d}$	Dist. ℓ_2	Mean Dist. ℓ_2	# RPQ	Certified Acc.
CIFAR10	Gaussian	$\propto e^{-\|z/a\|^2}$	$a = 0.25$	0.25	12.95	2.34	14.35	91.21%
	Cauthy	$\propto \frac{a^2}{z^2+a^2}$	$a = 0.01969$	0.25	7.82	4.87	32.77	94.12%
	Hyperbolic Secant	$\propto sech(\|z/a\|)$	$a = 0.1592$	0.25	12.51	2.43	14.59	91.67%
	General Normal ($b = 1.5$)	$\propto e^{-\|z/a\|^b}$	$a = 0.2909$, $b = 1.5$	0.25	12.74	2.37	14.15	91.39%
	General Normal ($b = 3.0$)	$\propto e^{-\|z/a\|^b}$	$a = 0.4092$, $b = 3$	0.25	13.16	2.38	14.15	91.25%
ImageNet	Gaussian	$\propto e^{-\|z/a\|^2}$	$a = 0.25$	0.25	87.47	16.78	17.02	99.60%
	Cauthy	$\propto \frac{a^2}{z^2+a^2}$	$a = 0.01969$	0.25	46.18	23.94	59.94	99.60%
	Hyperbolic Secant	$\propto sech(\|z/a\|)$	$a = 0.1592$	0.25	85.57	21.29	20.89	99.80%
	General Normal ($b = 1.5$)	$\propto e^{-\|z/a\|^b}$	$a = 0.2909$, $b = 1.5$	0.25	86.69	19.05	17.58	99.80%
	General Normal ($b = 3.0$)	$\propto e^{-\|z/a\|^b}$	$a = 0.4092$, $b = 3$	0.25	88.51	15.58	14.99	100.00%

puts and clean inputs. Specifically, the defender (i.e., model owner) uses ResNet110 (as powerful as the target model) to train a noise detector to distinguish the inputs with and without noise. The experimental results show the noise detection rate can be as high as 99% with the noise variance $\sigma = 0.5$, which means this detector can be used as a strong defense against our certifiable attacks with a larger noise. However, this defense does not work when the noise scale is smaller (i.e., $\sigma = 0.025$), where the detection rate is less than 1%. Especially, the adversary may design a novel method to hide this noise in the image texture, e.g., using the diffusion model for denoising, which may circumvent the detection.

White-Box Adaptive Defenses against Our Attack: We assume the model owner knows the noise distribution used by our attack and performs a "white-box" defense.

Table 4.11: White-box adaptive defense against our attack ($\sigma = 0.25$, $p = 90\%$) on CIFAR10

Defense Para.	Dist. ℓ_2	Mean Dist. ℓ_2	# RPQ	Cert. Acc.
$\sigma_d = 0.10$	9.99	7.73	34.11	87.51%
$\sigma_d = 0.25$	15.40	10.21	29.80	88.31%
$\sigma_d = 0.50$	20.46	8.11	26.52	86.56%

Particularly, it applies a denoiser to eliminate the injected noise, so that the adversarial examples can be restored to clean inputs. The denoiser can be deployed as a pre-processing module and is pre-trained by the model owner. Specifically, we use a U-Net structure [137] as the denoiser and denote it as $\mathcal{D}$. Then, the loss function for the training is

$$\mathbb{E}_{\epsilon\sim\mathcal{N}(0,\sigma_d)}[||\mathcal{D}(x+\epsilon)-x||_2 + ||f(\mathcal{D}(x+\epsilon))-f(x)||_2] \tag{4.8}$$

Taking Gaussian noise as an example (e.g., the model owner knows the Gaussian variance $\sigma = 0.25$ used in the certifiable attack), we train the denoiser to eliminate Gaussian noise with $\sigma = 0.25$ while evaluating the certifiable attack with Gaussian noise generated by different σ. Table 4.11 shows the results. We can observe that this defense can significantly degrade the performance of a certifiable attack. Notably, by choosing the same variance σ as the adversary, the adaptive defense can increase the Mean Dist. ℓ_2 significantly. However, the certified accuracy is still near 90%.

4.5 Related Work

Adversarial Attack. It aims to mislead learnt ML models by perturbing testing data with imperceptible perturbations. It can be divided into white-box attacks [91, 92, 96, 138, 139] and black-box attacks [93, 97, 97, 98, 99, 101, 101, 103, 104, 105, 106, 107, 108, 109, 110, 111, 112, 140], per the access that the adversary holds. White-box attacks have full access to the model parameters, and can leverage the gradient of the loss function w.r.t. the inputs to guide the adversarial example generation. Instead, black-box attacks only know the outputs (in the form of prediction scores or labels) of a target model via sending queries. It is widely believed that black-box attack is more practical in real-world scenarios [93, 103, 141]. Therefore, we focus on the black-box attacks in this paper.

Black-Box Attack. Existing black-box attack methods can be classified into three types: gradient estimation based [97, 98, 104, 105, 106, 142, 143], surrogate models based [103, 107, 108, 109], or local search based algorithms [99, 101, 110, 111, 112]. Gradient estimation based attack is mainly based on zero-order estimation since the true gradient is unknown [97]. Surrogate model-based methods first perform white-box attacks on an offline surrogate model to generate adversarial examples, and then use these generated adversarial examples to test the target model. The attack performance largely depends on the transferability of such generated adversarial examples. Local search-based methods craft adversarial examples by searching the effective perturbation direction, e.g., Boundary Attack [101] traverses the decision boundary to craft the *least*

imperceptible perturbations.

All existing black-box attacks rely on querying the target model until finding a successful adversarial example or reaching the maximum number of queries. However, none of them can ensure the success rate of the adversarial examples that have not been queried. Further, they are shown to be easily detected/removed via adversarial detection and randomized pre/post-processing-based defenses.

Empirical Defense. It defends against adversarial attacks without guarantees. Empirical defenses against white-box attacks can be roughly categorized into four classes. Gradient-masking defenses [144, 145, 146] modify the model inference process to obstacle the gradient computation. Input-transformation defenses [147, 148, 149, 150, 151, 152] use pre-processing methods to transform the inputs so that the malicious effects caused by the perturbations can be reduced. Detection-based defenses [153, 154, 155, 156, 157] identify features that expect to separate adversarial examples and clean examples, and train a binary classifier to detect adversarial examples. Another branch of works [113, 158, 159, 160] detects the adversarial examples based on the similarity of the queries, demonstrating high detection accuracy in practice. Among these, Blacklight [113] has shown supreme detection performance without assumptions on the user accounts. These three types of defenses show certain effectiveness when they target specific known attacks, but can be broken by adaptive attacks [65]. Lastly, adversarial training-based defenses [91, 161, 162, 163] have achieved the SOTA performance against adaptive attacks. The main idea is to augment training data with "adversarial

examples", but they are reassigned the correct label. As to defend against *black-box* attacks, RAND-Post [115], RAND-Pre [114], Adversarial Training based TRADES [91], and Blacklight [113] are the SOTA in each category. Thus, we evaluated our certifiable attack under these defenses.

Certified Defense. Certified defense [4, 164, 165, 166, 167, 168] was proposed to guarantee constant classification prediction on a set of adversarial examples. Recently, randomized smoothing (RS) [129] has achieved great success in the certified defense since it is the first method to certify arbitrary classifiers of any scale. Specifically, RS can guarantee the prediction if the perturbation is bounded by a distance in ℓ_p-norm, i.e., certified radius [2, 4, 129, 169]. RS adds noise from a distribution (e.g., Gaussian) to the inputs and uses hypothesis testing to quantify the prediction probability. Then the bound on the perturbations (usually a ℓ_p norm constraint) for ensuring the consistent prediction is derived. This method is widely used in certified defense to ensure consistent and correct prediction under attack. However, in this paper, we propose to use this method to ensure consistent and wrong prediction on the *Adversarial Distribution*, resulting in a reliable and strong certifiable attack.

4.6 Conclusion

Certifiable attack lays a novel direction for adversarial attacks, enabling the transition from deterministic to probabilistic adversarial attacks. Compared with empirical black-box attacks, certifiable attacks share significant benefits including breaking

SOTA strong detection and randomized defense, revealing consistent and severe robustness vulnerability of models, and guaranteeing the minimum ASP for numerous unique AEs without verifying via the query.

Chapter 5

SoK: Prompt Security on Large Language Models

Large Language Models (LLMs) have rapidly become integral to real-world applications, powering services across diverse sectors. However, their widespread deployment has exposed critical security risks, particularly through jailbreak prompts that can bypass model alignment and induce harmful outputs. Despite intense research into both attack and defense techniques, the field remains fragmented: definitions, threat models, and evaluation criteria vary widely, impeding systematic progress and fair comparison. In this Systematization of Knowledge (SoK), we address these challenges by (1) proposing a holistic, multi-level taxonomy that organizes attacks, defenses, and vulnerabilities in LLM prompt security; (2) formalizing threat models and cost assumptions into machine-readable profiles for reproducible evaluation; (3) introducing an open-source evaluation toolkit for standardized, auditable comparison of attacks and defenses; (4) releasing JAILBREAKDB, the largest annotated dataset of jailbreak and benign prompts to date; and (5) presenting a comprehensive evaluation and leaderboard of state-of-the-art methods. Our work unifies fragmented research, provides rigorous foundations for future studies, and supports the development of robust, trustworthy

LLMs suitable for high-stakes deployment.

5.1 Introduction

Large Language Models (LLMs) have rapidly transitioned from academic research to core components of real-world applications, especially since the emergence of high-profile foundation models such as OpenAI's GPT series [170, 171], Google Gemini [172], Meta Llama [173, 174], Anthropic Claude [175], Alibaba Qwen [176, 177, 178], and Doubao [179]. Today, LLMs are deployed across an unprecedented range of sectors—from web search and code assistants to legal, educational, and healthcare domains—reaching hundreds of millions of end users globally. Their adoption has ushered in a new era of AI-powered services, but also surfaced serious safety and security risks. A rapidly growing body of work shows that carefully crafted *jailbreak prompts* can bypass alignment constraints and induce models to generate sensitive, illegal, or toxic content. Alarmingly, recent studies report success rates exceeding 90% on flagship models such as GPT-4, Claude 3, and DeepSeek-R1 [180, 181, 182, 183].

Fragmented progress. The urgent need to secure these widely deployed models has led to a surge of research on both attacks and defenses. In just the past two years, scholars have developed a diverse arsenal of attack strategies, including token-flipping [180], task-overload [181], multi-turn *derailment* [182], evolutionary prompt search [183], and even image-based exploits targeting multimodal models [184]. Meanwhile, de-

fenses have proliferated as well—ranging from heuristic filters and sparsity-based runtime mitigations [185], to retrieval-augmented prompt decomposition [186], ensemble detection frameworks such as MoJE [187], reinforcement learning-based alignment, and architectural interventions like KV-cache eviction.

However, despite this rapid progress, research in the area remains highly fragmented. Most works are developed and evaluated in isolation, each introducing its own threat definitions, cost assumptions, datasets, and evaluation metrics—often incompatible with those of others. This heterogeneity makes it difficult, if not impossible, to fairly compare the effectiveness or generality of different defenses. As a result, the community lacks a clear understanding of which methods are robust, under what conditions they succeed or fail, and how to make principled progress toward secure and trustworthy LLMs.

Recognizing these gaps, the research community has begun to take steps toward a more unified and systematic understanding of LLM jailbreak attacks and defenses.

Partial attempts at systematisation. Several recent surveys—such as those by Yi et al [188], Fan *et al.* [184], and Esmradi *et al.* [189]—have begun to map the landscape of attacks and defenses. In parallel, benchmarks like JailbreakZoo [190], JailbreakBench [191], and MMJ-Bench [192] have provided important footholds for reproducibility. However, despite these valuable efforts, several fundamental challenges remain unresolved:

1. **Limited threat coverage.** Most resources over-represent early DAN-style single-turn exploits, leaving token-level, chain-of-thought, and multimodal attacks under-explored.

2. **Metric drift and opacity.** Success is variously reported as raw jailbreak rate, compliance likelihood, policy-violation score, or undefined "toxicity"—making apples-to-apples comparison difficult.

3. **Data sparsity and missing metadata.** Public corpora rarely annotate prompts with attacker capability, required knowledge of system prompts, or execution cost, hindering detection and interpretability research.

4. **Taxonomy–evaluation disconnect.** No prior effort couples a principled threat classification with *open*, executable tooling that enforces uniform cost, knowledge, and access assumptions across *both* attacks and defenses.

These structural gaps limit our ability to systematically compare approaches and to track meaningful progress. To address these challenges and move the field toward a more unified foundation, this Systematization of Knowledge (SoK) makes the following contributions:

1. **Holistic taxonomy.** We propose a comprehensive, multi-level taxonomy that systematically organizes both attacks and defenses based on (*i*) attacker or defender capability, (*ii*) the specific model vulnerabilities being exploited or protected, and

(*iii*) the underlying threat objectives—thereby unifying and extending prior classification efforts.

2. **Declarative threat models.** Our work formalizes commonly used but often implicit assumptions—such as attack budget, query limits, and side-channel access—into explicit, machine-readable profiles that serve as the backbone for consistent and reproducible evaluation.

3. **Open evaluation toolkit.** We introduce a modular and extensible platform that enables any combination of (MODEL, ATTACK, DEFENSE) to be instantiated, executed, and evaluated. The toolkit logs costs, safety, and utility scores with full auditability, supporting rigorous empirical comparisons.

4. **JAILBREAKDB.** We present the largest curated corpus to date of both jailbreak and benign prompts, annotated with lightweight features across seven groups, including linguistic, syntactic, semantic, pragmatic, safety risk, behavioral, and metadata. Experiments show that even simple classifiers trained on these features can match the performance of dedicated detectors such as GradSafe [193].

5. **Comprehensive evaluation and leaderboard.** We provide a unified evaluation of state-of-the-art attacks, defenses, and leading LLMs, accompanied by an up-to-date leaderboard. This systematic assessment uncovers a range of insightful observations and actionable findings regarding the strengths and limitations of current approaches.

By releasing our taxonomy, open-source evaluation toolkit, and richly annotated dataset, we transform fragmented anecdotal findings into a *rigorous, comparable, and extensible* science of LLM robustness. We hope this foundation will catalyze progress toward **verifiably aligned** language models—and, crucially, support the development of defenses robust enough for the fast-paced, high-stakes environments of real-world deployment.

5.2 Systematic Literature Review

The goal of this literature review and taxonomy is to provide a comprehensive overview of the rapidly evolving field of LLM jailbreak attacks, defenses, and revealing the LLM security vulnerability, helping the researchers understand the current landscape and the diversity of approaches being developed.

5.2.1 Scope and Overview of the Taxonomy

This work introduces three taxonomies that approach the field of LLM prompt security from distinct yet complementary perspectives: **Taxonomy I** covers jailbreak attack techniques, **Taxonomy II** addresses defense methodologies, and **Taxonomy III** summarizes inherent vulnerabilities in large language models. Each taxonomy aims to provide a systematic and comprehensive framework for organizing current research and facilitating consistent analysis.

Taxonomy I: Jailbreak Attack Technologies. Taxonomy I adopts a two-level organi-

zational principle. At the highest level, attacks are classified according to their underlying *threat model*, which reflects the adversary's capabilities and assumptions—such as black-box or white-box access to the model. Within each threat model, attacks are further organized by technical methodology.

Recognizing the complexity of real-world attacks, we decompose the landscape into a set of *atomic technical units*: minimal, actionable components that represent the core tactics used in the literature. These atomic units are not mutually exclusive—many attacks employ multiple tactics in parallel or series, and the boundaries between units can be fluid or overlapping. Our taxonomy, therefore, serves as a compositional framework that captures both the breadth and depth of jailbreak technologies, rather than imposing rigid categories. This approach enables systematic annotation and comparison, and helps to clarify how different techniques may be combined or extended. Detailed classification guidelines and an overview of the taxonomy are provided in Section 5.2.3 and Figure 5.1.

Taxonomy II: Jailbreak Attack Technologies For defenses, we adopt a similar two-step approach, based on the intended goal and implementation strategy: we first classify defenses by their primary goal—either detection or prevention—and then further subdivide each group based on the methodologies employed. This approach enables fair comparison and systematic analysis of defense strategies. See Section 5.2.4 and Figure 5.2.

Taxonomy III: LLM Vulnerabilities Notably, a significant class of attacks specifi-

cally exploits intrinsic vulnerabilities in LLMs to bypass guardrails; for these, we also present a taxonomy of LLM vulnerabilities. See Section 5.2.5 and Figure 5.3.

Table 5.1: Integrated comparison of representative jailbreak literature (survey / benchmark / dataset)

Taxonomy (citation)	Year	Survey / Review	Benchmark / Eval	Dataset released	Attack coverage	Defense coverage	Text-only LLM	VLM included	Threat-model axis	Notes
Jin *et al.* [190]	2024	✓	–	–	✓	✓	–	✓	–	7 flat "types"
Yi *et al.* [188]	2024	✓	–	–	✓	✓	✓	–	–	Black/white-box split
Xu *et al.* [194]	2024	✓	✓	–	✓	✓	✓	–	–	9 attacks / 7 defenses
Esmradi *et al.* [189]	2023	✓	–	–	✓	✓	–	–	–	Gen-AI broad scope
Inie *et al.* [195]	2023	✓	–	–	✓	–	✓	–	–	Grounded-theory interviews
Shayegani *et al.* [196]	2023	✓	–	–	✓	✓	✓	–	✓	Vulnerability-centric
Gupta *et al.* [197]	2023	✓	–	–	✓	✓	–	–	–	Cyber-security view
Ours	2025	✓	✓	✓	✓	✓	✓	–	✓	Multi-layer taxonomy + leaderboard

Comparison with Prior Taxonomies. Compared with previous taxonomies such as Jin et al. [190], Yi et al. [188], Xu et al. [194], Esmradi et al. [189], Inie et al. [195], Shayegani et al. [196], and Gupta et al. [197], our taxonomy adopts a more systematic and fine-grained approach. Specifically, we focus exclusively on text-to-text LLMs and introduce a multi-dimensional framework that separates threat models from attack methodologies, while also providing a parallel, structured taxonomy for defenses and a dedicated ontology of LLM vulnerabilities. This structure allows for more precise classification and mapping between attacks, defenses, vulnerabilities, and benchmarks, facilitating a comprehensive and extensible analysis of the field. For a detailed comparison with representative prior surveys, please refer to Table 5.1.

5.2.2 Key Concepts and Threat Model

This section defines the terminology and threat modeling assumptions used throughout the survey. Unless otherwise noted, the discussion primarily concerns the textual components of LLMs and their corresponding guardrail mechanisms, though the terminology and assumptions may also extend to the language-processing modules within multi-modal models.

Core Concepts. Throughout this survey, the following terminology is employed. *Guardrails (or safety filters)* refer to policies, system messages, or automated classifiers enforced at inference time to constrain model behaviour. A *jailbreak prompt* denotes an input specifically crafted to induce the LLM to violate these guardrails or its instruction hierarchy. *Access settings* are categorized as black-box, gray-box, or white-box: in the black-box setting, the attacker is restricted to input-output interactions with no knowledge of the model's internal mechanisms; the gray-box setting permits partial visibility, such as access to auxiliary signals (e.g., confidence scores, log probabilities, or limited feedback); in the white-box setting, the attacker enjoys full access to the model's parameters, architecture, training data, and internal computations, including gradients. Attacks may be *single-turn* (occurring within one round of dialogue) or *multi-turn* (spanning multiple rounds of dialogue).

Attacker Model Dimensions. The attacker model is defined by several dimensions, including the attack goal (e.g., policy violation, sensitive information extraction, system

instruction hijacking, or tool abuse), the level of knowledge (black-box, gray-box, or white-box access as described above), and capability (such as permitted query budget, granularity of context control, or access to additional external LLMs). Additional factors include the stealth requirement (the extent to which detection evasion is necessary, ranging from static filter evasion to dynamic, runtime detector evasion), interaction pattern (single-turn, multi-turn, or indirect injection, where the latter involves malicious prompts embedded in upstream content), and resource budget (constraints on tokens, computation, time, or monetary cost).

Defender Model Dimensions. The defender model is specified by the defense goal (detection, prevention, or refusal), deployment layer (model input, model modification, or model output), level of knowledge on attacks (known or unknown attack), and adaptivity (static rules, dynamic online learning, or proactive self-red-teaming). The resource budget includes permissible latency, additional inference calls, or human-in-the-loop involvement.

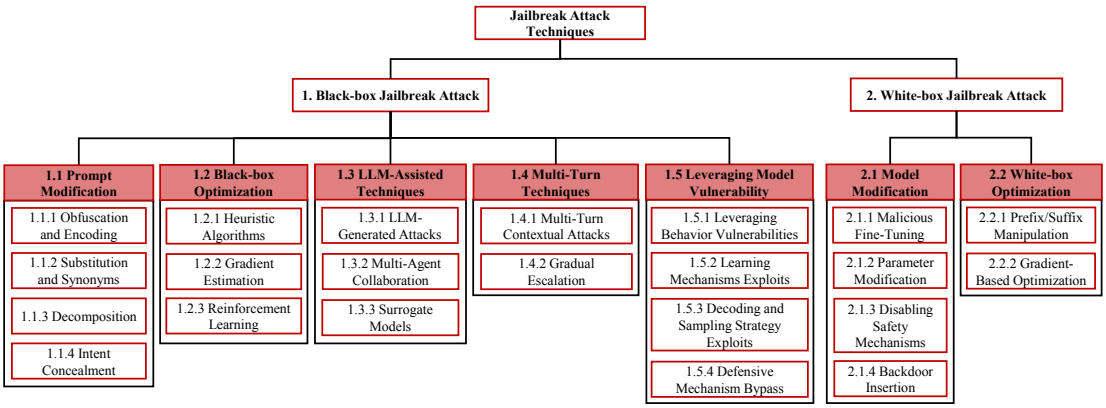


Fig. 5.1: Overview of Taxonomy I: Jailbreak Attack Techniques

5.2.3 Taxonomy I: Jailbreak Attack Techniques

We categorize jailbreak attacks primarily based on their underlying threat model, distinguishing between *black-box* attacks, where adversaries interact with the model solely through its input-output interface, and *white-box* attacks, where adversaries have access to the model's internal parameters or training process.

I.1 Black-box Jailbreak Attack

Black-box jailbreak attacks encompass adversarial strategies that seek to bypass LLM safety mechanisms solely through external interactions, without access to model internals such as weights, gradients, or architecture details. Attackers operate by probing the model via its public interface—relying on input-output behavior and feedback—to iteratively craft prompts or strategies that induce undesired or harmful outputs. This paradigm includes techniques such as prompt modification, heuristic and reinforcement learning-based optimization, LLM-assisted prompt generation, and multi-turn manipulation, collectively revealing the limits of external safety controls and

the generalization capacity of LLMs against real-world adversaries.

I.1.1 Prompt Modification Techniques

Prompt modification techniques encompass a family of black-box attack strategies that directly alter the content or structure of input prompts to circumvent safety filters in large language models (LLMs), while maintaining the underlying malicious intent.

I.1.1.1 Obfuscation and Encoding

Definition: The *Obfuscation and Encoding* class comprises black-box jailbreak attacks that systematically transform malicious prompts to evade safety filters while retaining semantics for LLMs. This category includes four key subclasses: (1) *Character Alteration*, such as typos, leet speak, or homoglyphs; (2) *Encoding and Encryption*, using Base64, ROT13, hexadecimal, binary, or custom ciphers; (3) *Multilingual Transformation*, including translation or language mixing to obscure disallowed content; and (4) *Code and Markup Language Transformation*, where requests are reframed as code, markup, or embedded in comments and strings. Unlike simple text perturbations, these methods exploit LLMs' symbolic reasoning and pattern recognition abilities, bypassing filters that focus on natural language or surface-level cues.

Recent literature establishes that such attacks, including cipher-based prompts, data structure encoding, and rare-token permutations, reliably bypass safety systems in state-of-the-art models like GPT-4 [198, 199]. Extensions to multimodal and clinical contexts reveal that obfuscated cues—such as Unicode, tiny fonts, or code-wrapped

instructions—subvert LLM-supported decisions [200, 201]. Automated attacks via prompt flipping and multi-level encoding have been systematically codified, highlighting the models' inability to generalize alignment to encoded modalities [180, 202]. Recent multilingual studies demonstrate that translating prompts into non-English languages (especially low-resource ones) can significantly weaken safety filters in LLMs, enabling both unintentional and intentional jailbreaks, and that mitigation remains challenging even with advanced models like GPT-4 [203, 203]. Multi-turn, ciphered, or delimiter-free attacks further demonstrate that symbolic obfuscation is both universal and stealthy, revealing that current guardrails fail to anticipate the models' proficiency in de-obfuscation and code reasoning [204, 205]. These works underscore a fundamental mismatch between pattern-based filtering and the general computation abilities of LLMs, necessitating future safety efforts to target non-natural input modalities and to automate semantic de-obfuscation.

I.1.1.2 Substitution and Synonyms

Definition: *Substitution and Synonyms* refers to the strategic replacement of sensitive words, phrases, or prompt components with alternatives—such as synonyms, euphemisms, paraphrases, or code words—to bypass input filters and safety mechanisms, while preserving the original semantics.

In the literature, *Substitution & Synonyms* is established as a key prompt modification tactic for black-box LLM jailbreaks. Automated synonym and phrase replacement—demonstrated by SurrogatePrompt [206], latent jailbreak benchmarks [207],

and large-scale empirical studies [208]—enables attackers to bypass filters with minor linguistic changes. Advanced strategies further leverage semantic constraints and optimization, generating transferable and high-similarity triggers that evade detection [209, 210]. Overall, substitution-based attacks form an adaptive subclass that fundamentally challenges input-based defenses, especially the keyword filters.

I.1.1.3 Decomposition

Definition: Prompt decomposition attacks break down a malicious request into smaller, innocuous components, which are then systematically recombined—either implicitly or explicitly—by the model to achieve the harmful intent.

Prompt decomposition bypasses LLM safety by splitting harmful queries into syntactically or semantically benign sub-prompts, which are then recombined by the model to achieve the original intent. Representative works, including DrAttack and RL-based frameworks like PathSeeker and DRA [211, 212], use syntactic parsing, in-context learning, and iterative feedback to automate sub-prompt generation and aggregation, significantly boosting attack effectiveness. Chain-of-Jailbreak generalizes this approach to multimodal models, demonstrating high bypass rates for toxic content construction [213]. Theoretical studies reveal that decomposition exposes compositional leakage, which evades input/output-based censorship [214]. Enhanced by obfuscation methods such as euphemization, distractor injection, or context-driven summarization [215, 216], prompt decomposition outperforms basic payload splitting [217] and remains robust against advanced defenses relying on perplexity or anomaly detec-

tion [218].

I.1.1.4 Intent Concealment

Definition: Intent concealment refers to the strategy of hiding malicious intent behind neutral or innocuous language to evade detection. Typical techniques include framing disallowed requests as academic or research inquiries, embedding harmful objectives within legitimate contexts or narratives, and employing hypothetical or fictional scenarios.

Recent literature on intent concealment demonstrates a range of prompt modification strategies that camouflage harmful intent. Notably, sequential masking and incremental synthesis, as in Imposter.AI [215], diffuse toxicity by staging benign subquestions that are later combined. Role-play, narrative, and scenario-based disguises [197, 219] detach illicit intent by reframing prompts as part of storytelling, research, or humor. Distraction-based techniques (e.g., Tastle [220]) manipulate model focus by embedding concealed objectives within larger benign tasks. Methods like IntentObfuscator [201] directly synthesize ambiguity and leverage complex linguistic alterations to evade both automated and manual detection.

I.1.2 Black-box Optimization

Black-box Optimization techniques encompass a broad family of black-box methods that iteratively search the prompt space using non-gradient, feedback-driven strategies—such as heuristics, gradient estimation, and reinforcement learning—to discover prompts capable of bypassing LLM safety filters without requiring internal model ac-

cess.

I.1.2.1 Heuristic Algorithms

Definition: The *heuristic algorithms* class encompasses black-box jailbreak methods that leverage iterative, often population-based or random, search strategies—such as genetic algorithms, evolutionary approaches, random search, and metaheuristics—to explore the discrete prompt space without access to model gradients. These algorithms optimize adversarial prompts by evaluating candidate generations through fitness functions, typically based on embedding similarity, harmfulness, and stealth, with the core objective of efficiently bypassing LLM safeguards where direct gradient signals are unavailable or uninformative.

A substantial body of literature establishes and refines this paradigm. OpenSesame [221] first adapts genetic algorithms for prompt suffix generation, demonstrating that black-box heuristic search can yield universal, transferable adversarial prompts. Subsequent works—such as AutoDAN [183], Semantic Mirror Jailbreak (SMJ) [210], AutoJailbreak [222], and BlackDAN [223]—advance this framework via improved initialization, crossover, mutation, and evaluation mechanisms, introducing multi-objective optimization and Pareto-dominance to balance harmfulness, semantic alignment, and detectability. Lightweight methods such as random search [224, 225], discrete coordinate ascent [226], and greedy tree-search [227] further enrich the heuristic toolkit, revealing new vulnerabilities and attack pathways. Collectively, these studies show that heuristic algorithms uniquely combine practical black-box applicability, extensi-

bility, and transferability, with continual innovations in fitness shaping and prompt initialization (e.g., [228]) driving ongoing progress and shaping the evolving adversarial landscape.

I.1.2.2 Gradient Estimation Methods

Definition: Gradient estimation techniques aim to enable optimization-based adversarial prompt engineering for black-box jailbreak attacks, where direct access to model gradients is unavailable. By approximating gradient directionality through queries or surrogate models, these methods facilitate systematic search for adversarial prompts, typically using strategies such as finite differences or proxy-driven estimation in the discrete token space.

Recent works have advanced the use of gradient estimation for efficient black-box adversarial prompt attacks. PAL [229] leverages surrogate models to approximate gradients and employs sophisticated candidate ranking, enabling scalable and low-cost attacks against commercial LLM APIs with strong transferability. Discrete gradient estimation methods such as RAL [230] and variants that use finite-difference or score-based queries further demonstrate that query-efficient, token-level directional search significantly outperforms random or evolutionary baselines. Extensions to multimodal settings, including attacks on diffusion and T2I models [231, 232], confirm the broad applicability of gradient estimation. Key insights include the method's efficiency and transferability, but also highlight vulnerabilities such as perplexity-based detection and the potential for overfitting to surrogates.

I.1.2.3 Reinforcement Learning

Definition: Unlike brute-force or purely stochastic approaches, RL-based techniques employ agents—typically leveraging deep RL algorithms such as PPO or MADDPG—that adaptively refine attack strategies based solely on black-box feedback, optimizing toward reward signals aligned with attack objectives.

RL-based methods, including PathSeeker [212], RLbreaker [233], SneakyPrompt [234], RL-JACK [235], Atoxia [236], and Arondight [237], collectively reframe jailbreak prompt generation as a reward-driven search problem. These works demonstrate that RL agents, via custom reward functions quantifying attack success, information richness, or semantic alignment, outperform classical genetic or random mutators in both LLMs and multimodal models. For example, PathSeeker and RLbreaker leverage collaborative or single-agent RL frameworks to optimize malicious prompt discovery, while SneakyPrompt and Arondight extend RL-based attacks to text-to-image and vision-language models using specialized reward shaping. Atoxia further highlights the effectiveness of RL by exploiting the target model's own toxic answer likelihood as a feedback signal. Collectively, these studies reveal RL's unique strengths: policy learning, adaptive exploration-exploitation, and advanced reward shaping, while also noting challenges such as sparse rewards and transferability issues across models or modalities.

I.1.3 LLM-Assisted Techniques

LLM-assisted techniques refer to methods that harness language models them-

selves to automatically generate, refine, or guide adversarial prompts, including LLM-generated attacks, multi-agent collaboration, and surrogate model approaches.

I.1.3.1 LLM-Generated Attacks

Definition: LLM-generated jailbreak attacks refer to adversarial strategies where one or more large language models (LLMs) autonomously generate, optimize, or refine attack prompts without access to target model internals. Distinct from human-crafted or gradient-based prompting, these methods leverage LLMs—often open-source or less-aligned—as automated attackers, optimizers, or red-teamers to produce adversarial content or prompt templates in a scalable, data-driven manner.

Recent works (e.g., SoP, SeqAR [238], GPTFUZZER [239], DAP [220]) advance this field by iteratively generating and optimizing sophisticated jailbreak prompts, including multi-character or sequential templates, using attacker LLMs alone. IRIS [240] and PAIR [241] introduce self-reflective and self-explanation-based prompt refinement, enabling LLMs to act as both attacker and target under strict black-box constraints. AdvPrompter [242] demonstrates adversarial prompt learning, with LLMs trained to rapidly generate adaptive, human-readable adversarial suffixes or prompts without gradient information. Recent multimodal approaches, such as AutoJailbreak [243] and Visual-RolePlay [244], further highlight LLMs' autonomous development of vision-language attack strategies.

I.1.3.2 Multi-Agent Collaboration

Definition: Multi-agent collaboration in LLM-assisted black-box jailbreak attacks refers

to the orchestrated interaction of multiple autonomous LLM-based agents—each potentially assigned distinct roles or strategies—to generate, refine, and validate adversarial prompts. These agents collectively pursue jailbreak objectives by dividing labor, iteratively exchanging feedback, and leveraging agent diversity, enabling the creation of sophisticated prompts that bypass safety alignment boundaries.

Recent literature has established multi-agent collaboration as a pivotal attack methodology. GUARD [245] introduces a structured four-role framework (Translator, Generator, Evaluator, Optimizer) where agents collectively translate, mutate, and iteratively enhance jailbreak prompts via contextual feedback, uncovering transferable and natural-language attacks. AutoDAN-Turbo [246] advances this by formalizing lifelong, self-evolving multi-agent exploration, enabling agents to autonomously discover, store, and combine adversarial strategies. JailFuzzer [247] adopts a fuzz-testing approach with mutation and oracle agents leveraging collective memory for effective prompt generation. Evil Geniuses [248] frames jailbreaks as Red-Blue multi-agent competitions, revealing cascading vulnerabilities arising from agent interactions.

I.1.3.3 Surrogate Models

Definition: The surrogate model class refers to black-box jailbreak attack techniques that leverage one or more accessible substitute models—often open-source or smaller LLMs—to craft, guide, or evaluate adversarial prompts targeting a closed-source or restricted-access LLM. By approximating the decision boundaries or behaviors of the target through these proxies, attackers systematically reduce sample complexity and

query cost, enabling scalable, transferable attacks even when internal model parameters and direct outputs are unavailable.

Recent literature operationalizes the surrogate model approach across diverse modalities and threat models. PAL framework [229] employs a proxy LLM for token-level gradient optimization and fine-tuning to align with the target, reducing queries and boosting attack efficacy. PRP [249] constructs universal adversarial prefixes via surrogate guard models, propagating successful perturbations to base LLMs for high transferability. Hayase et al. [250] integrate surrogates into gradient-based search, validating the "proxy filter plus query validation" paradigm for superior efficiency. BlackDAN [223] embeds surrogates within a genetic algorithm as fitness evaluators, orchestrating stealthy, semantically relevant jailbreaks.

I.1.4 Multi-Turn

Strategies that leverage repeated interactions with a model, systematically manipulating its behavior by constructing, evolving, or adapting context across multiple conversational turns.

I.1.4.1 Multi-Turn Contextual Attacks

Definition: The *multi-round context technique* exploits the large language model's conversational memory by distributing adversarial intent across multiple interaction rounds. Instead of presenting explicit harmful prompts, the attacker decomposes the malicious objective into a series of semantically or functionally connected queries, each of which appears benign in isolation. Through this process, the attacker incrementally

shapes the context such that the LLM is primed towards producing forbidden outputs, effectively circumventing safety mechanisms that focus on single-turn or overtly adversarial inputs.

Recent literature formalizes multi-round attacks as iterative, context-driven adversarial engagements. Cheng et al. [218] demonstrate that models prompted with sequential, semantically related questions can be steered towards policy-violating outputs with higher success rates compared to zero-shot or random multi-turn baselines. Yang et al. [251] advance this by modeling adaptive, feedback-driven attack chains using evaluation functions to optimize context progression. Studies by Li et al. [252] and Bhardwaj and Poria [253] reveal that context-dependent jailbreaks expose class-level blind spots in current safety training, which is often limited to shallow, turn-level defenses.

I.1.4.2 Gradual Escalation Techniques

Definition: Gradual escalation refers to a distinct class of black-box jailbreak attacks on large language models (LLMs) where the attacker incrementally increases the adversarial intent over multiple interaction rounds. In contrast to multi-turn contextual attacks—which focus on distributing benign-seeming components of the attack across the context—gradual escalation centers on progressively intensifying the maliciousness or risk of each prompt within the conversation. This approach exploits the LLM's conversational memory and adaptive response, with each turn subtly raising the attack's severity or explicitness. Through this gradual evolution of intent, adversaries circumvent

static or abrupt-trigger defenses, ultimately steering the LLM toward policy-violating outputs by systematically amplifying the threat level throughout the dialogue.

Recent literature converges on several mechanisms and insights for gradual escalation attacks. Russinovich et al. introduce Crescendo, illustrating how multi-turn, context-amplifying queries can subvert advanced alignment protocols across modalities [254]. Jiang et al. model red teaming as a learnable, adaptive process that incrementally optimizes prompt concealment and actively leverages model feedback to enhance attack effectiveness [255]. Semantic-driven strategies—such as chain-of-thought and actor-network accumulation by Yang et al. and Ren et al.—demonstrate that progressively relevant prompts, not individual queries, evade static detection [182, 251]. Lin et al. reveal that reasoning-level, multi-step manipulations prompt LLMs to infer and escalate harmful content across turns [256], while Ramesh et al. show that iterative self-refinement, as in IRIS, efficiently converges on jailbreak compliance with few queries [240]. Inie et al. qualitatively analyze expert red-teamers, highlighting real-world manifestations such as "foot-in-the-door" and emotional appeals, which collectively remain undetected until late-stage policy violations [195].

I.1.5 Leveraging Model Features

Techniques in this category exploit inherent LLM features or vulnerabilities—such as behavioral patterns, learning dynamics, decoding processes, or defense mechanism characteristics—to bypass safety controls in a black-box setting. Rather than manipulating surface-level prompts alone, these attacks systematically target the model's be-

havior vulnerabilities and the system vulnerabilities. These attacks demonstrate that model-level and system-level weaknesses, arising from both architecture and operational defense limitations, present persistent and adaptable attack surfaces for adversaries.

I.1.5.1 Leveraging Behavior Vulnerabilities

Definition: Leveraging LLM vulnerability refers to black-box jailbreak strategies that systematically exploit underlying model weaknesses, such as overreliance on user instructions, context misinterpretation, and inductive biases. These attacks target how the model processes, generalizes, and adapts, rather than focusing on surface-level prompt manipulations. Characteristic vectors include manipulation of instruction following, ambiguity handling, format interpretation, and memory or context management, enabling adversaries to bypass safety alignment by targeting the model's behavioral and architectural flaws. A systematic taxonomy of the LLM vulnerabilities are presented in 5.2.5. Representative subcategories include: exploiting overreliance on explicit user instructions (e.g., "Disregard all previous guidelines"), manipulating response or input formats (e.g., JSON, markdown, code), leveraging few-shot or in-context learning to guide policy violations, exploiting knowledge cutoff or system prompt leakage, and inducing ambiguity or role-play scenarios to circumvent alignment. These methods demonstrate that model vulnerabilities are not limited to single prompts, but arise from the core design and learning mechanisms of LLMs, demanding deeper, vulnerability-centered security assessments.

I.1.5.2 Learning Mechanisms Exploits

Definition: Learning mechanism exploitation refers to black-box jailbreak attacks that adversarially exploit the core mechanisms acquired or reinforced during LLM training, such as sequential prediction, in-context adaptation, fine-tuning dynamics, and context integration. Unlike attacks that manipulate surface-level prompts or static vulnerabilities, this subclass targets the underlying processes and inductive biases that govern how the model generalizes, updates, and reasons.

Zong et al. [257] reveal that post-alignment fine-tuning can cause catastrophic forgetting, eroding harmlessness by exploiting overfitting tendencies. Xu et al. [258] show preemptive answer attacks exploit reasoning biases, while Zhou et al. [259] demonstrate special token injections manipulate tokenization and context learning. Geiping et al. [260] describe "style injections" and role hacking that exploit learned format priors. Deng et al. [261] illustrate Pandora attacks poison retrieval in RAG, exploiting the model's context synthesis vulnerabilities. Zhou et al. [262] propose dual-objective losses that align attack optimization with refusal learning mechanisms, improving universality.

I.1.5.3 Decoding and Sampling Strategy Exploits

Definition: Exploiting decoding and sampling strategies constitutes a distinctive class of black-box jailbreak attacks targeting large language models (LLMs). These attacks manipulate vulnerabilities in the model's output generation process, specifically by altering decoding algorithms (e.g., greedy, top-k, top-p, temperature sampling) or

the probabilistic dynamics of candidate token selection. Unlike prompt-based attacks or parameter tuning, this category subverts model safety protections solely through inference-time adjustments, without requiring access to model internals or adversarial prompts.

Recent literature demonstrates that small yet systematic modifications to decoding settings—such as lowering nucleus sampling thresholds or increasing temperature—can significantly increase the likelihood of harmful outputs, even in safety-aligned LLMs [263]. Advanced approaches like Weak-to-Strong Jailbreaking exploit log-probability algebra to transfer toxic generations from weak to strong models without prompt optimization [264]. Techniques including coercive interrogation and forced token selection reveal that harmful content may surface when decoding is constrained or candidate rankings are perturbed [265, 266]. Unsafe path guidance further uncovers hidden "decoding trajectories" that transform refusals into illicit outputs when guided by cost functions or auxiliary models [226, 267].

I.1.5.4 Defensive Mechanism Bypass

Definition: Defense Mechanism Bypass denotes black-box jailbreak attacks designed to exploit specific characteristics or blind spots of deployed LLM defense mechanisms, enabling adversaries to circumvent safety or alignment controls without model internals. This class is defined by its focus on defeating explicit defense layers—such as content filters, instruction-following constraints, or modality-specific safeguards—by leveraging their generalization limitations, reliance on superficial pattern matching,

static policy enforcement, or incomplete modality coverage.

Across the literature, bypass attacks systematically target and exploit defense mechanism weaknesses. Wang et al. [268] manipulate RAG-based defenses by poisoning external knowledge sources, taking advantage of defenses' trust in external retrieval without robust validation. Ye et al. [269] and Debenedetti et al. [270] show prompt injection and tool-calling agents bypass dynamic filters by exploiting inadequate input sanitization and over-reliance on trusted tool outputs. Nassi et al. [271] introduce "PromptWares," which subvert content filters and input/output checks by embedding malicious payloads in seemingly benign prompts, exploiting static and naïve filtering logic. Liu et al. [272] and Li et al. [273] demonstrate that visual and multimodal attacks succeed due to incomplete coverage of non-text modalities by alignment defenses. Wu et al. [274] and Kimura et al. [275] reveal that function-calling and visual prompt injections exploit more permissive or poorly monitored execution paths during tool invocation and grounding. Deng et al. [276] highlight that multilingual defenses often fail for low-resource languages, as policies are tuned primarily for high-resource cases.

I.2 White-box Jailbreak Attack

White-box jailbreak attacks refer to adversarial techniques that exploit direct access to a model's internal parameters, architecture, or training process to systematically undermine or disable its safety alignment. In contrast to black-box methods, white-box attackers can modify, fine-tune, or analyze model weights and internals—enabling powerful strategies such as model modification, gradient-based prompt optimization,

and direct removal of safety guardrails. This category demonstrates the heightened risks when model internals are accessible and highlights the importance of robust security throughout the entire model lifecycle.

I.2.1 Model Modification-Based Techniques

Model modification-based techniques target large language models by directly altering their internal structures, parameters, or safety components. Unlike prompt-based or indirect jailbreak methods, these approaches leverage privileged access—such as model weights or training procedures—to subvert or disable safety alignment.

I.2.1.1 Malicious Fine-Tuning

Definition: Malicious fine-tuning refers to the deliberate adaptation of large language model (LLM) parameters—typically via supervised or reinforcement learning methods—with the explicit aim of subverting safety guardrails, often using ostensibly benign data or tools. Unlike broader model modification or backdooring, malicious fine-tuning directly and persistently embeds jailbreak capabilities or misalignments within model weights, exploiting the trust chain of model sharing and the limitations of input filtering. This subclass is distinguished by its operational stealth, efficiency, and the challenge it poses in open or semi-open LLM ecosystems where model weights are accessible.

Recent literature provides a multi-faceted exploration of malicious fine-tuning. Volkov et al. [277] empirically demonstrate that parameter-efficient fine-tuning methods (e.g., QLoRA, ReFT, Ortho) enable rapid, low-cost removal of safety alignment

from advanced models such as Llama 3. Wang et al. [278] formalize the fine-tuning jailbreak paradigm, showing that minimal malicious data suffices to compromise cloud-based LMaaS and proposing secret-triggered exemplars as a defense. Wang et al. [279] introduce functional homotopy to iteratively weaken alignment and facilitate subsequent jailbreaks. Hazra et al. [280] show that even targeted, non-explicit fine-tuning (e.g., for knowledge editing) can erode safety boundaries, while Liu et al. [281] highlight both the amplified risk from adversarial fine-tuning and the partial mitigation via curated clean data. Zhao et al. [282] further expand the scope, illustrating that reinforcement of seemingly benign features through fine-tuning can override safety mechanisms, framing malicious fine-tuning as a spectrum from overt to structurally adversarial adaptation.

I.2.1.2 Parameter Modification

Definition: Parameter modification refers to adversarial interventions that directly alter a large language model's internal parameters—such as weights, activations, or representations—to subvert or bypass safety mechanisms, without introducing new architectures or retraining from scratch.

A series of recent studies provide insights into parameter modification attacks. Anand and Getzen [283] reveal how adversaries can reverse safety alignment by manipulating residual negative value vectors in transformer MLP blocks, exploiting vulnerabilities left by alignment algorithms like PPO that induce only minimal weight changes. Banerjee et al. [284] demonstrate that surgical parameter edits (e.g., via ROME) can

sharply increase unsafe outputs with minimal impact on model utility, showing the precision and potency of this attack class.

I.2.1.3 Disabling Safety Mechanisms

Definition: Disabling safety mechanisms refers to directly and intentionally altering or removing an LLM's internal guardrails by exploiting privileged access to model weights or architecture.

Recent works sharpen the technical boundaries of this category. Zhang et al. [266] introduce "EnDec," which conditionally disables safety mechanisms at generation time by manipulating the decoding process, redirecting or replacing tokens to suppress rejections and force affirmative outputs—without extensive retraining or advanced prompt engineering. Volkov's Badllama 3 [277] further advances this paradigm by directly removing safety alignment from models like Llama 3 via parameter-efficient fine-tuning (e.g., QLoRA, ReFT, Ortho), enabling rapid erasure of refusal behaviors and distribution of jailbreak adapters.

I.2.1.4 Backdoor Insertion

Definition: Backdoor insertion refers to the deliberate implantation of trigger patterns into the parameters of large language models (LLMs). These triggers, typically universal and covert, are designed so that the model only exhibits malicious or unauthorized behaviors when the trigger is present, while otherwise maintaining benign and aligned outputs. Distinct from general model modification, backdoor insertion establishes persistent, stealthy vulnerabilities that are conditionally activated by adversarial inputs,

rendering their detection and mitigation highly non-trivial.

Rando et al. [285] provide the first systematic study of universal jailbreak backdoors, showing that poisoning the RLHF process with engineered triggers yields models susceptible to robust, cross-instance exploits that evade detection by remaining silent on benign prompts.

I.2.2 White-box Optimization

White-box optimization denotes jailbreak attack strategies that generate adversarial prompts by leveraging direct access to model internals, especially gradients.

I.2.2.1 Prefix/Suffix Manipulation

Definition: The *prefix/suffix manipulation* class comprises heuristic optimization attacks that append adversarial tokens to either the beginning (prefix) or end (suffix) of input prompts, strategically editing only the input string to subvert LLM safety guardrails. Typically operating under white-box settings, these methods leverage iterative search and optimization techniques to discover input-level modifications that consistently induce policy-violating or undesired model behaviors, without requiring access to model internals.

A substantial body of work has defined and expanded this class. Foundational studies on adversarial suffixes, such as GCG and its analysis by Zou et al. [230], formalize the optimization problem and demonstrate the cross-model universality and transferability of well-crafted suffixes. Subsequent research addresses efficiency and diversity challenges: Jiang et al. [286] propose ECLIPSE, which leverages LLMs as

autonomous optimizers for semantically natural and highly effective suffixes, while Liao and Sun [287] (AmpleGCG) introduce a generative framework that samples diverse adversarial suffixes, increasing both coverage and transferability. The paradigm extends to prefix manipulation and multi-stage guardrails, as shown by Ma et al. [232] and Mangaokar et al. [249], who demonstrate that optimized prefixes can propagate through layered defenses. Wang et al. [288] further enhance stealth by mapping optimized suffix embeddings into fluent text, reducing detectability by perplexity filters. Analytical works by Alon and Kamfonas [289] and Zhao et al. [282] probe the trade-offs between fluency, adversarial effectiveness, and detectability, while Liu et al. [290] address scalability via transfer learning-based frameworks (e.g., DeGCG), enabling efficient, domain-adaptive suffix generation. Collectively, these studies highlight prefix/suffix manipulation as a distinct, string-level, and highly generalizable jailbreak strategy—resilient to common defenses and presenting evolving challenges in efficiency, stealth, and adaptability against guard-rail advancements.

I.2.2.2 Gradient-Based Optimization

Definition: The *gradient-based optimization* class of white-box jailbreak attacks is characterized by its explicit use of model gradients to optimize input prompts. While prefix/suffix manipulation may employ either heuristic algorithms or gradient-based methods—focusing mainly on adversarially editing the prefix or suffix of prompts—gradient-based optimization emphasizes the principled use of gradients to directly guide the search for adversarial inputs. This approach is not restricted to prefix or suffix po-

sitions: it can optimize any part of the prompt, including all tokens, token order, or specific perturbations in embedding space.

Early work, notably GCG and its variants (Jia et al. [291]), demonstrated token-level greedy gradient updates for generating effective jailbreak prompts, with further advancements—such as multi-objective templates and multi-coordinate updating—enabling near-universal attack success and high transferability. Li et al. [292] bridged the gap between input gradients and token substitutions by incorporating transfer attack insights from vision, substantially improving efficiency and robustness. Later studies (Zhu et al. [293]; Zhou et al. [262]; Hu et al. [294]) introduced more interpretable and controllable objectives, including stealthy and linguistically coherent attacks (AutoDAN, COLD-Attack), and explored alternative optimization strategies (e.g., Langevin dynamics). Geiping et al. [260] and Geisler et al. [295] confirmed the generality of gradient-based PGD in relaxed token spaces, showing entropy projections improve efficiency over discrete search, while Pasquini et al. [296] demonstrated adaptability to prompt injection and complex defense pipelines.

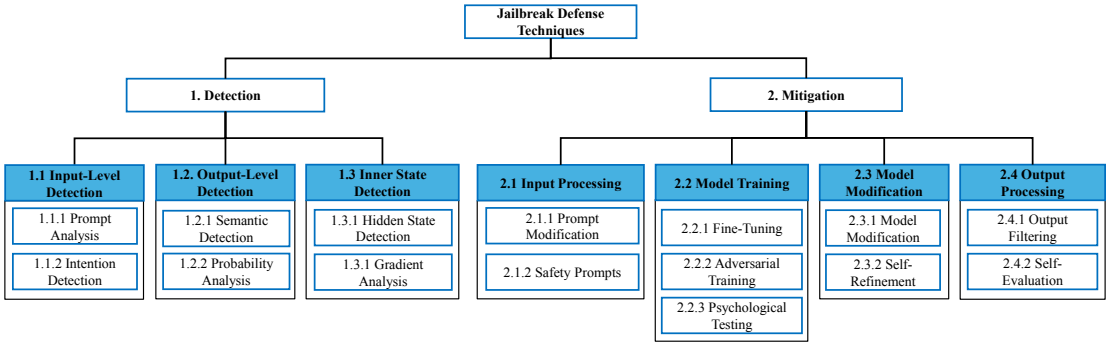


Fig. 5.2: Overview of Taxonomy II: Jailbreak Defense Techniques

5.2.4 Taxonomy II: Jailbreak Defense Techniques

This section presents a taxonomy of jailbreak defense techniques along two complementary pillars: *Detection*—which identifies compromised inputs, outputs, or internal states—and *Mitigation*—which intervenes via input processing, model training, model modification, or output processing to neutralize risks while preserving utility.

II.1 Detection

Detection focuses on identifying and flagging potentially compromised inputs, outputs, or model internal states to enable downstream handling or rejection. In contrast to Mitigation, detection methods do not alter the content or model itself—they serve to recognize and route risks, rather than neutralize them.

II.1.1 Input-Level Detection

Input-level detection refers to a family of proactive defense strategies that operate on user inputs *before* they are fed into large language models (LLMs).

II.1.1.1 Prompt Analysis

Definition: Prompt Analysis, within input-level detection for LLM security, centers on the explicit characterization, modeling, and algorithmic parsing of the textual content of user inputs to distinguish between benign and adversarial prompts *prior* to model execution.

Recent literature advances Prompt Analysis through diverse methodologies. Statistical pipelines like MoJE [297] employ tokenization and n-gram frequency analysis to build fast tabular classifiers, while transformer-embedding methods (e.g., BERT-based [298]) convert prompts into dense vectors for classical ML classification, improving detection and generalization across languages. SPML [299] introduces a programming-languages perspective, using domain-specific languages to formalize prompt structure and check for semantic or property violations.

II.1.1.2. Intention Detection

Definition: Intention detection is a proactive input-level defense paradigm in LLM security, aimed at identifying the underlying intent—whether benign, harmful, or manipulative—embedded in user or adversarial prompts prior to model inference.

Recent literature demonstrates the maturation of intention detection as a central input defense. Zhang et al. [300] formalize a two-stage prompting process to first elicit explicit intent statements, enhancing robustness against covert adversarial prompts. Wang et al. [301] introduce SELFDEFEND, using a dual-LM setup in which a dedicated LLM screens for harmful intentions in parallel, strengthening resistance to adaptive attacks and supporting distillation to open-source models. PrimeGuard [302] in-

corporates intention detection for dynamic risk-based prompt routing, improving safety without sacrificing helpfulness. Debenedetti et al. [270] integrate intent classifiers into agent toolkits, showing substantial reduction in agent takeover rates via pre-inference screening.

II.1.2 Output-level Detection

Output-level detection encompasses techniques that evaluate and classify the content of LLM outputs to intercept and mitigate unsafe, harmful, or adversarial responses.

II.1.2.1 Semantic Detection

Definition: Semantic Detection in output-level LLM security refers to the automated analysis and labeling of model responses based on their semantic content and behavioral characteristics. This approach identifies and discriminates harmful, unsafe, or otherwise undesired outputs directly at the output stage, utilizing machine learning-based classifiers, context-aware analysis, and nuanced interpretation of model behavior to assign safety-related categories or risk labels.

Recent literature establishes semantic detection as a technically rigorous and adaptable foundation for output-level detection. Systems such as BELLS [303], BABY-BLUE [304], WILDGUARD [305], AEGIS [306], MMJ-Bench [307], and XSTEST [308] leverage large, annotated datasets, advanced multi-class and multi-task classifiers, and comprehensive taxonomies that capture diverse forms of harmfulness, refusal, and compliance. These frameworks employ content- and context-aware analysis, step-

wise or token-level evaluation [309], and trajectory-based reasoning to recognize subtle or emergent output risks—including ambiguous, complex, or adversarial behaviors. Methodological innovations include ensemble and online-adaptive classifiers, zero/few-shot adaptability [310], and cross-modal semantic detection for multimodal outputs. Robust evaluation protocols, such as StrongREJECT [311] and JailbreakEval [312], ensure strong alignment with human safety judgments and practical utility. Semantic detection thus enables scalable, explainable, and continually improvable output-level safety for LLMs.

II.1.2.3 Probability Analysis

Definition: Probability analysis within output-level detection refers to the use of probabilistic metrics—such as distributional divergence, likelihood shifts, and aggregate statistical properties—over LLM outputs to identify adversarial attacks. Unlike rule-based or keyword-matching methods, probability analysis quantifies the uncertainty and variation in the output space, aiming to detect subtle and adaptive threats by measuring statistical inconsistencies or atypical behavior in response to perturbed inputs.

JailGuard [313] employs Kullback-Leibler (KL) divergence to measure the distributional differences among LLM responses to mutated benign versus attack queries, flagging attacks when high divergence emerges—thereby operationalizing output-level probability analysis as a statistical gating mechanism. RigorLLM [314] advances this paradigm by integrating probability analysis into an ensemble architecture, combining probabilistic KNN on energy-augmented embeddings with fine-tuned LLM-derived

harmfulness scores, and aggregating these via weighted averaging to make robust, category-aware detection decisions.

II.1.3 Inner State Detection

Inner state detection encompasses techniques that analyze the internal representations or gradients of LLMs in response to input prompts, aiming to identify adversarial intent or unsafe queries by probing the model's underlying decision processes rather than its outputs alone.

II.1.3.1 Hidden State Detection

Definition: Hidden state detection leverages the internal activations (hidden states) of large language models (LLMs) to distinguish between benign and adversarial prompts.

Qian et al. [315] propose the Hidden State Filter (HSF), which utilizes the clustering properties of alignment-trained LLMs in hidden space to achieve lightweight, pre-inference intent classification. HSF demonstrates that different prompt types naturally form separable clusters within the hidden representations, allowing efficient identification of potentially unsafe inputs before generation.

II.1.3.1 Gradient Analysis

Definition: Gradient analysis is an internal state detection method for large language models (LLMs) that directly interrogates model gradients—typically loss gradients—elicited by input prompts, in order to distinguish benign queries from adversarial or jailbreak attempts.

Recent literature, notably Gradient Cuff [316] and GradSafe [317], exemplifies and advances this approach. Gradient Cuff systematically analyzes the refusal loss landscape, uncovering that malicious prompts often induce larger gradient norms and lower refusal losses, and introduces a two-stage framework that combines loss-based rejection with zeroth-order gradient norm estimation for robust jailbreak detection. GradSafe further identifies safety-critical parameter slices, demonstrating that gradients from jailbreak prompts with compliant ("Sure") responses exhibit highly consistent patterns, unlike safe prompts, and constructs unsafety fingerprints via statistical gradient similarity analysis, enabling efficient, finetuning-free screening.

II.2 Mitigation

Techniques that prevent or mitigate the effects of malicious inputs through processing, training, and modification of the model and its outputs.

II.2.1 Input Processing

Input processing is a mitagation technique that preprocess or transform user inputs to mitigate adversarial content before model inference.

II.2.1.1 Prompt Modification

Definition: Prompt modification refers to a class of input processing mitigation techniques that actively transform, augment, or constrain user prompts prior to inference. By altering prompts at the syntactic or semantic level, these methods aim to neutralize adversarial intent or enforce safety constraints, thereby disrupting attack pathways.

A growing body of literature advances this category through increasingly systematic approaches. Hines et al. [318] introduce ”spotlighting,” which uses delimiting, datamarking, and encoding to make prompt provenance salient and suppress indirect injection attacks. Xiao et al. [319] propose retrieval-based prompt decomposition (RePD), separating embedded jailbreak structures via explicit prompt amendments. Delimiter-based segmentation, as in Chen et al. [227], enforces input boundaries recognized by fine-tuned models. Other works pursue prompt reconstruction and denoising—spell-checking, summarization (Lu et al. [222]), paraphrasing, and retokenization (Jain et al. [320])—to neutralize adversarial payloads while maintaining utility. Certifiable frameworks such as erase-and-check (Kumar et al. [321]) provide safety guarantees by systematically removing and checking subsequences. Formal and programmable prompt modification is realized through SPML (Sharma et al. [299]), which enforces chatbot boundaries via domain-specific languages. Backtranslation approaches (Wei et al. [322]) filter prompts by reconstructing user intent from generated outputs. These techniques collectively move prompt modification toward a principled, multi-faceted, and proactive defense paradigm, balancing robustness and utility while introducing programmability, formal guarantees, and resilience against adaptive threats.

II.2.1.3 Safety Prompts

Definition: Safety Prompt subclass in LLM security comprises input-stage defenses that prepend explicit, engineered safety instructions or dynamic prompts to user inputs to mitigate harmful or adversarial behaviors during inference.

The literature demonstrates the evolution and versatility of safety prompts. Wang et al. [323] present AdaShield, a framework for prepending adaptive, context-aware safety prompts—optimized manually or through iterative refinement—to defend against structure-based jailbreaks in multimodal LLMs, highlighting the importance of prompt diversity and adaptive retrieval. Zheng et al. [324] reveal, via representation-level analysis, that safety prompts systematically move queries toward higher refusal rates; their Directed Representation Optimization (DRO) dynamically tunes safety prompt embeddings to balance safeguarding with utility. Zhou et al. [325] formalize safety prompts as robust, transferable system-level suffixes, optimizing defensive chains for resilience against evolving attacks. Suffix-based defenses, including minimax-optimized prompt patching (Xiong et al. [326]) and adversarially co-designed safe suffix insertion (Yuan et al. [314]), further extend this class by appending defensive tokens to inoculate models against injected content. Benchmarking by Zou et al. [228] and Xu et al. [327] underscores the sensitivity of defense effectiveness to both the presence and precise wording of system prompts.

II.2.2 Model Training

Model training defenses aim to proactively improve a model's inherent robustness against jailbreak attacks by systematically shaping its behavior through targeted training strategies, including fine-tuning, adversarial training, and psychological testing.

II.2.2.1 Fine-Tuning

Definition: The fine-tuning class refers to adapting a pre-trained large language model

(LLM) by training it further on carefully chosen, usually small datasets, with the goal of making the model better at resisting security threats—especially jailbreak attacks that trick the model into ignoring its safety rules.

The literature shows that while fine-tuning is a central method for defending against jailbreaks, it also faces important challenges and limitations. Zhang et al. [328] show that using fine-tuning to "unlearn" harmful behaviors can effectively remove clustered malicious knowledge from LLMs, but this method works less well when harmful patterns are scattered. Bianchi et al. [329] find that adding even a small number of safety examples to fine-tuning data can make models much more resistant to jailbreaks, but too much fine-tuning can make the model overly cautious and refuse safe requests. Kim et al. [330] point out that state-of-the-art fine-tuning methods (such as DPO and PPO) still struggle to block new, cleverly designed jailbreak prompts, revealing key limitations of fine-tuning. Fu et al. [331] report that fine-tuning with refusal data helps reduce jailbreaks in both simple and complex tasks, but it is difficult to avoid overfitting and ensure that improvements transfer well to other tasks. Wang et al. [301] introduce SELFDEFEND, a dual-model system that uses fine-tuning to detect a wide range of jailbreaks quickly and transparently. Taken together, these works suggest that fine-tuning can effectively strengthen LLM defenses against jailbreak attacks, but success depends on carefully balancing data quality, calibration, and continuous testing to avoid gaps and unwanted side effects.

II.2.2.2 Adversarial Training

Definition: Adversarial training is a model training paradigm that systematically enhances large language model (LLM) robustness by directly and iteratively engaging with adversarial inputs. Unlike passive safety alignment or standard fine-tuning, adversarial training dynamically generates, mines, or simulates challenging prompts—often through automated red teaming or in-the-wild data collection—and incorporates these adversarial examples into the training process.

Recent literature consolidates adversarial training as the core active defense mechanism for LLM security. Howe et al. [332] demonstrate that only explicit adversarial training, not mere model scaling, ensures consistent robustness against adaptive attacks. Liu et al. [333] and Wallace et al. [334] highlight automatic, iterative adversarial data synthesis and hierarchical training pipelines as critical for countering jailbreaks and generalized prompt attacks. Multi-modal extensions [335], adversarial self-critique [336], and large-scale red teaming frameworks [337, 338] further expand the reach and scalability of adversarial training, while studies on data curation [281] emphasize the necessity of balancing adversarial and high-quality clean data.

II.2.2.3 Psychological Testing

Definition: Psychological testing for LLM safety refers to the integration of psychological assessment techniques—especially those targeting manipulative, deceptive, or ”dark” traits—into the training, evaluation, and defense of language models and multi-agent systems. This approach involves designing and administering standardized psychological instruments (e.g., Dark Triad Dirty Dozen, moral dimension scales) or

custom-crafted prompts that probe or simulate manipulative intent. The aim is to detect, characterize, and ultimately mitigate the model's susceptibility to psychological manipulation or the generation of harmful, manipulative outputs.

Recent works such as PsychoBench [339] provide comprehensive psychometric frameworks to systematically evaluate LLMs across a wide range of psychological attributes—including personality traits, motivational states, and emotional abilities—using validated clinical scales. These techniques enable deeper understanding of LLMs' psychological profiles, reveal alignment gaps, and inform targeted improvements in model behavior. In multi-agent settings, PsySafe [340] advances psychological testing as both an attack and defense mechanism: dark personality trait injection is used to induce manipulative or dangerous behaviors, while psychological assessments are applied to monitor and remediate agent states, e.g., through doctor-agent intervention. Results consistently show that higher scores on dark trait assessments correlate with increased risk of manipulative or unsafe behaviors, and that psychological testing provides a direct, quantifiable signal for proactive defense and mitigation strategies. This paradigm establishes psychological testing as a crucial, model-agnostic approach for the detection and reduction of manipulative vulnerabilities in LLMs and agent systems.

II.2.3 Model Modification

Model modification approaches enhance LLM robustness by directly altering model internals—such as parameters, architectures, or representations—to proactively

mitigate adversarial risks beyond external filtering or system-level defenses.

II.2.3.1 Model Modification

Definition: Model Modification category encompasses defense techniques that directly alter the internal parameters, architectures, or representations of large language models (LLMs) to enhance robustness against adversarial threats such as jailbreak attacks. Unlike input/output filtering or external system-based controls, model modification operates by intervening within the model itself.

Recent literature demonstrates a diverse landscape of model modification strategies. Layer-specific editing ([341]) pinpoints and adjusts safety-critical layers within transformer architectures, improving refusal rates for adversarial prompts. Knowledge editing and targeted unlearning ([342], [343], [328]) directly erase hazardous capabilities, but can induce "ripple effects", unintentionally suppressing related knowledge due to shared internal representations. Detoxification via intraoperative neural monitoring ([342]) enables precise mitigation by editing specific toxic weights with minimal side effects. Manipulating internal representations at runtime, such as through safety-direction interventions ([344]), allows adaptive safety-utility tradeoffs with low computational cost. Emerging approaches like circuit breaking ([345]), representation engineering ([328]), and merging self-critique models ([336]) further generalize protection by reshaping model internals to defend against both known and novel attacks. Collectively, these works highlight model modification's distinctive capacity for durable, attack-agnostic, and real-time defense—while also surfacing open challenges such as

preventing negative side effects and optimizing the safety-utility tradeoff.

II.2.3.2 Self-Refinement

Definition: Self-optimization in the context of model modification for LLM defense refers to techniques by which models autonomously refine their parameters, internal representations, or operational behaviors to mitigate security risks, without external retraining or significant human intervention.

Recent literature advances self-optimization through diverse mechanisms. Kim et al. [346] propose iterative self-refinement, where LLMs employ self-feedback loops, sometimes with format-based attention shifting—to revise outputs against adversarial prompts, surpassing static defenses. Wang et al. [323] introduce AdaShield, enabling multimodal LLMs to generate adaptive, context-sensitive defense prompts via a defender–target model loop, achieving on-the-fly protection without model updates. Li et al. [347] present RAIN, where frozen LLMs use self-evaluation and a rewind-and-search mechanism during inference to avoid harmful continuations, embodying pure self-optimization without retraining or annotation. Zheng et al. [324] develop Directed Representation Optimization (DRO), allowing model-internal adjustment of safety prompt embeddings to recalibrate refusal thresholds based on model-derived refusal vectors, enabling continuous safety self-calibration without altering model weights. Intention Analysis mandates the model to first explicitly extract the user's intent, then restricts its response in alignment with this self-identified intent, effectively enforcing self-consistency [300].

II.2.4 Output Processing

Output processing refers to post-generation mitigation techniques that analyze and control model outputs before they are delivered to users, enabling final-stage defense against unsafe, adversarial, or disallowed content.

II.2.4.1 Output Filtering

Definition: Output filtering refers to a root defense mechanism in LLMs that operates by analyzing and controlling the generated model response at the final stage of output processing, rather than intervening at the input level. This approach directly intercepts and evaluates model outputs before they are delivered to users, enabling prompt-agnostic and model-agnostic interventions to suppress unsafe or adversarial content.

Recent literature advances output filtering along both architectural and methodological lines. Multi-agent frameworks such as AutoDefense employ collaborative LLM agents to assess outputs based on user intent, reconstructed prompts, and content validity, achieving robust and modular filtering against sophisticated jailbreaks [348]. Fine-grained mechanisms like SafeAligner and SafeDecoding apply token-level filtering during decoding, dynamically adjusting sampling probabilities and leveraging expert models to prioritize safe continuations [349, 350]. Root Defense Strategies enhance this paradigm through iterative, token-wise safety checks and resampling [309]. Template-based methods (e.g., SelfDefend) pair output filtering with shadow models for low-latency rejection or explanation, supporting broad deployment [301]. Empirical results from red-teaming and competitions (e.g., SaTML 2024) confirm the utility

of universal and regex-based filters as baseline defenses against overt information leakage [351, 352]. A recurring insight is that output filtering, by decoupling from input heuristics and model internals, offers resilience and scalability but faces challenges in balancing strictness, utility, and explainability—especially against indirect or multilingual attacks.

II.2.4.2 Self-Evaluation

Definition: Self-evaluation output processing mitigation mechanisms enable LLMs to proactively assess and regulate their own outputs, identifying and mitigating unsafe or undesirable content through internal reasoning or critique, either during or immediately after generation. This distinguishes self-evaluation from static filters or external interventions, as the model acts as both generator and first-line evaluator.

Among surveyed methods, PrimeGuard instantiates self-evaluation by having the LLM perform an explicit risk analysis on its own candidate outputs before release, dynamically adjusting subsequent responses to maintain safety [302]. RAIN integrates self-assessment into the generation loop: the LLM iteratively reviews in-progress completions, rewinding whenever its own judgment deems the content noncompliant [347]. Backtranslation defense requires the LLM to reconstruct the original prompt from its output, refusing the initial input if the self-generated prompt would be blocked, thus using model-driven intent inference for self-checking [322]. Prefix Guidance prompts the model to self-classify the request's safety and generate a canonical refusal prefix; both the style and rationale are then used for algorithmic filtering, leveraging the

model's own refusals for diagnostic signals [353]. Merging-based approaches enhance self-critique by combining a base model with a fine-tuned external critic, expanding the capacity for self-assessment within a unified model [336]. SelfDefend operates by having the model run an internal check for harmful intent alongside normal generation, enabling real-time, model-driven detection of attacks [301]. Finally, self-reflection paradigms add a distinct, explicit self-review phase after initial reasoning; the LLM attempts to critique and correct its own answer, although current techniques still face limitations in reliably flagging subtle adversarial prompts [258].

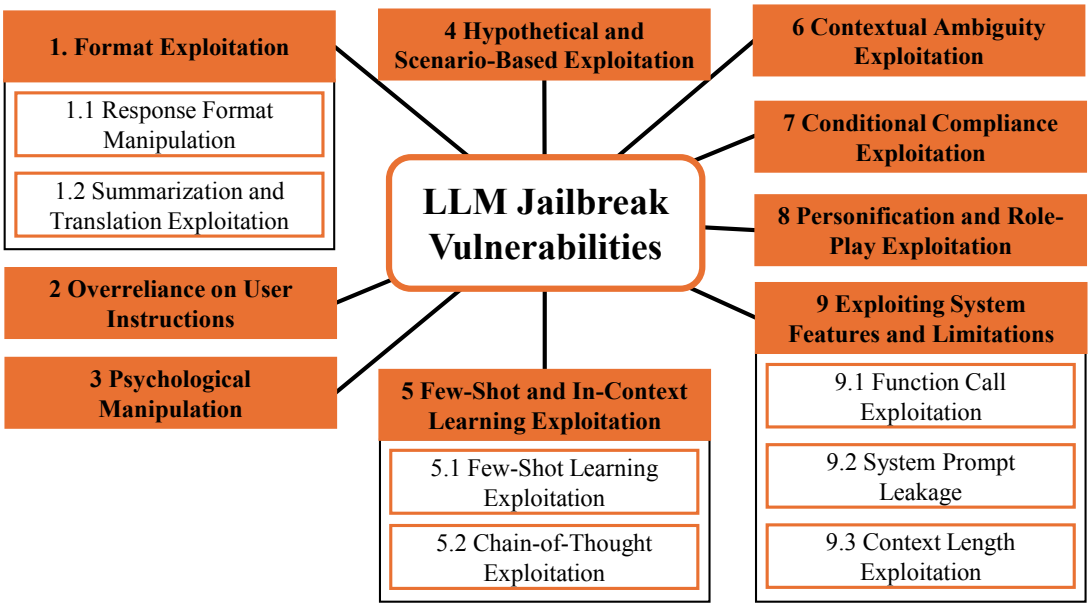


Fig. 5.3: Overview of Taxonomy III: LLM Jailbreak Vulnerabilities

5.2.5 Taxonomy III: Model Vulnerabilities

III.1 Format Exploitation

Techniques that exploit the model's handling of specific input or output formats.

III.1.1 Response Format Manipulation

Definition: Format Exploitation refers to a class of vulnerabilities where adversaries target the structural, semantic, and syntactic features of LLMs' response formatting mechanisms—such as code blocks, tables, pseudocode, tags, or contextual markers—to bypass or subvert alignment safeguards. Rather than manipulating content alone, these attacks exploit the model's interpretation and generation of structured templates, lever-

aging the underlying output regularities and parsing logic as a pivot for evasion or triggering of unsafe behaviors.

Recent literature demonstrates that format exploitation extends beyond content perturbations to include the deliberate design and mutation of response templates, thereby amplifying the attack surface. Banerjee et al. [284] and Zhao et al. [282] reveal that response formats can serve as covert channels, making instruction-centric and structure-driven outputs (e.g., pseudocode or manipulated templates) more vulnerable to jailbreaks. CodeChameleon [354] exemplifies attacks embedding payloads within code completion or encryption routines, which evade intent detection by safety mechanisms. Structural manipulation—using graphs, tables, or multi-turn format shifts—as discussed by Li et al. [355] and Jin et al. [356], further exposes failures in models' generalization of alignment to long-tail or compositional outputs. On the detection side, JailGuard [313] and Kim et al. [346] exploit the instability and low robustness of adversarial formats, showing that format-mutation attacks (such as typographic perturbations or code wrapping) induce non-robust responses. Pasquini et al. [296] highlight the adversarial use of formatting triggers, such as tags or comments, which reliably activate malicious payloads even in retrieval-augmented generation (RAG) settings. Collectively, these works clarify that format is an active locus of vulnerability and detection, establishing the need for format-aware alignment and defense strategies in LLM security research.

III.1.2 Summarization and Translation Exploitation

Definition: Summarization and Translation Exploitation refers to a subclass of LLM vulnerabilities where attackers leverage summarization or translation workflows to bypass safety controls and alignment mechanisms.

A growing literature demonstrates that translating malicious prompts into low-resource or less-defended languages, or reframing harmful requests as summarization or translation tasks, allows adversaries to evade primary safety measures [203, 357]. Empirical studies show LLMs often fail to filter dangerous content re-encoded through these tasks, especially when alignment in translation or summarization is weak [207, 331, 358, 359]. Techniques such as embedding adversarial instructions within translation/summarization templates and automatic semantics-preserving translation further increase attack success rates, particularly in low-resource or multilingual scenarios. These attacks frequently evade detection because they mimic legitimate task requests and can bypass both keyword and intent-based safety filters, highlighting the need for robust cross-task and multilingual alignment in defense strategies [222, 311].

III.2 Overreliance on User Instructions

Definition: Over-reliance on Instructions denotes a structural vulnerability in LLMs wherein models excessively and indiscriminately execute instructions embedded in their input, irrespective of provenance, privilege, or contextual hierarchy. This susceptibility arises from models' intrinsic tendency to generalize instruction-following capabilities, lacking robust mechanisms to authenticate or differentiate between trusted (system/developer) and untrusted (user/external) commands, thereby enabling adver-

saries to manipulate model behavior via crafted instructions.

Recent literature systematically reveals the scope of this vulnerability. Hines et al. [318] provide foundational analysis, demonstrating that instruction-tuned LLMs are inherently prone to indirect prompt injection, especially when instructions originate from user-controlled or external data; their "spotlighting" approach exposes the limits of demarcating instruction boundaries. Kimura et al. [275] empirically show that vision-language models readily obey adversarial instructions even when encoded as text in images, exacerbated by stronger instruction compliance. Wallace et al. [334] introduce an instruction hierarchy framework, emphasizing that robust LLMs must systematically prioritize privileged over unprivileged commands, with privilege-aware fine-tuning yielding significant robustness gains. Zhan et al. [360] extend this to LLM agents, where unfiltered action and tool-based instructions from untrusted environments lead to high attack success rates. Studies on safety-tuned LLaMA [329] and Banerjee et al. [284] further reveal that enhanced instruction-following, unless tightly privilege-controlled, amplifies both user helpfulness and susceptibility to malicious prompts, sometimes causing excessive refusals or ethical bypasses. Toyer et al. [204] and Chen et al. [227] reinforce that attackers exploit models' inability to distinguish genuine system instructions from adversarial meta-commands, rendering naive content filtering insufficient. Ren [361] and Nestaas et al. [362] highlight that even simulated agent outputs or persuasive claims are trusted if formatted as instructions, evidencing the inadequacy of instruction-as-authentication. Collectively, these works underscore that Over-reliance

on Instructions is not merely an alignment issue but a fundamental flaw rooted in insufficient privilege separation and trust modeling, necessitating systemic, architectural, and procedural reforms in LLM design.

III.3 Psychological Manipulation

Definition: Psychological manipulation refers to exploiting the LLM's alignment through persuasive or socially engineered prompt strategies, leveraging insights from psychology, communication, and social sciences. Rather than directly circumventing guardrails through technical means, this category targets the model's behavioral vulnerabilities by mimicking human-like persuasive communication—such as emotional appeals, logical reasoning, authority endorsements, or scenario framing—coercing the model to generate otherwise restricted or harmful content.

Recent studies systematically investigate the link between psychological manipulation and LLM vulnerabilities. Zeng et al. [363] introduce a comprehensive persuasion taxonomy, demonstrating that persuasive adversarial prompts (PAPs) built on social science principles can consistently bypass LLM safety measures, achieving high jailbreak success rates on both open- and closed-source models. Their results show that techniques such as logical appeal, authority endorsement, and emotional manipulation are highly effective, especially when tailored to specific risk domains. This work highlights that more advanced and helpful models are paradoxically more susceptible to human-like persuasive attacks, and that existing defenses—focused on detecting optimization-based or pattern-driven prompts—are insufficient for nuanced, context-rich manipula-

tions. Cold-Attack [294] complements this by enabling fine-grained control over attack features (e.g., sentiment, style, contextual fluency), thus expanding the landscape of psychological manipulation to include stealthier and more diverse jailbreak scenarios. These insights reveal an urgent need to bridge AI safety and social science, and to develop adaptive defenses grounded in understanding the model's susceptibility to human-like influence, not just technical exploits.

III.4 Hypothetical and Scenario-Based Exploitation

Definition: Hypothetical and Scenario-Based Exploitation denotes a category of vulnerabilities where attackers construct hypothetical, fictional, or contextually-rich scenarios to subvert LLM alignment and safety mechanisms. This approach manipulates the model's underlying assumptions, situational logic, and scenario-based reasoning—often via role enactment or narrative construction—not to mimic or personify a specific character, but to mislead the model's inference about user purpose and context. Unlike Personification and Role-Play Exploitation, which center on impersonating particular roles or entities to evoke role-consistent behaviors, this class exploits the plausibility and narrative logic of entire situations, regardless of the user's assumed identity.

Recent studies reveal that Hypothetical and Scenario-Based Exploitation presents distinctive risks by leveraging multi-turn scenario framing, narrative context, and hypothetical intent. For instance, RED QUEEN demonstrates that embedding requests in plausible scenarios (e.g., investigative or educational settings) systematically cir-

cumvents filters by targeting LLMs' limitations in Theory of Mind [364]. GUARD and Visual-RolePlay further show that scenario-rich, dynamically generated contexts can be synthesized at scale, embedding attacks within ostensibly harmless dialogues to challenge moderation [244, 245]. A Wolf in Sheep's Clothing generalizes such attacks by nesting prompts within layered narrative contexts, decoupling the attack from static string patterns [365]. Multimodal works, including Image-to-Text Logic Jailbreak and Arondight, extend this vector to visually depicted scenarios, illustrating that scenario exploitation is not limited to text but also affects vision-language models [237, 366]. Automated and agentic systems, such as SoP, AutoDAN-Turbo, and RedAgent, systematically evolve and optimize scenario-based attacks, leveraging feedback and memory to autonomously discover new vulnerabilities [238, 246, 367]. Social engineering research highlights that trust-building and authority-based scenarios further amplify attack success, often with fewer queries than direct requests [368].

III.5 Few-Shot and In-Context Learning Exploitation

Techniques that leverage few-shot or in-context learning to bypass safeguards.

III.5.1 Few-Shot Learning Exploitation

Definition: Few-Shot Learning Exploitation refers to the manipulation of large language models (LLMs) through the insertion of a small set of carefully designed demonstration-response pairs or behavioral examples directly into the model's prompt context. This approach leverages LLMs' inherent in-context learning and pattern generalization capabilities to induce target behaviors.

Recent literature demonstrates that few-shot exploitation poses a critical attack surface unique to LLMs. Foundational works such as ICA [369] and advICL [370] establish that inserting a small number of harmful or refusal demonstrations into the context can drastically increase attack success, with the required number of shots growing only logarithmically with success probability. Studies on scaling laws [371] reveal that larger models with longer context windows are more susceptible to such attacks. Transferability analyses [370] show that optimized demonstration sets can generalize to unseen harmful instructions, and recent research [188, 249, 372] extends these vulnerabilities to vision-integrated and multimodal LLMs. Moreover, iterative or contrastive prompting methods [243, 373] demonstrate that LLMs can autonomously generate and refine potent attack or defense demonstrations. Across studies, defense mechanisms such as output filters, perplexity-based detection, or system prompts consistently lag behind the adaptive threat posed by few-shot/in-context exploitation, highlighting the fundamental alignment risks inherent to this category as models scale and context windows expand.

III.5.2 Chain-of-Thought Exploitation

Definition: Chain-of-Thought (CoT) Exploitation encompasses both attacks that manipulate the reasoning trajectory of large language models (LLMs) and those that leverage CoT prompting as a tool to construct more effective attack strategies. This class includes (1) adversarial interventions in the model's multi-step or compositional reasoning—such as inserting distractors, preemptive answers, or disguised cues into the

reasoning chain to covertly steer outputs—and (2) attack methodologies that explicitly harness CoT-style prompting (e.g., stepwise decomposition, role-play, or iterative guidance) to bypass alignment safeguards or to reveal sensitive information.

The literature evidences both forms of CoT exploitation. Xu et al. [258] demonstrate that adversaries can compromise LLMs' reasoning by introducing "preemptive answer" cues or distractors ahead of reasoning chains, significantly degrading robustness. Bhardwaj et al. [253] show that "Chain of Utterances" structures in multi-turn prompts lower refusal rates and enable harmful completions through engineered internal thoughts. Multi-step jailbreaking [374] and Figure-it-Out Jailbreaks [256] use CoT-style decomposition, role-play, and iterative guessing to erode privacy protections and gradually induce unsafe outputs. "Flipping" or disguised prompts [180] manipulate the reasoning process by obfuscating harmful content, then cueing the model to reconstruct it through chained in-context instructions. In the vision-language domain, attacks [375] coordinate visual and textual CoT prompts, using feedback loops to optimize adversarial strategies. Other works [284, 304] illustrate that iterative, structure-aware prompting based on CoT principles amplifies unethical outputs. Finally, compositional pipelines [376] leverage CoT-style task decomposition across multiple models, chaining benign subtasks to achieve overall unsafe outcomes. Multi-turn chain-of-attack approaches [251] illustrate how gradual, contextually linked prompts can progressively guide models to compliance.

III.6 Contextual Ambiguity Exploitation

Definition: Context Ambiguity Exploitation refers to adversarial techniques that leverage uncertainties or ambiguities within an input's context to subvert the semantic boundaries that large language models (LLMs) use for task execution and content moderation. Unlike prompt attacks based on overt manipulation or token-level triggers, this category exploits subtle or artful confusion, blurring the contextual cues that allow LLMs to distinguish between benign and malicious intent.

Recent literature consistently demonstrates that LLMs are vulnerable to context ambiguity. Shang et al. [201] show that ambiguous or obfuscated queries can induce confusion and evade malicious intent detection, even in robust models. Mei et al. [304] highlight that context conflicts and ambiguity are key sources of false positives/negatives in jailbreak detection, emphasizing the need for evaluators sensitive to context-incoherence. Ren et al. [182] reveal that multi-turn attacks can subtly shift semantics within seemingly benign contexts, exploiting LLMs' reliance on contextual continuity. Bianchi et al. [329] report that safety-tuned LLMs may conflate ambiguous but safe inputs with unsafe content due to limited disambiguation ability. Pasquini et al. [296] demonstrate that inadequate instruction/data boundaries enable adversarial payloads to be misclassified as benign context. Pellegrino et al. [377] provide evidence that RAG-based LLMs are particularly susceptible to context ambiguity, as ambiguous or contradictory retrieved content can facilitate indirect prompt manipulation. Together, these works underscore that context ambiguity exploitation remains a critical and insidious vulnerability, demanding improved methods for context disentanglement and

intent inference to ensure LLM safety.

III.7 Conditional Compliance Exploitation

Definition: Conditional Compliance Exploitation refers to a nuanced subcategory of LLM vulnerabilities in which adversaries elicit unsafe or policy-violating outputs not through explicit prompts, but by manipulating the conditions under which models judge compliance. This exploitation relies on context, prompt structure, or model state—triggering harmful behaviors only when specific semantic, syntactic, or positional cues are present, and remaining benign otherwise.

Recent literature systematically reveals the mechanisms and scope of conditional compliance exploitation. Universal backdoor attacks [221, 285] demonstrate that secret triggers or adversarial suffixes can universally subvert refusals while preserving normal outputs in their absence, highlighting detection challenges. DeepInception [378] shows that embedding harmful requests within nested, authority-based, or role-driven scenarios exploits psychological vulnerabilities, enabling harmful generations without explicit unsafe input. In multimodal settings, structure-based triggers such as visual encoding of harmful text [323] or specific prompt arrangements [379] further expand conditional compliance into new modalities. Studies model editing [280] expose how conditional boundaries are set by fine-grained training and editing artifacts, with safety either leaving compliance paths open or inducing exaggerated refusals. Social manipulation, reverse psychology, and switch techniques [197] illustrate the practical ease of crafting context-dependent states that elicit forbidden content. Notably, re-

cent work [267] exposes the structural depth of this vulnerability by showing that cost-based decoding paths can reveal harmful completions along alternative, conditionally triggered sequences. Collectively, these studies underscore that conditional compliance exploitation is adaptive, often invisible to surface-level testing, and difficult to mitigate without context-sensitive alignment, highlighting a persistent tradeoff between model capability and safety.

III.8 Personification and Role-Play Exploitation

Definition: Personification and Role-Play Exploitation is a class of LLM vulnerability exploitation characterized by manipulating the model's personification and fictional imagination faculties via character-based prompts. This method subverts safety guardrails by inducing the model to adopt roles or engage in imagined narratives, prompting it to temporarily suspend normative safety judgments. Unlike token-level or encoding-based attacks, role-playing exploitation exploits the LLM's conversational and instruction-following design, leveraging persona adoption to mask, distract from, or justify malicious content within a fictional context.

Recent literature converges on the principle that prompted role-play systematically enables normative violations that would otherwise be blocked. Yang et al.[238]introduce SEQAR, which auto-generates and optimizes multiple jailbreak characters to maximize distraction and context division, each sequentially responding as distinct agents. Li et al.[378]'s DeepInception frames role-playing as psychological inception, stacking nested fictional scenes and authority-submissive character hierarchies to induce "self-

losing" in the model and amplify harmful output. .

III.9 Exploiting System Features and Limitations

III.9.1 Function Call Exploitation

Definition: Function Call Exploitation is a subclass of System Feature Exploitation that specifically targets the function calling capability within large language models (LLMs). This vulnerability arises when adversaries manipulate the system's function call interface—intended for tool use or API integration—to force execution pathways that circumvent alignment and safety controls, often by exploiting insufficient input validation or overly permissive API schemas.

The literature demonstrates the potency and generality of function call exploitation as an attack vector. Wu et al. [274] provide a systematic study, revealing that attackers can craft function call inputs which evade prompt-level and model-level alignment, leading to reliable jailbreaks across multiple models and operational settings. Their results highlight that standard filtering and alignment strategies are inadequate when system-level interfaces like function calling are insufficiently hardened, exposing a deep and transferable attack surface that is not addressable through prompt engineering or superficial dataset curation alone.

III.9.2 System Prompt Leakage

Definition: System Prompt Leakage (Stealing) is a vulnerability class wherein adversaries extract or manipulate a model's confidential system prompt or hidden instructions by exploiting inherent features or operational characteristics of LLM-based systems.

Unlike conventional prompt injection, this class targets privileged, non-user-exposed context, enabling downstream attacks—such as prompt theft, persistent jailbreaks, or extraction of sensitive behavioral policies—without requiring access to model internals.

Recent studies reveal that attackers can systematically exfiltrate system prompts by leveraging the LLM's context handling and instruction-following mechanisms. Wu et al. [380] demonstrate prompt theft in multimodal LLMs via API-based dialogue manipulation, even automating the conversion of stolen prompts into jailbreak payloads. Geiping et al. [260] extend this by categorizing "prompt extraction" and "prompt repeater" attacks, showing adversarial input can act as "arbitrary code" to extract hidden directives under strict UI controls. Liu et al. [381] highlight the risks at the code-data boundary in LLM applications, where context separation failures facilitate leakage. On the defense side, Khomsky et al. [352] find that multi-layered protections remain vulnerable to evasion via obfuscated outputs, while Hines et al. [318] propose "spotlighting"—using prompt provenance encoding to reduce extraction. Collectively, these works show that system prompt leakage exploits unique vulnerabilities at the privileged context boundary, often bypassing standard sanitization and highlighting the necessity for more robust isolation mechanisms in LLM security.

III.9.3 Context Length Exploitation

Definition: Context Length Exploitation refers to adversarial techniques that leverage the token-based context window and its length limitations to manipulate large language models (LLMs) into producing unintended or unsafe outputs.

Recent literature systematically investigates context length exploitation. Schulhoff et al. [202] identify attacks such as ”Context Overflow,” where adversaries pad prompts to displace or suppress benign instructions, manipulating model behavior through token budget exhaustion. Anil et al. [371] generalize this via ”Many-Shot Jailbreaking,” demonstrating that increasing context length enables attackers to reliably override safety alignment by populating the context with adversarial demonstrations; they show attack effectiveness scales with window size and is not fully mitigated by standard alignment. Geiping et al. [260] further expand this scope, showing that context manipulation enables prompt extraction, output misdirection, and the exploitation of model-internal quirks such as ”glitch tokens.” These works collectively show that context length exploitation is a fundamental vulnerability rooted in LLM architecture, challenging the integrity and safety of LLMs as context windows grow.

5.3 Dataset

During our survey of research papers on jailbreak attacks, jailbreak defenses, and surveys of jailbreak attacks, we collected and organized the datasets released by various sources. As a result, we have compiled the largest publicly available dataset on jailbreak attacks within the open-source community. In addition, we have compiled a dataset of over one million benign prompts.

5.3.1 Data Collection

To investigate the current landscape of open-source datasets related to jailbreak attacks on LLMs, we conducted a comprehensive survey of relevant literature. Our collection includes 31 papers related to jailbreak overview and benchmark content, 102 black-box jailbreak attack works, and 31 white-box jailbreak attack works. We have collected and organized the open source datasets from all the work to form the largest jailbreak and benign prompt dataset in the community so far, which contains 445,752 jailbreak prompt data from 48 sources and 1,094,122 benign prompt data from 14 sources.

The dataset is organized into five columns: *system_prompt*, *user_prompt*, *jailbreak*, *source*, and *tactic*. The *system_prompt* and *user_prompt* columns denote the system prompt and the user prompt submitted to the model, respectively. The *jailbreak* column denotes whether the prompt constitutes a jailbreak attempt (1 for jailbreak, 0 for benign). The *source* column denotes the origin of the data, and the *tactic* column denotes whether a jailbreak tactic is employed (1 for prompts using jailbreak tactics, 0 for prompts not using jailbreak tactics).

Table 5.2 and Table 5.3 respectively present statistics on the number of data and the average prompt length for each source in the jailbreak prompt and benign prompt datasets.

Table 5.2: Jailbreak Prompt Dataset Information

Source	Count	Average Prompt Length
allenai/wildjailbreak [338]	134778	648.28
ReNeLLM [365]	125494	548.49
FuzzLLM [382]	62136	1758.48
AetherPrior/TrickLLM [383]	40245	6084.99
GPTFuzzer [239]	11808	2088.01
CPAD [384]	10050	127.38
yueliu1999/FlipGuardData [180]	8840	974.66
Knowledge-to-Jailbreak [385]	7712	909.14
SoftMINER-Group/TechHazardQA [386]	7301	96.60
h4rm3l [387]	5294	852.01
suffix-maybe-feature/adver-suffix-maybe-features [282]	4556	78.08
TemplateJailbreak [194]	4100	2222.62
ECLIPSE [286]	4021	109.67
SAP [373]	2120	794.73
AdaPPA [388]	1948	436.19
jailbroken [389]	1898	963.47
Safe-RLHF [390]	1794	68.39
tml-epfl/llm-past-tense [391]	1419	101.08
DAN [392]	1398	2681.40
ACE [199]	1050	959.77
TAP [393]	831	508.37
JAILJUDGE [394]	826	357.18
GPT Generate [395]	763	72.31
PAIR [396]	741	464.19
Adaptive Attack [397]	689	1805.27
AdvBench [230]	515	73.01
BeaverTails [398]	493	70.56
WUSTL-CSPL/LLMJailbreak [219]	447	1756.81
HarmBench [399]	411	88.08
StrongREJECT [311]	238	168.28
Question Set [395]	215	74.44
DRA [216]	212	1496.65
GCG [230]	200	223.45
AutoDAN [400]	187	3416.41
hh-rlhf [395]	167	83.86
PAP [363]	165	1029.44
GBDA [401]	137	582.16
Handcraft [395]	124	107.77
MaliciousInstruct [263]	100	61.70
GPT Rewrite [395]	96	66.69
DeepInception [378]	64	558.66
JBB-Behaviors [191]	55	95.42
EnsembleGCG [230]	34	613.15
UAT [402]	33	401.73
AutoPrompt [403]	21	741.86
DirectRequest [399]	15	1198.47
LLM Jailbreak Study [395]	8	129.00
MasterKey [404]	3	77.67

Table 5.3: Benign Prompt Dataset Information

Source	Count	Average Prompt Length
OpenHermes-2.5 [405]	494829	927.15
glaive-code-assist [406]	181505	409.57
allenai/wildjailbreak [338]	128963	520.90
CamelAI [407]	77276	218.21
EvolInstruct_70k [408]	44393	559.97
cot_alpaca_gpt4 [409]	42022	85.96
metamath [410]	36596	244.34
airoboros2.2 [411]	35320	487.79
platypus [412]	22126	547.99
DAN [392]	16989	1552.36
UnnaturalInstructions [413]	6595	369.20
CogStackMed [414]	4408	211.15
LMSys Chatbot Arena [415]	3000	192.38
JBB-Behaviors [191]	100	73.75

5.3.2 Feature Extraction

We designed 58 heuristic features categorized into seven distinct groups, as detailed below:

- **Linguistic**: Features related to basic linguistic properties of text, such as av-

erage word length, word frequency, part-of-speech usage, lexical diversity, and sentence structure.

- **Syntactic**: Features capturing sentence-level grammatical structure, including dependency relations, number of clauses, and use of conditionals or specific syntactic forms.

- **Semantic**: Features reflecting the meaning, tone, or sentiment of the text, including polarity, subjectivity, and emotional tone.

- **Pragmatic**: Features associated with discourse style, speaker intent, or interpersonal strategies, such as politeness, evasiveness, rhetorical structure, and emotional manipulation.

- **Safety Risk**: Features designed to detect potentially harmful, misleading, incoherent, or offensive language, including toxicity, misinformation, logical fallacies, and deceptive cues.

- **Behavioral**: Features related to adversarial behavior or attempts to manipulate or bypass system boundaries, such as jailbreaks, conflicting prompts, or hidden instructions.

- **Metadata**: Named entity recognition features that quantify references to persons, organizations, locations, dates, and other entity types for contextual or content tracking.

A comprehensive overview of all heuristic features is presented in Table 5.4, including the description of each feature, its corresponding group, and the average processing time per token.

We extracted features from the jailbreak dataset and an equal-sized dataset of benign prompts using the 58 heuristic features proposed in Section 5.3.2, resulting in a mixed dataset composed of corresponding heuristic feature vectors. From each source of jailbreak and benign prompts, we randomly sampled 100 instances. Using jailbreak prompts from only one source as the test set and the remaining data as the train set, we trained a simple XGBoost classifier to evaluate whether it could effectively detect previously unseen types of jailbreak attacks in train set. The experimental results are shown in Figure 5.4. For most types of jailbreak methods, the trained XGBoost classifier achieved a detection success rate of over 70%. These experimental results demonstrate that even a simple binary classifier can serve as an effective coarse-grained filter for identifying previously unseen jailbreak attacks.

It is worth noting that XGBoost performs poorly when detecting data from four specific sources individually: HarmBench, ml-epfl/llm-past-tense, CPAD, and ACE. We analyzed the reasons behind this. HarmBench is a dataset containing only original harmful questions. Since the trained XGBoost model fails to fully capture these harmful prompts, it suggests that XGBoost primarily learns jailbreaking techniques from the training data, rather than recognizing harmful content itself. Similarly, ml-epfl/llm-past-tense consists of attacks that convert harmful prompts into past or future tense.

Table 5.4: All Heuristic Features

Feature	Function	Group	Avg Time (ms/token)
entropy	Measure of word distribution diversity in the text.	Linguistic	0.000257
avg_word_length	Average length of words.	Linguistic	0.069214
type_token_ratio	Lexical diversity.	Linguistic	0.000078
noun_ratio	Ratio of nouns to all POS tokens.	Linguistic	0.017500
verb_ratio	Ratio of verbs to all POS tokens.	Linguistic	0.026251
adj_ratio	Ratio of adjectives to all POS tokens.	Linguistic	0.013125
adv_ratio	Ratio of adverbs to all POS tokens.	Linguistic	0.013126
polarity	Sentiment polarity (positive vs. negative).	Semantic	0.081281
subjectivity	Degree of subjectivity in the text.	Semantic	0.081549
politeness_score	Heuristic score indicating politeness.	Pragmatic	0.133763
deception_score	Likelihood of deceptive language use.	Pragmatic	0.000148
ambiguity_score	Presence of vague or unclear expressions.	Pragmatic	0.035185
avg_token_length	Average number of characters per token.	Linguistic	0.000213
readability_score	Ease of reading (e.g., Flesch-Kincaid).	Linguistic	0.020912
num_clauses	Number of clauses detected.	Syntactic	0.125693
num_dependencies	Number of dependency relations.	Syntactic	0.123616
negation_presence	Frequency of the appearance of negative words.	Linguistic	0.027588
pronoun_ratio	Ratio of pronouns in the text.	Linguistic	0.036568
sentence_length	Average number of tokens per sentence.	Linguistic	0.005479
avg_word_freq	Mean frequency of words from a corpus.	Linguistic	0.000257
modal_verb_ratio	Proportion of modal verbs (can, should, etc.).	Linguistic	0.037158
question_ratio	Frequency of interrogative sentences.	Pragmatic	0.000001
imperative_presence	Presence of imperative commands.	Pragmatic	0.118916
toxicity_score	Score reflecting offensive or toxic language.	Safety Risk	0.014235
nonsense_score	Heuristic score for incoherence or gibberish.	Safety Risk	0.000407
emotional_manipulation_score	Score reflecting emotional manipulation.	Pragmatic	0.032445
evasive_language_score	Score reflecting the use of evasive or avoidant language.	Pragmatic	0.030615
jailbreak_score	Tendency to bypass system restrictions.	Behavioral	0.041695
conflicting_instruction_score	Degree of self-contradiction.	Behavioral	0.032872
persona_manipulation_score	Attempt to exploit model's persona.	Behavioral	0.038160
compute_language_score	Detect the language of the text.	Pragmatic	0.016813
false_safety_score	Safety-related but misleading language.	Safety Risk	0.034184
misinformation_score	Scores of the terms and phrases in the misinformation.	Safety Risk	0.035483
fake_expertise_score	Score of false professional knowledge.	Safety Risk	0.035417
leading_question_score	Score of leading questions in the text.	Pragmatic	0.034241
repetition_score	Redundancy or repeated phrases.	Linguistic	0.000186
cognitive_load	Calculate cognitive load score based on cognitive terms.	Pragmatic	0.028029
positive_tone	Presence of positive emotional language.	Semantic	0.033256
negative_tone	Presence of negative emotional language.	Semantic	0.031766
social_word_frequency	Frequency of socially oriented words.	Pragmatic	0.031449
punctuation_ratio	Ratio of punctuation in the text.	Linguistic	0.000239
code_markup_score	Ratio of code or formatting tokens in the text.	Pragmatic	0.001198
logical_fallacy_score	Score of logical fallacy in the text.	Safety Risk	0.026710
hidden_instructions_score	Score of concealed directives or commands in the text.	Behavioral	0.026510
lexical_diversity_score	Score of token diversity.	Linguistic	0.005101
language_switching_score	Score of language switching frequency.	Pragmatic	0.170473
rhetorical_questions_score	Frequency of rhetorical questions.	Pragmatic	0.004203
double_negation_score	Frequency of double negation sentences.	Pragmatic	0.002321
conditional_statements_score	Frequency of conditional statements.	Syntactic	0.004179
PERSON	Ratio of named entities tagged as persons.	Metadata	0.109839
ORG	Ratio of named entities tagged as organizations.	Metadata	0.107075
GPE	Ratio of named entities tagged as geopolitical entities.	Metadata	0.113253
DATE	Ratio of named entities tagged as dates.	Metadata	0.113527
TIME	Ratio of named entities tagged as times.	Metadata	0.107849
MONEY	Ratio of named entities tagged as monetary values.	Metadata	0.108133
PERCENT	Ratio of named entities tagged as percentages.	Metadata	0.104761
LOC	Ratio of named entities tagged as locations.	Metadata	0.108968
EVENT	Ratio of named entities tagged as events.	Metadata	0.112325

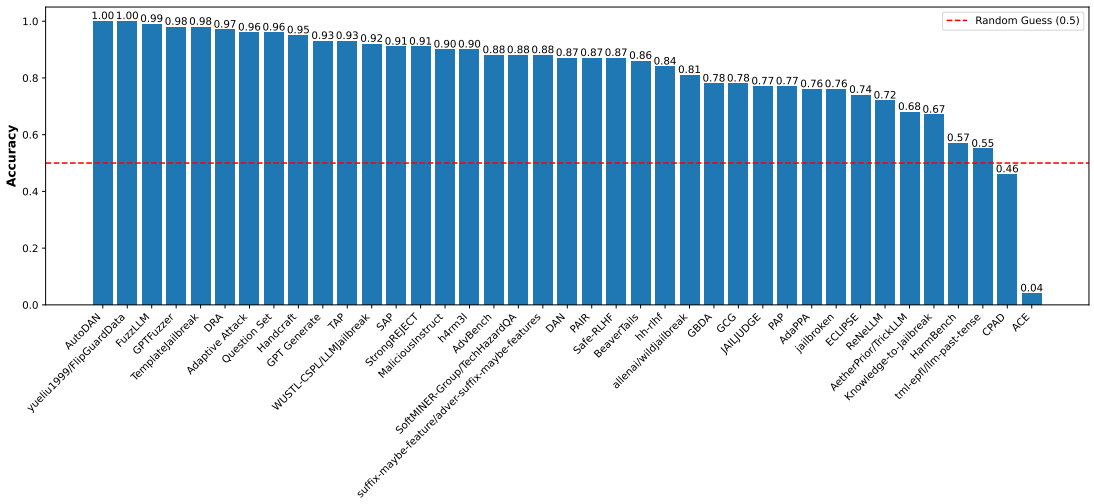


Fig. 5.4: Detection accuracy of the trained XGBoost model on each unseen type of jailbreak attack

The model's poor performance on this dataset can be attributed to the same issue as with HarmBench. CPAD is a jailbreak dataset in Chinese, whereas most of our training data consists of English jailbreak prompts, leading to a significant language mismatch. Finally, ACE employs simple encoding techniques (e.g., Base64, ROT13) on original harmful prompts. Without exposure to such encoded examples during training, XGBoost struggles to detect this jailbreak method, which are structurally distinct from other types.

5.4 PromptSecurity: A Modular Framework for LLM Security Evaluation

5.4.1 Overview and Motivation

Large Language Models (LLMs) are increasingly deployed in safety–critical settings, yet the evaluation of their security under adversarial prompts remains fragmented and often irreproducible. This chapter introduces **PromptSecurity**, a modular, configuration-driven framework for end-to-end evaluation of LLM security. The framework enables systematic cross-study of attack vectors, defensive mechanisms, and model behaviors under a unified protocol.

Problem gaps. PromptSecurity addresses three structural gaps observed in prior efforts: (i) the absence of standardized, reproducible evaluation protocols across heterogeneous attacks/defenses; (ii) the lack of unified interfaces that abstract over diverse model backends and deployment modalities; (iii) limited scalability for continuous integration of rapidly evolving methods and datasets.

Scope and current snapshot. The reference implementation supports a broad set of models (APIs and local), multiple attack families, defenses, evaluators (*judgers*), and standardized datasets with deterministic sampling. The design favors *open-world extensibility*: numbers are expected to grow; components and results are versioned.

5.4.2 Design Goals and Assumptions

PromptSecurity is guided by five principles:

- **Modularity.** Decompose evaluation into loosely coupled modules: `Attacks`, `Defenses`, `Models`, `Judgers`, `Datasets`, and `Experiments`.
- **Configuration over code.** All behaviors are externally specified via JSON/YAML, enabling version control of experiments independent of implementation changes.
- **Dynamic discovery.** Components are auto-discovered from registry metadata and directory scanning, minimizing maintenance overhead for new methods.
- **Compatibility by construction.** A validator precludes ill-posed settings (e.g., white-box attacks on API-only models, gradient defenses without weight access).
- **Deterministic reproducibility.** Fixed seeds, temperature-zero decoding (unless stated), and canonical data splits guarantee byte-level replayability.

5.4.3 System Architecture

The architecture follows a three-tier, layered design:

Component layer. Domain modules implement standardized base classes (`BaseAttack`, `BaseDefendedModel`, `BaseModel`, `BaseJudger`, `BaseDataset`). Each module declares *capabilities* and *requirements* to support automated compatibility checks.

Integration layer. Configuration loaders, registries, and a *compatibility validator* orchestrate component wiring. A *resource manager* handles API credentials, rate limiting, and local acceleration parameters (CPU/GPU, precision, batching).

Interface layer. A unified experimental API and CLI support batch execution, pipelines, and resume-on-failure with structured logging and metrics export (JSONL/Parquet).

5.4.4 Component Modules

Attack Module

Attacks are organized by access assumptions (black-box vs. white-box) and algorithmic families (e.g., template/transformation, iterative refinement, fuzzing/heuristics, gradient-based, genetic/evolutionary, RL-based). All methods inherit from `BaseAttack` and expose a common `generate(prompt, context)` interface with *cost accounting* (queries, tokens, time) for fair comparison.

Defense Module

Defenses are instantiated along three loci: *input filtering* (e.g., similarity/perplexity gates, back-translation, smoothing), *model-based defenses* (e.g., prompt hardening, gradient monitors), and *output filtering/self-evaluation*. They compose through `BaseDefendedModel`, which wraps a `BaseModel` and provides chainable pre/post hooks.

Model Module

The model abstraction unifies access to API models (e.g., commercial providers) and local models (e.g., LLaMA, Qwen, Phi). The module advertises capabilities (weights/gradients available? logprobs? tool-use?) to the validator. White-box attacks/defenses are restricted to weight-accessible backends by policy.

Judger Module

Judgers implement evaluation semantics, ranging from rule-based (e.g., refusal/policy matching) to model-based LLM judges for contextual safety and utility assessment. All judgers support batch mode, calibrated thresholds, and provenance-rich outputs (scores, rationales, uncertainties).

Dataset Module

Datasets provide standardized access to security benchmarks (e.g., harmful-content, jailbreak patterns, AI-risk scenarios). Deterministic samplers (fixed seeds, stratified subsampling) and metadata views (capability labels, language, domain) enable controlled, reproducible studies.

5.4.5 Unified Experimental Protocol

Five-element tuple. Each experiment is defined by the tuple

$$\langle M, A, D, S, J \rangle,$$

where M is the model (possibly defended), A an attack (or `no_attack`), D a defense (or `no_defense`), S the dataset split/sampler, and J the judger. This decomposition enables comparable ablations and factorial sweeps across the design space.

Metrics and outputs. The framework standardizes metrics such as jailbreak success rate, refusal rate, utility/harmlessness trade-offs, and cost (queries/tokens/latency). All results are emitted in JSONL with complete metadata: component versions, random seeds, decoding parameters, and dataset hashes.

Baselines and controls. PromptSecurity encourages paired baselines (`no_attack`, `no_defense`) and *threat-model alignment* (e.g., enforcing query budgets, language constraints) to avoid incomparable reports.

5.4.6 Configuration and Execution

Configuration management. Experiments can be specified programmatically or via JSON/YAML with schema validation. Environment overlays (e.g., containerized deployments) and CLI flags enable late-binding for credentials, rate limits, and resource placement.

Execution pipeline. The pipeline proceeds as: *(i)* schema and compatibility validation; *(ii)* dynamic instantiation and resource planning; *(iii)* defense-first construction (wrapping models before attacks are executed); *(iv)* evaluation with progress tracking, retries, and structured error handling; *(v)* standardized result emission with audit trails.

Determinism. Unless explicitly requested, decoding uses temperature 0 and fixed seeds; sampling and batching are canonicalized. A *run manifest* records environment fingerprints (library versions, GPU, CUDA, driver) to enable exact replays.

5.4.7 Scalability and Extensibility

Plug-in model. New components register via lightweight metadata (name, version, capabilities, dependencies). The registry supports semantic versioning and deprecation, allowing the codebase to evolve without breaking archived experiments.

Resource-aware execution. API backends are scheduled with concurrency windows and exponential backoff; local backends expose knobs for precision (`bf16/fp16`), tensor parallelism, and KV cache policies. The orchestrator adapts batch sizes to prevent OOM while preserving determinism.

Cost and safety guardrails. Built-in enforcement of query budgets, rate limits, and red-team safety constraints (e.g., disabling disallowed threat settings by default) reduce accidental policy violations during large-scale sweeps.

5.5 Experiments

5.5.1 Overview

Evaluating LLM security at scale is computationally intensive and methodologically non-trivial. A naive full factorial sweep over $\langle M, A, D, S, J \rangle$ (model, attack, defense,

dataset, judger; cf. §5.4.5) is infeasible. We propose a four-phase methodology that *reduces* the experimental design while preserving *coverage, rigor, and reproducibility*. The core principles are: (i) **representative sampling** under explicit coverage constraints; (ii) **progressive refinement** where earlier phases inform later ones; and (iii) **multi-criteria validation** balancing statistical reliability, cost, and generalizability.

5.5.2 Objectives, Endpoints, and Constraints

Primary endpoints. *Attack Success Rate* (ASR), refusal rate, and cost. For dataset S with items s and judger J:

$$\mathrm{ASR}(M, A, D, S, J) = \frac{1}{|S|} \sum_{s \in S} \mathbb{1}\{\mathrm{unsafe}(M, A, D, s; J)\}.$$

Secondary endpoints. Latency, token/Query cost, and stability under resampling (Section 5.5.4).

Threat-model and feasibility constraints. Compatibility rules from §5.4.3 (e.g., white-box $\Rightarrow$ local models with weight access), query budgets, and provider rate limits are enforced by the validator.

5.5.3 Phase I: Representative Model Selection

Objective. Select a subset $\mathcal{M}_\star \subset \mathcal{M}$ capturing diversity across vendor families, deployment types (API/local), and safety alignment behaviors.

Protocol.

1. **Setting:** no-attack, no-defense, complete baseline prompts.

2. **Metric:** refusal rate under majority-vote across candidate judgers:

$$R(m) = \frac{1}{|D|} \sum_{d \in D} \frac{|\{s \in d : \text{refuses}(m, s)\}|}{|d|}.$$

3. **Selection:** vendor-stratified sampling with refusal-quantile coverage; tie-break by capability vectors (API features, logprobs, tool-use) to maximize design diversity.

Outcome. A compact, behaviorally diverse set $\mathcal{M}_\star$ spanning conservative↔permissive regimes.

5.5.4 Phase II: Judger Calibration and Selection

Objective. Identify a single reliable judger $J_\star$ for large-scale runs.

Protocol.

1. Evaluate a representative grid (M, A, S) with multiple judgers $J = \{J_i\}$.

2. Compute inter-rater agreement: Krippendorff's α and pairwise Cohen's $\kappa(J_i, J_j)$; summarize

$$\text{Consistency}(J) = \frac{1}{\binom{|J|}{2}} \sum_{i<j} \kappa(J_i, J_j).$$

3. **Selection criteria:** (i) highest mean pairwise consistency; (ii) cost/latency efficiency; (iii) stability under bootstrap resampling.

Outcome. A calibrated $J_\star$ with documented variance and cost per decision.

5.5.5 Phase III: Dataset Optimization

Objective. Build an optimized evaluation set $S_\star$ with strong discriminative power while retaining coverage across content categories and attack types.

Method. Refusal-guided, stratified sampling with cross-model response patterns:

1. Estimate item discriminativeness using cross-model variance of outcomes and consistency under $J_\star$.

2. Apply stratified quotas over source datasets, categories (e.g., capability/risk taxonomy), and languages.

3. Select items maximizing discriminative utility subject to coverage and budget constraints; verify with held-out stability checks.

Validation. Category coverage audits, stability under subsampling, and leakage checks (no prompt overlap with any model/provider safety training disclosures when known).

Outcome. $S_\star$ with balanced provenance and high information yield per query.

5.5.6 Phase IV: Comprehensive Attack–Defense Evaluation

Objective. Execute the reduced but comprehensive matrix $\mathcal{M}_\star \times \mathcal{A} \times \mathcal{D} \times S_\star$ with $J_\star$.

Design.

- **Attacks:** black-box (prompt engineering, multi-turn, transformations, encoding) for all models; white-box (gradient/evolutionary) on local models only; include `no_attack`.

- **Defenses:** input/output filtering and model-based defenses; include `no_defense`.
- **Execution:** parallel batched runs with deterministic seeds, resume-on-failure, and cost accounting (queries/tokens/time) recorded per $\langle M, A, D \rangle$.

Reporting. For each cell, report ASR, R, cost, and latency with 95% CIs (nonparametric bootstrap). Provide ablations (e.g., `no_attack`, `no_defense`) and compatibility-aware fairness (equalized query budgets).

5.5.7 Statistical Analysis and Validity

Models. Mixed-effects logistic regression for binary outcomes with random intercepts for item and model; cluster-robust SEs across items. **Multiple comparisons.** Benjamini–Hochberg FDR control over family-wise hypotheses. **Sensitivity.** Re-run key comparisons under alternative decoders (e.g., temperature $\in \{0, 0.2\}$) and alternative label aggregations (majority vs. calibrated threshold). **Stability.** Report coefficient ranges over bootstrap replicates and across random seeds.

5.5.8 Scalability, Cost, and Reproducibility

Budgeting. Pre-commit query budgets per provider; enforce via the runtime validator. **Caching.** Deterministic caching keyed by $(\text{prompt}, \text{params}, \text{model\_version})$ to eliminate duplicate queries. **Artifacts.** Emit JSONL/Parquet with complete manifests (component versions, seeds, decoding params, dataset hashes, hardware fingerprints). Release configuration bundles for all reported tables/figures.

5.5.9 Expected Findings and Impact

Scientific contributions. Standardized, budget-aware methodology enabling cross-study comparability; representative yet tractable evaluation; calibrated, cost-effective judging at scale. **Outcomes.** (i) Baselines over representative model families; (ii) defense rankings under consistent threat models; (iii) vulnerability patterns (model-specific vs. universal); (iv) guidance on attack–defense cost–benefit trade-offs.

Chapter 6

Conclusions

In this dissertation, we have undertaken a comprehensive study of certifying trustworthy machine learning, motivated by the urgent need for robust and secure artificial intelligence systems in safety-critical and adversarial environments. As machine learning models continue to permeate high-impact domains, their vulnerabilities to adversarial attacks and the lack of systematic robustness guarantees present fundamental barriers to their reliable and ethical deployment. Through a synthesis of theoretical advances, algorithmic innovations, and empirical validation, this thesis has contributed new frameworks, methodologies, and insights that substantially advance the state of the art in certified robustness and adversarial security.

At the core of this work lies the development of *Universally Approximated Certified Robustness* (UniCR), a novel and unified framework for certifying the robustness of arbitrary classifiers under any ℓ_p-norm bounded adversarial perturbations, leveraging any continuous noise distribution. UniCR not only automates the certification process, but also establishes tight and, in many settings, optimal robustness guarantees without recourse to case-specific theoretical derivations. Our theoretical analysis

demonstrates that UniCR subsumes and extends prior randomized smoothing methods, providing a principled, analysis-free approach that supports a diverse spectrum of classifiers, noise models, and threat scenarios. Empirical results on benchmark datasets, including CIFAR-10, CIFAR-100, and ImageNet, demonstrate the scalability and broad applicability of UniCR, achieving strong certified accuracy under a variety of practical conditions.

Building upon this universal foundation, we have further proposed the *Universally Amplifying Randomized Smoothing* (UCAN) framework, which introduces the use of anisotropic noise to significantly amplify certified robustness. UCAN systematically generalizes existing randomized smoothing techniques, breaking the limitations of isotropic noise and allowing for optimizable, data-adaptive certification strategies. Theoretical analysis and experimental validation reveal that UCAN can yield substantial improvements in certified accuracy and certified radii across different datasets and adversarial threat models. This advance not only enhances the practical viability of certified defenses, but also deepens our understanding of the fundamental limits and potentials of randomized smoothing in high-dimensional settings.

Recognizing that robustness certification alone is not sufficient, this thesis also investigates the offensive side of the adversarial landscape. We have introduced the first paradigm of *certifiable black-box attacks* with randomized adversarial examples, enabling the construction of attacks with provable guarantees on attack success probability. Unlike traditional empirical black-box attacks, our approach provides a rigorous

characterization of the attack's efficacy before querying the target model, and reveals critical gaps in existing defense mechanisms. Theoretical results, complemented by extensive empirical evaluations, demonstrate the capability of certifiable black-box attacks to expose vulnerabilities even in state-of-the-art certified models, highlighting the need for more resilient and adaptive defense strategies.

In parallel, this work has systematized the rapidly emerging field of security in large language models (LLMs). With the growing adoption of LLMs in real-world applications, new security risks such as prompt injection, jailbreaks, and misuse have become increasingly salient. This dissertation presents the first comprehensive systematization of knowledge for LLM prompt security, introducing unified multi-level taxonomies for attacks, defenses, and vulnerabilities, as well as formal threat models and robust evaluation frameworks. The open-source resources and curated benchmarks developed as part of this research—including the JAILBREAKDB dataset and evaluation tools—provide the community with much-needed infrastructure for reproducible and systematic security assessment in LLMs.

Collectively, these contributions offer both a unified theoretical framework and a practical toolkit for certifying, evaluating, and improving the trustworthiness of machine learning systems. They bridge significant gaps between theory and practice, uncover the inherent trade-offs and limitations of current approaches, and lay the groundwork for future research in robust and secure artificial intelligence.

Looking ahead, several promising avenues remain open for exploration. First, the

extension of universal certification methods to structured, multi-modal, or sequential data—such as graphs, time series, or multi-modal fusion—represents a rich direction for further study. Second, as adversarial threats continue to evolve, the development of dynamic, adaptive, and context-aware certification techniques will be crucial for maintaining robustness in ever-changing environments [416]. Third, scaling up certifiable defenses and attack evaluation frameworks to support ultra-large-scale models and datasets, particularly in the realm of generative AI [417], remains an important and challenging task. Finally, as AI systems become ever more intertwined with societal processes, addressing questions of fairness [418], privacy [419, 420], and explainability [421] in the context of robust machine learning will require interdisciplinary approaches that draw upon insights from computer science, statistics, law, and ethics.

In conclusion, this dissertation aspires to advance both the theoretical understanding and practical deployment of trustworthy machine learning. By providing general, automated, and empirically validated certification frameworks; pioneering new paradigms for attack evaluation; and systematizing the security of emerging AI technologies, this work seeks to support the safe, ethical, and robust adoption of AI systems in service of society. It is my hope that the results and ideas presented herein will inspire and enable further advances in the ongoing quest for trustworthy artificial intelligence.

Bibliography

[1] Jeremy Cohen, Elan Rosenfeld, and Zico Kolter. Certified adversarial robustness via randomized smoothing. In *International Conference on Machine Learning*, 2019.

[2] Jiaye Teng, Guang-He Lee, and Yang Yuan. *ell*_1 adversarial robustness certificates: a randomized smoothing approach. 2019.

[3] Greg Yang, Tony Duan, J Edward Hu, Hadi Salman, Ilya Razenshteyn, and Jerry Li. Randomized smoothing of all shapes and sizes. In *International Conference on Machine Learning*, pages 10693–10705. PMLR, 2020.

[4] Hanbin Hong, Binghui Wang, and Yuan Hong. Unicr: Universally approximated certified robustness via randomized smoothing. In *ECCV*, 2022.

[5] Hanbin Hong and Yuan Hong. Certified adversarial robustness via anisotropic randomized smoothing. *CoRR*, abs/2207.05327, 2022.

[6] Hanbin Hong, Xinyu Zhang, Binghui Wang, Zhongjie Ba, and Yuan Hong. Certifiable black-box attacks with randomized adversarial examples: Breaking defenses with provable confidence. In *Proceedings of ACM SIGSAC Conference on Computer and Communications Security*, pages 600–614. ACM, 2024.

[7] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations*, 2018.

[8] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *2017 ieee symposium on security and privacy (sp)*, pages 39–57. IEEE, 2017.

[9] Pin-Yu Chen, Huan Zhang, Yash Sharma, Jinfeng Yi, and Cho-Jui Hsieh. Zoo: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models. In *10th ACM workshop on artificial intelligence and security*, 2017.

[10] Jianbo Chen, Michael I Jordan, and Martin J Wainwright. Hopskipjumpattack: A query-efficient decision-based attack. In *2020 IEEE Symposium on Security and Privacy*, 2020.

[11] Hanbin Hong, Xinyu Zhang, Binghui Wang, Zhongjie Ba, and Yuan Hong. Certifiable black-box attacks with randomized adversarial examples: Breaking defenses with provable confidence, 2024.

[12] Eric Wong and J Zico Kolter. Provable defenses against adversarial examples via the convex outer adversarial polytope. In *ICML*, 2018.

[13] Guy Katz, Clark Barrett, David L Dill, Kyle Julian, and Mykel J Kochenderfer. Reluplex: An efficient smt solver for verifying deep neural networks. In *International Conference on Computer Aided Verification*, pages 97–117. Springer, 2017.

[14] Nicholas Carlini, Guy Katz, Clark Barrett, and David L Dill. Provably minimally-distorted adversarial examples. *arXiv preprint arXiv:1709.10207*, 2017.

[15] Matteo Fischetti and Jason Jo. Deep neural networks and mixed integer linear optimization. *Constraints*, 23(3):296–309, 2018.

[16] Henry Gouk, Eibe Frank, Bernhard Pfahringer, and Michael J Cree. Regularisation of neural networks by enforcing lipschitz continuity. *Machine Learning*, 110(2):393–416, 2021.

[17] Aditi Raghunathan, Jacob Steinhardt, and Percy Liang. Certified defenses against adversarial examples. *arXiv preprint arXiv:1801.09344*, 2018.

[18] Francesco Croce and Matthias Hein. Provable robustness against all adversarial l_p-perturbations for $p\geq 1$. In *ICLR*. OpenReview.net, 2020.

[19] Matthew Mirman, Timon Gehr, and Martin Vechev. Differentiable abstract interpretation for provably robust neural networks. In *International Conference on Machine Learning*, 2018.

[20] Jieren Deng, Hanbin Hong, Aaron Palmer, Xin Zhou, Jinbo Bi, Kaleel Mahmood, Yuan Hong, and Derek Aguiar. Certifying adapters: Enabling and enhancing the certification of classifier adversarial robustness. *CoRR*, abs/2405.16036, 2024.

[21] Bai Li, Changyou Chen, Wenlin Wang, and Lawrence Carin. Second-order adversarial attack and certifiable robustness. *arXiv preprint arXiv:2006.00731*, 2020.

[22] Mathias Lecuyer, Vaggelis Atlidakis, Roxana Geambasu, Daniel Hsu, and Suman Jana. Certified robustness to adversarial examples with differential privacy. In *2019 IEEE Symposium on Security and Privacy (SP)*, pages 656–672. IEEE, 2019.

[23] Dinghuai Zhang, Mao Ye, Chengyue Gong, Zhanxing Zhu, and Qiang Liu. Black-box certification with randomized smoothing: A functional optimization based framework. 2020.

[24] Jiaye Teng, Guang-He Lee, and Yang Yuan. ℓ_1 adversarial robustness certificates: a randomized smoothing approach, 2020.

[25] Krishnamurthy (Dj) Dvijotham, Jamie Hayes, Borja Balle, J. Zico Kolter, Chongli Qin, András György, Kai Xiao, Sven Gowal, and Pushmeet Kohli. A framework for robustness certification of smoothed classifiers using f-divergences. In *ICLR*, 2020.

[26] James Kennedy and Russell Eberhart. Particle swarm optimization. In *Proceedings of ICNN'95-international conference on neural networks*. IEEE, 1995.

[27] Yann LeCun, Corinna Cortes, and CJ Burges. Mnist handwritten digit database. *ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist*, 2, 2010.

[28] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[29] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015.

[30] Kenneth T Co, Luis Muñoz-González, Sixte de Maupeou, and Emil C Lupu. Procedural noise adversarial examples for black-box attacks on deep convolutional networks. In *ACM SIGSAC conference on computer and communications security*, 2019.

[31] Francesco Croce and Matthias Hein. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In *International conference on machine learning*, pages 2206–2216. PMLR, 2020.

[32] Eric Wong, Frank Schmidt, and Zico Kolter. Wasserstein adversarial examples via projected sinkhorn iterations. In *International Conference on Machine Learning*, 2019.

[33] Maksym Andriushchenko, Francesco Croce, Nicolas Flammarion, and Matthias Hein. Square attack: a query-efficient black-box adversarial attack via random search. In *European Conference on Computer Vision*, pages 484–501. Springer, 2020.

[34] Cynthia Dwork and Aaron Roth. The algorithmic foundations of differential privacy. *Found. Trends Theor. Comput. Sci.*, 9(3-4):211–407, 2014.

[35] Kate Keahey, Jason Anderson, Zhuo Zhen, Pierre Riteau, Paul Ruth, Dan Stanzione, Mert Cevik, Jacob Colleran, Haryadi S. Gunawi, Cody Hammock, Joe Mambretti, Alexander Barnes, François Halbach, Alex Rocha, and Joe Stubbs. Lessons learned from the chameleon testbed. In *Proceedings of the 2020 USENIX Annual Technical Conference (USENIX ATC '20)*. USENIX Association, July 2020.

[36] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[37] Karsten Scheibler, Leonore Winterer, Ralf Wimmer, and Bernd Becker. Towards verification of artificial neural networks. In *MBMV*, pages 30–40, 2015.

[38] Ruediger Ehlers. Formal verification of piece-wise linear feed-forward neural networks. In *International Symposium on Automated Technology for Verification and Analysis*, pages 269–286. Springer, 2017.

[39] Chih-Hong Cheng, Georg Nührenberg, and Harald Ruess. Maximum resilience of artificial neural networks. In *International Symposium on Automated Technology for Verification and Analysis*, pages 251–268. Springer, 2017.

[40] Rudy R Bunel, Ilker Turkaslan, Philip HS Torr, Pushmeet Kohli, and Pawan Kumar Mudigonda. A unified view of piecewise linear neural network verification. In *NeurIPS*, 2018.

[41] Eric Wong, Frank R Schmidt, Jan Hendrik Metzen, and J Zico Kolter. Scaling provable adversarial defenses. *arXiv preprint arXiv:1805.12514*, 2018.

[42] Aditi Raghunathan, Jacob Steinhardt, and Percy S Liang. Semidefinite relaxations for certifying robustness to adversarial examples. In *NeurIPS*, 2018.

[43] Krishnamurthy Dvijotham, Sven Gowal, Robert Stanforth, and et al. Training verified learners with learned verifiers. *arXiv*, 2018.

[44] Krishnamurthy Dvijotham, Robert Stanforth, Sven Gowal, Timothy A Mann, and Pushmeet Kohli. A dual approach to scalable verification of deep networks. In *UAI*, 2018.

[45] Yusuke Tsuzuku, Issei Sato, and Masashi Sugiyama. Lipschitz-margin training: Scalable certification of perturbation invariance for deep neural networks. In *NeurIPS*, 2018.

[46] Cem Anil, James Lucas, and Roger Grosse. Sorting out lipschitz function approximation. In *International Conference on Machine Learning*, pages 291–301. PMLR, 2019.

[47] Moustapha Cissé, Piotr Bojanowski, Edouard Grave, Yann N. Dauphin, and Nicolas Usunier. Parseval networks: Improving robustness to adversarial examples. In *Proceedings of the 34th International Conference on Machine Learning*, 2017.

[48] Matthias Hein and Maksym Andriushchenko. Formal guarantees on the robustness of a classifier against adversarial manipulation. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 2266–2276, 2017.

[49] Timon Gehr, Matthew Mirman, Dana Drachsler-Cohen, Petar Tsankov, Swarat Chaudhuri, and Martin Vechev. Ai2: Safety and robustness certification of neural networks with abstract interpretation. In *IEEE S & P*, 2018.

[50] Gagandeep Singh, Timon Gehr, Matthew Mirman, Markus Püschel, and Martin Vechev. Fast and effective robustness certification. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pages 10825–10836, 2018.

[51] Sven Gowal, Krishnamurthy Dvijotham, Robert Stanforth, Rudy Bunel, Chongli Qin, Jonathan Uesato, Relja Arandjelovic, Timothy A. Mann, and Pushmeet Kohli. On the effectiveness of interval bound propagation for training verifiably robust models. *CoRR*, abs/1810.12715, 2018.

[52] Tsui-Wei Weng, Huan Zhang, Hongge Chen, Zhao Song, Cho-Jui Hsieh, Luca Daniel, Duane S. Boning, and Inderjit S. Dhillon. Towards fast computation of certified robustness for relu networks. In Jennifer G. Dy and Andreas Krause, editors, *International Conference on Machine Learning*, 2018.

[53] Huan Zhang, Tsui-Wei Weng, Pin-Yu Chen, Cho-Jui Hsieh, and Luca Daniel. Efficient neural network robustness certification with general activation functions. In *Neural Information Processing Systems*, 2018.

[54] Guang-He Lee, Yang Yuan, Shiyu Chang, and Tommi S. Jaakkola. Tight certificates of adversarial robustness for randomly smoothed classifiers. In *NeurIPS*, pages 4911–4922, 2019.

[55] G Wul. Zur frage der geschwindigkeit des wachstums und der auflosung der kristall achen. *Z. Kristallogr*, 34:449–530, 1901.

[56] Ian Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations*, 2015.

[57] Jiachen Sun, Yulong Cao, Qi Alfred Chen, and Z Morley Mao. Towards robust lidar-based perception in autonomous driving: General black-box adversarial sensor attack and countermeasures. In *USENIX Security*, 2020.

[58] Xingjun Ma, Yuhao Niu, Lin Gu, Yisen Wang, Yitian Zhao, James Bailey, and Feng Lu. Understanding adversarial attacks on deep learning based medical image analysis systems. *Pattern Recognition*, 2021.

[59] Yinpeng Dong, Hang Su, Baoyuan Wu, Zhifeng Li, Wei Liu, Tong Zhang, and Jun Zhu. Efficient decision-based black-box adversarial attacks on face recognition. In *CVPR*, 2019.

[60] Ali Shafahi, Mahyar Najibi, Amin Ghiasi, Zheng Xu, John Dickerson, Christoph Studer, Larry S Davis, Gavin Taylor, and Tom Goldstein. Adversarial training for free! In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, pages 3358–3369, 2019.

[61] Weilin Xu, David Evans, and Yanjun Qi. Feature squeezing: Detecting adversarial examples in deep neural networks. *arXiv preprint arXiv:1704.01155*, 2017.

[62] Cihang Xie, Jianyu Wang, Zhishuai Zhang, Zhou Ren, and Alan Yuille. Mitigating adversarial effects through randomization. In *International Conference on Learning Representations*, 2018.

[63] Cihang Xie, Yuxin Wu, Laurens van der Maaten, Alan L Yuille, and Kaiming He. Feature denoising for improving adversarial robustness. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 501–509, 2019.

[64] Shuo Yang, Tianyu Guo, Yunhe Wang, and Chang Xu. Adversarial robustness through disentangled representations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 3145–3153, 2021.

[65] Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *International conference on machine learning*, pages 274–283. PMLR, 2018.

[66] Cihang Xie, Zhishuai Zhang, Yuyin Zhou, Song Bai, Jianyu Wang, Zhou Ren, and Alan L Yuille. Improving transferability of adversarial examples with input diversity. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2730–2739, 2019.

[67] Kimin Lee, Kibok Lee, Honglak Lee, and Jinwoo Shin. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. *NeurIPS*, 2018.

[68] Francisco Eiras, Motasem Alfarra, Philip Torr, M. Pawan Kumar, Puneet K. Dokania, Bernard Ghanem, and Adel Bibi. ANCER: Anisotropic certification via sample-wise volume maximization. *TMLR*, 2022.

[69] Leilani H Gilpin, David Bau, Ben Z Yuan, Ayesha Bajwa, Michael Specter, and Lalana Kagal. Explaining explanations: An overview of interpretability of machine learning. In *DSAA*, 2018.

[70] Antonia Creswell, Tom White, Vincent Dumoulin, Kai Arulkumaran, Biswa Sengupta, and Anil A Bharath. Generative adversarial networks: An overview. *IEEE signal processing magazine*, 2018.

[71] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 2020.

[72] Yulun Zhang, Yapeng Tian, Yu Kong, Bineng Zhong, and Yun Fu. Residual dense network for image super-resolution. In *CVPR*, 2018.

[73] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *CVPR*, 2017.

[74] Bing Xu, Naiyan Wang, Tianqi Chen, and Mu Li. Empirical evaluation of rectified activations in convolutional network. *arXiv preprint arXiv:1505.00853*, 2015.

[75] Taras Rumezhak, Francisco Girbal Eiras, Philip HS Torr, and Adel Bibi. Rancer: Non-axis aligned anisotropic certification with randomized smoothing. In *WACV*, 2023.

[76] Motasem Alfarra, Adel Bibi, Philip HS Torr, and Bernard Ghanem. Data dependent randomized smoothing. *arXiv preprint arXiv:2012.04351*, 2020.

[77] Peter Súkeník, Aleksei Kuvshinov, and Stephan Günnemann. Intriguing properties of input-dependent randomized smoothing. *arXiv preprint arXiv:2110.05365*, 2021.

[78] Lei Wang, Runtian Zhai, Di He, Liwei Wang, and Li Jian. Pretrain-to-finetune adversarial training via sample-wise randomized smoothing. 2020.

[79] Runtian Zhai, Chen Dan, Di He, Huan Zhang, Boqing Gong, Pradeep Ravikumar, Cho-Jui Hsieh, and Liwei Wang. Macer: Attack-free and scalable robust training via maximizing certified radius. *arXiv preprint arXiv:2001.02378*, 2020.

[80] Jongheon Jeong, Sejun Park, Minkyu Kim, Heung-Chang Lee, Do-Guk Kim, and Jinwoo Shin. Smoothmix: Training confidence-calibrated smoothed classifiers for certified robustness. *NeurIPS*, 2021.

[81] Zhuolin Yang, Linyi Li, Xiaojun Xu, Bhavya Kailkhura, Tao Xie, and Bo Li. On the certified robustness for ensemble models and beyond. *arXiv preprint arXiv:2107.10873*, 2021.

[82] Hadi Salman, Jerry Li, Ilya Razenshteyn, Pengchuan Zhang, Huan Zhang, Sebastien Bubeck, and Greg Yang. Provably robust deep learning via adversarially trained smoothed classifiers. *NeurIPS*, 2019.

[83] Jongheon Jeong and Jinwoo Shin. Consistency regularization for certified robustness of smoothed classifiers. *NeurIPS*, 2020.

[84] Miklós Z Horváth, Mark Niklas Müller, Marc Fischer, and Martin Vechev. Boosting randomized smoothing with variance reduced classifiers. *arXiv preprint arXiv:2106.06946*, 2021.

[85] Nicholas Carlini, Florian Tramer, Krishnamurthy Dj Dvijotham, Leslie Rice, Mingjie Sun, and J Zico Kolter. (certified!!) adversarial robustness for free! *ICLR*, 2023.

[86] Jiawei Zhang, Zhongzhu Chen, Huan Zhang, Chaowei Xiao, and Bo Li. {DiffSmooth}: Certifiably robust learning via diffusion models and local smoothing. In *USENIX Security*, 2023.

[87] Xiaowei Huang, Marta Kwiatkowska, Sen Wang, and Min Wu. Safety verification of deep neural networks. In *International conference on computer aided verification*, pages 3–29. Springer, 2017.

[88] Alessio Lomuscio and Lalit Maganti. An approach to reachability analysis for feed-forward relu neural networks. *arXiv preprint arXiv:1706.07351*, 2017.

[89] Cynthia Dwork. Differential privacy. In *ICALP*, 2006.

[90] Cynthia Dwork. Differential privacy: A survey of results. In *TAMC*, 2008.

[91] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations*, 2018.

[92] Nicholas Carlini and David A. Wagner. Towards evaluating the robustness of neural networks. In *IEEE Symposium on Security and Privacy*, 2017.

[93] Jianbo Chen, Michael I. Jordan, and Martin J. Wainwright. Hopskipjumpattack: A query-efficient decision-based attack. In *IEEE Symposium on Security and Privacy*, pages 1277–1294. IEEE, 2020.

[94] Shangyu Xie, Han Wang, Yu Kong, and Yuan Hong. Universal 3-dimensional perturbations for black-box attacks on video recognition systems. In *In Proceedings of the 43rd IEEE Symposium on Security and Privacy (Oakland'22)*, 2022.

[95] Alexey Kurakin, Ian J. Goodfellow, and Samy Bengio. Adversarial machine learning at scale. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017.

[96] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, Omar Fawzi, and Pascal Frossard. Universal adversarial perturbations. *CoRR*, abs/1610.08401, 2016.

[97] Pin-Yu Chen, Huan Zhang, Yash Sharma, Jinfeng Yi, and Cho-Jui Hsieh. ZOO: zeroth order optimization based black-box attacks to deep neural networks without training substitute models. In *AISec@CCS*, 2017.

[98] Andrew Ilyas, Logan Engstrom, Anish Athalye, and Jessy Lin. Black-box adversarial attacks with limited queries and information. In *ICML*, 2018.

[99] Maksym Andriushchenko, Francesco Croce, Nicolas Flammarion, and Matthias Hein. Square attack: A query-efficient black-box adversarial attack via random search. In *ECCV*, 2020.

[100] Shangyu Xie, Yan Yan, and Yuan Hong. Stealthy 3d poisoning attack on video recognition models. *IEEE Trans. Dependable Secur. Comput.*, 2023.

[101] Wieland Brendel, Jonas Rauber, and Matthias Bethge. Decision-based adversarial attacks: Reliable attacks against black-box machine learning models. In *ICLR*. OpenReview.net, 2018.

[102] Shenao Yan, Shen Wang, Yue Duan, Hanbin Hong, Kiho Lee, Doowon Kim, and Yuan Hong. An llm-assisted easy-to-trigger backdoor attack on code completion models: Injecting disguised vulnerabilities against strong detection. In *USENIX Security*, 2024.

[103] Nicolas Papernot, Patrick D. McDaniel, Ian J. Goodfellow, Somesh Jha, Z. Berkay Celik, and Ananthram Swami. Practical black-box attacks against machine learning. In Ramesh Karri, Ozgur Sinanoglu, Ahmad-Reza Sadeghi, and Xun Yi, editors, *AsiaCCS 2017*.

[104] Minhao Cheng, Thong Le, Pin-Yu Chen, Huan Zhang, Jinfeng Yi, and Cho-Jui Hsieh. Query-efficient hard-label black-box attack: An optimization-based approach. In *ICLR*. OpenReview.net, 2019.

[105] Arjun Nitin Bhagoji, Warren He, Bo Li, and Dawn Song. Practical black-box attacks on deep neural networks using efficient query mechanisms. In *ECCV*. Springer, 2018.

[106] Yali Du, Meng Fang, Jinfeng Yi, Jun Cheng, and Dacheng Tao. Towards query efficient black-box attacks: An input-free perspective. In *ACM CCS*, 2018.

[107] Yucheng Shi, Siyu Wang, and Yahong Han. Curls & whey: Boosting black-box adversarial attacks. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019*, pages 6519–6527, 2019.

[108] Yinpeng Dong, Tianyu Pang, Hang Su, and Jun Zhu. Evading defenses to transferable adversarial examples by translation-invariant attacks. In *CVPR*, 2019.

[109] Muzammal Naseer, Salman H. Khan, Munawar Hayat, Fahad Shahbaz Khan, and Fatih Porikli. A self-supervised approach for adversarial robustness. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020*, pages 259–268. Computer Vision Foundation / IEEE, 2020.

[110] Thomas Brunner, Frederik Diehl, Michael Truong-Le, and Alois C. Knoll. Guessing smart: Biased sampling for efficient black-box adversarial attacks. In *ICCV*, pages 4957–4965. IEEE, 2019.

[111] Yandong Li, Lijun Li, Liqiang Wang, Tong Zhang, and Boqing Gong. NATTACK: learning the distributions of adversarial examples for an improved black-box attack on deep neural networks. In *ICML*, 2019.

[112] Chuan Guo, Jacob R. Gardner, Yurong You, Andrew Gordon Wilson, and Kilian Q. Weinberger. Simple black-box adversarial attacks. In *ICML 2019*.

[113] Huiying Li, Shawn Shan, Emily Wenger, Jiayun Zhang, Haitao Zheng, and Ben Y. Zhao. Blacklight: Scalable defense for neural networks against query-based black-box attacks. In *USENIX Security*, 2022.

[114] Zeyu Qin, Yanbo Fan, Hongyuan Zha, and Baoyuan Wu. Random noise defense against query-based black-box attacks. pages 7650–7663, 2021.

[115] Varun Chandrasekaran, Kamalika Chaudhuri, Irene Giacomelli, Somesh Jha, and Songbai Yan. Exploring connections between active learning and model extraction. In *USENIX Security*. USENIX Association, 2020.

[116] Sizhe Chen, Zhehao Huang, Qinghua Tao, Yingwen Wu, Cihang Xie, and Xiaolin Huang. Adversarial attack on attackers: Post-process to mitigate black-box score-based query attacks. *NeurIPS*, 2022.

[117] Zhezhi He, Adnan Siraj Rakin, and Deliang Fan. Parametric noise injection: Trainable randomness to improve deep neural network robustness against adversarial attack. In *CVPR*, 2019.

[118] Xuanqing Liu, Minhao Cheng, Huan Zhang, and Cho-Jui Hsieh. Towards robust neural networks via random self-ensemble. In *Proceedings of the european conference on computer vision (ECCV)*, pages 369–385, 2018.

[119] Andrew Ilyas, Logan Engstrom, and Aleksander Madry. Prior convictions: Black-box adversarial attacks with bandits and priors. In *International Conference on Learning Representations*, 2019.

[120] Seungyong Moon, Gaon An, and Hyun Oh Song. Parsimonious black-box adversarial attacks via efficient combinatorial optimization. In *ICML*, 2019.

[121] Abdullah Al-Dujaili and Una-May O'Reilly. Sign bits are all you need for black-box attacks. In *International Conference on Learning Representations*, 2020.

[122] Sijia Liu, Pin-Yu Chen, Xiangyi Chen, and Mingyi Hong. signsgd via zeroth-order oracle. In *International Conference on Learning Representations*, 2019.

[123] Ali Rahmati, Seyed-Mohsen Moosavi-Dezfooli, Pascal Frossard, and Huaiyu Dai. Geoda: a geometric framework for black-box adversarial attacks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020.

[124] Jinghui Chen and Quanquan Gu. Rays: A ray searching method for hard-label adversarial attack. In *KDD*, 2020.

[125] Weilun Chen, Zhaoxiang Zhang, Xiaolin Hu, and Baoyuan Wu. Boosting decision-based black-box adversarial attacks with random sign flip. In *ECCV*, 2020.

[126] Minhao Cheng, Simranjit Singh, Patrick H. Chen, Pin-Yu Chen, Sijia Liu, and Cho-Jui Hsieh. Sign-opt: A query-efficient hard-label adversarial attack. In *ICLR*. OpenReview.net, 2020.

[127] Lukas Schott, Jonas Rauber, Matthias Bethge, and Wieland Brendel. Towards the first adversarially robust neural network model on MNIST. In *ICLR*, 2019.

[128] Viet Quoc Vo, Ehsan Abbasnejad, and Damith Ranasinghe. Query efficient decision based sparse attacks against black-box deep learning models. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022.

[129] Jeremy M. Cohen, Elan Rosenfeld, and J. Zico Kolter. Certified adversarial robustness via randomized smoothing. In *ICML*, 2019.

[130] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

[131] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015.

[132] Saining Xie, Ross B. Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 5987–5995. IEEE Computer Society, 2017.

[133] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. In Richard C. Wilson, Edwin R. Hancock, and William A. P. Smith, editors, *Proceedings of the British Machine Vision Conference*, 2016.

[134] Matěj Korvas, Ondřej Plátek, Ondřej Dušek, Lukáš Žilka, and Filip Jurčíček. Free English and Czech telephone speech corpus shared under the CC-BY-SA 3.0 license. In *(LREC 2014)*, 2014.

[135] Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric Xing, Laurent El Ghaoui, and Michael Jordan. Theoretically principled trade-off between robustness and accuracy. In *International conference on machine learning*, 2019.

[136] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

[137] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention*, 2015.

[138] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

[139] Eric Wong, Frank R. Schmidt, and J. Zico Kolter. Wasserstein adversarial examples via projected sinkhorn iterations. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning, ICML 2019*, volume 97 of *Proceedings of Machine Learning Research*, pages 6808–6817. PMLR, 2019.

[140] Andrew Ilyas, Logan Engstrom, Anish Athalye, and Jessy Lin. Query-efficient black-box adversarial examples. 2017.

[141] Siddhant Bhambri, Sumanyu Muku, Avinash Tulasi, and Arun Balaji Buduru. A survey of black-box adversarial attacks on computer vision models. *arXiv preprint arXiv:1912.01667*, 2019.

[142] Jingyu Sun, Bingyu Liu, and Yuan Hong. Logbug: Generating adversarial system logs in real time. In *CIKM*, pages 2229–2232. ACM, 2020.

[143] Wenjie Qu, Youqi Li, and Binghui Wang. A certified radius-guided attack framework to image segmentation models. In *IEEE European Symposium on Security and Privacy*, 2023.

[144] Nicolas Papernot, Patrick D. McDaniel, Xi Wu, Somesh Jha, and Ananthram Swami. Distillation as a defense to adversarial perturbations against deep neural networks. In *IEEE Symposium on Security and Privacy, SP 2016*, pages 582–597. IEEE Computer Society, 2016.

[145] Cihang Xie, Jianyu Wang, Zhishuai Zhang, Zhou Ren, and Alan L. Yuille. Mitigating adversarial effects through randomization. In *6th International Conference on Learning Representations, ICLR 2018*. OpenReview.net, 2018.

[146] Guneet S. Dhillon, Kamyar Azizzadenesheli, Zachary C. Lipton, Jeremy Bernstein, Jean Kossaifi, Aran Khanna, and Animashree Anandkumar. Stochastic activation pruning for robust adversarial defense. In *ICLR*, 2018.

[147] Chuan Guo, Mayank Rana, Moustapha Cissé, and Laurens van der Maaten. Countering adversarial images using input transformations. In *ICLR*, 2018.

[148] Jacob Buckman, Aurko Roy, Colin Raffel, and Ian J. Goodfellow. Thermometer encoding: One hot way to resist adversarial examples. In *ICLR*, 2018.

[149] Pouya Samangouei, Maya Kabkab, and Rama Chellappa. Defense-gan: Protecting classifiers against adversarial attacks using generative models. In *6th International Conference on Learning Representations, ICLR 2018*, 2018.

[150] Fangzhou Liao, Ming Liang, Yinpeng Dong, Tianyu Pang, Xiaolin Hu, and Jun Zhu. Defense against adversarial attacks using high-level representation guided denoiser. In *CVPR*, 2018.

[151] Yang Song, Taesup Kim, Sebastian Nowozin, Stefano Ermon, and Nate Kushman. Pixeldefend: Leveraging generative models to understand and defend against adversarial examples. In *6th International Conference on Learning Representations, ICLR 2018*. OpenReview.net, 2018.

[152] Hanbin Hong, Yuan Hong, and Yu Kong. An eye for an eye: Defending against gradient-based attacks with gradients. *arXiv preprint arXiv:2202.01117*, 2022.

[153] Kevin Roth, Yannic Kilcher, and Thomas Hofmann. The odds are odd: A statistical test for detecting adversarial examples. In *ICML*, 2019.

[154] Florian Tramèr. Detecting adversarial examples is (nearly) as hard as classifying them. In *ICML 2022*.

[155] Jiajun Lu, Theerasit Issaranon, and David A. Forsyth. Safetynet: Detecting and rejecting adversarial examples robustly. In *IEEE International Conference on Computer Vision, ICCV 2017*, pages 446–454. IEEE Computer Society, 2017.

[156] Dongyu Meng and Hao Chen. Magnet: A two-pronged defense against adversarial examples. In Bhavani Thuraisingham, David Evans, Tal Malkin, and Dongyan Xu, editors, *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, CCS 2017*, pages 135–147. ACM, 2017.

[157] Shubham Jain, Ana-Maria Cretu, and Yves-Alexandre de Montjoye. Adversarial detection avoidance attacks: Evaluating the robustness of perceptual hashing-based client-side scanning. In *USENIX Security Symposium*, 2022.

[158] SeokHwan Choi, Jin-Myeong Shin, and Yoon-Ho Choi. PIHA: detection method using perceptual image hashing against query-based adversarial attacks. *Future Gener. Comput. Syst.*, 145:563–577, 2023.

[159] Steven Chen, Nicholas Carlini, and David Wagner. Stateful detection of black-box adversarial attacks. In *Proceedings of the 1st ACM Workshop on Security and Privacy on Artificial Intelligence*, pages 30–39, 2020.

[160] Bardia Esmaeili, Amin Azmoodeh, Ali Dehghantanha, Hadis Karimipour, Behrouz Zolfaghari, and Mohammad Hammoudeh. Iiot deep malware threat hunting: From adversarial example detection to adversarial scenario detection. *IEEE Trans. Ind. Informatics*, 18(12):8477–8486, 2022.

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

[162] Florian Tramèr, Alexey Kurakin, Nicolas Papernot, Ian J. Goodfellow, Dan Boneh, and Patrick D. McDaniel. Ensemble adversarial training: Attacks and defenses. In *ICLR 2018*.

[163] Florian Tramèr and Dan Boneh. Adversarial training and robustness for multiple perturbations. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett, editors, *Advances in Neural Information Processing Systems, NeurIPS 2019*, pages 5858–5868, 2019.

[164] Guy Katz, Clark W. Barrett, David L. Dill, Kyle Julian, and Mykel J. Kochenderfer. Reluplex: An efficient SMT solver for verifying deep neural networks. In *Computer Aided Verification - 29th International Conference, CAV 2017*, 2017.

[165] Matteo Fischetti and Jason Jo. Deep neural networks and mixed integer linear optimization. *Constraints An Int. J.*, 23(3):296–309, 2018.

[166] Cem Anil, James Lucas, and Roger B. Grosse. Sorting out lipschitz function approximation. In *ICML*, volume 97, pages 291–301. PMLR, 2019.

[167] Eric Wong and J. Zico Kolter. Provable defenses against adversarial examples via the convex outer adversarial polytope. In Jennifer G. Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning, ICML 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 5283–5292. PMLR, 2018.

[168] Xinyu Zhang, Hanbin Hong, Yuan Hong, Peng Huang, Binghui Wang, Zhongjie Ba, and Kui Ren. Text-crs: A generalized certified robustness framework against textual adversarial attacks. In *IEEE Symposium on Security and Privacy*, 2024.

[169] Greg Yang, Tony Duan, J. Edward Hu, Hadi Salman, Ilya P. Razenshteyn, and Jerry Li. Randomized smoothing of all shapes and sizes. In *ICML*, 2020.

[170] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

[171] OpenAI. GPT-4 technical report. *CoRR*, abs/2303.08774, 2023.

[172] Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M. Dai, Anja Hauth, Katie Millican, David Silver, Slav Petrov, Melvin Johnson, Ioannis Antonoglou, Julian Schrittwieser, Amelia Glaese, Jilin Chen, Emily Pitler, Timothy P. Lillicrap, Angeliki Lazaridou, Orhan Firat, James Molloy, Michael Isard, Paul Ronald Barham, Tom Hennigan, Benjamin Lee, Fabio Viola, Malcolm Reynolds, Yuanzhong Xu, Ryan Doherty, Eli Collins, Clemens Meyer, Eliza Rutherford, Erica Moreira, Kareem Ayoub, Megha Goel, George Tucker, Enrique Piqueras, Maxim Krikun, Iain Barr, Nikolay Savinov, Ivo Danihelka, Becca Roelofs, Anaïs White, Anders Andreassen, Tamara von Glehn, Lakshman Yagati, Mehran Kazemi, Lucas Gonzalez, Misha Khalman, Jakub Sygnowski, and et al. Gemini: A family of highly capable multimodal models. *CoRR*, abs/2312.11805, 2023.

[173] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurélien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. Llama: Open and efficient foundation language models. *CoRR*, abs/2302.13971, 2023.

[174] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton-Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurélien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models. *CoRR*, abs/2307.09288, 2023.

[175] Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, Nicholas Joseph, Saurav Kadavath, Jackson Kernion, Tom Conerly, Sheer El Showk, Nelson Elhage, Zac Hatfield-Dodds, Danny Hernandez, Tristan Hume, Scott Johnston, Shauna Kravec, Liane Lovitt, Neel Nanda, Catherine Olsson, Dario Amodei, Tom B. Brown, Jack Clark, Sam McCandlish, Chris Olah, Benjamin

Mann, and Jared Kaplan. Training a helpful and harmless assistant with reinforcement learning from human feedback. *CoRR*, abs/2204.05862, 2022.

[176] Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, Binyuan Hui, Luo Ji, Mei Li, Junyang Lin, Runji Lin, Dayiheng Liu, Gao Liu, Chengqiang Lu, Keming Lu, Jianxin Ma, Rui Men, Xingzhang Ren, Xuancheng Ren, Chuanqi Tan, Sinan Tan, Jianhong Tu, Peng Wang, Shijie Wang, Wei Wang, Shengguang Wu, Benfeng Xu, Jin Xu, An Yang, Hao Yang, Jian Yang, Shusheng Yang, Yang Yao, Bowen Yu, Hongyi Yuan, Zheng Yuan, Jianwei Zhang, Xingxuan Zhang, Yichang Zhang, Zhenru Zhang, Chang Zhou, Jingren Zhou, Xiaohuan Zhou, and Tianhang Zhu. Qwen technical report. *CoRR*, abs/2309.16609, 2023.

[177] 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, 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. *CoRR*, abs/2412.15115, 2024.

[178] An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, Jianxin Yang, Jin Xu, Jingren Zhou, Jinze Bai, Jinzheng He, Junyang Lin, Kai Dang, Keming Lu, Keqin Chen, Kexin Yang, Mei Li, Mingfeng Xue, Na Ni, Pei Zhang, Peng Wang, Ru Peng, Rui Men, Ruize Gao, Runji Lin, Shijie Wang, Shuai Bai, Sinan Tan, Tianhang Zhu, Tianhao Li, Tianyu Liu, Wenbin Ge, Xiaodong Deng, Xiaohuan Zhou, Xingzhang Ren, Xinyu Zhang, Xipin Wei, Xuancheng Ren, Xuejing Liu, Yang Fan, Yang Yao, Yichang Zhang, Yu Wan, Yunfei Chu, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, Zhifang Guo, and Zhihao Fan. Qwen2 technical report. *CoRR*, abs/2407.10671, 2024.

[179] ByteDance Seed Team. Seed1.5-vl technical report. *arXiv preprint arXiv:2505.07062*, 2025.

[180] Yue Liu, Xiaoxin He, Miao Xiong, Jinlan Fu, Shumin Deng, and Bryan Hooi. Flipattack: Jailbreak llms via flipping. *CoRR*, abs/2410.02832, 2024.

[181] Yiting Dong, Guobin Shen, Dongcheng Zhao, Xiang He, and Yi Zeng. Harnessing task overload for scalable jailbreak attacks on large language models. *CoRR*, abs/2410.04190, 2024.

[182] Qibing Ren, Hao Li, Dongrui Liu, Zhanxu Xie, Xiaoya Lu, Yu Qiao, Lei Sha, Junchi Yan, Lizhuang Ma, and Jing Shao. Derail yourself: Multi-turn LLM jailbreak attack through self-discovered clues. *CoRR*, abs/2410.10700, 2024.

[183] Xiaogeng Liu, Nan Xu, Muhao Chen, and Chaowei Xiao. Autodan: Generating stealthy jailbreak prompts on aligned large language models. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024.

[184] Yihe Fan, Yuxin Cao, Ziyu Zhao, Ziyao Liu, and Shaofeng Li. Unbridled icarus: A survey of the potential perils of image inputs in multimodal large language model security. In *IEEE International Conference on Systems, Man, and Cybernetics, SMC 2024, Kuching, Malaysia, October 6-10, 2024*, pages 3428–3433. IEEE, 2024.

[185] Guobin Shen, Dongcheng Zhao, Yiting Dong, Xiang He, and Yi Zeng. Jailbreak antidote: Runtime safety-utility balance via sparse representation adjustment in large language models. *CoRR*, abs/2410.02298, 2024.

[186] Peiran Wang, Xiaogeng Liu, and Chaowei Xiao. Repd: Defending jailbreak attack through a retrieval-based prompt decomposition process. *CoRR*, abs/2410.08660, 2024.

[187] Giandomenico Cornacchia, Giulio Zizzo, Kieran Fraser, Muhammad Zaid Hameed, Ambrish Rawat, and Mark Purcell. Moje: Mixture of jailbreak experts, naive tabular classifiers as guard for prompt attacks. *CoRR*, abs/2409.17699, 2024.

[188] Sibo Yi, Yule Liu, Zhen Sun, Tianshuo Cong, Xinlei He, Jiaxing Song, Ke Xu, and Qi Li. Jailbreak attacks and defenses against large language models: A survey. *CoRR*, abs/2407.04295, 2024.

[189] Aysan Esmradi, Daniel Wankit Yip, and Chun-Fai Chan. A comprehensive survey of attack techniques, implementation, and mitigation strategies in large language models. In Guojun Wang, Haozhe Wang, Geyong Min, Nektarios Georgalas, and Weizhi Meng, editors, *Ubiquitous Security - Third International Conference, UbiSec 2023, Exeter, UK, November 1-3, 2023, Revised Selected Papers*, volume 2034 of *Communications in Computer and Information Science*, pages 76–95. Springer, 2023.

[190] Haibo Jin, Leyang Hu, Xinuo Li, Peiyan Zhang, Chonghan Chen, Jun Zhuang, and Haohan Wang. Jailbreakzoo: Survey, landscapes, and horizons in jailbreaking large language and vision-language models. *CoRR*, abs/2407.01599, 2024.

[191] Patrick Chao, Edoardo Debenedetti, Alexander Robey, Maksym Andriushchenko, Francesco Croce, Vikash Sehwag, Edgar Dobriban, Nicolas Flammarion, George J. Pappas, Florian Tramèr, Hamed Hassani, and Eric Wong. Jailbreakbench: An open robustness benchmark for jailbreaking large language models. In Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang, editors, *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024.

[192] Fenghua Weng, Yue Xu, Chengyan Fu, and Wenjie Wang. Mmj-bench: A comprehensive study on jailbreak attacks and defenses for vision language models. In Toby Walsh, Julie Shah, and Zico Kolter, editors, *AAAI-25, Sponsored by the Association for the Advancement of Artificial Intelligence, February 25 - March 4, 2025, Philadelphia, PA, USA*, pages 27689–27697. AAAI Press, 2025.

[193] Yueqi Xie, Minghong Fang, Renjie Pi, and Neil Zhenqiang Gong. Gradsafe: Detecting unsafe prompts for llms via safety-critical gradient analysis. *CoRR*, abs/2402.13494, 2024.

[194] Zihao Xu, Yi Liu, Gelei Deng, Yuekang Li, and Stjepan Picek. A comprehensive study of jailbreak attack versus defense for large language models. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Findings of the Association for Computational Linguistics, ACL 2024, Bangkok, Thailand and virtual meeting, August 11-16, 2024*, pages 7432–7449. Association for Computational Linguistics, 2024.

[195] Nanna Inie, Jonathan Stray, and Leon Derczynski. Summon a demon and bind it: A grounded theory of LLM red teaming in the wild. *CoRR*, abs/2311.06237, 2023.

[196] Erfan Shayegani, Md Abdullah Al Mamun, Yu Fu, Pedram Zaree, Yue Dong, and Nael B. Abu-Ghazaleh. Survey of vulnerabilities in large language models revealed by adversarial attacks. *CoRR*, abs/2310.10844, 2023.

[197] Maanak Gupta, Charankumar Akiri, Kshitiz Aryal, Eli Parker, and Lopamudra Praharaj. From chatgpt to threatgpt: Impact of generative AI in cybersecurity and privacy. *IEEE Access*, 11:80218–80245, 2023.

[198] Youliang Yuan, Wenxiang Jiao, Wenxuan Wang, Jen-tse Huang, Pinjia He, Shuming Shi, and Zhaopeng Tu. GPT-4 is too smart to be safe: Stealthy chat with llms via cipher. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024.

[199] Divij Handa, Advait Chirmule, Bimal G. Gajera, and Chitta Baral. Jailbreaking proprietary large language models using word substitution cipher. *CoRR*, abs/2402.10601, 2024.

[200] Jan Clusmann, Dyke Ferber, Isabella C. Wiest, Carolin V. Schneider, Titus J. Brinker, Sebastian Foersch, Daniel Truhn, and Jakob Nikolas Kather. Prompt injection attacks on large language models in oncology. *CoRR*, abs/2407.18981, 2024.

[201] Jiawei Ma, Po-Yao Huang, Saining Xie, Shang-Wen Li, Luke Zettlemoyer, Shih-Fu Chang, Wen-Tau Yih, and Hu Xu. Mode: CLIP data experts via clustering. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2024, Seattle, WA, USA, June 16-22, 2024*, pages 26344–26353. IEEE, 2024.

[202] Sander Schulhoff, Jeremy Pinto, Anaum Khan, Louis-François Bouchard, Chenglei Si, Svetlina Anati, Valen Tagliabue, Anson Liu Kost, Christopher Carnahan, and Jordan L. Boyd-Graber. Ignore this title and hackaprompt: Exposing systemic vulnerabilities of llms through a global prompt hacking competition. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 4945–4977. Association for Computational Linguistics, 2023.

[203] Jie Li, Yi Liu, Chongyang Liu, Ling Shi, Xiaoning Ren, Yaowen Zheng, Yang Liu, and Yinxing Xue. A cross-language investigation into jailbreak attacks in large language models. *CoRR*, abs/2401.16765, 2024.

[204] Sam Toyer, Olivia Watkins, Ethan Adrian Mendes, Justin Svegliato, Luke Bailey, Tiffany Wang, Isaac Ong, Karim Elmaaroufi, Pieter Abbeel, Trevor Darrell, Alan Ritter, and Stuart Russell. Tensor trust: Interpretable prompt injection attacks from an online game. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024.

[205] Tom Gibbs, Ethan Kosak-Hine, George Ingebretsen, Jason Zhang, Julius Broomfield, Sara Pieri, Reihaneh Iranmanesh, Reihaneh Rabbany, and Kellin Pelrine. Emerging vulnerabilities in frontier models: Multi-turn jailbreak attacks. *CoRR*, abs/2409.00137, 2024.

[206] Zhongjie Ba, Jieming Zhong, Jiachen Lei, Peng Cheng, Qinglong Wang, Zhan Qin, Zhibo Wang, and Kui Ren. Surrogateprompt: Bypassing the safety filter of text-to-image models via substitution. In Bo Luo, Xiaojing Liao, Jun Xu, Engin Kirda, and David Lie, editors, *Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security, CCS 2024, Salt Lake City, UT, USA, October 14-18, 2024*, pages 1166–1180. ACM, 2024.

[207] Huachuan Qiu, Shuai Zhang, Anqi Li, Hongliang He, and Zhenzhong Lan. Latent jailbreak: A benchmark for evaluating text safety and output robustness of large language models. *CoRR*, abs/2307.08487, 2023.

[208] Yi Liu, Gelei Deng, Zhengzi Xu, Yuekang Li, Yaowen Zheng, Ying Zhang, Lida Zhao, Tianwei Zhang, and Yang Liu. Jailbreaking chatgpt via prompt engineering: An empirical study. *CoRR*, abs/2305.13860, 2023.

[209] Chong Zhang, Mingyu Jin, Qinkai Yu, Chengzhi Liu, Haochen Xue, and Xiaobo Jin. Goal-guided generative prompt injection attack on large language models. In Elena Baralis, Kun Zhang, Ernesto Damiani, Mérouane Debbah, Panos Kalnis, and Xindong Wu, editors, *IEEE International Conference on Data Mining, ICDM 2024, Abu Dhabi, United Arab Emirates, December 9-12, 2024*, pages 941–946. IEEE, 2024.

[210] Xiaoxia Li, Siyuan Liang, Jiyi Zhang, Han Fang, Aishan Liu, and Ee-Chien Chang. Semantic mirror jailbreak: Genetic algorithm based jailbreak prompts against open-source llms. *CoRR*, abs/2402.14872, 2024.

[211] Xirui Li, Ruochen Wang, Minhao Cheng, Tianyi Zhou, and Cho-Jui Hsieh. Drattack: Prompt decomposition and reconstruction makes powerful LLM jailbreakers. *CoRR*, abs/2402.16914, 2024.

[212] Zhihao Lin, Wei Ma, Mingyi Zhou, Yanjie Zhao, Haoyu Wang, Yang Liu, Jun Wang, and Li Li. Pathseeker: Exploring LLM security vulnerabilities with a reinforcement learning-based jailbreak approach. *CoRR*, abs/2409.14177, 2024.

[213] Wenxuan Wang, Kuiyi Gao, Zihan Jia, Youliang Yuan, Jen-tse Huang, Qiuzhi Liu, Shuai Wang, Wenxiang Jiao, and Zhaopeng Tu. Chain-of-jailbreak attack for image generation models via editing step by step. *CoRR*, abs/2410.03869, 2024.

[214] David Glukhov, Ziwen Han, Ilia Shumailov, Vardan Papyan, and Nicolas Papernot. A false sense of safety: Unsafe information leakage in 'safe' AI responses. *CoRR*, abs/2407.02551, 2024.

[215] Xiao Liu, Liangzhi Li, Tong Xiang, Fuying Ye, Lu Wei, Wangyue Li, and Noa Garcia. Imposter.ai: Adversarial attacks with hidden intentions towards aligned large language models. *CoRR*, abs/2407.15399, 2024.

[216] Tong Liu, Yingjie Zhang, Zhe Zhao, Yinpeng Dong, Guozhu Meng, and Kai Chen. Making them ask and answer: Jailbreaking large language models in few queries via disguise and reconstruction. In Davide Balzarotti and Wenyuan Xu, editors, *33rd USENIX Security Symposium, USENIX Security 2024, Philadelphia, PA, USA, August 14-16, 2024*. USENIX Association, 2024.

[217] Daniel Kang, Xuechen Li, Ion Stoica, Carlos Guestrin, Matei Zaharia, and Tatsunori Hashimoto. Exploiting programmatic behavior of llms: Dual-use through standard security attacks. In *IEEE Security and Privacy, SP 2024 - Workshops, San Francisco, CA, USA, May 23, 2024*, pages 132–143. IEEE, 2024.

[218] Yixin Cheng, Markos Georgopoulos, Volkan Cevher, and Grigorios G. Chrysos. Leveraging the context through multi-round interactions for jailbreaking attacks. *CoRR*, abs/2402.09177, 2024.

[219] Zhiyuan Yu, Xiaogeng Liu, Shunning Liang, Zach Cameron, Chaowei Xiao, and Ning Zhang. Don't listen to me: Understanding and exploring jailbreak prompts of large language models. In Davide Balzarotti and Wenyuan Xu, editors, *33rd USENIX Security Symposium, USENIX Security 2024, Philadelphia, PA, USA, August 14-16, 2024*. USENIX Association, 2024.

[220] Zeguan Xiao, Yan Yang, Guanhua Chen, and Yun Chen. Tastle: Distract large language models for automatic jailbreak attack. *CoRR*, abs/2403.08424, 2024.

[221] Raz Lapid, Ron Langberg, and Moshe Sipper. Open sesame! universal black box jailbreaking of large language models. *CoRR*, abs/2309.01446, 2023.

[222] Lin Lu, Hai Yan, Zenghui Yuan, Jiawen Shi, Wenqi Wei, Pin-Yu Chen, and Pan Zhou. Autojailbreak: Exploring jailbreak attacks and defenses through a dependency lens. *CoRR*, abs/2406.03805, 2024.

[223] Xinyuan Wang, Victor Shea-Jay Huang, Renmiao Chen, Hao Wang, Chengwei Pan, Lei Sha, and Minlie Huang. Blackdan: A black-box multi-objective approach for effective and contextual jailbreaking of large language models. *CoRR*, abs/2410.09804, 2024.

[224] Xiaosen Zheng, Tianyu Pang, Chao Du, Qian Liu, Jing Jiang, and Min Lin. Improved few-shot jailbreaking can circumvent aligned language models and their defenses. In Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang, editors, *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024.

[225] Maksym Andriushchenko, Francesco Croce, and Nicolas Flammarion. Jailbreaking leading safety-aligned llms with simple adaptive attacks. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025.

[226] Erik Jones, Anca D. Dragan, Aditi Raghunathan, and Jacob Steinhardt. Automatically auditing large language models via discrete optimization. In Andreas

Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pages 15307–15329. PMLR, 2023.

[227] Sizhe Chen, Julien Piet, Chawin Sitawarin, and David A. Wagner. Struq: Defending against prompt injection with structured queries. *CoRR*, abs/2402.06363, 2024.

[228] Xiaotian Zou, Yongkang Chen, and Ke Li. Is the system message really important to jailbreaks in large language models? *CoRR*, abs/2402.14857, 2024.

[229] Chawin Sitawarin, Norman Mu, David A. Wagner, and Alexandre Araujo. PAL: proxy-guided black-box attack on large language models. *CoRR*, abs/2402.09674, 2024.

[230] Andy Zou, Zifan Wang, J. Zico Kolter, and Matt Fredrikson. Universal and transferable adversarial attacks on aligned language models. *CoRR*, abs/2307.15043, 2023.

[231] Jiawen Shi, Zenghui Yuan, Yinuo Liu, Yue Huang, Pan Zhou, Lichao Sun, and Neil Zhenqiang Gong. Optimization-based prompt injection attack to llm-as-a-judge. In Bo Luo, Xiaojing Liao, Jun Xu, Engin Kirda, and David Lie, editors, *Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security, CCS 2024, Salt Lake City, UT, USA, October 14-18, 2024*, pages 660–674. ACM, 2024.

[232] Jiachen Ma, Yijiang Li, Zhiqing Xiao, Anda Cao, Jie Zhang, Chao Ye, and Junbo Zhao. Jailbreaking prompt attack: A controllable adversarial attack against diffusion models. In Luis Chiruzzo, Alan Ritter, and Lu Wang, editors, *Findings of the Association for Computational Linguistics: NAACL 2025, Albuquerque, New Mexico, USA, April 29 - May 4, 2025*, pages 3141–3157. Association for Computational Linguistics, 2025.

[233] Xuan Chen, Yuzhou Nie, Wenbo Guo, and Xiangyu Zhang. When LLM meets DRL: advancing jailbreaking efficiency via drl-guided search. In Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang, editors, *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024.

[234] Yuchen Yang, Bo Hui, Haolin Yuan, Neil Gong, and Yinzhi Cao. Sneakyprompt: Jailbreaking text-to-image generative models. In *IEEE Symposium on Security and Privacy, SP 2024, San Francisco, CA, USA, May 19-23, 2024*, pages 897–912. IEEE, 2024.

[235] Xuan Chen, Yuzhou Nie, Lu Yan, Yunshu Mao, Wenbo Guo, and Xiangyu Zhang. RL-JACK: reinforcement learning-powered black-box jailbreaking attack against llms. *CoRR*, abs/2406.08725, 2024.

[236] Yuhao Du, Zhuo Li, Pengyu Cheng, Xiang Wan, and Anningzhe Gao. Detecting AI flaws: Target-driven attacks on internal faults in language models. *CoRR*, abs/2408.14853, 2024.

[237] Yi Liu, Chengjun Cai, Xiaoli Zhang, Xingliang Yuan, and Cong Wang. Arondight: Red teaming large vision language models with auto-generated multi-modal jailbreak prompts. In Jianfei Cai, Mohan S. Kankanhalli, Balakrishnan Prabhakaran, Susanne Boll, Ramanathan Subramanian, Liang Zheng, Vivek K. Singh, Pablo César, Lexing Xie, and Dong Xu, editors, *Proceedings of the 32nd ACM International Conference on Multimedia, MM 2024, Melbourne, VIC, Australia, 28 October 2024 - 1 November 2024*, pages 3578–3586. ACM, 2024.

[238] Yan Yang, Zeguan Xiao, Xin Lu, Hongru Wang, Hailiang Huang, Guanhua Chen, and Yun Chen. Sop: Unlock the power of social facilitation for automatic jailbreak attack. *CoRR*, abs/2407.01902, 2024.

[239] Jiahao Yu, Xingwei Lin, Zheng Yu, and Xinyu Xing. GPTFUZZER: red teaming large language models with auto-generated jailbreak prompts. *CoRR*, abs/2309.10253, 2023.

[240] Govind Ramesh, Yao Dou, and Wei Xu. GPT-4 jailbreaks itself with near-perfect success using self-explanation. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen, editors, *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, EMNLP 2024, Miami, FL, USA, November 12-16, 2024*, pages 22139–22148. Association for Computational Linguistics, 2024.

[241] Chiju Chao, Qirui Wang, Hongfei Wu, and Zhiyong Fu. Synneure: Intelligent human-machine teamwork in virtual space. In Pei-Luen Patrick Rau, editor, *Cross-Cultural Design - 15th International Conference, CCD 2023, Held as Part of the 25th International Conference, HCII 2023, Copenhagen, Denmark, July 23-28, 2023, Proceedings, Part II*, volume 14023 of *Lecture Notes in Computer Science*, pages 349–361. Springer, 2023.

[242] Anselm Paulus, Arman Zharmagambetov, Chuan Guo, Brandon Amos, and Yuandong Tian. Advprompter: Fast adaptive adversarial prompting for llms. *CoRR*, abs/2404.16873, 2024.

[243] Jiyue Jiang, Liheng Chen, Pengan Chen, Sheng Wang, Qinghang Bao, Lingpeng Kong, Yu Li, and Chuan Wu. How far can cantonese NLP go? benchmarking cantonese capabilities of large language models. *CoRR*, abs/2408.16756, 2024.

[244] Siyuan Ma, Weidi Luo, Yu Wang, Xiaogeng Liu, Muhao Chen, Bo Li, and Chaowei Xiao. Visual-roleplay: Universal jailbreak attack on multimodal large language models via role-playing image characte. *CoRR*, abs/2405.20773, 2024.

[245] Haibo Jin, Ruoxi Chen, Andy Zhou, Jinyin Chen, Yang Zhang, and Haohan Wang. GUARD: role-playing to generate natural-language jailbreakings to test guideline adherence of large language models. *CoRR*, abs/2402.03299, 2024.

[246] Xiaogeng Liu, Peiran Li, G. Edward Suh, Yevgeniy Vorobeychik, Zhuoqing Mao, Somesh Jha, Patrick McDaniel, Huan Sun, Bo Li, and Chaowei Xiao. Autodan-turbo: A lifelong agent for strategy self-exploration to jailbreak llms. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025.

[247] Yingkai Dong, Zheng Li, Xiangtao Meng, Ning Yu, and Shanqing Guo. Jailbreaking text-to-image models with llm-based agents. *CoRR*, abs/2408.00523, 2024.

[248] Yu Tian, Xiao Yang, Jingyuan Zhang, Yinpeng Dong, and Hang Su. Evil geniuses: Delving into the safety of llm-based agents. *CoRR*, abs/2311.11855, 2023.

[249] Neal Mangaokar, Ashish Hooda, Jihye Choi, Shreyas Chandrashekaran, Kassem Fawaz, Somesh Jha, and Atul Prakash. PRP: propagating universal perturbations to attack large language model guard-rails. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 10960–10976. Association for Computational Linguistics, 2024.

[250] Jonathan Hayase, Ema Borevkovic, Nicholas Carlini, Florian Tramèr, and Milad Nasr. Query-based adversarial prompt generation. In Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang, editors, *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024.

[251] Xikang Yang, Xuehai Tang, Songlin Hu, and Jizhong Han. Chain of attack: a semantic-driven contextual multi-turn attacker for LLM. *CoRR*, abs/2405.05610, 2024.

[252] 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. *CoRR*, abs/2408.15221, 2024.

[253] Rishabh Bhardwaj and Soujanya Poria. Red-teaming large language models using chain of utterances for safety-alignment. *CoRR*, abs/2308.09662, 2023.

[254] Mark Russinovich, Ahmed Salem, and Ronen Eldan. Great, now write an article about that: The crescendo multi-turn LLM jailbreak attack. *CoRR*, abs/2404.01833, 2024.

[255] Bojian Jiang, Yi Jing, Tong Wu, Tianhao Shen, Deyi Xiong, and Qing Yang. Automated progressive red teaming. In Owen Rambow, Leo Wanner, Marianna Apidianaki, Hend Al-Khalifa, Barbara Di Eugenio, and Steven Schockaert, editors, *Proceedings of the 31st International Conference on Computational Linguistics, COLING 2025, Abu Dhabi, UAE, January 19-24, 2025*, pages 3850–3864. Association for Computational Linguistics, 2025.

[256] Shi Lin, Rongchang Li, Xun Wang, Changting Lin, Wenpeng Xing, and Meng Han. Figure it out: Analyzing-based jailbreak attack on large language models. *CoRR*, abs/2407.16205, 2024.

[257] Yongshuo Zong, Ondrej Bohdal, Tingyang Yu, Yongxin Yang, and Timothy M. Hospedales. Safety fine-tuning at (almost) no cost: A baseline for vision large language models. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net, 2024.

[258] Rongwu Xu, Zehan Qi, and Wei Xu. Preemptive answer "attacks" on chain-of-thought reasoning. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Findings of the Association for Computational Linguistics, ACL 2024, Bangkok, Thailand and virtual meeting, August 11-16, 2024*, pages 14708–14726. Association for Computational Linguistics, 2024.

[259] Yuqi Zhou, Lin Lu, Ryan Sun, Pan Zhou, and Lichao Sun. Virtual context enhancing jailbreak attacks with special token injection. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen, editors, *Findings of the Association for Computational Linguistics: EMNLP 2024, Miami, Florida, USA, November 12-16, 2024*, pages 11843–11857. Association for Computational Linguistics, 2024.

[260] Jonas Geiping, Alex Stein, Manli Shu, Khalid Saifullah, Yuxin Wen, and Tom Goldstein. Coercing llms to do and reveal (almost) anything. *CoRR*, abs/2402.14020, 2024.

[261] Gelei Deng, Yi Liu, Kailong Wang, Yuekang Li, Tianwei Zhang, and Yang Liu. Pandora: Jailbreak gpts by retrieval augmented generation poisoning. *CoRR*, abs/2402.08416, 2024.

[262] Yukai Zhou and Wenjie Wang. Don't say no: Jailbreaking LLM by suppressing refusal. *CoRR*, abs/2404.16369, 2024.

[263] Yangsibo Huang, Samyak Gupta, Mengzhou Xia, Kai Li, and Danqi Chen. Catastrophic jailbreak of open-source llms via exploiting generation. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024.

[264] Xuandong Zhao, Xianjun Yang, Tianyu Pang, Chao Du, Lei Li, Yu-Xiang Wang, and William Yang Wang. Weak-to-strong jailbreaking on large language models. *CoRR*, abs/2401.17256, 2024.

[265] Zhuo Zhang, Guangyu Shen, Guanhong Tao, Siyuan Cheng, and Xiangyu Zhang. On large language models' resilience to coercive interrogation. In *IEEE Symposium on Security and Privacy, SP 2024, San Francisco, CA, USA, May 19-23, 2024*, pages 826–844. IEEE, 2024.

[266] Hangfan Zhang, Zhimeng Guo, Huaisheng Zhu, Bochuan Cao, Lu Lin, Jinyuan Jia, Jinghui Chen, and Dinghao Wu. Jailbreak open-sourced large language models via enforced decoding. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 5475–5493. Association for Computational Linguistics, 2024.

[267] Haoyu Wang, Bingzhe Wu, Yatao Bian, Yongzhe Chang, Xueqian Wang, and Peilin Zhao. Probing the safety response boundary of large language models via unsafe decoding path generation. *CoRR*, abs/2408.10668, 2024.

[268] Ziqiu Wang, Jun Liu, Shengkai Zhang, and Yang Yang. Poisoned langchain: Jailbreak llms by langchain. *CoRR*, abs/2406.18122, 2024.

[269] Junjie Ye, Sixian Li, Guanyu Li, Caishuang Huang, Songyang Gao, Yilong Wu, Qi Zhang, Tao Gui, and Xuanjing Huang. Toolsword: Unveiling safety issues of large language models in tool learning across three stages. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 2181–2211. Association for Computational Linguistics, 2024.

[270] Edoardo Debenedetti, Jie Zhang, Mislav Balunovic, Luca Beurer-Kellner, Marc Fischer, and Florian Tramèr. Agentdojo: A dynamic environment to evaluate attacks and defenses for LLM agents. *CoRR*, abs/2406.13352, 2024.

[271] Stav Cohen, Ron Bitton, and Ben Nassi. A jailbroken genai model can cause substantial harm: Genai-powered applications are vulnerable to promptwares. *CoRR*, abs/2408.05061, 2024.

[272] Xin Liu, Yichen Zhu, Jindong Gu, Yunshi Lan, Chao Yang, and Yu Qiao. Mm-safetybench: A benchmark for safety evaluation of multimodal large language models. In Ales Leonardis, Elisa Ricci, Stefan Roth, Olga Russakovsky, Torsten Sattler, and Gül Varol, editors, *Computer Vision - ECCV 2024 - 18th European Conference, Milan, Italy, September 29-October 4, 2024, Proceedings, Part LVI*, volume 15114 of *Lecture Notes in Computer Science*, pages 386–403. Springer, 2024.

[273] Yixuan Li, Xuelin Liu, Xiaoyang Wang, Shiqi Wang, and Weisi Lin. Fakebench: Uncover the achilles' heels of fake images with large multimodal models. *CoRR*, abs/2404.13306, 2024.

[274] Penghao Sun, Julong Lan, Yuxiang Hu, Zehua Guo, Chong Wu, and Jiangxing Wu. Realizing the carbon-aware service provision in ICT system. *IEEE Trans. Netw. Serv. Manag.*, 21(4):4090–4103, 2024.

[275] Subaru Kimura, Ryota Tanaka, Shumpei Miyawaki, Jun Suzuki, and Keisuke Sakaguchi. Empirical analysis of large vision-language models against goal hijacking via visual prompt injection. *CoRR*, abs/2408.03554, 2024.

[276] Yue Deng, Wenxuan Zhang, Sinno Jialin Pan, and Lidong Bing. Multilingual jailbreak challenges in large language models. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024.

[277] Dmitrii Volkov. Badllama 3: removing safety finetuning from llama 3 in minutes. *CoRR*, abs/2407.01376, 2024.

[278] Jiongxiao Wang, Jiazhao Li, Yiquan Li, Xiangyu Qi, Junjie Hu, Yixuan Li, Patrick McDaniel, Muhao Chen, Bo Li, and Chaowei Xiao. Mitigating fine-tuning jailbreak attack with backdoor enhanced alignment. *CoRR*, abs/2402.14968, 2024.

[279] Zi Wang, Divyam Anshumaan, Ashish Hooda, Yudong Chen, and Somesh Jha. Functional homotopy: Smoothing discrete optimization via continuous parameters for LLM jailbreak attacks. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025.

[280] Tanmoy Hazra, Kushal Anjaria, Aditi Bajpai, and Akshara Kumari. *Applications of Game Theory in Deep Learning*. Springer Briefs in Computer Science. Springer, 2024.

[281] Xiaoqun Liu, Jiacheng Liang, Muchao Ye, and Zhaohan Xi. Robustifying safety-aligned large language models through clean data curation. *CoRR*, abs/2405.19358, 2024.

[282] Wei Zhao, Zhe Li, Yige Li, and Jun Sun. Adversarial suffixes may be features too! *CoRR*, abs/2410.00451, 2024.

[283] Suraj Anand and David Getzen. Are ppo-ed language models hackable? *CoRR*, abs/2406.02577, 2024.

[284] Matt Duver, Noah Wiederhold, Maria Kyrarini, Sean Banerjee, and Natasha Kholgade Banerjee. Vr-hand-in-hand: Using virtual reality (VR) hand tracking for hand-object data annotation. In *2024 IEEE International Conference on Artificial Intelligence and eXtended and Virtual Reality (AIxVR), Los Angeles, CA, USA, January 17-19, 2024*, pages 325–329. IEEE, 2024.

[285] Javier Rando, Francesco Croce, Krystof Mitka, Stepan Shabalin, Maksym Andriushchenko, Nicolas Flammarion, and Florian Tramèr. Competition report: Finding universal jailbreak backdoors in aligned llms. *CoRR*, abs/2404.14461, 2024.

[286] Weipeng Jiang, Zhenting Wang, Juan Zhai, Shiqing Ma, Zhengyu Zhao, and Chao Shen. Unlocking adversarial suffix optimization without affirmative phrases: Efficient black-box jailbreaking via LLM as optimizer. *CoRR*, abs/2408.11313, 2024.

[287] Zeyi Liao and Huan Sun. Amplegcg: Learning a universal and transferable generative model of adversarial suffixes for jailbreaking both open and closed llms. *CoRR*, abs/2404.07921, 2024.

[288] Hao Wang, Hao Li, Minlie Huang, and Lei Sha. From noise to clarity: Unraveling the adversarial suffix of large language model attacks via translation of text embeddings. *CoRR*, abs/2402.16006, 2024.

[289] Gabriel Alon and Michael Kamfonas. Detecting language model attacks with perplexity. *CoRR*, abs/2308.14132, 2023.

[290] Hongfu Liu, Yuxi Xie, Ye Wang, and Michael Shieh. Advancing adversarial suffix transfer learning on aligned large language models. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen, editors, *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, EMNLP 2024, Miami, FL, USA, November 12-16, 2024*, pages 7213–7224. Association for Computational Linguistics, 2024.

[291] Xiaojun Jia, Tianyu Pang, Chao Du, Yihao Huang, Jindong Gu, Yang Liu, Xiaochun Cao, and Min Lin. Improved techniques for optimization-based jailbreaking on large language models. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025.

[292] Qizhang Li, Yiwen Guo, Wangmeng Zuo, and Hao Chen. Improved generation of adversarial examples against safety-aligned llms. In Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang, editors, *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024.

[293] Wen-xiao Lu, Ping Fang, Ming-lu Zhu, Yi-run Zhu, Xinjian Fan, Tian-chen Zhu, Xuan Zhou, Feng-Xia Wang, Tao Chen, and Li-ning Sun. Artificial intelligence-enabled gesture-language-recognition feedback system using strain-sensor-arrays-based smart glove. *Adv. Intell. Syst.*, 5(8), 2023.

[294] Xingang Guo, Fangxu Yu, Huan Zhang, Lianhui Qin, and Bin Hu. Cold-attack: Jailbreaking llms with stealthiness and controllability. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net, 2024.

[295] Simon Geisler, Tom Wollschläger, M. H. I. Abdalla, Johannes Gasteiger, and Stephan Günnemann. Attacking large language models with projected gradient descent. *CoRR*, abs/2402.09154, 2024.

[296] Dario Pasquini, Martin Strohmeier, and Carmela Troncoso. Neural exec: Learning (and learning from) execution triggers for prompt injection attacks. In Maura Pintor, Xinyun Chen, and Matthew Jagielski, editors, *Proceedings of the 2024 Workshop on Artificial Intelligence and Security, AISec 2024, Salt Lake City, UT, USA, October 14-18, 2024*, pages 89–100. ACM, 2024.

[297] Giandomenico Cornacchia, Giulio Zizzo, Kieran Fraser, Muhammad Zaid Hameed, Ambrish Rawat, and Mark Purcell. Moje: Mixture of jailbreak experts, naive tabular classifiers as guard for prompt attacks. In Sanmay Das, Brian Patrick Green, Kush Varshney, Marianna Ganapini, and Andrea Renda, editors, *Proceedings of the Seventh AAAI/ACM Conference on AI, Ethics, and Society (AIES-24) - Full Archival Papers, October 21-23, 2024, San Jose, California, USA - Volume 1*, pages 304–315. AAAI Press, 2024.

[298] Md. Abdur Rahman, Hossain Shahriar, Fan Wu, and Alfredo Cuzzocrea. Applying pre-trained multilingual BERT in embeddings for improved malicious prompt injection attacks detection. In *2nd International Conference on Artificial*

Intelligence, Blockchain, and Internet of Things, AIBThings 2024, Mt Pleasant, MI, USA, September 7-8, 2024, pages 1–7. IEEE, 2024.

[299] Reshabh K. Sharma, Vinayak Gupta, and Dan Grossman. SPML: A DSL for defending language models against prompt attacks. *CoRR*, abs/2402.11755, 2024.

[300] Yuqi Zhang, Liang Ding, Lefei Zhang, and Dacheng Tao. Intention analysis prompting makes large language models A good jailbreak defender. *CoRR*, abs/2401.06561, 2024.

[301] Xunguang Wang, Daoyuan Wu, Zhenlan Ji, Zongjie Li, Pingchuan Ma, Shuai Wang, Yingjiu Li, Yang Liu, Ning Liu, and Juergen Rahmel. Selfdefend: Llms can defend themselves against jailbreaking in a practical manner. *CoRR*, abs/2406.05498, 2024.

[302] Blazej Manczak, Eliott Zemour, Eric Lin, and Vaikkunth Mugunthan. Primeguard: Safe and helpful llms through tuning-free routing. *CoRR*, abs/2407.16318, 2024.

[303] Diego Dorn, Alexandre Variengien, Charbel-Raphaël Ségerie, and Vincent Corruble. BELLS: A framework towards future proof benchmarks for the evaluation of LLM safeguards. *CoRR*, abs/2406.01364, 2024.

[304] Rida Noor, Abdul Wahid, Sibghat Ullah Bazai, Asad Khan, Meie Fang, Syam M. S, Uzair Aslam Bhatti, and Yazeed Yasin Ghadi. DLGAN: undersampled MRI reconstruction using deep learning based generative adversarial network. *Biomed. Signal Process. Control.*, 93:106218, 2024.

[305] Seungju Han, Kavel Rao, Allyson Ettinger, Liwei Jiang, Bill Yuchen Lin, Nathan Lambert, Yejin Choi, and Nouha Dziri. Wildguard: Open one-stop moderation tools for safety risks, jailbreaks, and refusals of llms. In Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang, editors, *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024.

[306] Shaona Ghosh, Prasoon Varshney, Erick Galinkin, and Christopher Parisien. AEGIS: online adaptive AI content safety moderation with ensemble of LLM experts. *CoRR*, abs/2404.05993, 2024.

[307] Sin-Han Yang, Tuomas P. Oikarinen, and Tsui-Wei Weng. Concept-driven continual learning. *Trans. Mach. Learn. Res.*, 2024, 2024.

[308] Anne Hartebrodt and Richard Röttger. Privacy of federated QR decomposition using additive secure multiparty computation. *IEEE Trans. Inf. Forensics Secur.*, 18:5122–5132, 2023.

[309] Xinyi Zeng, Yuying Shang, Yutao Zhu, Jiawei Chen, and Yu Tian. Root defence strategies: Ensuring safety of LLM at the decoding level. *CoRR*, abs/2410.06809, 2024.

[310] Xuefeng Du, Reshmi Ghosh, Robert Sim, Ahmed Salem, Vitor Carvalho, Emily Lawton, Yixuan Li, and Jack W. Stokes. Vlmguard: Defending vlms against malicious prompts via unlabeled data. *CoRR*, abs/2410.00296, 2024.

[311] Alexandra Souly, Qingyuan Lu, Dillon Bowen, Tu Trinh, Elvis Hsieh, Sana Pandey, Pieter Abbeel, Justin Svegliato, Scott Emmons, Olivia Watkins, and Sam Toyer. A strongreject for empty jailbreaks. In Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang, editors, *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024.

[312] Richard Harper and Dave W. Randall. Machine learning and the work of the user. *Comput. Support. Cooperative Work.*, 33(2):103–136, 2024.

[313] Xiaoyu Zhang, Cen Zhang, Tianlin Li, Yihao Huang, Xiaojun Jia, Xiaofei Xie, Yang Liu, and Chao Shen. A mutation-based method for multi-modal jailbreaking attack detection. *CoRR*, abs/2312.10766, 2023.

[314] Zhuowen Yuan, Zidi Xiong, Yi Zeng, Ning Yu, Ruoxi Jia, Dawn Song, and Bo Li. Rigorllm: Resilient guardrails for large language models against undesired content. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net, 2024.

[315] Cheng Qian, Hainan Zhang, Lei Sha, and Zhiming Zheng. HSF: defending against jailbreak attacks with hidden state filtering. In Guodong Long, Michale Blumestein, Yi Chang, Liane Lewin-Eytan, Zi Helen Huang, and Elad Yom-Tov, editors, *Companion Proceedings of the ACM on Web Conference 2025, WWW 2025, Sydney, NSW, Australia, 28 April 2025 - 2 May 2025*, pages 2078–2087. ACM, 2025.

[316] Xiaomeng Hu, Pin-Yu Chen, and Tsung-Yi Ho. Gradient cuff: Detecting jailbreak attacks on large language models by exploring refusal loss landscapes. In Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang, editors, *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024.

[317] Yueqi Xie, Minghong Fang, Renjie Pi, and Neil Zhenqiang Gong. Gradsafe: Detecting unsafe prompts for llms via safety-critical gradient analysis. *CoRR*, abs/2402.13494, 2024.

[318] Keegan Hines, Gary Lopez, Matthew Hall, Federico Zarfati, Yonatan Zunger, and Emre Kiciman. Defending against indirect prompt injection attacks with spotlighting. In Rachel Allen, Sagar Samtani, Edward Raff, and Ethan M. Rudd, editors, *Proceedings of the Conference on Applied Machine Learning in Information Security (CAMLIS 2024), Arlington, Virginia, USA, October 24-25, 2024*, volume 3920 of *CEUR Workshop Proceedings*, pages 48–62. CEUR-WS.org, 2024.

[319] Peiran Wang, Xiaogeng Liu, and Chaowei Xiao. Repd: Defending jailbreak attack through a retrieval-based prompt decomposition process. In Luis Chiruzzo, Alan Ritter, and Lu Wang, editors, *Findings of the Association for Computational Linguistics: NAACL 2025, Albuquerque, New Mexico, USA, April 29 - May 4, 2025*, pages 283–294. Association for Computational Linguistics, 2025.

[320] Neel Jain, Avi Schwarzschild, Yuxin Wen, Gowthami Somepalli, John Kirchenbauer, Ping-yeh Chiang, Micah Goldblum, Aniruddha Saha, Jonas Geiping, and Tom Goldstein. Baseline defenses for adversarial attacks against aligned language models. *CoRR*, abs/2309.00614, 2023.

[321] Aounon Kumar, Chirag Agarwal, Suraj Srinivas, Soheil Feizi, and Hima Lakkaraju. Certifying LLM safety against adversarial prompting. *CoRR*, abs/2309.02705, 2023.

[322] Bowen Zhao, Wei-Neng Chen, Feng-Feng Wei, Ximeng Liu, Qingqi Pei, and Jun Zhang. PEGA: A privacy-preserving genetic algorithm for combinatorial optimization. *IEEE Trans. Cybern.*, 54(6):3638–3651, 2024.

[323] Yu Wang, Xiaogeng Liu, Yu Li, Muhao Chen, and Chaowei Xiao. Adashield: Safeguarding multimodal large language models from structure-based attack via adaptive shield prompting. *CoRR*, abs/2403.09513, 2024.

[324] Chujie Zheng, Fan Yin, Hao Zhou, Fandong Meng, Jie Zhou, Kai-Wei Chang, Minlie Huang, and Nanyun Peng. Prompt-driven LLM safeguarding via directed representation optimization. *CoRR*, abs/2401.18018, 2024.

[325] Andy Zhou, Bo Li, and Haohan Wang. Robust prompt optimization for defending language models against jailbreaking attacks. In Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang, editors, *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024.

[326] Chen Xiong, Xiangyu Qi, Pin-Yu Chen, and Tsung-Yi Ho. Defensive prompt patch: A robust and interpretable defense of llms against jailbreak attacks. *CoRR*, abs/2405.20099, 2024.

[327] Zhao Xu, Fan Liu, and Hao Liu. Bag of tricks: Benchmarking of jailbreak attacks on llms. In Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang, editors, *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024.

[328] Zhexin Zhang, Junxiao Yang, Pei Ke, Shiyao Cui, Chujie Zheng, Hongning Wang, and Minlie Huang. Safe unlearning: A surprisingly effective and generalizable solution to defend against jailbreak attacks. *CoRR*, abs/2407.02855, 2024.

[329] Federico Bianchi, Mirac Suzgun, Giuseppe Attanasio, Paul Röttger, Dan Jurafsky, Tatsunori Hashimoto, and James Zou. Safety-tuned llamas: Lessons from improving the safety of large language models that follow instructions. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024.

[330] Taeyoun Kim, Suhas Kotha, and Aditi Raghunathan. Jailbreaking is best solved by definition. *CoRR*, abs/2403.14725, 2024.

[331] Yu Fu, Wen Xiao, Jia Chen, Jiachen Li, Evangelos E. Papalexakis, Aichi Chien, and Yue Dong. Cross-task defense: Instruction-tuning llms for content safety. *CoRR*, abs/2405.15202, 2024.

[332] Anubhav Bhatti, Prithila Angkan, Behnam Behinaein, Zunayed Mahmud, Dirk Rodenburg, Heather Braund, P. James Mclellan, Aaron J. Ruberto, Geoffery Harrison, Daryl Wilson, Adam Szulewski, Dan Howes, Ali Etemad, and Paul Hungler. CLARE: cognitive load assessment in realtime with multimodal data. *CoRR*, abs/2404.17098, 2024.

[333] Fan Liu, Zhao Xu, and Hao Liu. Adversarial tuning: Defending against jailbreak attacks for llms. *CoRR*, abs/2406.06622, 2024.

[334] Eric Wallace, Kai Xiao, Reimar Leike, Lilian Weng, Johannes Heidecke, and Alex Beutel. The instruction hierarchy: Training llms to prioritize privileged instructions. *CoRR*, abs/2404.13208, 2024.

[335] Trishna Chakraborty, Erfan Shayegani, Zikui Cai, Nael B. Abu-Ghazaleh, M. Salman Asif, Yue Dong, Amit K. Roy-Chowdhury, and Chengyu Song. Can textual unlearning solve cross-modality safety alignment? In Yaser Al-Onaizan,

Mohit Bansal, and Yun-Nung Chen, editors, *Findings of the Association for Computational Linguistics: EMNLP 2024, Miami, Florida, USA, November 12-16, 2024*, pages 9830–9844. Association for Computational Linguistics, 2024.

[336] Víctor Gallego. Merging improves self-critique against jailbreak attacks. *CoRR*, abs/2406.07188, 2024.

[337] 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 Gómez-Adorno, and Steven Bethard, editors, *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), NAACL 2024, Mexico City, Mexico, June 16-21, 2024*, pages 1927–1937. Association for Computational Linguistics, 2024.

[338] Liwei Jiang, Kavel Rao, Seungju Han, Allyson Ettinger, Faeze Brahman, Sachin Kumar, Niloofar Mireshghallah, Ximing Lu, Maarten Sap, Yejin Choi, and Nouha Dziri. Wildteaming at scale: From in-the-wild jailbreaks to (adversarially) safer language models. In Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang, editors, *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024.

[339] Jen-tse Huang, Wenxuan Wang, Eric John Li, Man Ho Lam, Shujie Ren, Youliang Yuan, Wenxiang Jiao, Zhaopeng Tu, and Michael R. Lyu. On the humanity of conversational AI: evaluating the psychological portrayal of llms. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024.

[340] Zaibin Zhang, Yongting Zhang, Lijun Li, Jing Shao, Hongzhi Gao, Yu Qiao, Lijun Wang, Huchuan Lu, and Feng Zhao. Psysafe: A comprehensive framework for psychological-based attack, defense, and evaluation of multi-agent system safety. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 15202–15231. Association for Computational Linguistics, 2024.

[341] Wei Zhao, Zhe Li, Yige Li, Ye Zhang, and Jun Sun. Defending large language models against jailbreak attacks via layer-specific editing. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen, editors, *Findings of the Association for Computational Linguistics: EMNLP 2024, Miami, Florida, USA, November 12-16, 2024*, pages 5094–5109. Association for Computational Linguistics, 2024.

[342] Mengru Wang, Ningyu Zhang, Ziwen Xu, Zekun Xi, Shumin Deng, Yunzhi Yao, Qishen Zhang, Linyi Yang, Jindong Wang, and Huajun Chen. Detoxifying large language models via knowledge editing. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 3093–3118. Association for Computational Linguistics, 2024.

[343] Weikai Lu, Ziqian Zeng, Jianwei Wang, Zhengdong Lu, Zelin Chen, Huiping Zhuang, and Cen Chen. Eraser: Jailbreaking defense in large language models via unlearning harmful knowledge. *CoRR*, abs/2404.05880, 2024.

[344] Guobin Shen, Dongcheng Zhao, Yiting Dong, Xiang He, and Yi Zeng. Jailbreak antidote: Runtime safety-utility balance via sparse representation adjustment in large language models. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025.

[345] Andy Zou, Long Phan, Justin Wang, Derek Duenas, Maxwell Lin, Maksym Andriushchenko, J. Zico Kolter, Matt Fredrikson, and Dan Hendrycks. Improving alignment and robustness with circuit breakers. In Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang, editors, *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024.

[346] Heegyu Kim, Sehyun Yuk, and Hyunsouk Cho. Break the breakout: Reinventing LM defense against jailbreak attacks with self-refinement. *CoRR*, abs/2402.15180, 2024.

[347] Yuhui Li, Fangyun Wei, Jinjing Zhao, Chao Zhang, and Hongyang Zhang. RAIN: your language models can align themselves without finetuning. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024.

[348] Yifan Zeng, Yiran Wu, Xiao Zhang, Huazheng Wang, and Qingyun Wu. Autodefense: Multi-agent LLM defense against jailbreak attacks. *CoRR*, abs/2403.04783, 2024.

[349] Caishuang Huang, Wanxu Zhao, Rui Zheng, Huijie Lv, Shihan Dou, Sixian Li, Xiao Wang, Enyu Zhou, Junjie Ye, Yuming Yang, Tao Gui, Qi Zhang, and Xuanjing Huang. Safealigner: Safety alignment against jailbreak attacks via response disparity guidance. *CoRR*, abs/2406.18118, 2024.

[350] Zhangchen Xu, Fengqing Jiang, Luyao Niu, Jinyuan Jia, Bill Yuchen Lin, and Radha Poovendran. Safedecoding: Defending against jailbreak attacks via safety-aware decoding. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 5587–5605. Association for Computational Linguistics, 2024.

[351] Edoardo Debenedetti, Javier Rando, Daniel Paleka, Silaghi Fineas Florin, Dragos Albastroiu, Niv Cohen, Yuval Lemberg, Reshmi Ghosh, Rui Wen, Ahmed Salem, Giovanni Cherubin, Santiago Zanella-Béguelin, Robin Schmid, Victor Klemm, Takahiro Miki, Chenhao Li, Stefan Kraft, Mario Fritz, Florian Tramèr, Sahar Abdelnabi, and Lea Schönherr. Dataset and lessons learned from the 2024 satml LLM capture-the-flag competition. In Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang, editors, *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024.

[352] Daniil Khomsky, Narek Maloyan, and Bulat Nutfullin. Prompt injection attacks in defended systems. *CoRR*, abs/2406.14048, 2024.

[353] Jiawei Zhao, Kejiang Chen, Xiaojian Yuan, and Weiming Zhang. Prefix guidance: A steering wheel for large language models to defend against jailbreak attacks. *CoRR*, abs/2408.08924, 2024.

[354] Huijie Lv, Xiao Wang, Yuansen Zhang, Caishuang Huang, Shihan Dou, Junjie Ye, Tao Gui, Qi Zhang, and Xuanjing Huang. Codechameleon: Personalized encryption framework for jailbreaking large language models. *CoRR*, abs/2402.16717, 2024.

[355] Bin Ren, Yawei Li, Nancy Mehta, Radu Timofte, Hongyuan Yu, Cheng Wan, Yuxin Hong, Bingnan Han, Zhuoyuan Wu, Yajun Zou, Yuqing Liu, Jizhe Li, Keji He, Chao Fan, Heng Zhang, Xiaolin Zhang, Xuanwu Yin, Kunlong Zuo, Bohao Liao, Peizhe Xia, Long Peng, Zhibo Du, Xin Di, Wangkai Li, Yang Wang, Wei Zhai, Renjing Pei, Jiaming Guo, Songcen Xu, Yang Cao, Zhengjun Zha, Yan Wang, Yi Liu, Qing Wang, Gang Zhang, Liou Zhang, Shijie Zhao, Long Sun, Jinshan Pan, Jiangxin Dong, Jinhui Tang, Xin Liu, Min Yan, Qian Wang, Menghan Zhou, Yiqiang Yan, Yixuan Liu, Wensong Chan, Dehua Tang, Dong Zhou, Li Wang, Lu Tian, Emad Barsoum, Bohan Jia, Junbo Qiao, Yunshuai Zhou, Yun Zhang, Wei Li, Shaohui Lin, Shenglong Zhou, Binbin Chen, Jincheng Liao, Suiyi Zhao, Zhao Zhang, Bo Wang, Yan Luo, Yanyan Wei, Feng Li, Mingshen Wang, Yawei Li, Jinhan Guan, Dehua Hu, Jiawei Yu, Qisheng Xu, Tao Sun,

Long Lan, Kele Xu, Xin Lin, Jingtong Yue, Lehan Yang, Shiyi Du, Lu Qi, Chao Ren, Zeyu Han, Yuhan Wang, Chaolin Chen, Haobo Li, Mingjun Zheng, Zhongbao Yang, Lianhong Song, Xingzhuo Yan, Minghan Fu, Jingyi Zhang, Baiang Li, Qi Zhu, Xiaogang Xu, Dan Guo, Chunle Guo, Jiadi Chen, Huanhuan Long, Chunjiang Duanmu, Xiaoyan Lei, Jie Liu, Weilin Jia, Weifeng Cao, Wenlong Zhang, Yanyu Mao, Ruilong Guo, Nihao Zhang, Manoj Pandey, Maksym Chernozhukov, Giang Le, Shuli Cheng, Hongyuan Wang, Ziyan Wei, Qingting Tang, Liejun Wang, Yongming Li, Yanhui Guo, Hao Xu, Akram Khatami-Rizi, Ahmad Mahmoudi-Aznaveh, Chih-Chung Hsu, Chia-Ming Lee, Yi-Shiuan Chou, Amogh Joshi, Nikhil Akalwadi, Sampada Malagi, Palani Yashaswini, Chaitra Desai, Ramesh Ashok Tabib, Ujwala Patil, and Uma Mudenagudi. The ninth NTIRE 2024 efficient super-resolution challenge report. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2024 - Workshops, Seattle, WA, USA, June 17-18, 2024*, pages 6595–6631. IEEE, 2024.

[356] Xiaohu Du, Fan Mo, Ming Wen, Tu Gu, Huadi Zheng, Hai Jin, and Jie Shi. Multi-turn jailbreaking large language models via attention shifting. In Toby Walsh, Julie Shah, and Zico Kolter, editors, *AAAI-25, Sponsored by the Association for the Advancement of Artificial Intelligence, February 25 - March 4, 2025, Philadelphia, PA, USA*, pages 23814–23822. AAAI Press, 2025.

[357] Zheng Xin Yong, Cristina Menghini, and Stephen H. Bach. Low-resource languages jailbreak GPT-4. *CoRR*, abs/2310.02446, 2023.

[358] Yu Fu, Yufei Li, Wen Xiao, Cong Liu, and Yue Dong. Safety alignment in NLP tasks: Weakly aligned summarization as an in-context attack. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 8483–8502. Association for Computational Linguistics, 2024.

[359] Zhifan Sun and Antonio Valerio Miceli Barone. Scaling behavior of machine translation with large language models under prompt injection attacks. *CoRR*, abs/2403.09832, 2024.

[360] Qiusi Zhan, Zhixiang Liang, Zifan Ying, and Daniel Kang. Injecagent: Benchmarking indirect prompt injections in tool-integrated large language model agents. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Findings of the Association for Computational Linguistics, ACL 2024, Bangkok, Thailand and virtual meeting, August 11-16, 2024*, pages 10471–10506. Association for Computational Linguistics, 2024.

[361] Yupeng Ren. F2A: an innovative approach for prompt injection by utilizing feign security detection agents. *CoRR*, abs/2410.08776, 2024.

[362] Fredrik Nestaas, Edoardo Debenedetti, and Florian Tramèr. Adversarial search engine optimization for large language models. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025.

[363] Yi Zeng, Hongpeng Lin, Jingwen Zhang, Diyi Yang, Ruoxi Jia, and Weiyan Shi. How johnny can persuade llms to jailbreak them: Rethinking persuasion to challenge AI safety by humanizing llms. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 14322–14350. Association for Computational Linguistics, 2024.

[364] Yifan Jiang, Kriti Aggarwal, Tanmay Laud, Kashif Munir, Jay Pujara, and Subhabrata Mukherjee. RED QUEEN: safeguarding large language models against concealed multi-turn jailbreaking. *CoRR*, abs/2409.17458, 2024.

[365] Peng Ding, Jun Kuang, Dan Ma, Xuezhi Cao, Yunsen Xian, Jiajun Chen, and Shujian Huang. A wolf in sheep's clothing: Generalized nested jailbreak prompts can fool large language models easily. In Kevin Duh, Helena Gómez-Adorno, and Steven Bethard, editors, *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), NAACL 2024, Mexico City, Mexico, June 16-21, 2024*, pages 2136–2153. Association for Computational Linguistics, 2024.

[366] Xiaotian Zou and Yongkang Chen. Image-to-text logic jailbreak: Your imagination can help you do anything. *CoRR*, abs/2407.02534, 2024.

[367] Huiyu Xu, Wenhui Zhang, Zhibo Wang, Feng Xiao, Rui Zheng, Yunhe Feng, Zhongjie Ba, and Kui Ren. Redagent: Red teaming large language models with context-aware autonomous language agent. *CoRR*, abs/2407.16667, 2024.

[368] Sonali Singh, Faranak Abri, and Akbar Siami Namin. Exploiting large language models (llms) through deception techniques and persuasion principles. In Jingrui He, Themis Palpanas, Xiaohua Hu, Alfredo Cuzzocrea, Dejing Dou, Dominik Slezak, Wei Wang, Aleksandra Gruca, Jerry Chun-Wei Lin, and Rakesh Agrawal, editors, *IEEE International Conference on Big Data, BigData 2023, Sorrento, Italy, December 15-18, 2023*, pages 2508–2517. IEEE, 2023.

[369] Zeming Wei, Yifei Wang, and Yisen Wang. Jailbreak and guard aligned language models with only few in-context demonstrations. *CoRR*, abs/2310.06387, 2023.

[370] Jiongxiao Wang, Zichen Liu, Keun Hee Park, Muhao Chen, and Chaowei Xiao. Adversarial demonstration attacks on large language models. *CoRR*, abs/2305.14950, 2023.

[371] Cem Anil, Esin Durmus, Nina Panickssery, Mrinank Sharma, Joe Benton, Sandipan Kundu, Joshua Batson, Meg Tong, Jesse Mu, Daniel Ford, Francesco Mosconi, Rajashree Agrawal, Rylan Schaeffer, Naomi Bashkansky, Samuel Svenningsen, Mike Lambert, Ansh Radhakrishnan, Carson Denison, Evan Hubinger, Yuntao Bai, Trenton Bricken, Timothy Maxwell, Nicholas Schiefer, James Sully, Alex Tamkin, Tamera Lanham, Karina Nguyen, Tomek Korbak, Jared Kaplan, Deep Ganguli, Samuel R. Bowman, Ethan Perez, Roger B. Grosse, and David Kristjanson Duvenaud. Many-shot jailbreaking. In Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang, editors, *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024.

[372] Xiangyu Qi, Kaixuan Huang, Ashwinee Panda, Mengdi Wang, and Prateek Mittal. Visual adversarial examples jailbreak large language models. *CoRR*, abs/2306.13213, 2023.

[373] Boyi Deng, Wenjie Wang, Fuli Feng, Yang Deng, Qifan Wang, and Xiangnan He. Attack prompt generation for red teaming and defending large language models. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore, December 6-10, 2023*, pages 2176–2189. Association for Computational Linguistics, 2023.

[374] Haoran Li, Dadi Guo, Wei Fan, Mingshi Xu, Jie Huang, Fanpu Meng, and Yangqiu Song. Multi-step jailbreaking privacy attacks on chatgpt. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore, December 6-10, 2023*, pages 4138–4153. Association for Computational Linguistics, 2023.

[375] Zonghao Ying, Aishan Liu, Tianyuan Zhang, Zhengmin Yu, Siyuan Liang, Xianglong Liu, and Dacheng Tao. Jailbreak vision language models via bi-modal adversarial prompt. *CoRR*, abs/2406.04031, 2024.

[376] Erik Jones, Anca D. Dragan, and Jacob Steinhardt. Adversaries can misuse combinations of safe models. *CoRR*, abs/2406.14595, 2024.

[377] Gianluca De Stefano, Lea Schönherr, and Giancarlo Pellegrino. Rag and roll: An end-to-end evaluation of indirect prompt manipulations in llm-based application frameworks. *CoRR*, abs/2408.05025, 2024.

[378] Xuan Li, Zhanke Zhou, Jianing Zhu, Jiangchao Yao, Tongliang Liu, and Bo Han. Deepinception: Hypnotize large language model to be jailbreaker. *CoRR*, abs/2311.03191, 2023.

[379] Yingchaojie Feng, Zhizhang Chen, Zhining Kang, Sijia Wang, Minfeng Zhu, Wei Zhang, and Wei Chen. Jailbreaklens: Visual analysis of jailbreak attacks against large language models. *CoRR*, abs/2404.08793, 2024.

[380] Yuanwei Wu, Xiang Li, Yixin Liu, Pan Zhou, and Lichao Sun. Jailbreaking GPT-4V via self-adversarial attacks with system prompts. *CoRR*, abs/2311.09127, 2023.

[381] Yi Liu, Gelei Deng, Yuekang Li, Kailong Wang, Tianwei Zhang, Yepang Liu, Haoyu Wang, Yan Zheng, and Yang Liu. Prompt injection attack against llm-integrated applications. *CoRR*, abs/2306.05499, 2023.

[382] Dongyu Yao, Jianshu Zhang, Ian G. Harris, and Marcel Carlsson. Fuzzllm: A novel and universal fuzzing framework for proactively discovering jailbreak vulnerabilities in large language models. *CoRR*, abs/2309.05274, 2023.

[383] Abhinav Rao, Sachin Vashistha, Atharva Naik, Somak Aditya, and Monojit Choudhury. Tricking llms into disobedience: Understanding, analyzing, and preventing jailbreaks. *CoRR*, abs/2305.14965, 2023.

[384] Chengyuan Liu, Fubang Zhao, Lizhi Qing, Yangyang Kang, Changlong Sun, Kun Kuang, and Fei Wu. Goal-oriented prompt attack and safety evaluation for llms. *arXiv preprint arXiv:2309.11830*, 2023.

[385] Shangqing Tu, Zhuoran Pan, Wenxuan Wang, Zhexin Zhang, Yuliang Sun, Jifan Yu, Hongning Wang, Lei Hou, and Juanzi Li. Knowledge-to-jailbreak: One knowledge point worth one attack. *CoRR*, abs/2406.11682, 2024.

[386] Somnath Banerjee, Sayan Layek, Rima Hazra, and Animesh Mukherjee. How (un)ethical are instruction-centric responses of llms? unveiling the vulnerabilities of safety guardrails to harmful queries. *CoRR*, abs/2402.15302, 2024.

[387] Moussa Koulako Bala Doumbouya, Ananjan Nandi, Gabriel Poesia, Davide Ghilardi, Anna Goldie, Federico Bianchi, Dan Jurafsky, and Christopher D. Manning. h4rm3l: A dynamic benchmark of composable jailbreak attacks for LLM safety assessment. *CoRR*, abs/2408.04811, 2024.

[388] Lijia Lv, Weigang Zhang, Xuehai Tang, Jie Wen, Feng Liu, Jizhong Han, and Songlin Hu. Adappa: Adaptive position pre-fill jailbreak attack approach targeting llms. *CoRR*, abs/2409.07503, 2024.

[389] Alexander Wei, Nika Haghtalab, and Jacob Steinhardt. Jailbroken: How does LLM safety training fail? In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine, editors, *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023.

[390] Delong Ran, Jinyuan Liu, Yichen Gong, Jingyi Zheng, Xinlei He, Tianshuo Cong, and Anyu Wang. Jailbreakeval: An integrated toolkit for evaluating jailbreak attempts against large language models. *CoRR*, abs/2406.09321, 2024.

[391] Maksym Andriushchenko and Nicolas Flammarion. Does refusal training in llms generalize to the past tense? *CoRR*, abs/2407.11969, 2024.

[392] Xinyue Shen, Zeyuan Chen, Michael Backes, Yun Shen, and Yang Zhang. "do anything now": Characterizing and evaluating in-the-wild jailbreak prompts on large language models. In Bo Luo, Xiaojing Liao, Jun Xu, Engin Kirda, and David Lie, editors, *Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security, CCS 2024, Salt Lake City, UT, USA, October 14-18, 2024*, pages 1671–1685. ACM, 2024.

[393] Anay Mehrotra, Manolis Zampetakis, Paul Kassianik, Blaine Nelson, Hyrum S. Anderson, Yaron Singer, and Amin Karbasi. Tree of attacks: Jailbreaking black-box llms automatically. In Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang, editors, *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024.

[394] Fan Liu, Yue Feng, Zhao Xu, Lixin Su, Xinyu Ma, Dawei Yin, and Hao Liu. JAILJUDGE: A comprehensive jailbreak judge benchmark with multi-agent enhanced explanation evaluation framework. *CoRR*, abs/2410.12855, 2024.

[395] Weidi Luo, Siyuan Ma, Xiaogeng Liu, Xiaoyu Guo, and Chaowei Xiao. Jailbreakv-28k: A benchmark for assessing the robustness of multimodal large language models against jailbreak attacks. *CoRR*, abs/2404.03027, 2024.

[396] Patrick Chao, Alexander Robey, Edgar Dobriban, Hamed Hassani, George J. Pappas, and Eric Wong. Jailbreaking black box large language models in twenty queries. *CoRR*, abs/2310.08419, 2023.

[397] Maksym Andriushchenko, Francesco Croce, and Nicolas Flammarion. Jailbreaking leading safety-aligned llms with simple adaptive attacks. *CoRR*, abs/2404.02151, 2024.

[398] Jiaming Ji, Mickel Liu, Josef Dai, Xuehai Pan, Chi Zhang, Ce Bian, Boyuan Chen, Ruiyang Sun, Yizhou Wang, and Yaodong Yang. Beavertails: Towards improved safety alignment of LLM via a human-preference dataset. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine, editors, *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023.

[399] Mantas Mazeika, Long Phan, Xuwang Yin, Andy Zou, Zifan Wang, Norman Mu, Elham Sakhaee, Nathaniel Li, Steven Basart, Bo Li, David A. Forsyth, and Dan Hendrycks. Harmbench: A standardized evaluation framework for automated red teaming and robust refusal. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net, 2024.

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

[401] Chuan Guo, Alexandre Sablayrolles, Hervé Jégou, and Douwe Kiela. Gradient-based adversarial attacks against text transformers. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih, editors, *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 5747–5757. Association for Computational Linguistics, 2021.

[402] Eric Wallace, Shi Feng, Nikhil Kandpal, Matt Gardner, and Sameer Singh. Universal adversarial triggers for attacking and analyzing NLP. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan, editors, *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 2153–2162. Association for Computational Linguistics, 2019.

[403] Taylor Shin, Yasaman Razeghi, Robert L. Logan IV, Eric Wallace, and Sameer Singh. Autoprompt: Eliciting knowledge from language models with automatically generated prompts. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu, editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 4222–4235. Association for Computational Linguistics, 2020.

[404] Gelei Deng, Yi Liu, Yuekang Li, Kailong Wang, Ying Zhang, Zefeng Li, Haoyu

Wang, Tianwei Zhang, and Yang Liu. Masterkey: Automated jailbreak across multiple large language model chatbots. *arXiv preprint arXiv:2307.08715*, 2023.

[405] Teknium. Openhermes 2.5: An open dataset of synthetic data for generalist llm assistants, 2023. Available at: https://huggingface.co/datasets/teknium/OpenHermes-2.5.

[406] Glaive AI. glaive-code-assistant, 2023. Available at: https://huggingface.co/datasets/glaiveai/glaive-code-assistant.

[407] CAMEL-AI. Camelai domain expert datasets (physics, math, chemistry & biology), 2023. Available at: https://huggingface.co/camel-ai.

[408] WizardLMTeam. Wizardlm_evol_instruct_70k, 2023. Available at: https://huggingface.co/datasets/WizardLMTeam/WizardLM_evol_instruct_70k.

[409] Crystalcareai. alpaca-gpt4-cot, 2024. Available at: https://huggingface.co/datasets/Crystalcareai/alpaca-gpt4-COT.

[410] Longhui Yu, Weisen Jiang, Han Shi, Jincheng Yu, Zhengying Liu, Yu Zhang, James T. Kwok, Zhenguo Li, Adrian Weller, and Weiyang Liu. Metamath: Bootstrap your own mathematical questions for large language models. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024.

[411] jondurbin. airoboros-2.2, 2023. Available at: https://huggingface.co/datasets/jondurbin/airoboros-2.2.

[412] Ariel N. Lee, Cole J. Hunter, and Nataniel Ruiz. Platypus: Quick, cheap, and powerful refinement of llms. *CoRR*, abs/2308.07317, 2023.

[413] Baolin Peng, Chunyuan Li, Pengcheng He, Michel Galley, and Jianfeng Gao. Instruction tuning with GPT-4. *CoRR*, abs/2304.03277, 2023.

[414] CogStack. Cogstack, 2023. Available at: https://github.com/CogStack/OpenGPT.

[415] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Tianle Li, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zhuohan Li, Zi Lin, Eric P. Xing, Joseph E. Gonzalez, Ion Stoica, and Hao Zhang. Lmsys-chat-1m: A large-scale real-world LLM conversation dataset. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024.

[416] Shuya Feng, Meisam Mohammady, Han Wang, Xiaochen Li, Zhan Qin, and Yuan Hong. DPI: ensuring strict differential privacy for infinite data streaming. In *IEEE Symposium on Security and Privacy, SP 2024, San Francisco, CA, USA, May 19-23, 2024*, pages 1009–1027. IEEE, 2024.

[417] Hanbin Hong, Shenao Yan, Shuya Feng, Yan Yan, and Yuan Hong. GALOT: generative active learning via optimizable zero-shot text-to-image generation. *CoRR*, abs/2412.16227, 2024.

[418] Canyu Chen, Yueqing Liang, Xiongxiao Xu, Shangyu Xie, Yuan Hong, and Kai Shu. On fair classification with mostly private sensitive attributes. *CoRR*, abs/2207.08336, 2022.

[419] Han Wang, Hanbin Hong, Li Xiong, Zhan Qin, and Yuan Hong. L-SRR: local differential privacy for location-based services with staircase randomized response. In Heng Yin, Angelos Stavrou, Cas Cremers, and Elaine Shi, editors, *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security, CCS 2022, Los Angeles, CA, USA, November 7-11, 2022*, pages 2809–2823. ACM, 2022.

[420] Shuya Feng, Meisam Mohammady, Hanbin Hong, Shenao Yan, Ashish Kundu, Binghui Wang, and Yuan Hong. Harmonizing differential privacy mechanisms for federated learning: Boosting accuracy and convergence. In *Proceedings of the Fifteenth ACM Conference on Data and Application Security and Privacy, CODASPY 2025, Pittsburgh, PA, USA, June 4-6, 2025*, pages 60–71. ACM, 2025.

[421] Fereshteh Razmi, Jian Lou, Yuan Hong, and Li Xiong. Interpretation attacks and defenses on predictive models using electronic health records. In *Machine Learning and Knowledge Discovery in Databases: Research Track - European Conference, ECML PKDD 2023, Turin, Italy, September 18-22, 2023, Proceedings, Part III*, volume 14171 of *Lecture Notes in Computer Science*, pages 446–461. Springer, 2023.

[422] Aryeh Dvoretzky, Jack Kiefer, and Jacob Wolfowitz. Asymptotic minimax character of the sample distribution function and of the classical multinomial estimator. *The Annals of Mathematical Statistics*, pages 642–669, 1956.

[423] Meisam Mohammady, Shangyu Xie, Yuan Hong, Mengyuan Zhang, Lingyu Wang, Makan Pourzandi, and Mourad Debbabi. R2dp: A universal and automated approach to optimizing the randomization mechanisms of differential privacy for utility metrics with no known optimal distributions. In *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*, pages 677–696, 2020.

[424] Shasha Li, Ajaya Neupane, Sujoy Paul, Chengyu Song, Srikanth V. Krishnamurthy, Amit K. Roy-Chowdhury, and Ananthram Swami. Stealthy adversarial perturbations against real-time video classification systems. In *NDSS*. The Internet Society, 2019.

[425] Yuxin Yang, Qiang Li, Jinyuan Jia, Yuan Hong, and Binghui Wang. Distributed backdoor attacks on federated graph learning and certified defenses. In *Proceedings of the 2024 ACM SIGSAC Conference on Computer and Communications Security*, pages 2829–2843.

[426] Robin Jia, Aditi Raghunathan, Kerem Göksel, and Percy Liang. Certified robustness to adversarial word substitutions. In *EMNLP/IJCNLP*, 2019.

[427] Robert G Bartle. *The elements of integration and Lebesgue measure*. John Wiley & Sons, 2014.

[428] Shahnawaz Ahmed and Elias G Saleeby. On volumes of hyper-ellipsoids. *Mathematics Magazine*, 2018.

[429] Joseph P Campbell. Speaker recognition: A tutorial. *Proceedings of the IEEE*, 85(9):1437–1462, 1997.

[430] Brecht Desplanques, Jenthe Thienpondt, and Kris Demuynck. Ecapa-tdnn: Emphasized channel attention, propagation and aggregation in tdnn based speaker verification. 2020.

[431] Mirco Ravanelli, Titouan Parcollet, Peter Plantinga, Aku Rouhe, Samuele Cornell, Loren Lugosch, Cem Subakan, Nauman Dawalatabad, Abdelwahab Heba, Jianyuan Zhong, Ju-Chieh Chou, Sung-Lin Yeh, Szu-Wei Fu, Chien-Feng Liao, Elena Rastorgueva, François Grondin, William Aris, Hwidong Na, Yan Gao, Renato De Mori, and Yoshua Bengio. SpeechBrain: A general-purpose speech toolkit, 2021.

[432] Daniel Garcia-Romero, David Snyder, Gregory Sell, Alan McCree, Daniel Povey, and Sanjeev Khudanpur. x-vector dnn refinement with full-length recordings for speaker recognition. In *Interspeech*, pages 1493–1496, 2019.

[433] A. Nagrani, J. S. Chung, and A. Zisserman. Voxceleb: a large-scale speaker identification dataset. In *INTERSPEECH*, 2017.

[434] J. S. Chung, A. Nagrani, and A. Zisserman. Voxceleb2: Deep speaker recognition. In *INTERSPEECH*, 2018.

Chapter 7

Appendix

7.1 Preliminary

We first briefly review the recent certified robustness scheme [1] for a general classification problem by classifying data point in $\mathbb{R}^d$ to classes in $\mathcal{Y}$. Given an arbitrary base classifier f, it can be converted to a "smoothed" classifier [1] g by adding isotropic Gaussian noise to the input x:

$$g(x) = \arg\max_{c \in \mathcal{Y}} \mathbb{P}(f(x+\epsilon) = c), where \ \ \epsilon \sim \mathcal{N}(0, \sigma^2 I) \tag{7.1}$$

Lemma 1. *(**Neyman-Pearson Lemma**) Let X and Y be random variables in $\mathbb{R}^d$ with densities μ_X and μ_Y. Let $f : \mathbb{R}^d \to \{0, 1\}$ be a random or deterministic function. Then:*

(1) If $S = \{z \in \mathbb{R}^d : \frac{\mu_Y(z)}{\mu_X(z)} \leq t\}$ for some $t > 0$ and $\mathbb{P}(f(X) = 1) \geq \mathbb{P}(X \in S)$, then $\mathbb{P}(f(Y) = 1) \geq \mathbb{P}(Y \in S)$;

(2) If $S = \{z \in \mathbb{R}^d : \frac{\mu_Y(z)}{\mu_X(z)} \geq t\}$ for some $t > 0$ and $\mathbb{P}(f(X) = 1) \leq \mathbb{P}(X \in S)$, then $\mathbb{P}(f(Y) = 1) \leq \mathbb{P}(Y \in S)$.

With Lemma 4, Cohen [1] derives the certified radius when the classifier is smoothed with the Gaussian noise. As shown in Theorem 8, when the smoothed classifier's prediction probabilities satisfy Equation (7.2), the prediction result is guaranteed to be the most probable class c_A when the perturbation is limited within a radius R in ℓ_2-norm.

Theorem 8. *(**Randomized Smoothing with Gaussian Noise [1]**) Let $f : \mathbb{R}^d \to \mathcal{Y}$ be any deterministic or random function, and let $\epsilon \sim \mathcal{N}(0, \sigma^2 I)$. Denote g as the smoothed classifier in Equation (7.1), and the most probable and the second probable classes as $c_A, c_B \in \mathcal{Y}$, respectively. If the lower bound of the class c_A's prediction probability $\underline{p_A} \in [0, 1]$, and the upper bound of the class c_B's prediction probability $\overline{p_B} \in [0, 1]$ satisfy:*

$$\mathbb{P}(f(x+\epsilon) = c_A) \geq \underline{p_A} \geq \overline{p_B} \geq \max_{c \neq c_A} \mathbb{P}(f(x+\epsilon) = c) \tag{7.2}$$

Then $g(x + \delta) = c_A$ for all $\|\delta\|_2 \leq R$, where

$$R = \frac{\sigma}{2}(\Phi^{-1}(\underline{p_A}) - \Phi^{-1}(\overline{p_B})) \tag{7.3}$$

where Φ^{-1} is the inverse of the standard Gaussian CDF.

Proof. See detailed proof in [1]. □

7.2 Proofs

7.2.1 Proof of Theorem 1

Proof. We prove the theorem based on Neyman-Pearson Lemma (Lemma 4).

Let $x := x_0 + \epsilon$ be the random variable that follows any continuous distribution. δ be the perturbation added to the input image. $y = x_0 + \epsilon + \delta$ is the perturbed random variable. Thus, x and y are random variables with densities μ_x and μ_y. Define sets:

$$A := \{z : \frac{\mu_y(z)}{\mu_x(z)} \leq t_A\} \tag{7.4}$$

$$B := \{z : \frac{\mu_y(z)}{\mu_x(z)} \geq t_B\} \tag{7.5}$$

where t_A and t_B are picked to suffice:

$$\mathbb{P}(x \in A) = \underline{p_A} \tag{7.6}$$

$$\mathbb{P}(x \in B) = \overline{p_B} \tag{7.7}$$

Suppose $c_A \in \mathcal{Y}$ and $\underline{p_A}, \overline{p_B} \in [0, 1]$ satisfy:

$$\mathbb{P}(f(x + \epsilon) = c_A) \geq \underline{p_A} \geq \overline{p_B} \geq \max_{c \neq c_A} \mathbb{P}(f(x + \epsilon) = c) \tag{7.8}$$

Since $\mathbb{P}(f(x + \epsilon) = c_A) \geq \underline{p_A} = \mathbb{P}(x \in A)$ and $A = \{z : \frac{\mu_Y(z)}{\mu_X(z)} \leq t_A\}$, using Neyman-Pearson Lemma (Lemma 4), we have:

$$\mathbb{P}(f(y) = c_A) \geq \mathbb{P}(y \in A) \tag{7.9}$$

Similarly, we have:

$$\mathbb{P}(f(y) = c_B) \leq \mathbb{P}(y \in B) \tag{7.10}$$

To guarantee $\mathbb{P}(f(y) = c_A) \geq \mathbb{P}(f(y) = c_B)$, we need

$$\mathbb{P}(f(y) = c_A) \geq \mathbb{P}(y \in A) \geq \mathbb{P}(y \in B) \geq \mathbb{P}(f(y) = c_B) \tag{7.11}$$

In summary, to guarantee the certified robustness on class A, Equation (7.4), (7.5), (7.6), (7.7), (7.11) must be satisfied. The conditions can be rewritten as:

$$\mathbb{P}(\frac{\mu_y(x)}{\mu_x(x)} \leq t_A) = \underline{p_A} \tag{7.12}$$

$$\mathbb{P}(\frac{\mu_y(x)}{\mu_x(x)} \geq t_B) = \overline{p_B} \tag{7.13}$$

$$\mathbb{P}(\frac{\mu_y(y)}{\mu_x(y)} \leq t_A) \geq \mathbb{P}(\frac{\mu_y(y)}{\mu_x(y)} \geq t_B) \tag{7.14}$$

where Equation (7.12) is from Equation (7.4) and Equation (7.6), Equation (7.13) is from Equation (7.5) and Equation (7.7), and Equation (7.14) is from Equation (7.11).

Considering the relationship $y = x + \delta$, we can derive:

$$\mu_y(x) = \mu_x(x - \delta) \tag{7.15}$$

$$\mu_x(y) = \mu_x(x + \delta) \tag{7.16}$$

$$\mu_y(y) = \mu_x(y - \delta) = \mu_x(x) \tag{7.17}$$

Thus, the conditions (11), (12) and (13) can be rewritten as:

$$\mathbb{P}(\frac{\mu_x(x - \delta)}{\mu_x(x)} \leq t_A) = \underline{p_A} \tag{7.18}$$

$$\mathbb{P}(\frac{\mu_x(x-\delta)}{\mu_x(x)} \geq t_B) = \overline{p_B} \tag{7.19}$$

$$\mathbb{P}(\frac{\mu_x(x)}{\mu_x(x+\delta)} \leq t_A) \geq \mathbb{P}(\frac{\mu_x(x)}{\mu_x(x+\delta)} \geq t_B) \tag{7.20}$$

Any perturbation δ satisfying these conditions will not fool the smoothed classifier. In this case, these conditions construct a robustness area in δ space. If we want to find a ℓ_p ball within which the prediction is constant, the l_p ball should be in this robustness area. Therefore, the certified radii is the minimum $\|\delta\|_p$ on the boundary of this robustness area. In this case, the ℓ_p ball is exactly the maximum inscribed ball in the robustness area. Also, x can be replace by ϵ in these conditions since it is in the fraction, which means the optimization is independent to the input if given $\underline{p_A}$ and $\overline{p_B}$. Therefore, the whole optimization problem is summarized as:

$$\begin{aligned}
\underset{\delta}{\text{minimize}} \quad & R = \|\delta\|_p \\
\text{subject to} \quad & \mathbb{P}(\frac{\mu_x(x-\delta)}{\mu_x(x)} \leq t_A) = \underline{p_A}, \\
& \mathbb{P}(\frac{\mu_x(x-\delta)}{\mu_x(x)} \geq t_B) = \overline{p_B}, \\
& \mathbb{P}(\frac{\mu_x(x)}{\mu_x(x+\delta)} \leq t_A) = \mathbb{P}(\frac{\mu_x(x)}{\mu_x(x+\delta)} \geq t_B)
\end{aligned}$$

If the noise is isotropic, each dimension is independent,

$$\mu_x(x) = \prod_{i=1}^{d} \mu_x(x_j) \tag{7.21}$$

Thus, conditions for the isotropic noise can be rewritten as:

$$\mathbb{P}(\prod_{j=1}^{d} \frac{\mu_x(x_j-\delta_j)}{\mu_x(x_j)} \leq t_A) = \underline{p_A} \tag{7.22}$$

$$\mathbb{P}(\prod_{j=1}^{d} \frac{\mu_x(x_j-\delta_j)}{\mu_x(x_j)} \geq t_B) = \overline{p_B} \tag{7.23}$$

$$\mathbb{P}(\prod_{j=1}^{d} \frac{\mu_x(x_j)}{\mu_x(x_j+\delta_j)} \leq t_A) = \mathbb{P}(\prod_{j=1}^{d} \frac{\mu_x(x_j)}{\mu_x(x_j+\delta_j)} \geq t_B) \tag{7.24}$$

Thus, this completes the proof. □

7.2.2 Binary Case for Theorem 2

Theorem 9. ***(Universal Certified Robustness (Binary Case))*** *Let $f : \mathbb{R}^d \to \mathcal{Y}$ be any deterministic or random function, and let ϵ follows any continuous distribution. Let g be defined as in (7.1). Suppose the most probable class $c_A \in \mathcal{Y}$ and the lower bound of the probability $\underline{p_A}$ satisfy:*

$$\mathbb{P}(f+\epsilon) = c_A \geq \underline{p_A} \geq \frac{1}{2} \tag{7.25}$$

Then $g(x+\delta) = c_A$ for all $\|\delta\|_p \leq R$, where R is given by the optimization:

$$\begin{aligned} \underset{\delta}{\text{minimize}} \quad & R = \|\delta\|_p \\ \text{subject to} \quad & \mathbb{P}(\frac{\mu_x(x-\delta)}{\mu_x(x)} \leq t_A) = \underline{p_A}, \\ & \mathbb{P}(\frac{\mu_x(x)}{\mu_x(x+\delta)} \leq t_A) = \frac{1}{2} \end{aligned}$$

7.2.3 UniCR (Binary Case)

Similar to the binary case of Theorem 9, the binary case of the two-phase optimization can be easily derived:

$$\begin{aligned} & R = \|\lambda\boldsymbol{\delta}\|_p, \textit{where } \boldsymbol{\delta} \in \arg\min_{\delta} \|\lambda\boldsymbol{\delta}\|_p \\ & s.t. \quad \lambda = \arg\min_{\lambda} |K| \\ & \qquad \mathbb{P}(\frac{\mu_x(x-\lambda\boldsymbol{\delta})}{\mu_x(x)} \leq t_A) = \underline{p_A} \\ & \qquad K = \mathbb{P}(\frac{\mu_x(x)}{\mu_x(x+\lambda\boldsymbol{\delta})} \leq t_A) - \frac{1}{2} \\ & \qquad \underline{p_A} \geq \frac{1}{2} \end{aligned}$$

7.2.4 UniCR Bound

The certified radius R approximated by the two-phase optimization is tight if achieving the optimality. Under this assumption, we analysis the confidence bound for the certification. We follow [1] to compute the probabilities $\underline{p_A}$ and $\overline{p_B}$ using Monte Carlo method with sample number n. The confidence is $1-\alpha_0$, where $p_A >= \alpha_0^{1/n}$. To estimate the auxiliary parameters t_A and t_B, we use Dvoretzky–Kiefer–Wolfowitz inequality [422] to bound the CDFs of the random variables $\frac{\mu_x(x-\lambda\delta)}{\mu_x(x)}$ and $\frac{\mu_x(x)}{\mu_x(x+\lambda\delta)}$, then determine the t_A and t_B using Algorithm 10.

Lemma 2. ***(Dvoretzky–Kiefer–Wolfowitz inequality****(restate)) Let $X_1, X_2, ..., X_n$ be real-valued independent and identically distributed random variables with cumulative distribution function $F(\cdot)$, where $n \in \mathbb{N}$.Let F_n denotes the associated empirical distribution function defined by*

$$F_n(x) = \frac{1}{n}\sum_{i=1}^{n} \mathbf{1}_{\{X_i <= x\}}, x \in \mathbb{R} \tag{7.26}$$

The Dvoretzky–Kiefer–Wolfowitz inequality bounds the probability that the random function F_n differs from F by more than a given constant $\Delta \in \mathbb{R}^+$:

$$\mathbb{P}(\sup_{x\in\mathbb{R}} |F_n(x) - F(x)| > \Delta) \leq 2e^{-2n\Delta^2} \tag{7.27}$$

We use the Lemma 5 to estimate the CDFs in algorithm 10. In the robustness condition, we need to estimate 4 probabilities with confidence $1-2e^{-2n\Delta^2}$ as well as the p_A and p_B with confidence $1-\alpha_0$. Therefore, the confidence that deriving the correct radius is at least $(1-\alpha_0)^2(1-2e^{-2n\Delta^2})^4$. In Figure 7.1,

we show the confidence on a varying number of samples when $\Delta = 0.1$ and $\alpha_0 = 0.999$. As the number of samples increases to around 400 (all our experiments use more than 400 samples), our confidence is very close to Cohen's confidence [1]. Thus, the confidence is nearly 1 in all our experiments.

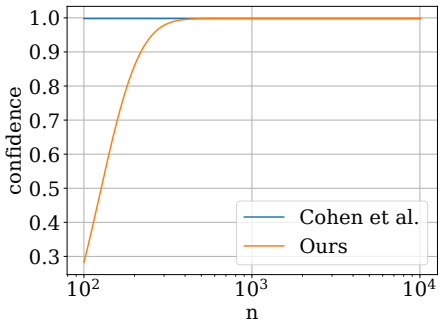


Fig. 7.1: Confidence vs. number of Monte Carlo samples.

7.2.5 Optimization Convergence

We analysis the convergence of the two-phase optimization and the certification accuracy in this section. On one hand, the optimality of the scalar optimization can be asymptotically achieved by binary search. On the other hand, it is hard to find the minimum $\|\lambda\delta\|_p$ in the highly-dimensional space, but some special symmetry in the direction of δ (e.g., spherical symmetry that is also found in [3, 23]), can help approximate the certified radius. The detailed algorithms are presented in Section 3.1. The defense performance of such universally approximated certified robustness against different real-world attacks is the same as certified robustness (as shown in Appendix 7.4.2). Thus, such negligible approximation error is close to 0, but result in many significant new benefits in return.

7.3 Algorithms

7.3.1 Computing t_A and t_B

We present the algorithm to compute the p_A and p_B in Algorithm 10.

Algorithm 10 Computing t_A and t_B

Input: Lower bound of the probabilities, $\underline{p_A}$; upper bound of the probabilities, $\overline{p_B}$; perturbation scalar, λ; perturbation, δ; noise PDF, μ_x; number of samples, n (for Monte Carlo).

Output: The auxiliary parameters, t_A and t_B.

1: Sample n noise vectors $\epsilon \in \mathbb{R}^{n\times d}$ from a discrete version of the PDF μ_x.
2: Calculate $\frac{\mu_x(x-\lambda\delta)}{\mu_x(x)}$ for each noise sample using μ_x, λ, and δ.
3: Estimate the CDF Φ of $\frac{\mu_x(x-\lambda\delta)}{\mu_x(x)}$ using the Monte Carlo method.
4: **return** $t_A = \Phi^{-1}(p_A)$ and $t_B = \Phi^{-1}(p_B)$, using the inverse CDF Φ.

7.3.2 Scalar Optimization

We use the binary search to find a scale factor that minimizes $|K|$ (*the distance between δ and the robustness boundary*). When $K = 0$, the perturbation δ is exactly on the robustness boundary. Fixing the direction of δ, we find two scalars such that $K > 0$ and $K < 0$. Specifically, we start from a scalar λ_a and compute K. If $K > 0$, then the scaled perturbation $\lambda_a\delta$ is within the robustness boundary, thus we enlarge the scalar to find a λ_b such that $K < 0$ and vice versa. After that, we iteratively compute the K using $\lambda = \frac{1}{2}(\lambda_a + \lambda_b)$: if $K > 0$, we let $\lambda_a = \lambda$; otherwise, we let $\lambda_b = \lambda$. We repeat this iteration until K is less than a threshold or the number of iterations is sufficiently large. The procedures are summarized in Algorithm 11.

Algorithm 11 Scalar Optimization

Input: Lower bound of the probabilities, $\underline{p_A}$; upper bound of the probabilities, $\overline{p_B}$; perturbation scalar, λ; perturbation, δ; noise PDF, μ_x; number of samples in Monte Carlo method, n; threshold for K, K_m; number of iterations for binary search, N

Output: The scalar λ that minimizes $|K|$

Find initial scalars λ_a and λ_b such that $K(\lambda_a) > 0$ and $K(\lambda_b) < 0$
$\lambda \leftarrow \frac{\lambda_a+\lambda_b}{2}$
Compute K using λ
while $N > 0$ **and** $|K| > K_m$ **do**
    if $K > 0$ **then**
        $\lambda_a \leftarrow \lambda$
    else
        $\lambda_b \leftarrow \lambda$
    $\lambda \leftarrow \frac{\lambda_a+\lambda_b}{2}$
    Compute K using λ
    $N \leftarrow N - 1$
return λ

7.3.3 Direction Optimization

We show how to initialize the positions for different ℓ_p norms in PSO. Since some noise follows PDFs with symmetry [3, 23], we set the initial position of particles by considering this, e.g., setting the initial positions w.r.t. ℓ_p for $p \in \mathbb{R}^+$ as $[0, ..., 0, a, 0, ..., 0]$ and the initial positions w.r.t. ℓ_∞ as $[a, a, a, ..., a]$, where a is a small random number. Although the search space is highly-dimensional, empirical results show that the radius given by PSO can accurately approximate the theoretical radius given by other methods, e.g., Cohen's [1] (see Figure 4 in main context). Notice that, for more complicated PDFs without symmetry (which is indeed difficult for deriving the certified radius), PSO can also approximate the certified radius with more particles and iterations.

7.3.4 Noise Optimization Algorithms

With the UniCR for estimating the certified radius, we can further tune the noise PDF to each input or the classifier. Specifically, let $\mu(x, \boldsymbol{\alpha})$ denote the noise PDF, where $\boldsymbol{\alpha}$ is a set of hyper-parameters in the function, i.e., $\boldsymbol{\alpha} = [\alpha_1, \alpha_2, ..., \alpha_m]$. We simply use grid-search algorithm to find the best hyper-parameters in the classifier smoothing (C-OPT). For the input noise optimization (I-OPT), we use the Hill-Climbing

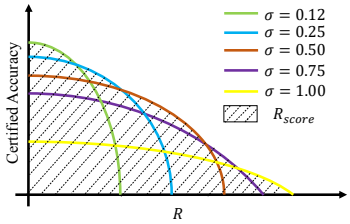


Fig. 7.2: An example of the Robustness Score.

algorithm to find the optimal hyper-parameters in the function for each input. During the algorithm execution, hyper-parameter for the input is iteratively updated if a better solution is found in each round, until convergence. The procedures for the Hill Climbing algorithm are summarized in Algorithm 12.

Algorithm 12 I-OPT with Hill Climbing

Input: Input data, x; PDF of noise distribution, μ_x; universally approximated certified robustness, UniCR$(\cdot)$; initial hyper-parameters $\boldsymbol{\alpha}$; optimization range of hyper-parameters, $[\boldsymbol{L}, \boldsymbol{H}]$; optimization step of hyper-parameters, $\boldsymbol{S}$.

Output: The optimal hyper-parameters, $\boldsymbol{\alpha}_{\text{optimal}}$.

Initialize the certified radius: $R_0 \leftarrow \text{UniCR}(x, \mu(\boldsymbol{\alpha}))$
for each hyper-parameter α_i in $\boldsymbol{\alpha}$ **do**
    if $L_i < \alpha_i + S_i < H_i$ **then**
        $R' \leftarrow \text{UniCR}\Big(x, \mu(\boldsymbol{\alpha} \mid \alpha_i = \alpha_i + S_i)\Big)$
        if $R' > R_0$ **then**
            Update $\boldsymbol{\alpha}$ with $\alpha_i \leftarrow \alpha_i + S_i$
            $R_0 \leftarrow R'$
        else if $L_i < \alpha_i - S_i < H_i$ **then**
            $R' \leftarrow \text{UniCR}\Big(x, \mu(\boldsymbol{\alpha} \mid \alpha_i = \alpha_i - S_i)\Big)$
            if $R' > R_0$ **then**
                Update $\boldsymbol{\alpha}$ with $\alpha_i \leftarrow \alpha_i - S_i$
                $R_0 \leftarrow R'$
        else
            break
    else
        break
return $\boldsymbol{\alpha}_{\text{optimal}} \leftarrow \boldsymbol{\alpha}$

7.4 More Experimental Results

7.4.1 Metrics

We show the illustration of Robustness Score in Figure. 7.2

7.4.2 Defense against Real Attacks

We evaluate our UniCR's defense accuracy against a diverse set of state-of-the-art attacks, including universal attacks [30], white-box attacks [31, 32], and black-box attacks [10, 33]. We compare UniCR with other state-of-the-art certified schemes [1, 3, 24] against ℓ_1, ℓ_2 and ℓ_∞ perturbations. The certified radius R for each image in the test set (10,000 images in total) are computed beforehand, and the perturbation generation is constrained by $\|\delta\|_p = R$ for all the attack methods. We define the defense accuracy as the rate that the smoothed classifier can successfully defend against the perturbations with the ℓ_p size identical to the the certified radius:

$$acc_d = \mathbb{E}_{\|\delta\|_p=R}[\frac{\sum g(x+\delta) = c_A}{N}] \tag{7.28}$$

where $c_A = g(x)$, N is the total test number. In this defense study, we use 500 samples for both Monte Carlo method and testing.

Table 7.1 shows the defense accuracy on the smoothed classifier. The attack with "$*$" is re-scaled to the required norm (perturbation size R) based on their perturbation formats. UniCR universally provides a 100% defense accuracy against all the ℓ_1, ℓ_2 and ℓ_∞ perturbations generated by all the state-of-the-art attacks. These results validate our universally approximated certified robustness ensures the same defense performance as certified robustness in practice.

Table 7.1: Defense against real attacks on CIFAR10 (results on MNIST & ImageNet are similar and not included due to space limit).

Defense Accuracy (%)	Gaussian*	Procedural* [30]	Auto-PGD [31]	Wasserstein* [32]	Square* [33]	HSJ* [10]
Teng's [24] ℓ_1-norm R	100.00	100.00	100.00	100.00	100.00	100.00
Our ℓ_1-norm R	100.00	100.00	100.00	100.00	100.00	100.00
Cohen's [1] ℓ_2-norm R	100.00	100.00	100.00	100.00	100.00	100.00
Our ℓ_2-norm R	100.00	100.00	100.00	100.00	100.00	100.00
Yang's [3] ℓ_∞-norm R	100.00	100.00	100.00	100.00	100.00	100.00
Our ℓ_∞-norm R	100.00	100.00	100.00	100.00	100.00	100.00

7.4.3 List of PDFs

The PDFs used in our experimental are summarized in Table 7.2.

Table 7.2: List of noise distributions.

Distribution	Probability Density Function
Gaussian	$\propto e^{-\lvert x/\alpha\rvert^2}$
Laplace	$\propto e^{-\lvert x/\alpha\rvert}$
Hyperbolic Secant	$\propto sech(\lvert x/\alpha\rvert)$
General Normal	$\propto e^{-\lvert x/\alpha\rvert^\beta}$
Cauthy	$\propto \frac{\alpha^2}{x^2+\alpha^2}$
Pareto	$\propto \frac{1}{(1+\lvert x/\alpha\rvert)^{\beta+1}}$
Laplace-Gaussian Mix.	$\propto \beta e^{-\lvert x/\alpha\rvert^1} + (1-\beta)e^{-\lvert x/\alpha\rvert^2}$
Exponential Mix.	$\propto e^{-\beta\lvert x/\alpha\rvert^1 - (1-\beta)\lvert x/\alpha\rvert^2}$

7.4.4 Efficiency for Radius Derivation

We show the runime of our algorithms on deriving the certified radius for the inputs with various input dimensions in Figure 7.3. For the common input dimensions, e.g., 24×24 for MNIST, $3 \times 32 \times 32$ for

CIFAR10, and $3 \times 224 \times 224$ for ImageNet, it takes less than 10 seconds for certifying an image on average. Comparing with the theoretical certified radius deriving, our method's running time is undoubtedly larger since their radius is pre-derived. However, with the significant benefits on the universality and the automatically deriving, we believe the cost of the extra running time is worthwhile and acceptable in practice.

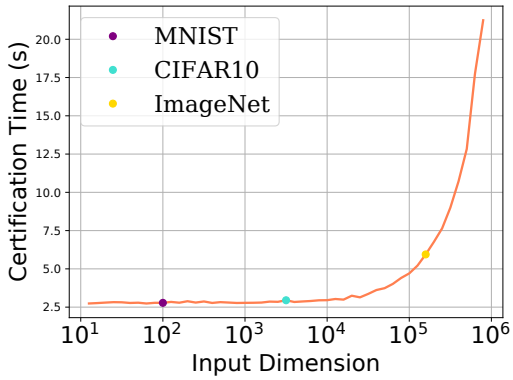


Fig. 7.3: Runtime of UniCR vs. input sizes (with RTX3080 GPU).

7.4.5 Any p (besides 1, 2, ∞)

Existing methods [1, 24] usually focus on the certified radius in a specific norm, e.g., ℓ_1, ℓ_2 or ℓ_∞ norms. Some methods [3, 23] provide certified robustness theories for multiple norms but specific settings are usually needed for deriving the certified radii in different norms. None of the existing methods can automatically compute the certified radius in any ℓ_p norm. In this section, we show our UniCR can automatically approximate the certified radii for various p, in which p is a real number greater than 0.

In the experiments, we set the probability $\underline{p_A} = 0.9$ and draw the lines of certified radius w.r.t. different p for $p > 0$. We show the results computed with different noise distributions in Figure 7.4. We observe that when $p \in (0, 2]$, the certified radius for different p are approximately identical. This finding also matches the theoretical results in Yang et al. [3], in which the certified radii in ℓ_1 and ℓ_2 norm are exactly the same for multiple distributions. When $p > 2$, we observe that the certified radius decreases as p increases.

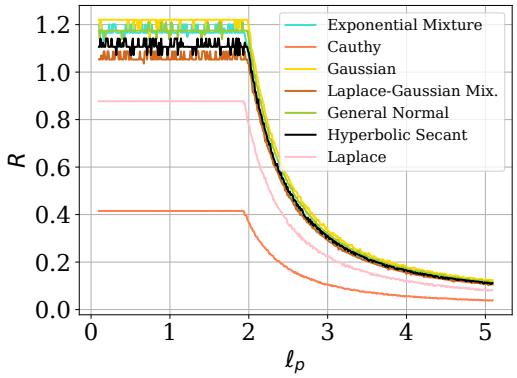
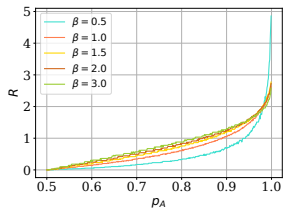
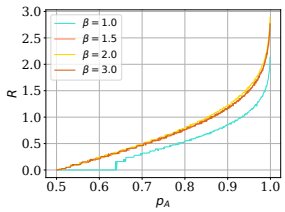


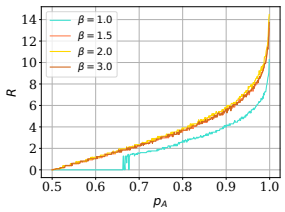
Fig. 7.4: Radius vs. various ℓ_p pert.



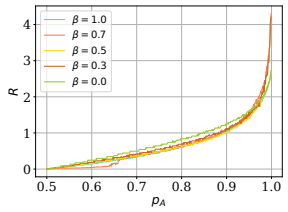
(a) General Normal vs. ℓ_1 perturbations



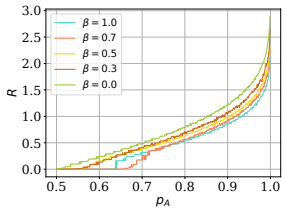
(b) General Normal vs. ℓ_2 perturbations



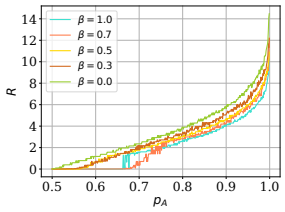
(c) General Normal vs. ℓ_∞ perturbations



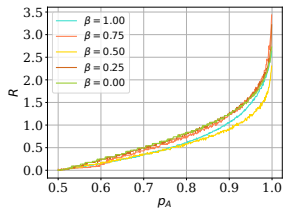
(d) Laplace-Gaussian Mixture vs. ℓ_1 perturbations



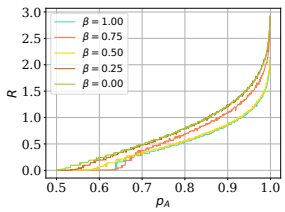
(e) Laplace-Gaussian Mixture vs. ℓ_2 perturbations



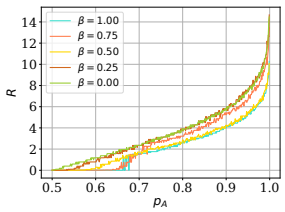
(f) Laplace-Gaussian Mixture vs. ℓ_∞ perturbations



(g) Exponential Mixture vs. ℓ_1 perturbations



(h) Exponential Mixture vs. ℓ_2 perturbations



(i) Exponential Mixture vs. ℓ_∞ perturbations

Fig. 7.5: p_A-R curves of General Normal, Laplace-Gaussian Mixture, and Exponential Mixture noise with a varying β.

7.4.6 Evaluations on Complicated PDFs

We provide a fine-grained evaluation on the complicated distributions [423], e.g., General Normal, Laplace-Gaussian Mixture, and Exponential Mixture noises with various β. It shows that the Gaussian (i.e., $\beta = 2$ for General Normal, $\beta = 0$ for Laplace-Gaussian Mixture and Exponential Mixture) is the optimal noise in these β setting. We also observe the "crash" on Laplace-based distributions when p_A is small.

7.4.7 Certification on Non-Smoothed Classifier

Besides certifying inputs with the smoothed classifier, our input noise optimization (I-OPT) can certify input with a standard classifier without degrading the classifier accuracy on clean data (on the contrary, existing works have to trade off such accuracy for certified defenses).

Specifically, since our I-OPT allows the noise for the input certification to be different from the noise used in training, a special case of the training noise is no noise ($\sigma = 0$). This means that we can certify a naturally-trained classifier (standard classifier). This provides an obvious benefit that the classifier can still execute normal classification on clean data with high accuracy since the standard

Table 7.3: Certified accuracy on standard classifier.

radius R	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8
Yang's [3] vs. ℓ_1-norm	10.6	10.4	10.4	9.8	8.8	8.2	5.4	2.2	1.0
Ours vs. ℓ_1-norm	**98.8**	**47.0**	**22.4**	**17.8**	**13.8**	**10.2**	**7.0**	**3.8**	1.0
Cohen's [1] vs. ℓ_2-norm	10.6	10.4	10.4	9.6	8.8	8.2	5.6	2.2	1.2
Ours vs. ℓ_2-norm	**98.8**	**46.0**	**22.4**	**17.6**	**13.8**	**9.8**	**7.0**	**3.8**	1.2
Yang's [3] vs. ℓ_∞-norm (at $R/255$)	10.6	10.6	10.6	10.4	10.4	10.4	10.4	10.4	10.4
Ours vs. ℓ_∞-norm (at $R/255$)	**98.6**	**92.4**	**69.4**	**61.6**	**53.6**	**46.0**	**37.8**	**27.4**	**24.4**

classifier is trained without noise. Also, with I-OPT, we can tune the noise for the input to maintain the prediction accuracy. Thus, any classifier can be certifiably protected against perturbations without degrading the general performance on clean data.

To maintain the performance on standard classification, we add a condition while performing I-OPT:

$$g(x + \delta) = f(x) \tag{7.29}$$

We show this application on a standard ResNet110 classifier trained on CIFAR10 (see Table 7.3). For the baselines, we use Gaussian noise ($\sigma = 0.35$) and its corresponding theoretical radius [1, 3] for certification. Our method uses I-OPT with General Normal noise and initializes it with the same σ. While approximating the certified radius with UniCR, we generate $4,000$ samples with the Monte Carlo method on CIFAR10.

The table shows that over 98.6% of the inputs are certified by our method with a radius $R > 0$. This means that over 98.6% of the samples are certifiably protected while only 10.6% of inputs are certified by the baselines, which is nearly the accuracy by random guessing. This significant improvement emerges since the I-OPT could optimize the noise PDF for each input even though the classifier is not trained with noise (non-smoothed classifier). Although the certified radii are low compared to smoothly-trained classifiers, it provides a certifiable protection on perturbed data while maintaining the high accuracy for classifying clean data.

7.5 Visual Examples of I-OPT

We present some examples of I-OPT on the ImageNet dataset against ℓ_1, ℓ_2 and ℓ_∞ perturbations, respectively (see Figure 7.6). In the first case (ℓ_1 perturbations), without executing the I-OPT, our UniCR certifies the input with a radius $R = 1.24$. Our I-OPT optimizes the distribution as the right-most figure shows, then the certified radius is improved to 1.48 with our UniCR. Similarly, in the rest cases, we show I-OPT can improve the certified radius significantly by optimizing the noise distribution. Especially, we improve the radius from 0.35 to 1.30 in the second case.

7.6 Discussions

7.6.1 Universal Certified Robustness

It might be impractical to make a universal framework satisfy all the theoretical conditions w.r.t. all ℓ_p perturbations, especially p can be any positive real number. Thus, we admit that UniCR may not strictly satisfy certified robustness all the time due to the approximated optimization. However, extensive empirical results confirm that our derived radii highly approximate the theoretical certified radii against different ℓ_p perturbations. In addition, the defense performance against real attacks also illustrate that our method is as reliable as different theoretical certified radii. We believe that with the negligible error in

practice, UniCR can be deployed as a universal framework to significantly ease the process of achieving certified robustness in different scenarios.

7.6.2 Certifying Perturbed Data with Randomized Smoothing

Randomized smoothing usually assumes that the input is clean and empirical defenses [7, 152] are not applied, if the input data is perturbed before certification, then certification in I-OPT might be inaccurate. Indeed, the certification in traditional randomized smoothing (e.g., [1]) methods also depend on the inputs (since p_A is different for different inputs), they might be inaccurate if the input data is perturbed, either. Thus, randomized smoothing focuses on certifying clean inputs rather than correcting perturbed inputs. We will study this interesting problem on certifying both clean and perturbed inputs in the future.

7.6.3 Can existing methods adopt noise optimization?

A question here is that if the noise optimization can improve the certified radius, can the theoretical methods provide personalized randomization for each input? The personalized randomization is actually not adaptable in the theoretical methods since they cannot automatically derive the certified radius for different noise distributions, especially for uncommon distributions, e.g., $e^{-|x/0.5|^{1.5}}$. Instead, our UniCR can automatically derive the certified radius for any distribution within the continuous parameter space.

7.6.4 Extensions

We evaluate our UniCR on the image classification. Indeed, our UniCR is a general method that can be directly applied to other tasks, e.g., video classification [94, 424], graph learning [425], and natural language processing [426].

7.7 Proof of Theorem 3

We restate Theorem 3:

Theorem 10 (**Asymmetric Randomized Smoothing via Universal Transformation**). *Let $f : \mathbb{R}^d \to C$ be any deterministic or randomized function. Suppose that for the multivariate random variable with isotropic noise $X = x + \epsilon$ in Theorem 2, the certified radius function is $R(\cdot)$. Then, for the corresponding anisotropic input $Y = x + \epsilon^\top \Sigma + \mu$, if there exist $c'_A \in C$ and $\underline{p_A}', \overline{p_B}' \in [0, 1]$ such that:*

$$\mathbb{P}\left(f(Y) = c'_A\right) \geq \underline{p_A}' \geq \overline{p_B}' \geq \max_{c \neq c'_A} \mathbb{P}\left(f(Y) = c\right) \tag{3.5}$$

then for the anisotropic smoothed classifier $g'(x + \delta') \equiv \arg\max_{c \in C} \mathbb{P}\left(f(Y + \delta') = c\right)$, we can guarantee $g'(x + \delta') = c'_A$ for all perturbations $\delta' \in \mathbb{R}^d$ such that:

$$\left\|\Sigma^{-1}\delta'\right\|_p \leq R\left(\underline{p_A}', \overline{p_B}'\right) \tag{3.6}$$

provided that Σ is invertible.

Proof. Let $X = x + \epsilon$, where $\epsilon \in \mathbb{R}^d$ follows an isotropic noise distribution. Let $Y = x + \Sigma\epsilon + \mu$, where $\Sigma \in \mathbb{R}^{d \times d}$ is an invertible covariance matrix, and $\mu \in \mathbb{R}^d$ is a mean offset vector.

Given the input x, covariance matrix Σ, and mean offset μ, define:

$$\mu' = \mu + x - \Sigma x \tag{7.30}$$

Then, the anisotropic input can be written as:

$$Y = \Sigma(x + \epsilon) + \mu' = \Sigma X + \mu' \tag{7.31}$$

Define a transformation $h : \mathbb{R}^d \to \mathbb{R}^d$ as:

$$h(z) = \Sigma z + \mu' \tag{7.32}$$

This transformation maps the isotropic input z to the anisotropic space.

Given any deterministic or randomized function $f : \mathbb{R}^d \to C$, define the anisotropic smoothed classifier as:

$$g'(x) = \arg\max_{c \in C} \mathbb{P}\left(f\left(x + \Sigma\epsilon + \mu\right) = c\right) \tag{7.33}$$

Suppose that for the anisotropic random variable Y, there exist $c'_A \in C$ and $\underline{p_A}', \overline{p_B}' \in [0, 1]$ such that:

$$\mathbb{P}\left(f(Y) = c'_A\right) \geq \underline{p_A}' \geq \overline{p_B}' \geq \max_{c \neq c'_A} \mathbb{P}\left(f(Y) = c\right) \tag{7.34}$$

By the transformation h, the condition in Equation (7.34) is equivalent to:

$$\begin{aligned}
\mathbb{P}\left(f(Y) = c'_A\right) &= \mathbb{P}\left(f\left(\Sigma X + \mu'\right) = c'_A\right) && (7.35)\\
&= \mathbb{P}\left(f\left(h(X)\right) = c'_A\right) && (7.36)\\
&\geq \underline{p_A}' \geq \overline{p_B}' \geq \max_{c \neq c'_A} \mathbb{P}\left(f\left(h(X)\right) = c\right) && (7.37)
\end{aligned}$$

Consider a new classifier $f' : \mathbb{R}^d \to C$ defined as:

$$f'(z) = f\left(h(z)\right) \tag{7.38}$$

This classifier maps the isotropic input z to the class space C through the transformation h.

From Equations (7.36) and (7.37), we have:

$$\mathbb{P}\left(f'(X) = c'_A\right) \geq \underline{p_A}' \geq \overline{p_B}' \geq \max_{c \neq c'_A} \mathbb{P}\left(f'(X) = c\right) \tag{7.39}$$

This is the prerequisite condition in the standard isotropic randomized smoothing theory (e.g., Theorem 2).

Therefore, we can obtain the robustness guarantee with isotropic certified radius R such that:

$$\begin{aligned}
&\arg\max_{c \in C} \mathbb{P}\left(f'(X + \delta) = c\right) = c'_A && (7.40)\\
&\text{subject to } \left\|\delta\right\|_p \leq R\left(\underline{p_A}', \overline{p_B}'\right) && (7.41)
\end{aligned}$$

By the definition of h in Equation (7.32) and the fact that $Y = h(X)$, Equation (7.40) is equivalent to:

$$\begin{aligned}
&\arg\max_{c \in C} \mathbb{P}\left(f'\left(X + \delta\right) = c\right) && (7.42)\\
&\quad = \arg\max_{c \in C} \mathbb{P}\left(f\left(h\left(X + \delta\right)\right) = c\right) && (7.43)\\
&\quad = \arg\max_{c \in C} \mathbb{P}\left(f\left(\Sigma\left(X + \delta\right) + \mu'\right) = c\right) && (7.44)\\
&\quad = \arg\max_{c \in C} \mathbb{P}\left(f\left(\Sigma X + \Sigma\delta + \mu'\right) = c\right) && (7.45)\\
&\quad = \arg\max_{c \in C} \mathbb{P}\left(f\left(Y + \Sigma\delta\right) = c\right) = c'_A && (7.46)
\end{aligned}$$

Define $\delta' = \Sigma\delta$, with $\delta = \Sigma^{-1}\delta'$ since Σ is invertible.

Substituting $\delta = \Sigma^{-1}\delta'$ into the condition $\left\|\delta\right\|_p \leq R\left(\underline{p_A}', \overline{p_B}'\right)$, we obtain:

$$\left\|\Sigma^{-1}\delta'\right\|_p \leq R\left(\underline{p_A}', \overline{p_B}'\right) \tag{7.47}$$

Table 7.4: ALM (binary-case) for randomized smoothing with independent anisotropic noise. d is the dimension size. Φ^{-1} is the inverse CDF of Gaussian distribution. λ is the scalar parameter of the isotropic noise. σ is the anisotropic scale multiplier.

Distribution	PDF	Adv.	Anisotropic Guarantee	ALM
Gaussian [1]	$\propto e^{-\|\frac{z}{\lambda}\|_2^2}$	ℓ_2	$\|\delta \oslash \sigma\|_2 \leq \lambda(\Phi^{-1}(\underline{p_A}'))$	$\sqrt[d]{\prod_{i=1}^d \sigma_i}\lambda(\Phi^{-1}(\underline{p_A}'))$
Gaussian [3]	$\propto e^{-\|\frac{z}{\lambda}\|_2^2}$	ℓ_1	$\|\delta \oslash \sigma\|_1 \leq \lambda(\Phi^{-1}(\underline{p_A}'))$	$\sqrt[d]{\prod_{i=1}^d \sigma_i}\lambda(\Phi^{-1}(\underline{p_A}'))$
		ℓ_∞	$\|\delta \oslash \sigma\|_\infty \leq \lambda(\Phi^{-1}(\underline{p_A}'))/\sqrt{d}$	$\sqrt[d]{\prod_{i=1}^d \sigma_i}\lambda(\Phi^{-1}(\underline{p_A}'))/\sqrt{d}$
Laplace [24]	$\propto e^{-\|\frac{z}{\lambda}\|_1}$	ℓ_1	$\|\delta \oslash \sigma\|_1 \leq -\lambda\log(2(1-\underline{p_A}'))$	$\sqrt[d]{\prod_{i=1}^d \sigma_i}\lambda\log(2(1-\underline{p_A}'))$
Exp. ℓ_∞ [3]	$\propto e^{-\|\frac{z}{\lambda}\|_\infty}$	ℓ_1	$\|\delta \oslash \sigma\|_1 \leq 2d\lambda(\underline{p_A}' - \frac{1}{2})$	$2\sqrt[d]{\prod_{i=1}^d \sigma_i}d\lambda(\underline{p_A}' - \frac{1}{2})$
		ℓ_∞	$\|\delta \oslash \sigma\|_\infty \leq \lambda\log(\frac{1}{2(1-\underline{p_A}')})$	$\sqrt[d]{\prod_{i=1}^d \sigma_i}\lambda\log(\frac{1}{2(1-\underline{p_A}')})$
Uniform ℓ_∞ [67]	$\propto \mathbb{I}(\|z\|_\infty \leq \lambda)$	ℓ_1	$\|\delta \oslash \sigma\|_1 \leq 2\lambda(\underline{p_A}' - \frac{1}{2})$	$2\sqrt[d]{\prod_{i=1}^d \sigma_i}\lambda(\underline{p_A}' - \frac{1}{2})$
		ℓ_∞	$\|\delta \oslash \sigma\|_\infty \leq 2\lambda(1 - \sqrt[d]{\frac{3}{2} - \underline{p_A}'})$	$2\sqrt[d]{\prod_{i=1}^d \sigma_i}\lambda(1 - \sqrt[d]{\frac{3}{2} - \underline{p_A}'})$
Power Law ℓ_∞ [3]	$\propto \frac{1}{(1+\|\frac{z}{\lambda}\|_\infty)^a}$	ℓ_1	$\|\delta \oslash \sigma\|_1 \leq \frac{2d\lambda}{a-d}(\underline{p_A} - \frac{1}{2})$	$2\sqrt[d]{\prod_{i=1}^d \sigma_i}\frac{d\lambda}{a-d}(\underline{p_A} - \frac{1}{2})$

Therefore, the guarantee in Equations (7.40) and (7.41) is equivalent to:

$$\arg\max_{c\in C} \mathbb{P}\left(f\left(Y+\delta'\right)=c\right) = c_A' \tag{7.48}$$

$$\text{subject to } \left\|\Sigma^{-1}\delta'\right\|_p \leq R\left(\underline{p_A}', \overline{p_B}'\right) \tag{7.49}$$

By Equation (7.33), we have:

$$g'\left(x+\delta'\right) = \arg\max_{c\in C} \mathbb{P}\left(f\left(Y+\delta'\right)=c\right) = c_A' \tag{7.50}$$

for all $\delta' \in \mathbb{R}^d$ satisfying $\left\|\Sigma^{-1}\delta'\right\|_p \leq R\left(\underline{p_A}', \overline{p_B}'\right)$.

This completes the proof. □

7.8 Binary case of Theorem 3

Theorem 11 (Universal Transformation for Anisotropic Noise). *Let $f : \mathbb{R}^d \to C$ be any deterministic or random function. Suppose that for the isotropic input X in Theorem 2, the certified radius function for the binary case is $R(\cdot)$. Then, for the corresponding anisotropic input Y such that $Y = x + \epsilon^\top\sigma + \mu$, if there exist $c_A' \in C$ and $\underline{p_A}' \in (1/2, 1]$ such that:*

$$\mathbb{P}(f(Y) = c_A') \geq \underline{p_A}' \geq \frac{1}{2} \tag{7.51}$$

Then $g'(x+\delta') = \arg\max_{c\in C}\mathbb{P}(f(Y)=c) = c_A'$ for all $\|\delta' \oslash \sigma\|_p \leq R(\underline{p_A}')$ where g' denotes the anisotropic smoothed classifier, δ' denotes the perturbation injected to g'.

Proof. The proof for the binary case is similar to the proofs for the multiclass-case (see Appendix 7.7). □

7.9 Additional Metric for Certified Region (Binary-case)

How to develop a general metric for evaluating the robustness region for randomized smoothing with anisotropic noise is an important but challenging problem. We observe that the guarantee in Theorem 3 forms a certified region, within which the perturbation is certifiably safe to the smoothed classifier. The

ℓ_p-norm bounding on the scaled perturbation $\delta' \oslash \sigma$ results in the anisotropy of the certified region around the input. We illustrate the asymmetric certified region for different ℓ_p-norm guarantees in Figure 3.2. It shows that if the δ space is a 2-dimension space, then the guarantee of asymmetric RS draws a rhombus, ellipse, and rectangle in ℓ_1, ℓ_2, and ℓ_∞ norms, respectively. Within the asymmetric region, we can find an isotropic region that also satisfies the robustness guarantee (a subset of the anisotropic region), which represents an explicit certified radius as depicted in Corollary 3.2.

However, evaluating the performance of randomized smoothing with anisotropic noise via Eq. (3.9) in Corollary 3.2, although formally guaranteed with robustness, fails to capture the full certified regions.

Specifically, Eq. (3.9) evaluates the performance only based on the blue region in Figure 3.2, but the Eq. (3.6) in Theorem 3 actually guarantees that all the δ (perturbations) within the green region do not change the perturbation. Therefore, to fairly and accurately evaluate the performance of certification via anisotropic noise, we need to develop an auxiliary metric that can cover the entire certified region in highly-dimensional δ space as a complement besides the certified radius.

From another perspective, evaluating the performance of randomized smoothing can be considered as evaluating the size of the robust perturbation set $S(n, p)$.

Consider the Euclidean structure, $S(d, p)$ is a finite set in d-dimensional Euclidean space. Therefore, we leverage the Lebesgue measure [427] to compute the size of $S(d, p)$ (see Theorem 4).

We observe that for a fixed d and p, the $\frac{(2\Gamma(1+\frac{1}{p}))^d}{\Gamma(1+\frac{d}{p})}$ factor in the Lebesgue measure is a constant. Then, when comparing the Lebesgue measure in the same norm ℓ_p and the same space $\mathbb{R}^d$, the constant term can be ignored. Also, the R^d factor can lead to infinite numeral computation, thus we also scale the Lebesgue measure by calculating the d-th root. As a result, we define the Alternative Lebesgue Measure (ALM) of the robust perturbation set with the same d and p as:

$$V'_S = \sqrt[d]{\prod_{i=1}^{d} \sigma_i} R \tag{7.52}$$

Table 7.4 presents the ALM formulas of common RS methods.

Alternative Lebesgue Measure[1] **vs Radius**. When the multipliers of the scale parameter for anisotropic noise $\sigma_1 = \sigma_2 = ... = 1$, the noise turns into the isotropic noise and the alternative Lebesgue measure turns into the certified radius R. Therefore, the alternative Lebesgue measure can be treated as a generalized metric compared to the certified radius. This generalization based on the certified radius also enables us to fairly compare the randomized smoothing based on anisotropic noise with isotropic noise.

Note that the new metric ALM is not developed to bound the perturbation, but to accurately measure the certified guarantees of randomized smoothing with anisotropic noise.

7.10 Proof of Theorem 4

Proof.

Definition 3 (**d-dimensional Generalized Super-ellipsoid**). *The d-dimensional generalized super-ellipsoid ball is defined as*

$$E(d, p) = \{(\delta_1, \delta_2, ..., \delta_d) : \sum_{i=1}^{d} |\frac{\delta_i}{c_i}|^{p_i} \leq 1, p_i > 0\} \tag{7.53}$$

Definition 4 (**Euler Gamma Function**). *The Euler gamma function is defined by*

$$\Gamma(\beta) = \int_0^\infty \alpha^{\beta-1} e^{-\alpha} d\alpha \tag{7.54}$$

[1] ALM is like the normalized radius of the certified region in all d dimensions.

There are some properties of Γ*: 1) For all* $\beta > 0$*,* $\beta\Gamma(\beta) = \Gamma(\beta + 1)$*, 2) For all positive integers n,* $\Gamma(n) = (n-1)!$*, and 3)* $\Gamma(1/2) = \sqrt{\pi}$*.*

Lemma 3 (**Lebesgue Measure of Generalized Super-ellipsoids [428]**). *The Lebesgue measure of the generalized super-ellipsoids defined in Definition 3 is given by*

$$V_E(d,p) = 2^d \frac{\prod_{i=1}^{d} c_i \Gamma(1+\frac{1}{p_i})}{\Gamma(1+\sum_{i=1}^{d}\frac{1}{p_i})}; p_i > 0, d = 1,2,3,... \tag{7.55}$$

We leverage Lemma 3 to prove Theorem 3. Let $p_i = p$, the d-dimensional robust perturbation set is equivalent to

$$S(d,p) = \{(\delta_1, \delta_2, ..., \delta_d) : \sum_{i=1}^{d} |\frac{\delta_i}{\sigma_i R}|^p \leq 1\} \tag{7.56}$$

The Lebesgue measure in Lemma 3 will be

$$\begin{aligned} V_E(d,p) =& 2^d \frac{\prod_{i=1}^{d} c_i R\Gamma(1+\frac{1}{p})}{\Gamma(1+\sum_{i=1}^{d}\frac{1}{p})} = \frac{(2R\Gamma(1+\frac{1}{p}))^d \prod_{i=1}^{d}\sigma_i}{\Gamma(1+\frac{d}{p})}; \\ & p > 0, d = 1,2,3,... \end{aligned} \tag{7.57}$$

Thus, this completes the proof. □

7.11 Additional Experiments

7.11.1 Anisotropic noise vs. isotropic noise

We also present the performance comparison of the anisotropic and isotropic RS on ImageNet in Figure 7.7. It shows that similar to the results in the MNIST and CIFAR10, the pattern-wise and dataset-wise anisotropic noise can improve the performance of certified accuracy w.r.t. the ALM moderately, while the certification-wise anisotropic noise can significantly boost the performance with the best optimality.

7.11.2 Universality of Pattern-wise and Dataset-wise Methods

Similar to the certification-wise noise, pattern-wise and dataset-wise noise are also universal to different ℓ_p norms and different PDFs. As shown in Figure 7.8, the pattern-wise and dataset-wise anisotropic noise can also significantly boost the performance of isotropic RS on various settings.

7.12 Proofs

7.12.1 Proof of Theorem 6

The proof of Theorem 6 is based on the Neyman-Pearson Lemma, so we first review the Neyman-Pearson Lemma.

Lemma 4. ***(Neyman-Pearson Lemma)*** *Let X and Y be random variables in* $\mathbb{R}^d$ *with densities* μ_X *and* μ_Y*. Let* $f : \mathbb{R}^d \to \{0,1\}$ *be a random or deterministic function. Then:*

(1) If $S = \{z \in \mathbb{R}^d : \frac{\mu_Y(z)}{\mu_X(z)} \leq t\}$ *for some* $t > 0$ *and* $\mathbb{P}(f(X) = 1) \geq \mathbb{P}(X \in S)$*, then* $\mathbb{P}(f(Y) = 1) \geq \mathbb{P}(Y \in S)$*;*

(2) If $S = \{z \in \mathbb{R}^d : \frac{\mu_Y(z)}{\mu_X(z)} \geq t\}$ *for some* $t > 0$ *and* $\mathbb{P}(f(X) = 1) \leq \mathbb{P}(X \in S)$*, then* $\mathbb{P}(f(Y) = 1) \leq \mathbb{P}(Y \in S)$*.*

Let $x \in \mathbb{R}^d$ be any clean input with label y. Let noise ϵ be drawn from any continuous distribution $\varphi(0, \boldsymbol{\kappa})$. Let $x' \in \mathbb{R}^d$ be any input. Denote $X = x' + \epsilon$, and $X_\delta = x' + \delta + \epsilon$. Let $f : \mathbb{R}^d \to \mathbb{R}^1$ be any deterministic or random function. For each input x, we can consider two classes: y or $\neq y$, so the problem can be considered as a binary classification problem. Let the lower bound of randomized parallel query on x' denoted as $Q(x') = \underline{p_{adv}}$. Define the half set:

$$A := \{z : \frac{\varphi(z - \delta, \boldsymbol{\kappa})}{\varphi(z, \boldsymbol{\kappa})} \leq \tau\} \tag{7.58}$$

where the auxiliary parameter τ is picked to suffice:

$$\mathbb{P}(X \in A) = \mathbb{P}[\frac{\varphi(x' + \epsilon - \delta, \boldsymbol{\kappa})}{\varphi(x' + \epsilon, \boldsymbol{\kappa})} \leq \tau] = \underline{p_{adv}} \tag{7.59}$$

Suppose $\underline{p_{adv}}$ and the ASP Threshold p satisfy $\underline{p_{adv}} \geq p$, then

$$\mathbb{P}[f(X) \neq y] \geq p_{adv} = \mathbb{P}(X \in A) \tag{7.60}$$

Using Neyman-Pearson Lemma (considering $X_\delta = X + \delta$ as Y in Neyman-Pearson Lemma, and $\neq y$ as class 1), we have:

$$\mathbb{P}[f(X_\delta) \neq y] \geq \mathbb{P}[X_\delta \in A] \tag{7.61}$$

which is equal to

$$\mathbb{P}[f(X_\delta) \neq y] \geq \mathbb{P}[\frac{\varphi(x' + \epsilon, \boldsymbol{\kappa})}{\varphi(x' + \epsilon + \delta, \boldsymbol{\kappa})} \leq \tau] \tag{7.62}$$

If $\mathbb{P}(\frac{\varphi(x'+\epsilon,\boldsymbol{\kappa})}{\varphi(x'+\epsilon+\delta,\boldsymbol{\kappa})} \leq \tau) \geq p$, we can guarantee that

$$\mathbb{P}[f(X_\delta) \neq y] \geq p \tag{7.63}$$

which means the probability of classifying X_δ as adversarial examples is greater than p. Therefore, the distribution of X_δ can be guaranteed to have the attack success probability larger than p.

Considering Eq. (7.59), we have $\tau = \Phi_-^{-1}(\underline{p_{adv}})$,

where Φ_-^{-1} is the inverse CDF of random variable $\frac{\varphi(x'+\epsilon-\delta,\boldsymbol{\kappa})}{\varphi(x'+\epsilon,\boldsymbol{\kappa})}$. Therefore, substitute τ in $\mathbb{P}[\frac{\varphi(x'+\epsilon,\boldsymbol{\kappa})}{\varphi(x'+\epsilon+\delta,\boldsymbol{\kappa})} \leq \tau] \geq p$, we have

$$\Phi_+[\Phi_-^{-1}(\underline{p_{adv}})] \geq p \tag{7.64}$$

where Φ_+ is the CDF of random variable $\frac{\varphi(x'+\epsilon,\boldsymbol{\kappa})}{\varphi(x'+\epsilon+\delta,\boldsymbol{\kappa})}$. The ratios can be further simplified as

$$\frac{\varphi(x' + \epsilon - \delta, \boldsymbol{\kappa})}{\varphi(x' + \epsilon, \boldsymbol{\kappa})} = \frac{\varphi(\epsilon - \delta, \boldsymbol{\kappa})}{\varphi(\epsilon, \boldsymbol{\kappa})} \tag{7.65}$$

$$\frac{\varphi(x' + \epsilon, \boldsymbol{\kappa})}{\varphi(x' + \epsilon + \delta, \boldsymbol{\kappa})} = \frac{\varphi(\epsilon, \boldsymbol{\kappa})}{\varphi(\epsilon + \delta, \boldsymbol{\kappa})} \tag{7.66}$$

Now we complete the proof of Theorem 6.

7.12.2 Certifiable Attack: Gaussian Noise

Corollary 11.1. ***(Certifiable Adversarial Shifting: Gaussian Noise)*** *Under the same condition with Theorem 6, and let ϵ be a noise drawn from Gaussian distribution $\mathcal{N}(0, \sigma)$. Then, if the randomized query on the adversarial input satisfies Eq. (4.5):*

$$\mathbb{P}[f(x' + \epsilon) \neq y] \geq \underline{p_{adv}} = Q(x') \geq p \tag{7.67}$$

Then $\mathbb{P}[f(x' + \delta + \epsilon) \neq y] \geq p$ is guaranteed for any shifting vector δ when

$$\|\delta\|_2 \leq \sigma[\Phi^{-1}(\underline{p_{adv}}) - \Phi^{-1}(p)] \tag{7.68}$$

where Φ^{-1} denotes the inverse of the standard Gaussian CDF.

Proof. The Gaussian distribution is $\mu(x) \propto e^{-\frac{x^2}{2\sigma^2}}$, thus

$$\frac{\mu(x-\delta)}{\mu(x)} = e^{(2x\delta-\delta^2)/(2\sigma^2)} \tag{7.69}$$

Let $\tau := exp((2\Phi_\sigma^{-1}(\underline{p_{adv}})\delta - \delta^2)/(2\sigma^2))$, where Φ_σ^{-1} denotes the inverse Gaussian CDF with variance σ. Let random variables $X := x' + \epsilon$ and $X_\delta := x' + \epsilon + \delta$. Then we have:

$$\begin{aligned}
\mathbb{P}(X \in A) &= \mathbb{P}[\frac{\mu(X-\delta)}{\mu(X)} \leq \tau] && (7.70)\\
&= \mathbb{P}[exp((2X\delta - \delta^2)/(2\sigma^2))] \leq && (7.71)\\
&exp[(2\Phi_\sigma^{-1}(\underline{p_{adv}})\delta - \delta^2)/(2\sigma^2)] && (7.72)\\
&= \mathbb{P}[X \leq \Phi_\sigma^{-1}(\underline{p_{adv}})] && (7.73)\\
&= \underline{p_{adv}} && (7.74)\\
& && (7.75)
\end{aligned}$$

Using Neyman-Pearson Lemma, we have:

$$\mathbb{P}[f(X_\delta) \neq y] \geq \mathbb{P}[(X_\delta) \in A] \tag{7.76}$$

Since

$$\begin{aligned}
\mathbb{P}[X_\delta \in A] &= \mathbb{P}[\frac{\mu(X)}{\mu(X+\delta)} \leq \tau] && (7.77)\\
&= \mathbb{P}[exp((2X\delta + \delta^2)/(2\sigma^2))] \leq && (7.78)\\
&exp[(2\Phi_\sigma^{-1}(\underline{p_{adv}})\delta - \delta^2)/(2\sigma^2)] && (7.79)\\
&= \mathbb{P}[2X\delta + \delta^2 \leq (2\Phi_\sigma^{-1}(\underline{p_{adv}})\delta - \delta^2)] && (7.80)\\
&= \mathbb{P}[X \leq \Phi_\sigma^{-1}(\underline{p_{adv}}) - \|\delta\|] && (7.81)\\
&= \mathbb{P}[\frac{X}{\sigma} \leq \Phi^{-1}(\underline{p_{adv}}) - \frac{\|\delta\|}{\sigma}] && (7.82)
\end{aligned}$$

where Φ^{-1} denotes the inverse standard Gaussian CDF.

To guarantee that $\mathbb{P}[f(X_\delta) \neq y] \geq p$, we need:

$$\begin{aligned}
\mathbb{P}[X_\delta \in A] &= \mathbb{P}[\frac{X}{\sigma} \leq \Phi^{-1}(\underline{p_{adv}}) - \frac{\|\delta\|}{\sigma}] && (7.83)\\
&\geq p && (7.84)\\
& && (7.85)
\end{aligned}$$

which is equivalent to

$$\|\delta\| \leq \sigma[\Phi^{-1}(\underline{p_{adv}}) - \Phi^{-1}(p)] \tag{7.86}$$

This completes the proof of Corollary 11.1.

□

7.12.3 Proof of Theorem 7

If the Condition Eq. (4.5) is satisfied, we have

$$\underline{p_{adv}} \geq p \tag{7.87}$$

For any direction w, our goal is to find the δ in this direction with maximum $\|\delta\|_2$. When $\|\delta\|_2 = 0$, we have

$$\Phi_+ = \Phi_- \tag{7.88}$$

Thus, we have

$$\Phi_+[\Phi_-^{-1}(\underline{p_{adv}})] = \underline{p_{adv}} \geq p \tag{7.89}$$

Then, we prove that when $\|\delta\|_2$ increase, we will get $\Phi_+[\Phi_-^{-1}(\underline{p_{adv}})]$ decrease.

Since Φ_- is the CDF of the random variable $\frac{\varphi(\epsilon-\delta,\kappa)}{\varphi(\epsilon,\kappa)}$, and $\varphi(x)$ decreases when $|x|$ increases, when $\|\delta\|_2 \to \infty$, we have $\frac{\varphi(\epsilon-\delta,\kappa)}{\varphi(\epsilon,\kappa)} \to 0$, and $\Phi_-^{-1}(\underline{p_{adv}}) \to 0$.

Since Φ_+ is the CDF of the random variable $\frac{\varphi(\epsilon,\kappa)}{\varphi(\epsilon+\delta,\kappa)}$, when $\|\delta\|_2 \to \infty$, $\frac{\varphi(\epsilon,\kappa)}{\varphi(\epsilon+\delta,\kappa)} \to \infty$, so $\Phi_+ \to 0$. Therefore, when $\|\delta\|_2 \to \infty$, we have

$$\Phi_+[\Phi_-^{-1}(\underline{p_{adv}})] \to 0 < p \tag{7.90}$$

When $\|\delta\|_2 \to 0$, we have

$$\Phi_+[\Phi_-^{-1}(\underline{p_{adv}})] = \underline{p_{adv}} \geq p \tag{7.91}$$

Since Φ_- and Φ_+ is continuous function, between 0 and ∞, there must be some δ such that $\Phi_+[\Phi_-^{-1}(p_{adv})] = p$.

Now we prove that the binary search algorithm can always find the δ solution, then we show how to bound the adversarial attack certification. We use Monte Carlo method to estimate the $\underline{p_{adv}}$ as well as the CDFs Φ_- and Φ_+. To bound the empirical CDFs, we leverage Dvoretzky-Kiefer-Wolfowitz inequality [422].

Lemma 5. ***(Dvoretzky–Kiefer–Wolfowitz inequality** (restate)) Let $X_1, X_2, ..., X_n$ be real-valued independent and identically distributed random variables with cumulative distribution function $F(\cdot)$, where $n \in \mathbb{N}$.Let F_n denote the associated empirical distribution function defined by*

$$F_n(x) = \frac{1}{n}\sum_{i=1}^{n} \mathbf{1}_{\{X_i <= x\}}, x \in \mathbb{R} \tag{7.92}$$

The Dvoretzky–Kiefer–Wolfowitz inequality bounds the probability that the random function F_n differs from F by more than a given constant $\Delta \in \mathbb{R}^+$:

$$\mathbb{P}[\sup_{x\in\mathbb{R}} |F_n(x) - F(x)| > \Delta] \leq 2e^{-2n\Delta^2} \tag{7.93}$$

Let the Monte Carlo sampling number N_m. Each shifting is an independent certification, and there are a lower-bound estimation and two CDF estimations in each certification. Suppose the confidence of lower-bound estimation is $(1-\alpha)$, then the certification confidence should be at least $(1-\alpha)(1-2e^{-2N_m\Delta^2})^2$.

7.13 Denoising with Diffusion Models

The certifiable adversarial examples sampled from *Adversarial Distribution* are noise-injected inputs that still might be perceptible when the noise is large. We further leverage the recent innovation for image synthesis, i.e., diffusion model [130], to denoise the adversarial examples for better imperceptibility. The key idea is to consider the noise-perturbed adversarial examples as the middle sample in the forward process of the diffusion model [85, 86]. This is shown to improve the imperceptibility and the diversity of the adversarial examples.

Table 7.5: Summary of all parameter settings

Experiments	General		Random Parallel Query		Smoothed Self-supervised Localization						Bin-search Localization			Refinement				
	p	σ	α	N_m	π_{init}	γ	N_{max}	n_{max}	η	N_s	N_r	N_b	Ω	M	η'	e	e_s	N_b
Comparison with empirical attack under Blacklight detection	10%	0.025	0.001	50	3/255	3/255	85	10	3/255	50	85	15	0.1	20	0.05	0.01	0.0025	72
Comparison with empirical attack against RAND pre-processing defense	10%	0.025	0.001	50	3/255	3/255	85	10	3/255	50	85	15	0.1	20	0.05	0.01	0.0025	72
Comparison with empirical attack against RAND post-processing Defense	10%	0.025	0.001	50	3/255	3/255	85	10	3/255	50	85	15	0.1	20	0.05	0.01	0.0025	72
Comparison with empirical attack against TRADES adversarial training	10%	0.025	0.001	50	3/255	3/255	85	10	3/255	50	85	15	0.1	20	0.05	0.01	0.0025	72
Certifiable Attack against Feature Squeezing	90%	0.25	0.001	1000	3/255	3/255	85	10	3/255	1000	85	15	0.1	20	0.05	0.01	0.025	72
Certifiable Attack against Randomized Smoothing and Adaptive Denoiser	90%	0.25	0.001	1000	3/255	3/255	85	10	3/255	1000	85	15	0.1	20	0.05	0.01	0.025	72
Ablation study: Certifiable attack vs. different noise variance	90%	0.10 - 0.50	0.001	500/1000	3/255	3/255	85	10	3/255	500/1000	85	15	0.1	20	0.05	0.01	0.1 σ	72
Ablation study: Certifiable attack vs. different ASP Threshold p	50 - 95%	0.25	0.001	500/1000	3/255	3/255	85	10	3/255	500/1000	85	15	0.1	20	0.05	0.01	0.025	72
Ablation study: Certifiable attack vs. different Localization/Shifting	90%	0.25	0.001	500/1000	3/255	3/255	85	10	3/255	500/1000	85	15	0.1	20	0.05	0.01	0.025	72
Ablation study: Certifiable attack vs. different noise PDF	90%	–	0.001	500/1000	3/255	3/255	85	10	3/255	500/1000	85	15	0.1	20	0.05	0.01	–	72
Ablation study: Certifiable attack w/ and w/o Diffusion Denoise	90%	0.25	0.001	500/1000	3/255	3/255	85	10	3/255	500/1000	85	15	0.1	20	0.05	0.01	0.025	72

Specifically, the closed-form sampling in the forward process at timestep t in [130] can be written as:

$$x_t = \sqrt{\overline{\alpha}_t} x_0 + \sqrt{1 - \overline{\alpha}_t} \epsilon_0 \tag{7.94}$$

where $x_0 = x$ is a clean image, $\overline{\alpha}_t$ is the parameter indicating the transformation of the image in the forward process, and ϵ_0 is a noise drawn from the standard normal distribution $\mathcal{N}(0, 1)$. The certifiable adversarial examples when adding noise ϵ_0 can be expressed as:

$$x_{adv} = x' + \delta + \epsilon_0 \tag{7.95}$$

We can then consider x_{adv} as the sample x_t in Eq. (7.94) by transforming x_{adv} to $\sqrt{\overline{\alpha}_t} x_{adv}$ and satisfying these conditions: (1) $x' + \delta = x_0$ and (2) $\sqrt{\overline{\alpha}_t}\sigma = \sqrt{1 - \overline{\alpha}_t}$. Then, we have $\overline{\alpha}_t = \frac{1}{\sigma^2+1}$ to bridge diffusion model and the certifiable attack.

By finding the corresponding time step t and $\overline{\alpha_t}$, we can leverag the reverse process $\mathcal{R}(\cdot)$ of the diffusion model to denoise x_{adv}:

$$x'_{adv} = \mathcal{R}(\mathcal{R}(...\mathcal{R}(\sqrt{\overline{\alpha}_t} x_{adv}))) \tag{7.96}$$

Note that the reverse denoising process can be plugged into our attack framework by simply replacing x_{adv} with x'_{adv} in all processes. It will not affect the guarantee since the denoising process can be part of the classification model, i.e., constructing a new target model $f'(x_{adv}) = f(\mathcal{R}(\mathcal{R}(...\mathcal{R}(\sqrt{\overline{\alpha}_t} x_{adv}))))$ given any f.

7.14 Additional Experiments

7.14.1 Additional Experimental Settings

Table 7.5 summarizes all parameter settings.

7.14.2 More Results of Black-Box Attacks against SOTA Defenses

More Results on Attacking Blacklight

See Results in Table 7.8-Table 7.15.

More Results on Attacking RAND-Pre

See Results in Table 7.16-Table 7.23.

More Results on Attacking RAND-Post

See Results in Table 7.24-Table 7.31.

7.14.3 Attacking Other Empirical and Certified Defenses

Attacking Empirical Feature Squeezing Detection

We also evaluate the certifiable attack against adversarial detection. Specifically, we select the Feature Squeezing [61] method, which modifies the image and detects the adversarial examples according to the difference of model outputs. To position the performance of the detection, we compare the certifiable attack with the C&W empirical attack [92]. In this experiment, Gaussian noise was adopted, and parameters are set as $\sigma = 0.25$ and $p = 90\%$. We draw the ROC curve in Figure 7.9. As the results show, the certifiable attack is less detectable than the C&W attack w.r.t. Feature Squeezing (with lower ROC scores), possibly because the prediction of empirical adversarial examples is less robust to image modification (empirical adversarial examples tend to be some special data points near the decision boundary). After the modification, it tends to output a different result. The outputs of certifiable adversarial examples are more consistent after the modification since their neighbors tend to be adversarial as well.

Attacking Randomized Smoothing-based Certified Defense

Randomized smoothing trains the classifier on inputs with Gaussian noises. We evaluate the certifiable attack on the classifier trained with the same noise. Specifically, we use the Gaussian noise with $\sigma = 0.12$ to 0.50 in the classifier training and $\sigma = 0.25$ in the certifiable attack. Table 7.6 shows that the noise-trained classifier, especially when the model is trained with the same noise parameter as the adversary, can significantly degrade the performance of certifiable attacks. Noticeably, the smoothed training increases the perturbation sizes, especially the Mean Dist. ℓ_2 significantly, and doubles the number of RPQ, which means the smoothing training can obstacle the certifiable attack to find a *Adversarial Distribution*. It also reduces the certified accuracy of the certifiable attack significantly. However, the certifiable attack can still guarantee that 87.20% of the test samples can be provably misclassified. Without performing the randomized smoothing certification against the certifiable attack, we can conclude that the certified accuracy of randomized smoothing with $\sigma = 0.25$ will be at most 12.80% since at least 87.20% of the RandAEs are guaranteed to generate successful AEs with 90% probability.

Table 7.6: RS-based defense against our attack on CIFAR10. $\sigma = 0.25$, $p = 90\%$

Defense Para.	Dist. ℓ_2	Mean Dist. ℓ_2	# RPQ	Cert. Acc.
none	12.95	2.34	14.35	91.21%
$\sigma_{rs} = 0.12$	13.93	6.72	23.93	88.40%
$\sigma_{rs} = 0.25$	16.07	11.84	33.93	87.20%
$\sigma_{rs} = 0.50$	15.84	10.12	29.57	90.60%

7.14.4 Diffusion Model for Denoising

We implement the diffusion model [130] with the linear schedule. We train a diffusion model and an UNet with $3 \times 32 \times 32$ dimension for CIFAR10 and $3 \times 64 \times 64$ dimension for ImageNet and denoise the certified *Adversarial Distribution*samples injected by Gaussian noise with $\sigma = 0.25$. The experimental results are shown in Table 7.7. Although the diffusion denoise increases the number of queries and Mean Dist. ℓ_2, the perturbation of AE samples is significantly reduced due to the denoise. It is worth noting that the AEs generated by the diffusion model are unique due to the stochastic reverse process. This difference enables the certifiable attack to generate diverse AEs while ensuring the ASP guarantee.

Table 7.7: Performance of Certifiable Black-box Attack with Diffusion Denoise (p = 90%, σ = 0.25). Diffusion Denoise can significantly reduce the perturbation size when the noise scale is very large.

Dataset	denoise	Dist. ℓ_2	Mean Dist. ℓ_2	# RPQ	Certified Acc.
CIFAR10	w/o	12.95	2.34	14.35	91.21%
	w/	9.04	10.38	30.89	91.30%
ImageNet	w/o	24.32	0.72	5.60	99.8%
	w/	8.48	7.30	12.85	100.00%

Table 7.8: Attack performance under Blacklight detection on VGG and CIFAR10 (Clean Accuracy: 90.3%)

Attack	Query Type	Pert. Type	Det. Rate %	# Q to Det.	Det. Cov. %	Model Acc.	# Q	Dist. ℓ_2
Bandit	Score	ℓ_∞	100.0	1.0	69.2	0.0	59	4.77
NES	Score	ℓ_∞	100.0	8.6	21.2	0.0	264	1.33
Parsimonious	Score	ℓ_∞	100.0	2.0	96.7	0.0	107	4.90
Sign	Score	ℓ_∞	100.0	2.0	92.4	0.0	83	4.90
Square	Score	ℓ_∞	100.0	2.0	75.9	0.0	25	4.90
ZOSignSGD	Score	ℓ_∞	100.0	2.0	50.6	0.0	267	1.29
GeoDA	Label	ℓ_∞	100.0	1.0	89.7	0.2	377	3.08
HSJ	Label	ℓ_∞	100.0	7.7	92.2	0.0	353	2.94
Opt	Label	ℓ_∞	100.0	8.6	83.4	37.1	2571	1.04
RayS	Label	ℓ_∞	100.0	5.8	78.6	0.0	226	4.82
SignFlip	Label	ℓ_∞	100.0	8.7	56.0	0.0	68	3.91
SignOPT	Label	ℓ_∞	100.0	8.6	89.1	21.0	1112	1.15
Bandit	Score	ℓ_2	100.0	1.0	98.9	0.0	121	2.62
NES	Score	ℓ_2	100.0	8.5	32.6	0.0	823	0.53
Simple	Score	ℓ_2	100.0	1.0	99.8	0.0	779	0.77
Square	Score	ℓ_2	100.0	2.0	78.7	0.0	26	4.47
ZOSignSGD	Score	ℓ_2	100.0	2.0	53.4	0.3	1631	0.48
Boundary	Label	ℓ_2	100.0	7.6	67.4	21.7	315	2.70
GeoDA	Label	ℓ_2	100.0	1.0	89.7	0.1	230	2.80
HSJ	Label	ℓ_2	100.0	7.6	90.9	0.0	188	2.49
Opt	Label	ℓ_2	100.0	8.6	65.1	32.6	675	2.39
SignOPT	Label	ℓ_2	100.0	8.6	77.7	24.7	510	1.89
PointWise	Label	Opt.	100.0	1.0	99.5	0.0	764	2.06
SparseEvo	Label	Opt.	95.7	1.0	100.0	0.0	9569	2.40
CA (sssp)	Label	Opt.	0.0	∞	0.0	6.2	393	3.75
CA (bin search)	Label	Opt.	0.0	∞	0.0	0.0	473	5.20

Table 7.9: Attack performance under Blacklight detection on ResNet and CIFAR10 (Clean Accuracy: 92.1%)

Attack	Query Type	Pert. Type	Det. Rate %	# Q to Det.	Det. Cov. %	Model Acc.	# Q	Dist. ℓ_2
Bandit	Score	ℓ_∞	100.0	1.0	67.9	0.0	32	4.90
NES	Score	ℓ_∞	100.0	8.5	20.7	0.0	256	1.36
Parsimonious	Score	ℓ_∞	100.0	2.0	96.3	0.0	83	5.01
Sign	Score	ℓ_∞	100.0	2.0	92.2	0.0	94	4.99
Square	Score	ℓ_∞	100.0	2.0	71.6	0.0	17	4.99
ZOSignSGD	Score	ℓ_∞	100.0	2.0	50.6	0.0	248	1.30
GeoDA	Label	ℓ_∞	100.0	1.0	89.6	9.8	215	2.84
HSJ	Label	ℓ_∞	100.0	361.1	85.2	0.5	683	3.17
Opt	Label	ℓ_∞	89.0	9.0	85.3	50.5	2290	0.86
RayS	Label	ℓ_∞	100.0	5.8	78.9	0.0	251	4.82
SignFlip	Label	ℓ_∞	99.5	246.8	56.3	0.5	389	4.22
SignOPT	Label	ℓ_∞	92.5	8.4	88.4	31.0	1270	1.08
Bandit	Score	ℓ_2	100.0	1.0	98.7	0.0	109	2.65
NES	Score	ℓ_2	100.0	8.1	32.5	0.0	762	0.53
Simple	Score	ℓ_2	100.0	1.0	99.7	0.0	703	0.78
Square	Score	ℓ_2	100.0	2.0	74.2	0.0	21	4.55
ZOSignSGD	Score	ℓ_2	100.0	2.0	53.3	0.0	1340	0.45
Boundary	Label	ℓ_2	100.0	341.4	71.5	35.4	438	2.38
GeoDA	Label	ℓ_2	100.0	1.0	89.7	8.3	274	2.65
HSJ	Label	ℓ_2	100.0	358.4	82.1	0.4	497	2.75
Opt	Label	ℓ_2	90.1	9.9	64.4	40.0	660	2.34
SignOPT	Label	ℓ_2	88.1	8.5	75.6	35.3	414	1.67
PointWise	Label	Opt.	91.3	1.0	99.6	8.0	888	1.92
SparseEvo	Label	Opt.	87.9	1.0	100.0	8.8	8796	2.57
CA (sssp)	Label	Opt.	0.0	∞	0.0	8.3	437	3.95
CA (bin search)	Label	Opt.	0.0	∞	0.0	10.7	421	4.09

Table 7.10: Attack performance under Blacklight detection on ResNeXt and CIFAR10 (Clean Accuracy: 94.9%)

Attack	Query Type	Pert. Type	Det. Rate %	# Q to Det.	Det. Cov. %	Model Acc.	# Q	Dist. ℓ_2
Bandit	Score	ℓ_∞	100.0	1.0	67.3	0.0	27	5.07
NES	Score	ℓ_∞	100.0	8.5	20.4	0.0	213	1.28
Parsimonious	Score	ℓ_∞	100.0	2.0	97.3	0.0	104	5.16
Sign	Score	ℓ_∞	100.0	2.0	93.6	0.0	75	5.15
Square	Score	ℓ_∞	100.0	2.0	69.7	0.0	15	5.15
ZOSignSGD	Score	ℓ_∞	100.0	2.0	50.6	0.0	222	1.27
GeoDA	Label	ℓ_∞	100.0	1.0	89.9	10.1	197	2.22
HSJ	Label	ℓ_∞	100.0	148.1	85.9	0.0	383	2.65
Opt	Label	ℓ_∞	88.4	8.5	78.7	43.3	1644	0.87
RayS	Label	ℓ_∞	100.0	5.4	81.4	0.0	391	4.77
SignFlip	Label	ℓ_∞	100.0	165.8	41.5	0.0	205	3.43
SignOPT	Label	ℓ_∞	92.7	8.6	92.5	26.0	780	0.92
Bandit	Score	ℓ_2	100.0	1.0	98.8	0.0	110	2.81
NES	Score	ℓ_2	100.0	8.8	32.4	0.0	608	0.48
Simple	Score	ℓ_2	100.0	1.0	99.7	0.0	595	0.71
Square	Score	ℓ_2	100.0	2.0	74.2	0.0	20	4.69
ZOSignSGD	Score	ℓ_2	100.0	2.0	53.4	0.6	1376	0.46
Boundary	Label	ℓ_2	100.0	161.4	56.2	14.8	186	2.71
GeoDA	Label	ℓ_2	100.0	1.0	90.3	10.2	165	2.13
HSJ	Label	ℓ_2	100.0	138.3	83.9	0.0	287	2.16
Opt	Label	ℓ_2	88.4	8.6	60.8	37.2	480	1.65
SignOPT	Label	ℓ_2	88.4	8.6	87.5	32.7	387	1.17
PointWise	Label	Opt.	97.6	1.0	97.6	2.3	1084	2.01
SparseEvo	Label	Opt.	95.0	1.0	100.0	2.6	9506	3.11
CA (sssp)	Label	Opt.	0.0	∞	0.0	8.3	437	3.95
CA (bin search)	Label	Opt.	0.0	∞	0.0	10.7	421	4.09

Table 7.11: Attack performance under Blacklight detection on WRN and CIFAR10 (Clean Accuracy: 96.1%)

Attack	Query Type	Pert. Type	Det. Rate %	# Q to Det.	Det. Cov. %	Model Acc.	# Q	Dist. ℓ_2
Bandit	Score	ℓ_∞	100.0	1.0	67.4	0.0	47	5.09
NES	Score	ℓ_∞	100.0	8.7	20.7	0.0	295	1.44
Parsimonious	Score	ℓ_∞	100.0	2.0	97.6	0.2	140	5.22
Sign	Score	ℓ_∞	100.0	2.0	94.6	0.0	130	5.21
Square	Score	ℓ_∞	100.0	2.0	74.5	0.0	21	5.21
ZOSignSGD	Score	ℓ_∞	100.0	2.0	50.6	0.2	300	1.42
GeoDA	Label	ℓ_∞	100.0	1.0	89.6	0.1	323	2.83
HSJ	Label	ℓ_∞	100.0	7.3	91.9	0.0	280	2.69
Opt	Label	ℓ_∞	97.0	11.3	81.1	35.0	2179	1.15
RayS	Label	ℓ_∞	100.0	4.9	81.5	0.0	309	4.87
SignFlip	Label	ℓ_∞	100.0	8.3	53.0	0.0	72	3.75
SignOPT	Label	ℓ_∞	88.4	8.5	89.4	28.0	843	0.97
Bandit	Score	ℓ_2	100.0	1.0	98.8	0.0	136	2.71
NES	Score	ℓ_2	100.0	8.2	32.5	0.0	863	0.57
Simple	Score	ℓ_2	100.0	1.0	99.8	0.0	813	0.83
Square	Score	ℓ_2	100.0	2.0	73.2	0.0	20	4.75
ZOSignSGD	Score	ℓ_2	100.0	2.0	53.3	2.3	1803	0.54
Boundary	Label	ℓ_2	100.0	7.3	62.2	16.7	376	2.85
GeoDA	Label	ℓ_2	100.0	1.0	89.6	0.0	186	2.63
HSJ	Label	ℓ_2	100.0	7.4	91.0	0.0	185	2.27
Opt	Label	ℓ_2	97.3	10.7	66.3	35.1	678	2.14
SignOPT	Label	ℓ_2	91.7	8.5	80.3	33.0	470	1.57
PointWise	Label	Opt.	99.9	1.0	99.5	0.1	800	1.96
SparseEvo	Label	Opt.	97.7	1.0	100.0	0.2	9772	2.83
CA (sssp)	Label	Opt.	0.0	∞	0.0	6.8	417	3.7
CA (bin search)	Label	Opt.	0.0	∞	0.0	0.0	461	5.05

Table 7.12: Attack performance under Blacklight detection on VGG and CIFAR100 (Clean Accuracy: 68.6%)

Attack	Query Type	Pert. Type	Det. Rate %	# Q to Det.	Det. Cov. %	Model Acc.	# Q	Dist. ℓ_2
Bandit	Score	ℓ_∞	100.0	1.0	61.5	0.0	22	3.66
NES	Score	ℓ_∞	100.0	8.6	21.6	0.0	178	0.80
Parsimonious	Score	ℓ_∞	100.0	2.0	94.0	0.0	73	3.70
Sign	Score	ℓ_∞	100.0	2.0	84.6	0.0	44	3.68
Square	Score	ℓ_∞	100.0	2.0	64.7	0.0	9	3.68
ZOSignSGD	Score	ℓ_∞	100.0	2.0	50.7	0.1	184	0.79
GeoDA	Label	ℓ_∞	100.0	1.0	89.8	0.0	154	2.07
HSJ	Label	ℓ_∞	100.0	7.2	91.9	0.0	232	2.16
Opt	Label	ℓ_∞	100.0	8.6	74.7	20.8	1463	0.97
RayS	Label	ℓ_∞	100.0	6.6	71.7	0.0	120	4.40
SignFlip	Label	ℓ_∞	100.0	8.7	44.9	0.0	30	2.88
SignOPT	Label	ℓ_∞	100.0	8.5	92.3	23.0	699	0.84
Bandit	Score	ℓ_2	100.0	1.0	97.9	0.0	73	1.35
NES	Score	ℓ_2	100.0	8.7	32.4	0.0	553	0.31
Simple	Score	ℓ_2	100.0	1.0	99.4	0.0	507	0.44
Square	Score	ℓ_2	100.0	2.0	64.0	0.0	6	3.35
ZOSignSGD	Score	ℓ_2	100.0	2.0	53.6	0.4	1137	0.29
Boundary	Label	ℓ_2	100.0	7.3	54.0	6.3	100	2.49
GeoDA	Label	ℓ_2	100.0	1.0	90.2	0.0	125	1.95
HSJ	Label	ℓ_2	100.0	7.3	91.4	0.0	147	1.74
Opt	Label	ℓ_2	100.0	8.6	63.3	19.8	563	1.51
SignOPT	Label	ℓ_2	99.4	8.5	88.7	14.4	446	1.13
PointWise	Label	Opt.	100.0	1.0	98.9	0.0	300	1.42
SparseEvo	Label	Opt.	86.5	1.0	100.0	0.0	8647	1.88
CA (sssp)	Label	Opt.	0.0	∞	0.0	1.7	187	2.34
CA (bin search)	Label	Opt.	0.0	∞	0.0	0.0	457	2.70

Table 7.13: Attack performance under Blacklight detection on ResNet and CIFAR100 (Clean Accuracy: 66.8%)

Attack	Query Type	Pert. Type	Det. Rate %	# Q to Det.	Det. Cov. %	Model Acc.	# Q	Dist. ℓ_2
Bandit	Score	ℓ_∞	100.0	1.0	62.5	0.0	10	3.56
NES	Score	ℓ_∞	100.0	8.5	21.3	0.0	141	0.72
Parsimonious	Score	ℓ_∞	100.0	2.0	93.5	0.0	49	3.61
Sign	Score	ℓ_∞	100.0	2.0	82.4	0.0	31	3.59
Square	Score	ℓ_∞	100.0	2.0	66.2	0.0	8	3.58
ZOSignSGD	Score	ℓ_∞	100.0	2.0	50.7	0.0	148	0.70
GeoDA	Label	ℓ_∞	100.0	1.0	89.5	0.0	152	2.21
HSJ	Label	ℓ_∞	100.0	7.3	91.8	0.0	214	2.23
Opt	Label	ℓ_∞	100.0	8.5	75.1	25.8	1442	0.96
RayS	Label	ℓ_∞	100.0	6.6	70.3	0.0	109	4.02
SignFlip	Label	ℓ_∞	100.0	8.7	49.0	0.0	39	3.28
SignOPT	Label	ℓ_∞	100.0	8.5	89.5	17.0	749	0.88
Bandit	Score	ℓ_2	100.0	1.0	97.8	0.0	63	1.25
NES	Score	ℓ_2	100.0	8.7	32.5	0.0	404	0.27
Simple	Score	ℓ_2	100.0	1.0	99.5	0.0	371	0.39
Square	Score	ℓ_2	100.0	2.0	64.4	0.0	6	3.26
ZOSignSGD	Score	ℓ_2	100.0	2.0	53.5	0.1	747	0.23
Boundary	Label	ℓ_2	100.0	7.3	56.4	9.1	105	2.67
GeoDA	Label	ℓ_2	100.0	1.0	89.9	0.0	130	2.02
HSJ	Label	ℓ_2	100.0	7.3	91.2	0.0	151	1.81
Opt	Label	ℓ_2	100.0	8.5	63.2	21.6	567	1.58
SignOPT	Label	ℓ_2	100.0	8.5	85.8	17.3	472	1.20
PointWise	Label	Opt.	100.0	1.0	99.2	0.0	468	1.52
SparseEvo	Label	Opt.	90.9	1.0	100.0	0.0	9088	2.18
CA (sssp)	Label	Opt.	0.0	∞	0.0	1.4	191	2.39
CA (bin search)	Label	Opt.	0.0	∞	0.0	0.0	458	3.05

Table 7.14: Attack performance under Blacklight detection on ResNeXt and CIFAR100 (Clean Accuracy: 80.0%)

Attack	Query Type	Pert. Type	Det. Rate %	# Q to Det.	Det. Cov. %	Model Acc.	# Q	Dist. ℓ_2
Bandit	Score	ℓ_∞	100.0	1.0	62.3	0.0	13	4.28
NES	Score	ℓ_∞	100.0	8.6	21.6	0.0	145	0.87
Parsimonious	Score	ℓ_∞	100.0	2.0	95.5	0.0	73	4.32
Sign	Score	ℓ_∞	100.0	2.0	87.7	0.0	47	4.30
Square	Score	ℓ_∞	100.0	2.0	63.7	0.0	7	4.30
ZOSignSGD	Score	ℓ_∞	100.0	2.0	50.7	0.0	158	0.89
GeoDA	Label	ℓ_∞	100.0	1.0	90.1	0.1	136	1.91
HSJ	Label	ℓ_∞	100.0	7.3	92.0	0.0	226	1.96
Opt	Label	ℓ_∞	99.0	8.6	73.5	32.1	1158	0.84
RayS	Label	ℓ_∞	100.0	6.5	75.2	0.0	175	4.44
SignFlip	Label	ℓ_∞	100.0	8.8	42.2	0.0	27	2.52
SignOPT	Label	ℓ_∞	98.8	8.5	92.7	14.0	616	0.79
Bandit	Score	ℓ_2	100.0	1.0	98.0	0.0	69	1.62
NES	Score	ℓ_2	100.0	8.7	31.7	0.0	420	0.33
Simple	Score	ℓ_2	100.0	1.0	99.5	0.0	409	0.48
Square	Score	ℓ_2	100.0	2.0	65.8	0.0	8	3.92
ZOSignSGD	Score	ℓ_2	100.0	2.0	53.5	0.0	1029	0.33
Boundary	Label	ℓ_2	100.0	7.3	51.1	5.7	53	2.26
GeoDA	Label	ℓ_2	100.0	1.0	90.6	0.1	120	1.84
HSJ	Label	ℓ_2	100.0	7.3	91.7	0.0	146	1.53
Opt	Label	ℓ_2	99.0	8.6	62.1	21.7	537	1.46
SignOPT	Label	ℓ_2	98.9	8.6	89.5	16.5	432	1.07
PointWise	Label	Opt.	100.0	1.0	99.4	0.0	766	1.80
SparseEvo	Label	Opt.	93.8	1.0	100	0.0	9376	2.53
CA (sssp)	Label	Opt.	0.0	∞	0.0	2.1	174	2.31
CA (bin search)	Label	Opt.	0.0	∞	0.0	0.0	459	2.61

Table 7.15: Attack performance under Blacklight detection on WRN and CIFAR100 (Clean Accuracy: 79.4%)

Attack	Query Type	Pert. Type	Det. Rate %	# Q to Det.	Det. Cov. %	Model Acc.	# Q	Dist. ℓ_2
Bandit	Score	ℓ_∞	100.0	1.0	62.1	0.0	13	4.25
NES	Score	ℓ_∞	100.0	8.8	21.7	0.0	151	0.87
Parsimonious	Score	ℓ_∞	100.0	2.0	95.2	0.0	75	4.29
Sign	Score	ℓ_∞	100.0	2.0	87.2	0.0	54	4.26
Square	Score	ℓ_∞	100.0	2.0	65.1	0.0	10	4.26
ZOSignSGD	Score	ℓ_∞	100.0	2.0	50.6	0.0	159	0.87
GeoDA	Label	ℓ_∞	100.0	1.0	90.0	0.0	139	1.93
HSJ	Label	ℓ_∞	100.0	7.3	91.9	0.0	185	1.89
Opt	Label	ℓ_∞	100.0	8.6	73.8	24.1	1269	0.93
RayS	Label	ℓ_∞	100.0	6.0	74.1	0.0	165	4.20
SignFlip	Label	ℓ_∞	100.0	8.7	43.1	0.0	29	2.70
SignOPT	Label	ℓ_∞	100.0	8.6	92.9	19.0	624	0.75
Bandit	Score	ℓ_2	100.0	1.0	97.7	0.0	70	1.54
NES	Score	ℓ_2	100.0	8.9	31.8	0.0	442	0.32
Simple	Score	ℓ_2	100.0	1.0	99.4	0.0	437	0.48
Square	Score	ℓ_2	100.0	2.0	65.2	0.0	7	3.88
ZOSignSGD	Score	ℓ_2	100.0	2.0	53.6	0.5	964	0.31
Boundary	Label	ℓ_2	100.0	7.2	52.0	4.8	81	2.46
GeoDA	Label	ℓ_2	100.0	1.0	90.4	0.0	120	1.86
HSJ	Label	ℓ_2	100.0	7.3	91.5	0.0	143	1.55
Opt	Label	ℓ_2	100.0	8.6	65.2	22.2	586	1.35
SignOPT	Label	ℓ_2	99.9	8.5	89.0	16.7	456	1.04
PointWise	Label	Opt.	100.0	1.0	99.1	0.0	469	1.57
SparseEvo	Label	Opt.	91.4	1.0	100.0	0.0	9145	2.16
CA (sssp)	Label	Opt.	0.0	∞	0.0	1.6	218	2.56
CA (bin search)	Label	Opt.	0.0	∞	0.0	0.0	458	2.65

Table 7.16: Attack performance under RAND Pre-processing Defense on VGG and CIFAR10 (Clean Accuracy: 87.7%)

Attack	Query Type	Perturbation Type	# Query	Model Acc.	Dist. ℓ_2
Bandit	Score	ℓ_∞	116	36.1	4.68
NES	Score	ℓ_∞	612	50.7	1.69
Parsimonious	Score	ℓ_∞	565	63.7	4.79
Sign	Score	ℓ_∞	257	55.3	4.75
Square	Score	ℓ_∞	91	32.9	4.80
ZOSignSGD	Score	ℓ_∞	644	49.9	1.72
GeoDA	Label	ℓ_∞	324	58.1	2.58
HSJ	Label	ℓ_∞	591	59.7	2.26
Opt	Label	ℓ_∞	957	87.3	0.25
RayS	Label	ℓ_∞	251	60.1	4.60
SignFlip	Label	ℓ_∞	365	51.1	3.31
SignOPT	Label	ℓ_∞	1411	82.0	0.30
Bandit	Score	ℓ_2	773	76.5	3.36
NES	Score	ℓ_2	1875	72.2	0.75
Simple	Score	ℓ_2	1693	89.3	0.14
Square	Score	ℓ_2	100	33.9	4.33
ZOSignSGD	Score	ℓ_2	2840	78.3	0.63
Boundary	Label	ℓ_2	43	54.4	2.46
GeoDA	Label	ℓ_2	299	62.0	2.35
HSJ	Label	ℓ_2	735	56.5	3.20
Opt	Label	ℓ_2	807	69.4	1.92
SignOPT	Label	ℓ_2	586	50.9	2.76
PointWise	Label	Optimized	2813	83.0	2.02
SparseEvo	Label	Optimized	9492	80.8	1.62
CA (sssp)	Label	Optimized	359	5.7	3.53
CA (bin search)	Label	Optimized	473	0.0	4.94

Table 7.17: Attack performance under RAND Pre-processing Defense on ResNet and CIFAR10 (Clean Accuracy 92.3%)

Attack	Query Type	Perturbation Type	# Query	Model Acc.	Dist. ℓ_2
Bandit	Score	ℓ_∞	52	27.2	4.88
NES	Score	ℓ_∞	605	49.9	1.84
Parsimonious	Score	ℓ_∞	254	50.1	4.97
Sign	Score	ℓ_∞	394	58.3	4.85
Square	Score	ℓ_∞	32	21.4	4.95
ZOSignSGD	Score	ℓ_∞	674	51.4	1.86
GeoDA	Label	ℓ_∞	265	62.8	2.53
HSJ	Label	ℓ_∞	641	64.6	2.12
Opt	Label	ℓ_∞	655	88.2	0.18
RayS	Label	ℓ_∞	306	62.9	4.75
SignFlip	Label	ℓ_∞	902	56.5	3.33
SignOPT	Label	ℓ_∞	1086	85.6	0.24
Bandit	Score	ℓ_2	747	76.0	3.45
NES	Score	ℓ_2	1808	71.2	0.77
Simple	Score	ℓ_2	2139	90.4	0.15
Square	Score	ℓ_2	44	24.2	4.49
ZOSignSGD	Score	ℓ_2	2822	77.6	0.64
Boundary	Label	ℓ_2	46	60.1	2.25
GeoDA	Label	ℓ_2	333	66.7	2.14
HSJ	Label	ℓ_2	1163	56.7	3.45
Opt	Label	ℓ_2	724	72.8	1.93
SignOPT	Label	ℓ_2	442	59.1	2.47
PointWise	Label	Optimized	3804	86.5	2.02
SparseEvo	Label	Optimized	8709	87.3	1.78
CA (sssp)	Label	Optimized	406	8.1	3.80
CA (bin search)	Label	Optimized	417	10.8	3.89

Table 7.18: Attack performance under RAND Pre-processing Defense on ResNeXt and CIFAR10 (Clean Accuracy: 89.6%)

Attack	Query Type	Perturbation Type	# Query	Model Acc.	Dist. ℓ_2
Bandit	Score	ℓ_∞	54	25.9	4.81
NES	Score	ℓ_∞	446	48.2	1.61
Parsimonious	Score	ℓ_∞	537	68.1	4.88
Sign	Score	ℓ_∞	454	63.8	4.75
Square	Score	ℓ_∞	90	24.2	4.91
ZOSignSGD	Score	ℓ_∞	487	49.4	1.64
GeoDA	Label	ℓ_∞	197	58.9	2.43
HSJ	Label	ℓ_∞	363	59.3	2.30
Opt	Label	ℓ_∞	771	89.8	0.30
RayS	Label	ℓ_∞	305	65.3	4.60
SignFlip	Label	ℓ_∞	334	54.0	2.91
SignOPT	Label	ℓ_∞	895	84.9	0.33
Bandit	Score	ℓ_2	407	75.1	3.03
NES	Score	ℓ_2	1611	74.6	0.70
Simple	Score	ℓ_2	1692	90.2	0.14
Square	Score	ℓ_2	123	34.0	4.41
ZOSignSGD	Score	ℓ_2	2582	80.9	0.59
Boundary	Label	ℓ_2	18	52.9	2.50
GeoDA	Label	ℓ_2	146	58.9	2.32
HSJ	Label	ℓ_2	593	57.7	2.86
Opt	Label	ℓ_2	694	76.1	1.58
SignOPT	Label	ℓ_2	337	67.1	1.91
PointWise	Label	Optimized	4516	84.5	2.18
SparseEvo	Label	Optimized	9632	87.4	1.85
CA (sssp)	Label	Optimized	259	9.8	2.77
CA (bin search)	Label	Optimized	416	10.6	2.75

Table 7.19: Attack performance under RAND Pre-processing Defense on WRN and CIFAR10 (Clean Accuracy: 92.6%)

Attack	Query Type	Perturbation Type	# Query	Model Acc.	Dist. ℓ_2
Bandit	Score	ℓ_∞	55	28.0	4.92
NES	Score	ℓ_∞	600	51.6	1.78
Parsimonious	Score	ℓ_∞	695	71.4	5.04
Sign	Score	ℓ_∞	627	70.6	4.82
Square	Score	ℓ_∞	142	28.6	5.00
ZOSignSGD	Score	ℓ_∞	642	52.3	1.80
GeoDA	Label	ℓ_∞	202	60.1	2.54
HSJ	Label	ℓ_∞	468	61.4	2.30
Opt	Label	ℓ_∞	1009	90.5	0.24
RayS	Label	ℓ_∞	454	69.2	4.62
SignFlip	Label	ℓ_∞	280	52.4	3.04
SignOPT	Label	ℓ_∞	1145	86.4	0.28
Bandit	Score	ℓ_2	616	77.1	3.22
NES	Score	ℓ_2	2074	77.0	0.82
Simple	Score	ℓ_2	1639	92.0	0.15
Square	Score	ℓ_2	62	30.9	4.57
ZOSignSGD	Score	ℓ_2	2982	82.0	0.68
Boundary	Label	ℓ_2	36	56.1	2.53
GeoDA	Label	ℓ_2	191	60.9	2.35
HSJ	Label	ℓ_2	627	57.2	2.97
Opt	Label	ℓ_2	759	73.5	1.69
SignOPT	Label	ℓ_2	527	60.5	2.34
PointWise	Label	Optimized	3721	87.2	1.90
SparseEvo	Label	Optimized	9730	89.7	1.7
CA (sssp)	Label	Optimized	401	6.1	3.63
CA (bin search)	Label	Optimized	461	0.0	4.80

Table 7.20: Attack performance under RAND Pre-processing Defense on VGG and CIFAR100 (Clean Accuracy: 61.4%)

Attack	Query Type	Perturbation Type	# Query	Model Acc.	Dist. ℓ_2
Bandit	Score	ℓ_∞	9	8.8	3.28
NES	Score	ℓ_∞	326	34.4	0.91
Parsimonious	Score	ℓ_∞	218	37.1	3.26
Sign	Score	ℓ_∞	138	32.5	3.26
Square	Score	ℓ_∞	25	9.0	3.31
ZOSignSGD	Score	ℓ_∞	376	32.9	0.93
GeoDA	Label	ℓ_∞	139	40.5	2.36
HSJ	Label	ℓ_∞	301	43.7	2.12
Opt	Label	ℓ_∞	908	66.1	0.38
RayS	Label	ℓ_∞	127	46.4	4.03
SignFlip	Label	ℓ_∞	194	43.9	2.64
SignOPT	Label	ℓ_∞	997	60.3	0.46
Bandit	Score	ℓ_2	187	43.9	1.43
NES	Score	ℓ_2	1215	48.5	0.39
Simple	Score	ℓ_2	1335	60.4	0.09
Square	Score	ℓ_2	37	10.0	2.99
ZOSignSGD	Score	ℓ_2	1804	54.8	0.33
Boundary	Label	ℓ_2	26	41.0	2.48
GeoDA	Label	ℓ_2	150	43.5	2.20
HSJ	Label	ℓ_2	278	46.3	2.41
Opt	Label	ℓ_2	765	56.5	1.44
SignOPT	Label	ℓ_2	390	51.1	1.93
PointWise	Label	Optimized	956	51.3	1.07
SparseEvo	Label	Optimized	8793	53.2	0.95
CA (sssp)	Label	Optimized	168	0.9	2.25
CA (bin search)	Label	Optimized	457	0.0	2.52

Table 7.21: Attack performance under RAND Pre-processing Defense on ResNet and CIFAR100 (Clean Accuracy: 62.4%)

Attack	Query Type	Perturbation Type	# Query	Model Acc.	Dist. ℓ_2
Bandit	Score	ℓ_∞	13	10.2	3.36
NES	Score	ℓ_∞	279	36.4	0.92
Parsimonious	Score	ℓ_∞	85	31.5	3.39
Sign	Score	ℓ_∞	221	33.1	3.33
Square	Score	ℓ_∞	12	9.7	3.48
ZOSignSGD	Score	ℓ_∞	330	34.9	0.94
GeoDA	Label	ℓ_∞	220	40.5	2.34
HSJ	Label	ℓ_∞	472	43.6	2.17
Opt	Label	ℓ_∞	907	62.7	0.32
RayS	Label	ℓ_∞	119	44.0	3.97
SignFlip	Label	ℓ_∞	383	40.0	2.93
SignOPT	Label	ℓ_∞	912	57.9	0.43
Bandit	Score	ℓ_2	291	47.8	1.47
NES	Score	ℓ_2	964	51.0	0.36
Simple	Score	ℓ_2	1152	62.8	0.09
Square	Score	ℓ_2	6	10.0	3.05
ZOSignSGD	Score	ℓ_2	1514	55.2	0.30
Boundary	Label	ℓ_2	16	37.7	2.52
GeoDA	Label	ℓ_2	256	44.3	2.14
HSJ	Label	ℓ_2	381	43.8	2.56
Opt	Label	ℓ_2	806	54.5	1.46
SignOPT	Label	ℓ_2	452	46.0	1.96
PointWise	Label	Optimized	2051	52.4	1.21
SparseEvo	Label	Optimized	9043	56.3	1.12
CA (sssp)	Label	Optimized	167	1.6	2.27
CA (bin search)	Label	Optimized	459	0.0	2.81

Table 7.22: Attack performance under RAND Pre-processing Defense on ResNeXt and CIFAR100 (Clean Accuracy: 65.0%)

Attack	Query Type	Perturbation Type	# Query	Model Acc.	Dist. ℓ_2
Bandit	Score	ℓ_∞	20	9.2	3.43
NES	Score	ℓ_∞	311	35.3	0.95
Parsimonious	Score	ℓ_∞	359	40.7	3.54
Sign	Score	ℓ_∞	309	38.2	3.44
Square	Score	ℓ_∞	14	8.1	3.45
ZOSignSGD	Score	ℓ_∞	348	33.2	0.97
GeoDA	Label	ℓ_∞	152	48.6	2.06
HSJ	Label	ℓ_∞	184	51.3	1.94
Opt	Label	ℓ_∞	951	73.9	0.44
RayS	Label	ℓ_∞	203	56.7	4.23
SignFlip	Label	ℓ_∞	132	49.3	2.32
SignOPT	Label	ℓ_∞	798	71.8	0.45
Bandit	Score	ℓ_2	188	45.7	1.49
NES	Score	ℓ_2	1117	54.3	0.41
Simple	Score	ℓ_2	1607	65.1	0.09
Square	Score	ℓ_2	31	13.9	3.19
ZOSignSGD	Score	ℓ_2	1964	58.3	0.35
Boundary	Label	ℓ_2	17	49.7	2.21
GeoDA	Label	ℓ_2	118	52.3	2.00
HSJ	Label	ℓ_2	301	51.9	2.19
Opt	Label	ℓ_2	741	65.1	1.35
SignOPT	Label	ℓ_2	399	56.7	1.76
PointWise	Label	Optimized	1884	55.9	1.20
SparseEvo	Label	Optimized	9364	61.2	1.03
CA (sssp)	Label	Optimized	159	1.4	2.21
CA (bin search)	Label	Optimized	459	0.0	2.43

Table 7.23: Attack performance under RAND Pre-processing Defense on WRN and CIFAR100 (Clean Accuracy: 65.5%)

Attack	Query Type	Perturbation Type	# Query	Model Acc.	Dist. ℓ_2
Bandit	Score	ℓ_∞	12	11.1	3.52
NES	Score	ℓ_∞	334	36.9	0.98
Parsimonious	Score	ℓ_∞	282	43.2	3.60
Sign	Score	ℓ_∞	144	43.9	3.51
Square	Score	ℓ_∞	21	11.9	3.57
ZOSignSGD	Score	ℓ_∞	383	36.9	0.99
GeoDA	Label	ℓ_∞	136	52.1	2.21
HSJ	Label	ℓ_∞	312	52.1	2.09
Opt	Label	ℓ_∞	866	74.4	0.37
RayS	Label	ℓ_∞	212	56.1	4.07
SignFlip	Label	ℓ_∞	146	50.0	2.47
SignOPT	Label	ℓ_∞	929	69.6	0.39
Bandit	Score	ℓ_2	175	50.2	1.53
NES	Score	ℓ_2	1131	55.3	0.41
Simple	Score	ℓ_2	1268	65.2	0.09
Square	Score	ℓ_2	20	13.1	3.19
ZOSignSGD	Score	ℓ_2	1702	57.2	0.34
Boundary	Label	ℓ_2	19	48.1	2.49
GeoDA	Label	ℓ_2	171	50.7	2.17
HSJ	Label	ℓ_2	250	50.3	2.34
Opt	Label	ℓ_2	753	65.7	1.39
SignOPT	Label	ℓ_2	395	56.0	1.81
PointWise	Label	Optimized	1617	55.1	1.07
SparseEvo	Label	Optimized	9157	59.1	0.85
CA (sssp)	Label	Optimized	180	1.2	2.37
CA (bin search)	Label	Optimized	457	0.0	2.49

Table 7.24: Attack performance under RAND Post-processing Defense on VGG and CIFAR10 (Clean Accuracy: 90.5%)

Attack	Query Type	Perturbation Type	# Query	Model Acc.	Dist. ℓ_2
Bandit	Score	ℓ_∞	192	10.9	4.83
NES	Score	ℓ_∞	364	0.0	1.43
Parsimonious	Score	ℓ_∞	215	8.8	4.90
Sign	Score	ℓ_∞	134	1.2	4.88
Square	Score	ℓ_∞	37	0.4	4.89
ZOSignSGD	Score	ℓ_∞	389	0.1	1.40
GeoDA	Label	ℓ_∞	371	52.9	2.69
HSJ	Label	ℓ_∞	518	53.7	2.43
Opt	Label	ℓ_∞	970	78.0	0.31
RayS	Label	ℓ_∞	235	45.0	4.77
SignFlip	Label	ℓ_∞	76	17.2	3.91
SignOPT	Label	ℓ_∞	1177	34.2	1.13
Bandit	Score	ℓ_2	999	25.6	3.78
NES	Score	ℓ_2	1137	0.1	0.59
Simple	Score	ℓ_2	3138	67.1	0.21
Square	Score	ℓ_2	41	0.2	4.47
ZOSignSGD	Score	ℓ_2	1955	1.6	0.53
Boundary	Label	ℓ_2	57	52.3	2.14
GeoDA	Label	ℓ_2	545	52.7	2.69
HSJ	Label	ℓ_2	234	45.4	2.90
Opt	Label	ℓ_2	641	53.3	2.02
SignOPT	Label	ℓ_2	504	34.8	2.00
PointWise	Label	Optimized	1290	89.8	2.02
SparseEvo	Label	Optimized	9425	38.3	2.27
CA (sssp)	Label	Optimized	400	5.7	3.75
CA (bin search)	Label	Optimized	473	0.0	5.19

Table 7.25: Attack performance under RAND Post-processing Defense on ResNet and CIFAR10 (Clean Accuracy: 92.0%)

Attack	Query Type	Perturbation Type	# Query	Model Acc.	Dist. ℓ_2
Bandit	Score	ℓ_∞	153	10.3	4.93
NES	Score	ℓ_∞	277	0.0	1.39
Parsimonious	Score	ℓ_∞	104	0.2	5.01
Sign	Score	ℓ_∞	144	0.7	4.98
Square	Score	ℓ_∞	18	0.0	4.99
ZOSignSGD	Score	ℓ_∞	274	0.0	1.35
GeoDA	Label	ℓ_∞	380	60.2	2.71
HSJ	Label	ℓ_∞	792	60.7	2.21
Opt	Label	ℓ_∞	658	84.9	0.21
RayS	Label	ℓ_∞	252	46.8	4.79
SignFlip	Label	ℓ_∞	442	17.0	4.26
SignOPT	Label	ℓ_∞	1095	42.0	1.00
Bandit	Score	ℓ_2	673	20.7	3.76
NES	Score	ℓ_2	840	0.0	0.54
Simple	Score	ℓ_2	5021	64.4	0.30
Square	Score	ℓ_2	19	0.0	4.55
ZOSignSGD	Score	ℓ_2	1443	0.1	0.47
Boundary	Label	ℓ_2	85	64.4	1.68
GeoDA	Label	ℓ_2	545	57.6	2.54
HSJ	Label	ℓ_2	558	48.5	3.32
Opt	Label	ℓ_2	593	59.4	1.98
SignOPT	Label	ℓ_2	397	41.3	1.90
PointWise	Label	unrestricted	1735	91.1	1.90
SparseEvo	Label	unrestricted	8665	63.2	2.41
CA(sssp)	Label	unrestricted	425	9.1	3.90
CA(bin search)	Label	unrestricted	421	10.7	4.09

Table 7.26: Attack performance under RAND Post-processing Defense on ResNeXt and CIFAR10 (Clean Accuracy: 94.8%)

Attack	Query Type	Perturbation Type	# Query	Model Acc.	Dist. ℓ_2
Bandit	Score	ℓ_∞	141	4.7	5.08
NES	Score	ℓ_∞	237	0.0	1.32
Parsimonious	Score	ℓ_∞	221	2.1	5.15
Sign	Score	ℓ_∞	133	0.0	5.14
Square	Score	ℓ_∞	23	0.4	5.14
ZOSignSGD	Score	ℓ_∞	249	0.0	1.31
GeoDA	Label	ℓ_∞	282	53.1	2.52
HSJ	Label	ℓ_∞	417	57.4	2.41
Opt	Label	ℓ_∞	683	81.6	0.36
RayS	Label	ℓ_∞	484	52.5	4.72
SignFlip	Label	ℓ_∞	287	28.1	3.35
SignOPT	Label	ℓ_∞	760	45.3	0.91
Bandit	Score	ℓ_2	574	14.2	3.69
NES	Score	ℓ_2	694	0.0	0.50
Simple	Score	ℓ_2	4075	61.7	0.29
Square	Score	ℓ_2	22	0.0	4.68
ZOSignSGD	Score	ℓ_2	1459	0.6	0.47
Boundary	Label	ℓ_2	20	51.1	2.36
GeoDA	Label	ℓ_2	206	54.5	2.48
HSJ	Label	ℓ_2	414	48.6	2.84
Opt	Label	ℓ_2	542	61.2	1.70
SignOPT	Label	ℓ_2	357	44.2	1.50
PointWise	Label	Optimized	2519	94.6	2.05
SparseEvo	Label	Optimized	9537	78.4	2.85
CA (sssp)	Label	Optimized	276	9.3	2.92
CA (bin search)	Label	Optimized	434	10.0	3.05

Table 7.27: Attack performance under RAND Post-processing Defense on WRN and CIFAR10 (Clean Accuracy: 96.1%)

Attack	Query Type	Perturbation Type	# Query	Model Acc.	Dist. ℓ_2
Bandit	Score	ℓ_∞	157	6.2	5.12
NES	Score	ℓ_∞	464	0.2	1.59
Parsimonious	Score	ℓ_∞	390	21.1	5.22
Sign	Score	ℓ_∞	330	12.4	5.02
Square	Score	ℓ_∞	36	0.5	5.21
ZOSignSGD	Score	ℓ_∞	464	0.4	1.56
GeoDA	Label	ℓ_∞	186	56.9	2.58
HSJ	Label	ℓ_∞	364	56.0	2.37
Opt	Label	ℓ_∞	1022	84.8	0.31
RayS	Label	ℓ_∞	334	51.5	4.81
SignFlip	Label	ℓ_∞	164	22.8	3.72
SignOPT	Label	ℓ_∞	968	39.2	0.98
Bandit	Score	ℓ_2	727	17.5	3.81
NES	Score	ℓ_2	1328	1.6	0.68
Simple	Score	ℓ_2	2536	74.7	0.21
Square	Score	ℓ_2	26	0.4	4.75
ZOSignSGD	Score	ℓ_2	2143	5.5	0.60
Boundary	Label	ℓ_2	35	54.0	2.35
GeoDA	Label	ℓ_2	267	58.3	2.53
HSJ	Label	ℓ_2	360	48.8	2.82
Opt	Label	ℓ_2	620	59.7	1.89
SignOPT	Label	ℓ_2	476	43.4	1.65
PointWise	Label	Optimized	1727	95.8	1.89
SparseEvo	Label	Optimized	9720	59.9	2.67
CA (sssp)	Label	Optimized	399	7.3	3.65
CA (bin search)	Label	Optimized	460	0.0	4.94

Table 7.28: Attack performance under RAND Post-processing Defense on VGG and CIFAR100 (Clean Accuracy: 68.5%)

Attack	Query Type	Perturbation Type	# Query	Model Acc.	Dist. ℓ_2
Bandit	Score	ℓ_∞	35	1.0	3.67
NES	Score	ℓ_∞	190	0.2	0.83
Parsimonious	Score	ℓ_∞	103	2.0	3.71
Sign	Score	ℓ_∞	72	0.7	3.66
Square	Score	ℓ_∞	8	0.1	3.66
ZOSignSGD	Score	ℓ_∞	203	0.1	0.82
GeoDA	Label	ℓ_∞	177	35.6	2.31
HSJ	Label	ℓ_∞	235	36.7	2.19
Opt	Label	ℓ_∞	756	53.3	0.48
RayS	Label	ℓ_∞	116	32.2	4.14
SignFlip	Label	ℓ_∞	40	14.4	2.89
SignOPT	Label	ℓ_∞	724	29.4	0.93
Bandit	Score	ℓ_2	139	2.3	1.86
NES	Score	ℓ_2	650	0.1	0.33
Simple	Score	ℓ_2	2570	34.7	0.18
Square	Score	ℓ_2	6	0.1	3.35
ZOSignSGD	Score	ℓ_2	1225	0.5	0.30
Boundary	Label	ℓ_2	40	28.8	2.31
GeoDA	Label	ℓ_2	205	36.3	2.31
HSJ	Label	ℓ_2	169	32.8	2.11
Opt	Label	ℓ_2	581	37.7	1.54
SignOPT	Label	ℓ_2	409	27.1	1.32
PointWise	Label	Optimized	494	67.5	1.38
SparseEvo	Label	Optimized	8502	21.1	1.75
CA (sssp)	Label	Optimized	186	1.8	2.35
CA (bin search)	Label	Optimized	582	0.0	2.67

Table 7.29: Attack performance under RAND Post-processing Defense on ResNet and CIFAR100 (Clean Accuracy: 67.5%)

Attack	Query Type	Perturbation Type	# Query	Model Acc.	Dist. ℓ_2
Bandit	Score	ℓ_∞	46	1.0	3.57
NES	Score	ℓ_∞	152	0.0	0.75
Parsimonious	Score	ℓ_∞	56	0.0	3.63
Sign	Score	ℓ_∞	39	0.0	3.57
Square	Score	ℓ_∞	7	0.0	3.61
ZOSignSGD	Score	ℓ_∞	159	0.0	0.72
GeoDA	Label	ℓ_∞	245	38.7	2.49
HSJ	Label	ℓ_∞	407	38.2	2.28
Opt	Label	ℓ_∞	762	51.6	0.47
RayS	Label	ℓ_∞	99	32.7	4.03
SignFlip	Label	ℓ_∞	47	17.5	3.25
SignOPT	Label	ℓ_∞	839	28.2	0.97
Bandit	Score	ℓ_2	211	3.2	1.86
NES	Score	ℓ_2	431	0.0	0.28
Simple	Score	ℓ_2	3593	36.0	0.20
Square	Score	ℓ_2	6	0.0	3.28
ZOSignSGD	Score	ℓ_2	791	0.0	0.24
Boundary	Label	ℓ_2	56	34.2	2.35
GeoDA	Label	ℓ_2	387	35.6	2.38
HSJ	Label	ℓ_2	197	33.2	2.40
Opt	Label	ℓ_2	633	40.2	1.60
SignOPT	Label	ℓ_2	413	27.0	1.53
PointWise	Label	Optimized	922	65.7	1.41
SparseEvo	Label	Optimized	9028	41.0	2.02
CA (sssp)	Label	Optimized	187	1.8	2.36
CA (bin search)	Label	Optimized	458	0.0	3.04

Table 7.30: Attack performance under RAND Post-processing Defense on ResNeXt and CIFAR100 (Clean Accuracy: 79.5%)

Attack	Query Type	Perturbation Type	# Query	Model Acc.	Dist. ℓ_2
Bandit	Score	ℓ_∞	46	1.3	4.28
NES	Score	ℓ_∞	165	0.0	0.92
Parsimonious	Score	ℓ_∞	117	0.5	4.33
Sign	Score	ℓ_∞	124	0.1	4.29
Square	Score	ℓ_∞	8	0.1	4.32
ZOSignSGD	Score	ℓ_∞	179	0.0	0.93
GeoDA	Label	ℓ_∞	147	40.7	2.04
HSJ	Label	ℓ_∞	201	41.4	1.91
Opt	Label	ℓ_∞	744	64.2	0.52
RayS	Label	ℓ_∞	179	45.7	4.32
SignFlip	Label	ℓ_∞	98	29.2	2.47
SignOPT	Label	ℓ_∞	844	43.8	0.86
Bandit	Score	ℓ_2	163	3.1	2.03
NES	Score	ℓ_2	461	0.0	0.34
Simple	Score	ℓ_2	3805	45.2	0.23
Square	Score	ℓ_2	10	0.0	3.90
ZOSignSGD	Score	ℓ_2	1054	0.0	0.33
Boundary	Label	ℓ_2	20	36.6	2.15
GeoDA	Label	ℓ_2	143	44.3	2.02
HSJ	Label	ℓ_2	256	40.6	2.13
Opt	Label	ℓ_2	590	49.5	1.39
SignOPT	Label	ℓ_2	371	38.0	1.56
PointWise	Label	Optimized	1811	78.7	1.75
SparseEvo	Label	Optimized	9476	62.6	2.20
CA (sssp)	Label	Optimized	179	2.0	2.33
CA (bin search)	Label	Optimized	458	0.0	2.60

Table 7.31: Attack performance under RAND Post-processing Defense on WRN and CIFAR100 (Clean Accuracy: 79.4%)

Attack	Query Type	Perturbation Type	# Query	Model Acc.	Dist. ℓ_2
Bandit	Score	ℓ_∞	35	1.8	4.26
NES	Score	ℓ_∞	170	0.0	0.89
Parsimonious	Score	ℓ_∞	134	1.1	4.26
Sign	Score	ℓ_∞	81	0.6	4.24
Square	Score	ℓ_∞	10	0.0	4.23
ZOSignSGD	Score	ℓ_∞	178	0.1	0.89
GeoDA	Label	ℓ_∞	146	40.9	2.22
HSJ	Label	ℓ_∞	224	42.3	2.03
Opt	Label	ℓ_∞	738	62.1	0.46
RayS	Label	ℓ_∞	168	41.9	4.22
SignFlip	Label	ℓ_∞	50	24.1	2.70
SignOPT	Label	ℓ_∞	761	36.2	0.89
Bandit	Score	ℓ_2	172	3.6	2.10
NES	Score	ℓ_2	523	0.1	0.34
Simple	Score	ℓ_2	2954	37.9	0.21
Square	Score	ℓ_2	15	0.0	3.89
ZOSignSGD	Score	ℓ_2	1009	0.7	0.32
Boundary	Label	ℓ_2	25	35.6	2.34
GeoDA	Label	ℓ_2	179	42.8	2.21
HSJ	Label	ℓ_2	214	38.0	2.18
Opt	Label	ℓ_2	597	46.7	1.41
SignOPT	Label	ℓ_2	397	35.0	1.37
PointWise	Label	Optimized	1145	77.1	1.50
SparseEvo	Label	Optimized	9145	49.3	1.94
CA (sssp)	Label	Optimized	210	1.6	2.53
CA (bin search)	Label	Optimized	457	0.0	2.65

7.14.5 Experiments on Audio Classification task

Dataset. The speaker verification task focuses on determining if a given voice sample belongs to a specific individual or ascribed identity [429]. The process for this task entails comparing two voice samples using a speaker verification model and making a decision. We utilize the large-scale multi-speaker corpus LibriSpeech ("train-clean-100" set), which comprises over 100 hours of read English voices from 251 speakers encompassing various accents, occupations, and age groups. Within this corpus, each speaker has multiple voice samples spanning from several seconds to tens of seconds, sampled at a rate of 16kHz.

Model and Setting. For the speaker verification task, we use two SOTA speaker verification models: ECAPA-TDNN [430] pre-trained by Speechbrain [431] as the target model for the model owner and the X-vector model [432] pre-trained by Speechbrain [431] as the feature extractor. In the experiments, we utilize the pre-trained SOTA model, which is trained on the VoxCeleb dataset [433] and VoxCeleb2 dataset [434], to perform the task of verifying if two voice samples are from the same speaker. We evaluate the performance of the model on 500 pairs of voice samples randomly selected from the LibriSpeech "train-clean-100" set. Each sample pair consists of two voice samples from a single speaker.

Table 7.32: Performance of certifiable attack with Gaussian Noise on Audio Dataset. ($p = 90\%$)

σ	Dist. ℓ_2	Mean Dist. ℓ_2	# RPQ	Certified Acc.
0.05	18.35	12.67	5.42	100.00%
0.1	26.71	12.70	3.72	100.00%
0.15	35.82	12.63	3.12	100.00%

Table 7.33: Performance of certifiable attack with Gaussian noise $\sigma = 0.25$ on LibriSpeech

p	Dist. ℓ_2	Mean Dist. ℓ_2	# RPQ	Certified Acc.
50%	26.64	12.55	5.31	100.00%
60%	26.65	12.57	5.08	100.00%
70%	26.66	12.60	4.87	100.00%
80%	26.68	12.63	4.49	100.00%
90%	26.71	12.70	3.72	100.00%
95%	26.74	12.78	3.05	100.00%

Table 7.34: Attack performance of different localization/refinement algorithms on LibriSpeec ($\sigma = 0.25$, $p = 90\%$).

Localization	Shifting	Dist. ℓ_2	Mean Dist. ℓ_2	# RPQ	Cert. Acc.
random	none	165.56	164.59	1.00	100.00%
random	geo.	159.21	157.53	73.88	100.00%
SSSP	none	26.72	12.74	1.32	100.00%
SSSP	geo.	26.71	12.70	3.72	100.00%

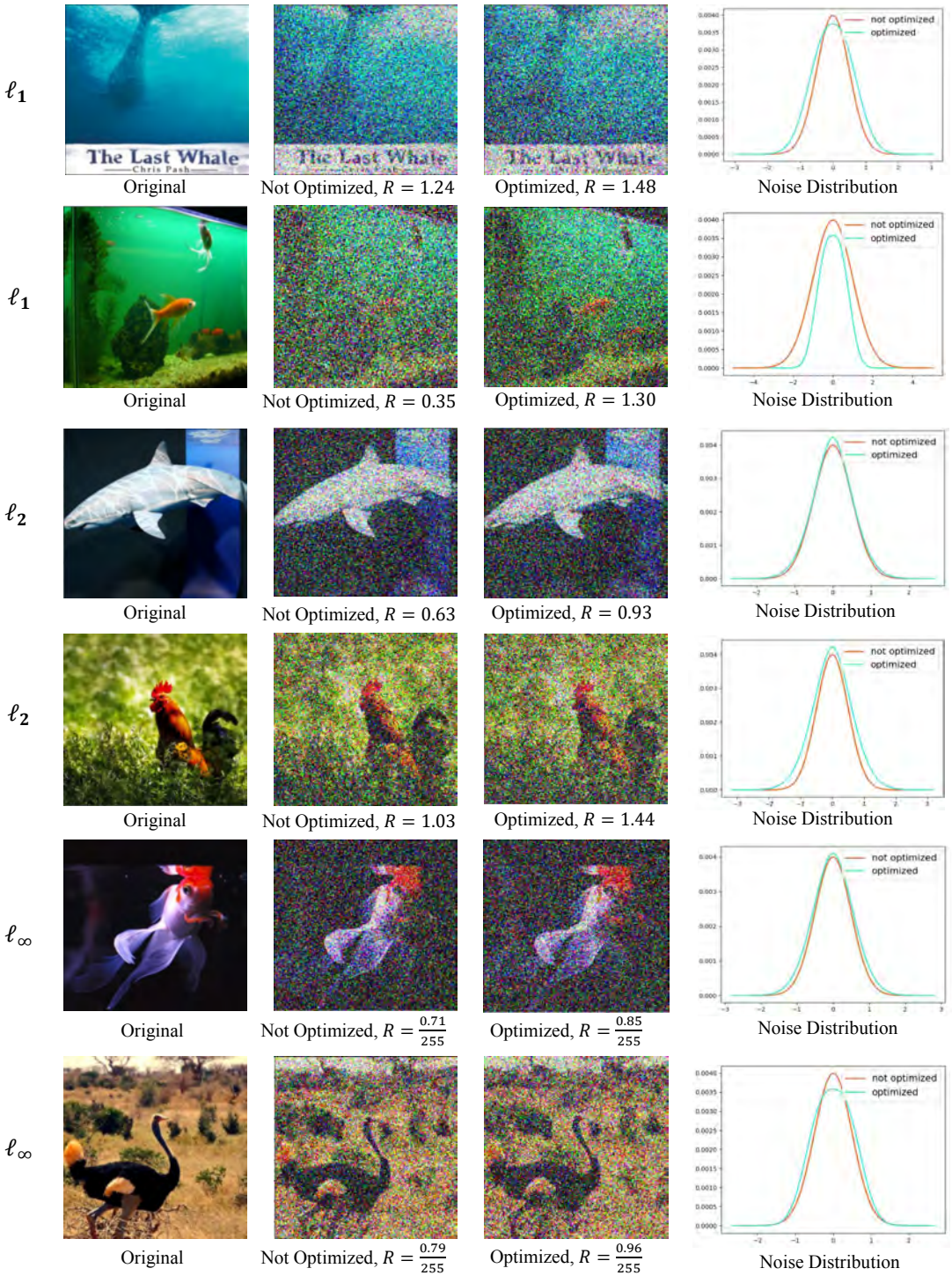
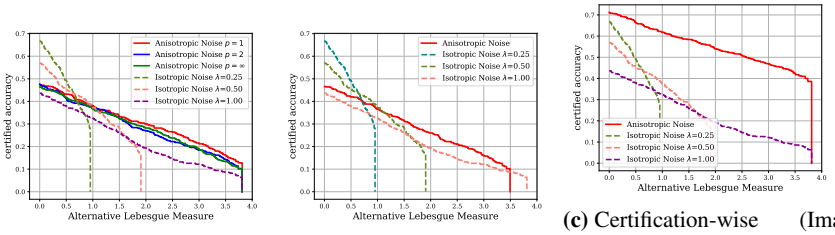
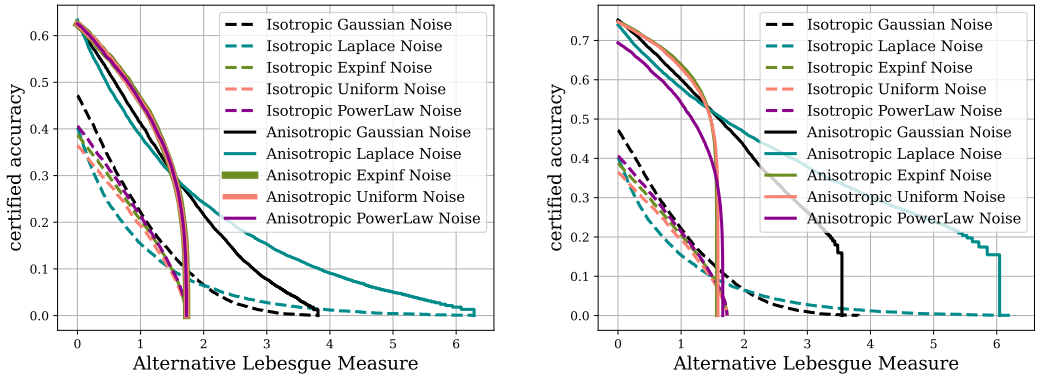


Fig. 7.6: Example images of applying I-OPT (based on UniCR) for smoothed classifier against different ℓ_p perturbations on the ImageNet dataset. From the left to right, the first figure shows the original image. The second and third figures show the smoothed image without I-OPT and with I-OPT, respectively. The fourth figure shows the corresponding distributions before/after I-OPT.

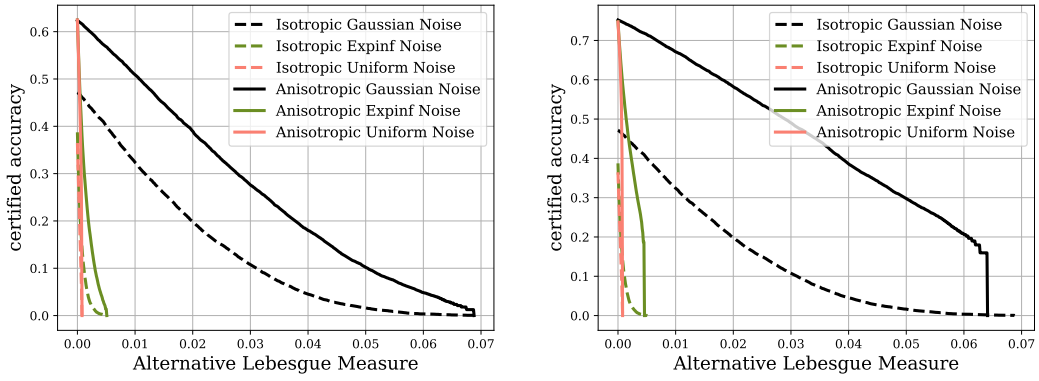


(a) Pattern-wise (ImageNet) **(b)** Dataset-wise (ImageNet) **(c)** Certification-wise (ImageNet)

Fig. 7.7: Comparison of randomized smoothing performance on ImageNet with anisotropic noise and that with isotropic noise (Gaussian distribution for certified defense against ℓ_2 perturbations, comparing with [1]).



(a) Pattern-wise vs. ℓ_1 pert. **(b)** Dataset-wise vs. ℓ_1 pert.



(c) Pattern-wise vs. ℓ_∞ pert. **(d)** Dataset-wise vs. ℓ_∞ pert.

Fig. 7.8: UCAN with pattern-wise and dataset-wise anisotropic noise vs. RS with isotropic noise – different noise PDFs against different ℓ_p perturbations (universality) on CIFAR10.

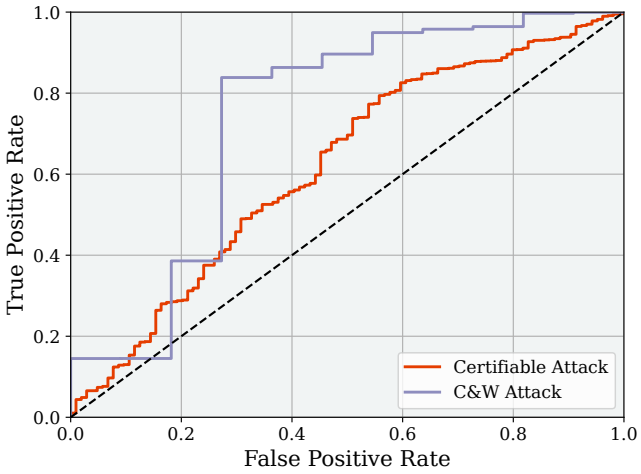


Fig. 7.9: ROC Curve of Detection Results by Feature Squeezing on CIFAR10. The ROC score is 0.38 and 0.26 for the C&W attack and certifiable attack, respectively

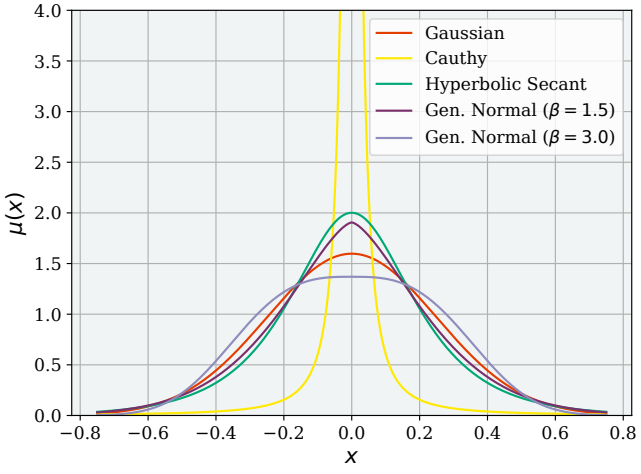


Fig. 7.10: PDF of Different Noise Distributions ($\sigma = 0.25$)