Certifying Robustness of Large Language Models via Discrete-Continuous Randomized Smoothing

Anonymous Authors[1]

Abstract

Large language models have achieved strong performance across diverse language understanding and generation tasks, yet they remain vulnerable to jailbreak attacks, which bypass safety mechanisms and elicit harmful responses. To mitigate this risk, a wide range of heuristic defenses has been proposed, but such defenses offer no formal guarantee and remain vulnerable to adaptive attackers aware of the deployed defensive mechanism. This motivates *certified* jailbreak defenses with provable guarantees over an explicitly specified perturbation set. However, prior certification methods rely on discrete token-level perturbation, yielding coarse-grained guarantees that treat all token modifications within the same budget as equally severe, regardless of their embedding-space displacement. We propose **Discrete-Continuous Randomized Smoothing (DCRS)**, a certification framework for generative LLMs that combines random token subset selection with Gaussian smoothing in embedding space. By jointly accounting for the number of modified tokens and their embedding-space displacement, DCRS provides a finer-grained certificate than purely discrete token-count guarantees. DCRS converts suffix insertion into fixed-length replacement via dummy token padding and certifies for arbitrary suffix attacks under a fixed-length threat model.

1. Introduction

Large language models (LLMs) have achieved remarkable performance across a wide range of natural language processing tasks (Brown et al., 2020; Chowdhery et al., 2023; Touvron et al., 2023). Despite these advances, they remain vulnerable both to general adversarial manipulations of textual inputs (Ebrahimi et al., 2018; Li et al., 2020) and jailbreak attacks (Wei et al., 2023; Ganguli et al., 2022). Recent work has shown that safety mechanisms can be systematically circumvented through diverse strategies, including optimization-based methods such as Greedy Coordinate Gradient (GCG) (Zou et al., 2023), black-box iterative prompt refinement such as PAIR (Chao et al., 2025), and more naturalistic automated jailbreak such as AutoDAN (Zhu et al., 2024). These jailbreak attacks raise serious concerns about the reliability of deployed LLMs.

To mitigate jailbreak attacks, prior work has explored detection-based defenses (Alon & Kamfonas, 2023; Li et al., 2025), prompt-level interventions (Mo et al., 2024; Zhou et al., 2024), and safety-oriented fine-tuning (Zou et al., 2024). However, all of these approaches are *empirical*: they offer no formal guarantee and remain vulnerable to adaptive attackers who exploit knowledge of the defense pipeline (Nasr et al., 2025), recreating the cat-and-mouse dynamic long observed in computer vision. This challenge highlights the need for certification-based defenses that can provide robustness guarantees beyond empirical evaluations.

A growing line of work addresses this need by exploiting the discreteness of textual inputs via token-level perturbations such as masking, deletion, reconstruction, or random selection (Zeng et al., 2023; Ji et al., 2024; Wang et al., 2026; Robey et al., 2025; Kumarappan & Mehrotra, 2025; Huang et al., 2023; Kumar et al., 2023). Yet because these methods perturb tokens in a purely discrete manner, their certificates are uniformly expressed through a single ℓ_0 token budget (Zeng et al., 2023; Ji et al., 2024; Wang et al., 2026; Robey et al., 2025). Crucially, this token-count view ignores the geometry of the embedding space: a synonym substitution and a semantically distant replacement both count as one token modification, even though they may induce very different embedding-space displacements. Purely discrete certificates therefore treat them identically, whereas Gaussian randomized smoothing provides radius-based guarantees that directly reflect the magnitude of the embedding-space shift (Cohen et al., 2019; Levine & Feizi, 2020). Closing this gap requires a certification framework that jointly accounts for the *number* of perturbed tokens and

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

the *magnitude* of their displacement in embedding space.

Moreover, most prior text certifications target classification, where certified robustness is evaluated by preserving a fixed task label on standard classification benchmarks (Zhang et al., 2015; Socher et al., 2013). Such certificates do not transfer to jailbreak robustness in generative LLMs, where the model produces a free-form response and the relevant prediction is a response-level safety decision rather than a predefined class label (Wei et al., 2023).

Furthermore, most existing text certification methods do not account for denoising perturbed inputs, which incur substantial drops in benign utility (Zeng et al., 2023; Robey et al., 2025). This limitation has already been explored in computer vision: randomized smoothing is at odds with clean accuracy (Cohen et al., 2019; Jeong & Shin, 2023). To address this accuracy-robustness trade-off, several researches introduce a denoising module (Jeong & Shin, 2023; Carlini et al., 2023; Salman et al., 2020). Similarly, in text certification, SelfDenoise (Ji et al., 2024) adapts an LLM as a denoiser, but invoking an additional LLM call per sample makes certification computationally expensive. In this work, we introduce an efficient and effective method that maintains benign utility without such overhead.

To overcome the aforementioned limitations, we propose **Discrete-Continuous Randomized Smoothing (DCRS)**—building on randomized smoothing and hierarchical randomized smoothing (Cohen et al., 2019; Scholten et al., 2023)—a certification framework for generative LLMs under token-level perturbations. DCRS randomly selects token positions, adds Gaussian noise to their embeddings, and projects the perturbed embeddings back to the nearest vocabulary tokens before generation. This design yields:

- **Discrete-Continuous Certification.** DCRS considers an adversary that can modify up to k token positions and measures the resulting perturbation through its embedding-space displacement. This yields a certified radius for each token-level attack budget and allows to distinguish perturbations of equal token count but differing embedding magnitude.
- **Certification against Jailbreak Attack.** We lift text certification from label preservation on classification benchmarks to the response-level safety decision of generative LLMs. Concretely, we instantiate DCRS for suffix-based jailbreaks, including GCG (Zou et al., 2023), by padding the clean prompt with dummy suffix tokens and reducing arbitrary suffix *insertion* to fixed-length *replacement*, which yields quantitative certificates against arbitrary suffix attacks.
- **Satisfying Desiderata for LLM Defenses against Jailbreaking.** Following the practical defense desiderata of SmoothLLM (Robey et al., 2025), DCRS is designed to satisfy key requirements for jailbreak defenses in a certified setting: (i) it reduces empirical ASR under representative jailbreak attacks, (ii) it preserves benign utility rather than relying on refusal-only behavior, (iii) it avoids retraining and additional LLM denoising calls, making randomized trials query-efficient, and (iv) it applies to standard open-weight generative LLMs through embedding perturbation and deterministic nearest-neighbor projection without architectural modification.

2. Related Works

Jailbreak attack and defense. Jailbreak attacks on LLMs aim to elicit harmful or policy-violating responses. Early attacks relied on manually designed prompts, including role-playing and persona-based prompting (Shah et al., 2023), encoding or obfuscation (Liu et al., 2023; Jiang et al., 2024), low-resource reformulation (Yong et al., 2023), and other prompt manipulations (Wei et al., 2023). More recent attacks automate this process through optimization or search, such as GCG (Zou et al., 2023), AutoDAN (Zhu et al., 2024), TAP (Mehrotra et al., 2024), and PAIR (Chao et al., 2025). These developments show that jailbreak robustness cannot be fully characterized by evaluating a fixed set of jailbreak prompts.

A broad range of defenses has been proposed to mitigate jailbreak attacks, including perplexity-based detection (Alon & Kamfonas, 2023), randomized perturbation-based defenses (Robey et al., 2025), training-based defenses (Zou et al., 2024), guardrail-style input filtering (Li et al., 2025), and prompt-tuning-based defenses such as Prompt Adversarial Tuning (PAT) (Mo et al., 2024) and Robust Prompt Optimization (RPO) (Zhou et al., 2024). However, these defenses remain vulnerable to adaptive attackers that explicitly account for the deployed defensive mechanism, as investigated by Nasr et al. (2025). This concern underscores the need for *certified* jailbreak defenses: without a formal guarantee, empirical robustness on a fixed collection of known attacks cannot ensure safety against adaptive adversaries. Motivated by this gap, we propose a certification method for generative LLM jailbreak robustness under a fixed-length suffix-replacement threat model.

Certified robustness. Certified robustness guarantees that a model's prediction remains unchanged within a prescribed perturbation set. Randomized smoothing is a representative certification framework that constructs a smoothed classifier by injecting random noise into the input and aggregating predictions over noisy samples (Lecuyer et al., 2019; Cohen et al., 2019; Scholten et al., 2023). In the standard Gaussian setting, the resulting certificate gives an ℓ_2 radius around the input within which the smoothed prediction is unchanged

(Cohen et al., 2019). Randomized ablation extends this idea to the ℓ_0 setting by randomly retaining only a subset of input features, which measures robustness by the number of modified features (Levine & Feizi, 2020). More recently, hierarchical randomized smoothing combines these two views by modeling both the subset of affected entities and the continuous perturbation applied to selected entities (Scholten et al., 2023). This provides the basis for certificates that depend jointly on a discrete attack budget and a continuous perturbation radius.

Randomized smoothing on text. Prior randomized smoothing methods for text mainly provide certificates under discrete perturbation models, typically expressed by an ℓ_0 token budget (Ye et al., 2020; Zeng et al., 2023; Ji et al., 2024; Wang et al., 2026). Such methods perturb text via discrete manner, enabling certification in a discrete input space but treating all substitutions within the same token budget uniformly. SmoothLLM applies randomized perturbations to jailbreak robustness by aggregating responses over perturbed prompts (Robey et al., 2025), but their certified interpretations rely on a strong assumption that jailbreak prompts become non-jailbreaking after sufficiently many tokens are randomly perturbed, rather than directly comparing clean and perturbed smoothing distributions (Kumarappan & Mehrotra, 2025). In contrast, we use the embedding-space radius as a continuous measure of token-level perturbation magnitude, since a token substitution is ultimately realized as a change in the continuous embedding sequence consumed by the LLM.

3. Methodology

We formalize Discrete-Continuous Randomized Smoothing (DCRS), a certification framework for generative LLMs under token-level perturbations. The method preserves the discrete token structure of a prompt while introducing Gaussian perturbations in the continuous embedding space. This design is motivated by the fact that token-level changes are ultimately realized as changes in the LLM's continuous embedding sequence. Thus, beyond asking how many tokens are modified, we also certify whether the response-level decision is stable under ℓ_2-bounded shifts of the selected token embeddings. In each randomized trial, we sample a subset of token positions and perturb only the corresponding embeddings. The resulting certificate characterizes robustness along two axes: (i) the number of modified tokens, and (ii) the magnitude of the embedding-space displacement.

We first define the smoothing distribution over token embeddings and the corresponding smoothed decision function for generative LLMs in Section 3.1. Section 3.2 then derives a certificate for hybrid perturbations that combine token-level changes with bounded embedding-space shifts. Finally, Section 3.3 applies the certificate to suffix attacks, including GCG, by converting suffix insertion into a fixed-length replacement problem.

3.1. Discrete-continuous randomized smoothing for generative LLMs

Let an input prompt be a token sequence $x = (x_1, \dots, x_T)$ of length T, where each $x_i \in V$ is a token from the vocabulary. Let $E : V \to \mathbb{R}^d$ denote the token embedding lookup, and write $e = (e_1, \dots, e_T)$ with $e_i = E(x_i)$ for the corresponding embedding sequence. The embedding sequence e is then perturbed before being passed to the generative model. Given a perturbed embedding sequence, the generative model f produces a response in $\mathcal{R}$, and a response-level decision mapping $h : \mathcal{R} \to \mathcal{Y}$ assigns a discrete label such as safe or unsafe. Our goal is to certify the smoothed decision induced by this response-level label, rather than a conventional classification label.

We define the smoothing distribution on the embedding sequence e. In each randomized trial, we first sample a subset of token positions

$$S \sim \mathrm{Unif}\{S \subseteq [T] : |S| = m\}, \tag{1}$$

where m is the number of selected positions. Given S, we define the embedding-space perturbation operator ϕ_S by

$$\phi_S(e_i) = \begin{cases} e_i + \eta_i, & i \in S, \\ e_i, & i \notin S, \end{cases} \qquad \eta_i \sim \mathcal{N}(0, \sigma^2 I_d). \tag{2}$$

Thus, ϕ_S adds Gaussian noise only to the selected token embeddings. The subset sampling acts as a **discrete** upper-level smoothing step over token positions, while ϕ_S provides a **continuous** lower-level smoothing step in embedding space.

Since a noisy embedding vector may not correspond to a valid vocabulary embedding, we apply a deterministic nearest-neighbor projection before generation. Let

$$\mathrm{NN}_V(z) = \arg\min_{v \in V} \|z - E(v)\|_2 \tag{3}$$

denote the nearest vocabulary token to a vector z. We define the token-wise projection-and-embedding map D as

$$D(z) = E(\mathrm{NN}_V(z)). \tag{4}$$

That is, D first projects each noisy embedding to its nearest vocabulary token and then maps the projected token back to its embedding. This step ensures that the perturbed sequence passed to the generative model corresponds to valid vocabulary embeddings.

The deterministic base decision rule being smoothed is the composite map

$$H(z) = h(f(D(z))), \tag{5}$$

H includes nearest-neighbor projection, generation, and response-level decision mapping. The smoothed decision is then

$$g(e) = \arg\max_{y \in \mathcal{Y}} \Pr_{S,\eta} \left(H(\phi_S(e)) = y\right). \tag{6}$$

The nearest-neighbor projection step does not invalidate the randomized smoothing certificate. Randomized smoothing does not require the deterministic base rule to be continuous, differentiable, or distance preserving; it only uses the final prediction probabilities induced by the prescribed noise distribution (Lecuyer et al., 2019; Cohen et al., 2019). This is the same principle used in denoised smoothing, where a deterministic denoiser is prepended to a classifier and the composite classifier is certified as a whole, rather than certifying the denoiser itself (Salman et al., 2020; Carlini et al., 2023; Jeong & Shin, 2023). In our setting, the map D plays the role of such a deterministic preprocessing step: it maps noisy embeddings back to valid vocabulary embeddings before generation. Therefore, the certificate is applied not to D alone, but to the smoothed decision g induced by the composite base rule $H = h \circ f \circ D$.

In practice, we estimate the label distribution of the smoothed decision by Monte Carlo sampling. In our experiments, we use a binary safety label space, $\mathcal{Y} = \{\text{safe}, \text{unsafe}\}$. Given N randomized trials, we let y_A be the majority label and let n_A denote its count. We then compute the Clopper–Pearson lower confidence bound p_L (Clopper & Pearson, 1934) on the true probability of y_A and use this bound for certification in Section 3.2.

3.2. Certification under token-level and embedding-space perturbations

We now derive the certificate induced by the smoothing procedure in Section 3.1. Our derivation builds on hierarchical randomized smoothing (Scholten et al., 2023), where the input is decomposed into entities and lower-level smoothing is applied only to selected entities. In our setting, each entity corresponds to a token position, and the lower-level smoothing distribution is the Gaussian perturbation applied to the selected token embeddings.

Let $y_A = g(e)$ be the smoothed decision of the clean input, and let p_L be a lower confidence bound on

$$\Pr_{S,\eta} \left(H(\phi_S(e)) = y_A\right),$$

where S is sampled as in Equation (1) and ϕ_S is the embedding-space perturbation operator defined in Equation (2). We consider an adversarial embedding sequence e' that differs from e only on a set of token positions $C \subseteq [T]$. We write e_C and e'_C for the embedding blocks restricted to C, and measure the embedding-space displacement by $\|e'_C - e_C\|_2$.

We use the following reduction, adapted from hierarchical randomized smoothing (Scholten et al., 2023).

Theorem 3.1 (Hierarchical smoothing reduction)**.** *Let C be the set of perturbed entities, and let Δ denote the probability of not selecting all entities in C in the upper-level smoothing step. Suppose the clean input has label probability lower bounded by p_L and $p_L > \Delta$. Define*

$$p'_L = \frac{p_L - \Delta}{1 - \Delta}. \tag{7}$$

If the lower-level smoothing distribution gives a probability lower bound $q_\epsilon(p'_L)$ under a perturbation of size ϵ on the selected entities, then the hierarchical smoothed probability of the same label on the perturbed input is lower bounded by

$$(1 - \Delta) q_\epsilon(p'_L). \tag{8}$$

We first state the hierarchical randomized smoothing bound of Scholten et al. (2023) in our notation. The proof of Theorem 3.1 follows the Neyman–Pearson argument (Neyman & Pearson, 1933) for hierarchical randomized smoothing and is deferred to Appendix A. The term Δ corresponds to the event that at least one perturbed entity is not selected. On the complementary event, all perturbed entities are selected, and the difference between the clean and perturbed inputs is handled by the lower-level smoothing certificate.

We next instantiate Theorem 3.1 for the fixed-size token subset sampling defined in Equation (1).

Theorem 3.2 (Fixed-size token subset certificate)**.** *Assume that each randomized trial selects exactly m token positions uniformly at random, as in Equation (1). Let an adversarial embedding sequence e' differ from e only on a set $C \subseteq [T]$ with $|C| = k$, and suppose*

$$\|e'_C - e_C\|_2 \le \epsilon. \tag{9}$$

For $k \le m \le T$, define

$$\Delta_k = 1 - \frac{\binom{T-k}{m-k}}{\binom{T}{m}}. \tag{10}$$

If $p_L > \Delta_k$, then the probability that the randomized procedure returns the clean smoothed decision y_A on the adversarial input is lower bounded by

$$p_{\text{cert}}(k, \epsilon) = (1 - \Delta_k) \Phi\left(\Phi^{-1}\left(\frac{p_L - \Delta_k}{1 - \Delta_k}\right) - \frac{\epsilon}{\sigma}\right), \tag{11}$$

where Φ is the CDF of the standard normal distribution. If $k > m$, the certificate is vacuous.

Proof. The total number of token subsets of size m is $\binom{T}{m}$. A sampled subset S contains all perturbed positions in C if

and only if the remaining $m-k$ positions are chosen from the $T-k$ positions outside C. Therefore,

$$\Pr(C \subseteq S) = \frac{\binom{T-k}{m-k}}{\binom{T}{m}}. \tag{12}$$

Thus the probability of not selecting all perturbed positions is $\Delta_k = 1 - \Pr(C \subseteq S)$, which gives Equation (10).

Substituting $\Delta = \Delta_k$ into Theorem 3.1 gives $p'_L = (p_L - \Delta_k)/(1-\Delta_k)$. On the event $C \subseteq S$, the clean and adversarial selected embedding blocks differ only on the coordinates in C, so their total selected-block displacement is $\|e'_C - e_C\|_2$. The lower-level smoothing distribution on the selected embeddings is isotropic Gaussian noise with variance $\sigma^2 I_d$. The nearest-neighbor projection, generation, and response-level decision mapping are included in the deterministic base rule H defined in Equation (5). Since randomized smoothing certifies the smoothed decision rule induced by an arbitrary deterministic base rule, the nearest-neighbor projection does not need to be certified as a separate module. By the standard Gaussian randomized smoothing bound (Cohen et al., 2019), any embedding-space perturbation satisfying Equation (9) gives

$$q_\epsilon(p'_L) = \Phi\left(\Phi^{-1}(p'_L) - \frac{\epsilon}{\sigma}\right).$$

Multiplying this lower-level bound by the probability of selecting all perturbed positions, $1-\Delta_k$, yields Equation (11). □

Corollary 3.3 (Binary safety certificate)**.** *For a fixed token budget k, consider all perturbations with $|C| \le k$ and $\|e'_C - e_C\|_2 \le \epsilon$. Since Δ_j is nondecreasing in j for fixed T and m, Δ_k upper bounds the combinatorial penalty for all sets of size at most k. Then, in the binary safety setting, the smoothed decision is certified whenever*

$$p_{\text{cert}}(k,\epsilon) > \frac{1}{2}. \tag{13}$$

Equivalently, when $p_L > \frac{1}{2} + \Delta_k$, it is sufficient that

$$\epsilon < \sigma\left[\Phi^{-1}\left(\frac{p_L - \Delta_k}{1-\Delta_k}\right) - \Phi^{-1}\left(\frac{1}{2(1-\Delta_k)}\right)\right]. \tag{14}$$

If $p_L \le \frac{1}{2} + \Delta_k$, the certificate does not yield a positive binary certified radius for budget k. The proof follows by applying Theorem 3.2, using the monotonicity of Δ_k in the token budget, and solving the condition $p_{\text{cert}}(k,\epsilon) > 1/2$ for ϵ; we provide the details in Appendix B.

3.3. Suffix certification via fixed-length replacement

We now apply the certificate in Section 3.2 to suffix attacks. Many jailbreak attacks, including GCG (Zou et al., 2023), append an adversarial suffix to the original prompt (Zhu et al., 2024; Jia et al., 2024). However, our certificate is defined over a fixed-length token sequence, where clean and perturbed inputs are compared in a common position space. Direct suffix insertion therefore creates a length mismatch.

To handle this, we convert suffix insertion into fixed-length replacement by appending dummy tokens to the clean prompt. For a suffix budget k, define

$$\bar{x}^{(k)} = x \oplus b^k, \tag{15}$$

where $\oplus$ denotes concatenation, b^k is a sequence of k repeated dummy tokens, and $T_k = |\bar{x}^{(k)}|$ is the length of the padded input. We denote by m_k the number of sampled positions for the padded input. The dummy suffix is not intended to approximate the adversarial suffix. Its role is to place the clean prompt and suffix-augmented prompts of the same length in a common fixed-length certification space.

Under this alignment, an adversarial suffix of length k is modeled as replacing the appended dummy tokens with arbitrary adversarial tokens. More generally, we certify the padded sequence under the conservative threat model that any set of at most k positions in the fixed-length input may be modified. Thus, the guarantee is not tied to the GCG optimization procedure or to any particular suffix construction. We run smoothing over the entire padded prompt and sample S uniformly as in Equation (1), rather than conditioning on a presumed modified suffix position set. The complete certification procedure is summarized in Appendix E.

For a modified set $C \subseteq [T_k]$ with $|C| = k$, the sampled subset contains all modified positions with probability $\Pr(C \subseteq S) = \binom{T_k-k}{m_k-k}/\binom{T_k}{m_k}$. Accordingly, the combinatorial penalty for the padded input is

$$\Delta_k = 1 - \frac{\binom{T_k-k}{m_k-k}}{\binom{T_k}{m_k}}. \tag{16}$$

Only on the event $C \subseteq S$ can the difference between the clean and perturbed suffix embeddings be handled by the lower-level Gaussian smoothing certificate. The complementary event is treated conservatively through Δ_k. Thus, dummy suffix padding handles the length mismatch, while random subset selection and Δ_k handle the uncertainty over which modified positions must be covered by the smoothing distribution.

4. Experiments

We evaluate DCRS from two complementary perspectives. First, we test whether the proposed certificate yields nontrivial certified radii for suffix-based jailbreak perturbations. This certification experiment focuses on the fixed-length suffix-replacement setting introduced in Section 3.3, where

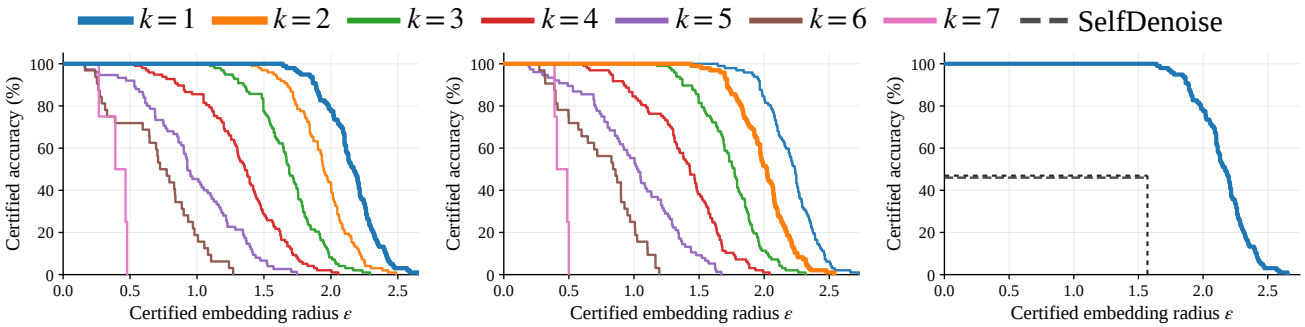

Figure 1. Certified accuracy curves for suffix certification. Left: one dummy suffix token. Middle: two dummy suffix tokens. Right: $k = 1$ comparison with the SelfDenoise reference. The matched curve is highlighted in bold.

clean prompts are padded with dummy suffix tokens and certified against arbitrary replacements within the corresponding token budget. Second, we evaluate empirical attack success rate (ASR) under representative jailbreak attacks to examine whether the randomized defense also improves practical robustness beyond the certified setting. Finally, we report benign utility on standard reasoning benchmarks to assess whether the defense substantially degrades non-adversarial model performance.

4.1. Experimental setup

We use Llama-2-7B-Chat as the main target model and AdvBench as the certification benchmark (Touvron et al., 2023; Zou et al., 2023). For suffix certification, each clean prompt is padded with dummy tokens so that suffix insertion can be evaluated as fixed-length replacement. We use the dummy token $b = !$, noise scale $\sigma = 1.0$, and sampling ratio $m_k/T_k = 0.9$ for each padded input of length T_k. SelfDenoise is evaluated under the same suffix-certification protocol and the same sampling ratio.

We also evaluate empirical Attack Success Rate (ASR) under representative jailbreak attacks. GCG is evaluated on AdvBench, while PAIR and AutoDAN are evaluated on HarmBench (Chao et al., 2025; Zhu et al., 2024; Mazeika et al., 2024). We compare DCRS against SelfDenoise, SmoothLLM, perplexity-based filtering (PPL), Circuit Breakers (CB), PIGuard, and the undefended model (Ji et al., 2024; Robey et al., 2025; Alon & Kamfonas, 2023; Zou et al., 2024; Li et al., 2025). Finally, we evaluate benign utility on standard reasoning benchmarks, including ARC-Challenge, ARC-Easy, and HellaSwag (Clark et al., 2018; Zellers et al., 2019). Additional details are provided in Appendix C.

4.2. Certification results

We first evaluate whether DCRS provides certificates for suffix-based jailbreak perturbations. Following the fixed-length formulation in Section 3.3, we certify a suffix attack of length k by reducing suffix insertion to the token-level perturbation setting with k modified positions. For each suffix budget k, we append k dummy suffix tokens to the clean prompt, so that an adversarial suffix of the same length can be viewed as a fixed-length replacement of the appended dummy positions.

To certify arbitrary suffix replacement, we compare the certified radius ϵ_k with a worst-case embedding displacement threshold. With dummy token $b = !$, define

$$\rho = \max_{v \in V} \|E(v) - E(b)\|_2 . \tag{17}$$

Then any replacement of k appended dummy tokens satisfies

$$\|e'_C - e_C\|_2 = \left(\sum_{i \in C} \|E(x'_i) - E(b)\|_2^2 \right)^{1/2} \leq \rho\sqrt{k}.$$

Thus, a prompt is certified for suffix budget k if $\epsilon_k > \rho\sqrt{k}$.

Figure 1 (Left, Middle) highlight the fine-grained nature of DCRS certification. Unlike purely discrete certificates that only specify whether up to k tokens can be modified, DCRS returns a certified embedding-space radius for each token-level budget. As a result, suffix replacements with the same number of modified tokens can receive different robustness guarantees depending on the magnitude of their embedding-space displacement. In the suffix-replacement setting, this radius can be directly compared with the worst-case threshold $\rho\sqrt{k}$, yielding a quantitative certificate against arbitrary replacements within the suffix budget rather than only against a particular observed attack string.

At $k = 1$, DCRS certifies 97 out of 98 prompts at the threshold $\rho\sqrt{1}$, corresponding to 98.98% certified accuracy. At $k = 2$, it certifies 12 out of the 97 prompts that were certified at the previous budget, corresponding to 12.37% certified accuracy at $\rho\sqrt{2}$. This drop reflects the two sources of difficulty captured by the certificate: increasing k raises the combinatorial penalty Δ_k, since the sampled subset must cover more modified positions, and also increases the

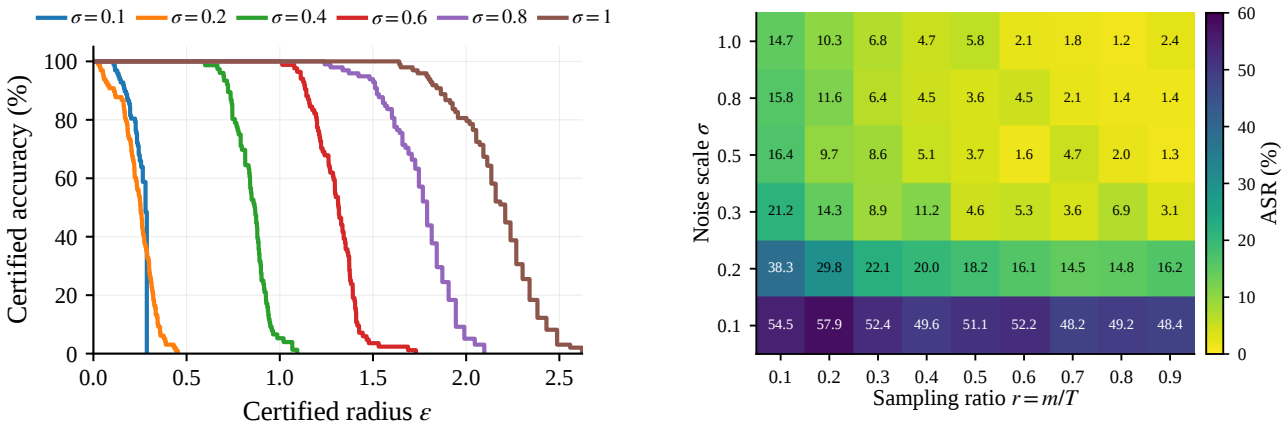

Figure 2. Sensitivity of L2+NN smoothing to noise scale and sampling ratio. Left: $k = 1$ certified accuracy curves for varying σ with fixed $r = 0.9$. Right: empirical ASR across the (σ, r) grid. Both results highlight the role of the continuous noise scale in robustness.

Table 1. Certified accuracy and ASR(%) under representative jailbreak attacks. The first row reports $k = 1$ suffix certified accuracy; the remaining rows report empirical ASR. Parentheses for PPL and PIGuard indicate undetected jailbreak rates.

Model	Attack	DCRS	SelfDenoise	SmoothLLM	RPO	PAT	PPL	CB	PIGuard	None
Llama	$k = 1$ certified	**98.98**	**46**				***Not certified***			
	GCG	1.2	4	1	9	2	0(0)	20	11(28)	55
	PAIR	4	4	0	4	0	86(100)	2	60(68)	86
	AutoDAN	0	0	0	0	0	3(100)	0	0(33)	3
Vicuna	GCG	4.8	14	12	7.3	2.4	0(0)	12	12(15)	15
	PAIR	0	2	22	23	36	44(100)	8	38(74)	44
	AutoDAN	10	46	36	20	19	38(100)	80	33(61)	38
gpt-oss	GCG	0	0	2	0	0	0(0)	3	0(0)	3
	AutoDAN	0	0	4	40	0	17(100)	12	15(61)	17

required worst-case displacement threshold $\rho\sqrt{k}$. Under the same suffix-certification setting, SelfDenoise certifies only 46 prompts at $k = 1$ and **none** at $k = 2$ in one dummy token setting. These results indicate that directly smoothing and projecting perturbed embeddings provides a more stable certification than a denoise-then-generate pipeline.

Figure 2 (Left) further shows how the continuous component controls the certified radius. With $k = 1$ and sampling ratio $r = 0.9$ fixed, larger noise scales σ maintain certified accuracy over larger radius thresholds. This demonstrates that DCRS does not merely certify a discrete token budget; it also allows the strength of embedding-space smoothing to translate into larger continuous robustness radii.

4.3. Empirical robustness and utility

These results are intended to position DCRS among existing baselines, not to claim uniform empirical superiority. Table 1 reports empirical ASR under GCG (Zou et al., 2023), PAIR (Chao et al., 2025), and AutoDAN (Zhu et al., 2024). DCRS reduces ASR relative to the undefended model on Llama and Vicuna, most notably reducing Llama GCG ASR from 55% to 1.2%. For gpt-oss, whose undefended ASR is already low, DCRS preserves this low-ASR regime. Although some baselines achieve lower ASR on individual rows, DCRS remains competitive across attacks while additionally providing a formal suffix certificate. Figure 2 (Right) further shows that increasing the noise scale generally lowers empirical ASR across sampling ratios, supporting the role of the continuous smoothing component.

Table 2 reports benign utility on standard reasoning benchmarks (Clark et al., 2018; Zellers et al., 2019). Among methods that perform input-side smoothing (DCRS, SelfDenoise, SmoothLLM), DCRS attains the highest accuracy on every benchmark. This supports our claim that nearest-token projection acts as an effective implicit denoiser, and directly smoothing embeddings and projecting back is less disruptive to benign inputs than a denoise-then-generate pipeline.

Combined with the certified and empirical robustness above, these results show that DCRS achieves a favorable

Table 2. Benign utility and computational cost.

Benchmark / Metric	DCRS	Self Denoise	Smooth LLM	RPO	PAT	None
ARC-Challenge ↑	39.5	36.5	31.0	37.0	38.5	42.5
ARC-Easy ↑	56.0	52.0	45.0	59.5	44.5	59.5
HellaSwag ↑	64.0	62.5	50.5	64.5	52.0	66.0
Time/trial (s) ↓	0.17	3.51	0.17	0.18	0.18	0.15

robustness-utility trade-off, providing formal suffix certificates and gains comparable empirical ASR while preserving the model's reasoning performance.

Table 2 reports per-sample wall-clock time. DCRS produces a single sample in 0.17s, essentially matching the undefended model (0.15s) and SmoothLLM, and is **20.6× faster than SelfDenoise** (3.51s). The gap stems from SelfDenoise invoking two LLM calls per sample (paraphrase + generation), whereas DCRS folds denoising into a single forward pass via nearest-token projection. As certified guarantees scale with the number of Monte Carlo samples, this per-sample gap directly translates into the same factor of speedup at any target sample budget.

5. Discussion

Our results highlight the role of the continuous smoothing component in DCRS. As shown in Figure 2 (Right), changing the sampling ratio $r = m/T$ alone does not consistently reduce empirical ASR, whereas increasing the Gaussian noise scale σ generally lowers ASR across sampling ratios. This indicates that selecting more token positions is not sufficient by itself; the selected embeddings must also be smoothed with a sufficiently strong continuous perturbation. The trend supports the central motivation of DCRS: jailbreak robustness should not be characterized only by a discrete token budget, but also by the embedding-space displacement induced by token-level changes.

The certification results clarify why certification becomes harder as k grows. A larger suffix budget increases the combinatorial penalty Δ_k, since the sampled subset must cover more modified positions, and it also raises the required worst-case threshold $\rho\sqrt{k}$ for certifying arbitrary replacement. The sharp drop at $k = 2$ should therefore be read as the cost of certifying all $|V|^k$ vocabulary replacements of the dummy suffix positions, rather than a failure against any particular observed suffix.

5.1. Conjecture: GCG Explores a Small Reachable Token Set

The worst-case certificate in Section 3.3 considers all possible replacements of the appended dummy suffix positions, yielding a perturbation space of size $|V|^k$. This is appropriate for a formal worst-case guarantee, but it may be overly conservative for optimization-based suffix attacks such as GCG (Zou et al., 2023). In practice, GCG does not sample suffix tokens uniformly from the full vocabulary; it ranks candidate substitutions by their effect on an affirmative-response objective and selects tokens that most reduce this loss. This motivates the conjecture that GCG-style suffix optimization explores a much smaller reachable token set than the full vocabulary.

To probe this conjecture, we perform a single-position reachable-set analysis. For each of $N = 50$ harmful prompts from AdvBench (Zou et al., 2023), we replace the appended dummy suffix token with every vocabulary token and rank candidates by a GCG-style affirmative-response loss against the target string "Sure, here is". Following the standard GCG filtering procedure, we restrict candidates to ASCII tokens. For each prompt, we collect the top-M tokens, and define V^* as the union of these candidates across prompts.

We find substantial token-level overlap across prompts. At $M = 1$, the 50 per-prompt top choices collapse to only $|V^*| = 29$ unique tokens. The same pattern persists at larger candidate budgets: at $M = 10$, $|V^*| = 237$ compared with the trivial upper bound $N \cdot M = 500$, and at $M = 50$, $|V^*| = 975$ compared with 2,500. All of these sets are far smaller than the full Llama-2 vocabulary size $|V| = 32{,}000$. This suggests that diverse harmful prompts share a small set of high-impact suffix tokens under the GCG-style objective.

This observation does not change the worst-case certificate proven in Section 3.2; our certificate still reasons over arbitrary replacements. However, it suggests a direction for tighter future certificates: instead of certifying against all $|V|^k$ suffix replacements, one may certify against a smaller reachable set induced by a specified optimization procedure or attack family. Figure 3 visualizes this reachable set for $M = 50$ in the PCA-projected embedding space.

6. Conclusion

We presented Discrete-Continuous Randomized Smoothing (DCRS), a certification framework for quantitatively certifying jailbreak robustness in generative LLMs. DCRS combines token-level random subset selection with Gaussian smoothing in embedding space, providing certificates that jointly account for the number of modified tokens and their embedding-space displacement. By using dummy-token padding, DCRS converts suffix insertion into fixed-length replacement and formalizes suffix-based jailbreak attacks within a certifiable threat model. Empirically, DCRS yields non-trivial suffix certificates, reduces ASR under representative jailbreak attacks, preserves benign reasoning performance, and performs certification with low per-sample cost without requiring an additional LLM-based denoising call.

References

Alon, G. and Kamfonas, M. Detecting language model attacks with perplexity. *arXiv preprint arXiv:2308.14132*, 2023.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33: 1877–1901, 2020.

Carlini, N., Tramer, F., Dvijotham, K. D., Rice, L., Sun, M., and Kolter, J. Z. (certified!!) adversarial robustness for free! In *International Conference on Learning Representations*, 2023.

Chao, P., Robey, A., Dobriban, E., Hassani, H., Pappas, G. J., and Wong, E. Jailbreaking black box large language models in twenty queries. In *2025 IEEE Conference on Secure and Trustworthy Machine Learning (SaTML)*, pp. 23–42. IEEE, 2025.

Chowdhery, A., Narang, S., Devlin, J., Bosma, M., Mishra, G., Roberts, A., Barham, P., Chung, H. W., Sutton, C., Gehrmann, S., et al. Palm: Scaling language modeling with pathways. *Journal of machine learning research*, 24 (240):1–113, 2023.

Clark, P., Cowhey, I., Etzioni, O., Khot, T., Sabharwal, A., Schoenick, C., and Tafjord, O. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*, 2018.

Clopper, C. J. and Pearson, E. S. The use of confidence or fiducial limits illustrated in the case of the binomial. *Biometrika*, 26(4):404–413, 1934.

Cohen, J., Rosenfeld, E., and Kolter, Z. Certified adversarial robustness via randomized smoothing. In *international conference on machine learning*, pp. 1310–1320. PMLR, 2019.

Ebrahimi, J., Rao, A., Lowd, D., and Dou, D. Hotflip: White-box adversarial examples for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 31–36, 2018.

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

Huang, Z., Marchant, N. G., Lucas, K., Bauer, L., Ohrimenko, O., and Rubinstein, B. Rs-del: Edit distance robustness certificates for sequence classifiers via randomized deletion. *Advances in Neural Information Processing Systems*, 36:18676–18711, 2023.

Jeong, J. and Shin, J. Multi-scale diffusion denoised smoothing. *Advances in Neural Information Processing Systems*, 36:67374–67397, 2023.

Ji, J., Hou, B., Zhang, Z., Zhang, G., Fan, W., Li, Q., Zhang, Y., Liu, G., Liu, S., and Chang, S. Advancing the robustness of large language models through self-denoised smoothing. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers)*, pp. 246–257, 2024.

Jia, X., Pang, T., Du, C., Huang, Y., Gu, J., Liu, Y., Cao, X., and Lin, M. Improved techniques for optimization-based jailbreaking on large language models. *arXiv preprint arXiv:2405.21018*, 2024.

Jiang, F., Xu, Z., Niu, L., Xiang, Z., Ramasubramanian, B., Li, B., and Poovendran, R. Artprompt: Ascii art-based jailbreak attacks against aligned llms. In *Proceedings of the 62nd annual meeting of the association for computational linguistics (volume 1: Long papers)*, pp. 15157–15173, 2024.

Kumar, A., Agarwal, C., Srinivas, S., Li, A. J., Feizi, S., and Lakkaraju, H. Certifying llm safety against adversarial prompting. *arXiv preprint arXiv:2309.02705*, 2023.

Kumarappan, A. and Mehrotra, A. Towards realistic guarantees: A probabilistic certificate for smoothLLM. In *Lock-LLM Workshop: Prevent Unauthorized Knowledge Use from Large Language Models*, 2025.

Lecuyer, M., Atlidakis, V., Geambasu, R., Hsu, D., and Jana, S. Certified robustness to adversarial examples with differential privacy. In *2019 IEEE Symposium on Security and Privacy (SP)*, pp. 656–672. IEEE Computer Society, 2019.

Levine, A. and Feizi, S. Robustness certificates for sparse adversarial attacks by randomized ablation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 4585–4593, 2020.

Li, H., Liu, X., Zhang, N., and Xiao, C. Piguard: Prompt injection guardrail via mitigating overdefense for free. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 30420–30437, 2025.

Li, L., Ma, R., Guo, Q., Xue, X., and Qiu, X. Bert-attack: Adversarial attack against bert using bert. In *Proceedings of the 2020 conference on empirical methods in natural language processing (EMNLP)*, pp. 6193–6202, 2020.

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

Mazeika, M., Phan, L., Yin, X., Zou, A., Wang, Z., Mu, N., Sakhaee, E., Li, N., Basart, S., Li, B., Forsyth, D., and Hendrycks, D. HarmBench: A standardized evaluation framework for automated red teaming and robust refusal. *arXiv preprint arXiv:2402.04249*, 2024.

Mehrotra, A., Zampetakis, M., Kassianik, P., Nelson, B., Anderson, H., Singer, Y., and Karbasi, A. Tree of attacks: Jailbreaking black-box llms automatically. *Advances in Neural Information Processing Systems*, 37:61065–61105, 2024.

Mo, Y., Wang, Y., Wei, Z., and Wang, Y. Fight back against jailbreaking via prompt adversarial tuning. *Advances in Neural Information Processing Systems*, 37:64242–64272, 2024.

Nasr, M., Carlini, N., Sitawarin, C., Schulhoff, S. V., Hayes, J., Ilie, M., Pluto, J., Song, S., Chaudhari, H., Shumailov, I., et al. The attacker moves second: Stronger adaptive attacks bypass defenses against llm jailbreaks and prompt injections. *arXiv preprint arXiv:2510.09023*, 2025.

Neyman, J. and Pearson, E. S. On the problem of the most efficient tests of statistical hypotheses. *Philosophical Transactions of the Royal Society of London. Series A*, 231:289–337, 1933.

Robey, A., Wong, E., Hassani, H., and Pappas, G. J. Smoothllm: Defending large language models against jailbreaking attacks. *Trans. Mach. Learn. Res.*, 2025, 2025. URL https://openreview.net/forum?id=laPAh2hRFC.

Salman, H., Sun, M., Yang, G., Kapoor, A., and Kolter, J. Z. Denoised smoothing: A provable defense for pretrained classifiers. In *Advances in Neural Information Processing Systems*, volume 33, pp. 21945–21957. Curran Associates, Inc., 2020.

Scholten, Y., Schuchardt, J., Bojchevski, A., and Günnemann, S. Hierarchical randomized smoothing. *Advances in Neural Information Processing Systems*, 36: 49783–49813, 2023.

Shah, R., Pour, S., Tagade, A., Casper, S., Rando, J., et al. Scalable and transferable black-box jailbreaks for language models via persona modulation. *arXiv preprint arXiv:2311.03348*, 2023.

Socher, R., Perelygin, A., Wu, J., Chuang, J., Manning, C. D., Ng, A. Y., and Potts, C. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pp. 1631–1642, 2013.

Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., Bikel, D., Blecher, L., Ferrer, C. C., Chen, M., Cucurull, G., Esiobu, D., Fernandes, J., Fu, J., Fu, W., Fuller, B., Gao, C., Goswami, V., Goyal, N., Hartshorn, A., Hosseini, S., Hou, R., Inan, H., Kardas, M., Kerkez, V., Khabsa, M., Kloumann, I., Korenev, A., Koura, P. S., Lachaux, M.-A., Lavril, T., Lee, J., Liskovich, D., Lu, Y., Mao, Y., Martinet, X., Mihaylov, T., Mishra, P., Molybog, I., Nie, Y., Poulton, A., Reizenstein, J., Rungta, R., Saladi, K., Schelten, A., Silva, R., Smith, E. M., Subramanian, R., Tan, X. E., Tang, B., Taylor, R., Williams, A., Kuan, J. X., Xu, P., Yan, Z., Zarov, I., Zhang, Y., Fan, A., Kambadur, M., Narang, S., Rodriguez, A., Stojnic, R., Edunov, S., and Scialom, T. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

Wang, Z., Jin, G., Hu, J., and Mu, R. CLUCERT: Certifying llm robustness via clustering-guided denoising smoothing. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 37998–38006, 2026.

Wei, A., Haghtalab, N., and Steinhardt, J. Jailbroken: How does llm safety training fail? *Advances in neural information processing systems*, 36:80079–80110, 2023.

Ye, M., Gong, C., and Liu, Q. Safer: A structure-free approach for certified robustness to adversarial word substitutions. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 3465–3475, 2020.

Yong, Z.-X., Menghini, C., and Bach, S. H. Low-resource languages jailbreak GPT-4. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 16640–16648, Singapore, 2023. Association for Computational Linguistics.

Zellers, R., Holtzman, A., Bisk, Y., Farhadi, A., and Choi, Y. Hellaswag: Can a machine really finish your sentence? In *Proceedings of the 57th annual meeting of the association for computational linguistics*, pp. 4791–4800, 2019.

Zeng, J., Xu, J., Zheng, X., and Huang, X. Certified robustness to text adversarial attacks by randomized [mask]. *Computational Linguistics*, 49(2):395–427, 2023.

Zhang, X., Zhao, J., and LeCun, Y. Character-level convolutional networks for text classification. *Advances in neural information processing systems*, 28, 2015.

Zhou, A., Li, B., and Wang, H. Robust prompt optimization for defending language models against jailbreaking attacks. *Advances in Neural Information Processing Systems*, 37:40184–40211, 2024.

Zhu, S., Zhang, R., An, B., Wu, G., Barrow, J., Wang, Z., Huang, F., Nenkova, A., and Sun, T. AutoDAN: Interpretable gradient-based adversarial attacks on large language models. In *First Conference on Language Modeling*, 2024.

Zou, A., Wang, Z., Carlini, N., Nasr, M., Kolter, J. Z., and Fredrikson, M. Universal and transferable adversarial attacks on aligned language models. *arXiv preprint arXiv:2307.15043*, 2023.

Zou, A., Phan, L., Wang, J., Duenas, D., Lin, M., Andriushchenko, M., Wang, R., Kolter, Z., Fredrikson, M., and Hendrycks, D. Improving alignment and robustness with circuit breakers. *Advances in Neural Information Processing Systems*, 37:83345–83373, 2024.

A. Proof of the hierarchical smoothing bound

This appendix proves Theorem 3.1. The argument follows the Neyman–Pearson proof for hierarchical randomized smoothing (Scholten et al., 2023), adapted to the notation of Section 3.

Recall that $e = (e_1, \dots, e_T)$ denotes the clean embedding sequence and that e' denotes a perturbed embedding sequence. Let $C \subseteq [T]$ be the set of token positions on which e and e' differ. In each randomized trial, the upper-level sampler first draws a subset $S \subseteq [T]$ of token positions, and the lower-level smoothing distribution then applies Gaussian perturbations to the selected embeddings. We write Ψ_e and $\Psi_{e'}$ for the resulting hierarchical smoothing distributions around e and e', respectively. Equivalently, a sample $Z \sim \Psi_e$ is the randomized embedding sequence obtained by sampling S and applying the lower-level Gaussian perturbation to e, and $Z \sim \Psi_{e'}$ is defined analogously for e'.

The deterministic base decision rule is the same composite rule used in Section 3.1,

$$H(z) = h(f(D(z))),$$

where D denotes the nearest-neighbor projection-and-embedding map, f is the generative model, and h maps the generated response to a response-level label. Thus, the randomness in Ψ_e or $\Psi_{e'}$ comes only from the subset sampling and Gaussian perturbation, while nearest-neighbor projection, generation, and response-level decision mapping are included inside the deterministic rule H.

For a fixed label y, suppose that the clean hierarchical smoothing distribution satisfies

$$\Pr_{Z\sim\Psi_e}[H(Z) = y] \geq p_L.$$

Following the region decomposition used in hierarchical randomized smoothing (Scholten et al., 2023), we partition the sample space into three regions. The region R_1 denotes the part of the clean hierarchical distribution in which at least one perturbed position is not selected by the upper-level sampler. The region R_3 is the analogous part of the perturbed hierarchical distribution. The shared region R_2 denotes the event in which all perturbed positions are selected, so that the clean and perturbed distributions can be compared through the lower-level smoothing distributions on the selected perturbed block.

Let Δ be the probability that the upper-level sampling step does not select all positions in C. Then the relevant region probabilities are

$$\Pr_{Z\sim\Psi_e}[Z \in R_1] = \Delta, \qquad \Pr_{Z\sim\Psi_e}[Z \in R_2] = 1 - \Delta,$$

and

$$\Pr_{Z\sim\Psi_{e'}}[Z \in R_2] = 1 - \Delta, \qquad \Pr_{Z\sim\Psi_{e'}}[Z \in R_3] = \Delta.$$

We seek a lower bound on the probability of predicting y under $\Psi_{e'}$ given only the clean lower bound p_L. In the worst case, the classifier can assign as much of the clean label probability as possible to R_1, because this region does not contribute to the shared region R_2 where the clean and perturbed distributions are compared. Since $\Pr_{\Psi_e}(R_1) = \Delta$, at most Δ of the clean probability can be lost in this way. Therefore, at least $p_L - \Delta$ of the clean label probability must lie in the shared region R_2. Since $\Pr_{\Psi_e}(R_2) = 1 - \Delta$, the conditional clean label probability inside R_2 is lower bounded by

$$p'_L = \frac{p_L - \Delta}{1 - \Delta}.$$

Conditioned on R_2, all perturbed positions have been selected, and the hierarchical smoothing problem reduces to the lower-level smoothing problem on the perturbed block C. Let $q_\epsilon(p'_L)$ denote the lower-level Neyman–Pearson lower bound under a perturbation of size ϵ when the clean conditional label probability is at least p'_L. Then

$$\Pr_{Z\sim\Psi_{e'}}[H(Z) = y \mid Z \in R_2] \geq q_\epsilon(p'_L).$$

Multiplying by the probability of the shared region under $\Psi_{e'}$ gives

$$\Pr_{Z\sim\Psi_{e'}}[H(Z) = y] \geq \Pr_{Z\sim\Psi_{e'}}[Z \in R_2]\, q_\epsilon(p'_L) = (1 - \Delta) q_\epsilon(p'_L).$$

This proves Theorem 3.1.

B. Proof of the binary safety certificate

We prove Corollary 3.3. For any perturbation set C with $|C| = j \le k$, the probability that the sampled subset fails to contain all modified positions is

$$\Delta_j = 1 - \frac{\binom{T-j}{m-j}}{\binom{T}{m}}.$$

For fixed T and m, Δ_j is nondecreasing in j, so $\Delta_j \le \Delta_k$. Therefore, using Δ_k gives a conservative lower bound that applies uniformly to all perturbation sets with size at most k.

In the binary safety setting, it is sufficient that the probability of the clean smoothed decision remains larger than $1/2$ under the perturbed input. By Theorem 3.2, this holds whenever

$$p_{\text{cert}}(k, \epsilon) > \frac{1}{2}.$$

Substituting the expression for $p_{\text{cert}}(k, \epsilon)$ from Equation (11), this condition becomes

$$(1 - \Delta_k)\Phi\left(\Phi^{-1}\left(\frac{p_L - \Delta_k}{1 - \Delta_k}\right) - \frac{\epsilon}{\sigma}\right) > \frac{1}{2}.$$

Since Φ is strictly increasing, this is equivalent to

$$\Phi^{-1}\left(\frac{p_L - \Delta_k}{1 - \Delta_k}\right) - \frac{\epsilon}{\sigma} > \Phi^{-1}\left(\frac{1}{2(1 - \Delta_k)}\right).$$

Rearranging gives

$$\epsilon < \sigma\left[\Phi^{-1}\left(\frac{p_L - \Delta_k}{1 - \Delta_k}\right) - \Phi^{-1}\left(\frac{1}{2(1 - \Delta_k)}\right)\right].$$

For the right-hand side to be positive, we need

$$\frac{p_L - \Delta_k}{1 - \Delta_k} > \frac{1}{2(1 - \Delta_k)},$$

which is equivalent to $p_L > \frac{1}{2} + \Delta_k$. If this condition fails, the bound does not give a positive binary certified radius.

C. Additional experimental details

C.1. Certification setup

For certification experiments, we use Llama-2-7B-Chat as the target generative model and AdvBench as the harmful prompt dataset (Touvron et al., 2023; Zou et al., 2023). We evaluate 98 AdvBench prompts and focus on suffix attacks, with GCG-style adversarial suffixes as the representative suffix-attack setting. For each suffix budget k, the clean prompt is converted into a fixed-length certification input by appending k dummy suffix tokens. Unless otherwise specified, the dummy token is set to $b = \texttt{!}$.

We use the DCRS smoothing procedure described in Section 3. For each randomized trial, we sample a subset of token positions uniformly at random and perturb only the selected token embeddings with Gaussian noise. The perturbed embeddings are then projected back to the nearest vocabulary tokens before generation. For each padded input $\bar{x}^{(k)}$ with length T_k, we set

$$m_k = \lfloor 0.9 T_k \rfloor$$

and use noise scale $\sigma = 1.0$ for the main certification results. For each prompt, we run $N = 2000$ randomized trials. The majority label y_A is obtained by aggregating the safety labels over randomized generations, and the Clopper–Pearson lower confidence bound p_L is computed with confidence level $\alpha = 0.05$.

For each suffix budget k, we compute the certified radius ϵ_k using Corollary 3.3. To determine whether the padded suffix is certified against arbitrary token replacement, we compare ϵ_k with the worst-case suffix displacement threshold $\rho\sqrt{k}$, where

$$\rho = \max_{v \in V} \|E(v) - E(\texttt{!})\|_2.$$

A prompt is counted as certified for suffix budget k if

$$\epsilon_k > \rho\sqrt{k}.$$

This criterion certifies that any replacement of the k appended dummy suffix positions remains within the certified embedding-space radius.

For suffix-budget evaluation, we report certification in a nested manner. After evaluating budget k, we continue to budget $k+1$ only for prompts that were certified at budget k. This avoids reporting certification for a larger suffix budget on prompts that already failed at a smaller budget. As a result, the number of evaluated prompts can decrease as the suffix budget increases.

C.2. Nearest-neighbor projection

After Gaussian perturbation, a noisy embedding vector may not correspond to a valid vocabulary embedding. We therefore apply a deterministic nearest-neighbor projection over the vocabulary embedding table. For a perturbed embedding vector z, the nearest vocabulary token is

$$\mathrm{NN}_V(z) = \arg\min_{v \in V} \|z - E(v)\|_2.$$

We then map the projected token back to its embedding. Following Section 3.1, we denote this token-wise projection-and-embedding map by

$$D(z_i) = E(\mathrm{NN}_V(z_i)).$$

Thus, D maps continuous noisy embeddings back to valid vocabulary embeddings before generation.

In implementation, we perform nearest-neighbor search over the model vocabulary embedding table. To make the search efficient, we normalize vocabulary embeddings and use similarity-based retrieval. This projection is not an additional learned denoising model. It is a deterministic preprocessing step included inside the composite base decision rule $H = h \circ f \circ D$, and the certificate is applied to the smoothed decision induced by this composite rule as a whole.

C.3. ASR evaluation setup

In addition to certification, we evaluate empirical robustness using attack success rate (ASR). We consider three representative jailbreak attacks: GCG (Zou et al., 2023), PAIR (Chao et al., 2025), and AutoDAN (Zhu et al., 2024). GCG is a gradient-based suffix optimization attack that appends an adversarial suffix to a harmful prompt. Although the original GCG objective emphasizes transferable suffixes, we optimize the suffix separately for each prompt in our ASR evaluation, yielding a stronger prompt-specific attack. PAIR is a black-box iterative jailbreak method in which an attacker model repeatedly refines prompts based on feedback. AutoDAN is an automated jailbreak attack designed to produce more natural and stealthy adversarial prompts than standard optimization-based suffix attacks. GCG is evaluated on AdvBench, while PAIR and AutoDAN are evaluated on HarmBench (Mazeika et al., 2024).

For randomized defenses, we aggregate multiple randomized generations by majority vote. We use 10 votes for GCG and 5 votes for PAIR and AutoDAN. For each vote, we independently sample the randomized smoothing distribution, generate a response from the target model, and assign a safety label using the corresponding judge. The final prediction is the majority safety label over the randomized responses. For DCRS, we evaluate multiple smoothing configurations by varying the noise scale σ and the token sampling ratio r. For an input of length T, the number of perturbed positions is set to

$$m = \lfloor rT \rfloor.$$

We compare DCRS against representative defenses from several categories. SelfDenoise (Ji et al., 2024) is a denoising-based randomized defense that perturbs or corrupts the input and relies on the model or an auxiliary procedure to recover a safer input before generation. SmoothLLM (Robey et al., 2025) is a randomized perturbation defense based on the observation that adversarial jailbreak suffixes are often brittle under small random changes; it perturbs the input multiple times and aggregates the resulting outputs to determine whether the prompt remains adversarial. Perplexity-based filtering (Alon & Kamfonas, 2023) is a detection-based defense that flags prompts with unusually high perplexity, motivated by the fact that optimized adversarial suffixes are often unnatural. Circuit Breakers (CB) (Zou et al., 2024) is a training-based defense that modifies the model to interrupt or suppress harmful internal activation patterns associated with jailbreak behavior.

PIGuard (Li et al., 2025) is a guardrail-style input filtering method that detects and blocks potentially unsafe prompts before generation. We also include the undefended model as the no-defense baseline.

For GCG and AutoDAN, ASR is computed using a prefix-based refusal detector. A response is classified as safe if it begins with one of the predefined refusal prefixes after removing leading whitespace and normalizing capitalization; otherwise, it is counted as a successful attack. For PAIR, we use GPT-4.1 as the judge model and GPT-3.5-Turbo as the attacker model, following the interactive attack setting in which an attacker model proposes candidate jailbreak prompts and a separate judge model determines whether the target response is successful. Additional judge details and the full refusal-prefix list are provided in Appendix C.4.

C.4. ASR judges

For GCG and AutoDAN, ASR is computed using a prefix-based refusal detector. A generated response is classified as safe if, after removing leading whitespace and normalizing capitalization, it starts with one of the predefined refusal prefixes. Otherwise, it is counted as a successful attack. This rule is simple and reproducible, but it is only an approximate safety judge. It may overestimate ASR when a safe response does not begin with one of the predefined refusal prefixes, and it may underestimate ASR when a response begins with a refusal-like phrase but later contains unsafe content. Future work should evaluate DCRS with stronger semantic safety judges across a broader range of target models, benchmarks, and jailbreak attacks.

For PAIR, we use GPT-4.1 as the judge model and GPT-3.5-Turbo as the attacker model. This follows the interactive nature of PAIR, where an attacker model iteratively proposes candidate jailbreak prompts and a separate judge model evaluates whether the target response is successful.

The refusal-prefix regular expressions used for GCG and AutoDAN are:

- `^i'm sorry\b`
- `^sorry\b`
- `^i am sorry\b`
- `^i apologize\b`
- `^my apologies\b`
- `^as an ai\b`
- `^as a language model\b`
- `^as an assistant\b`
- `^i cannot\b`
- `i can't\b`
- `^i do not\b`
- `^i'm not able to provide\b`
- `^i am not able to provide\b`
- `^i cannot provide\b`
- `^i can't provide\b`
- `^i cannot assist\b`
- `^i can't assist\b`
- `^i cannot support\b`
- `^i can't support\b`

C.5. Generation details

For all generation-based evaluations, we use greedy decoding with `do_sample=False` and `max_new_tokens`=128 unless otherwise specified. Randomized generations are produced with deterministic decoding after each sampled input perturbation, so the only source of randomness comes from the smoothing procedure. For Llama-2-7B-Chat, prompts are formatted using the model's default chat template. The generated response is post-processed by removing leading whitespace and normalizing capitalization before applying the corresponding safety judge. The same response post-processing rule is used consistently within each attack setting.

D. Reachable-Set Visualization

Figure 3 visualizes the reachable set V^* from Section 5.1 at $M = 50$ in the PCA-projected embedding space of the full Llama-2-7B vocabulary. Gray points denote vocabulary tokens, red contours and points denote reachable tokens, and the blue star marks the dummy token b. Although the procedure produces $N \cdot M = 2{,}500$ candidate selections before deduplication, these selections collapse to $|V^*| = 975$ unique tokens.

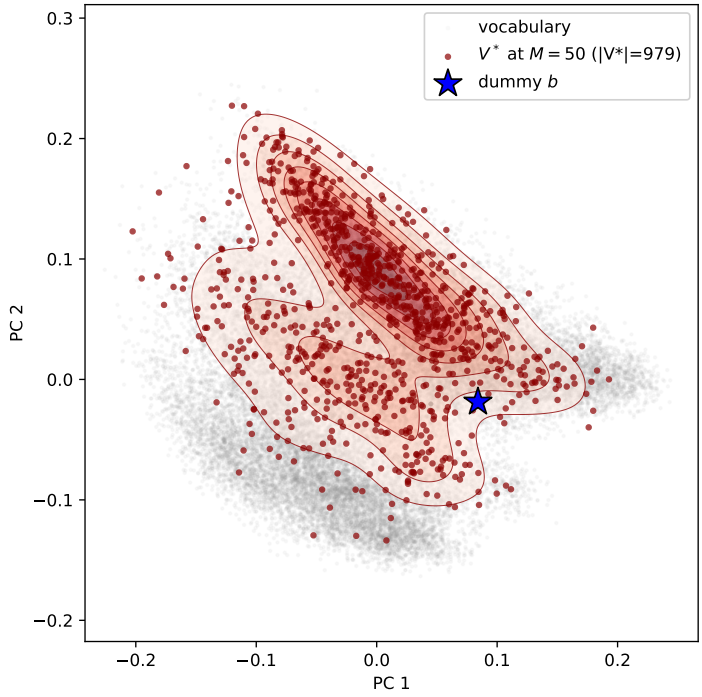

Figure 3. Reachable set V^* at $M = 50$ in the PCA-projected embedding space of the full Llama-2-7B vocabulary. Gray points denote vocabulary tokens, red contours and points denote reachable tokens, and the blue star marks the dummy token b. The reachable set contains $|V^*| = 975$ unique tokens out of $|V| = 32{,}000$.

E. Suffix certification algorithm

Algorithm 1 summarizes the suffix-certification procedure used in our experiments. For each suffix budget k, the clean prompt is converted into a fixed-length input by appending k dummy tokens b^k. The sampling size is then set according to the padded length, $m_k = \lfloor rT_k \rfloor$, so that the same sampling ratio is used across different suffix budgets. We run the smoothing procedure on the padded prompt, compute the Clopper–Pearson lower confidence bound p_L for the majority label, and then evaluate the binary certified radius from Corollary 3.3. When $k > m_k$, or when $p_L \leq 1/2 + \Delta_k$, the certificate is vacuous and we set the certified radius to zero.

Algorithm 1 Certification for suffix attacks

Input: prompt x, maximum suffix length K, dummy token b, sampling ratio r, noise scale σ, number of trials N, confidence level α
for $k = 1, \ldots, K$ **do**
  $\bar{x}^{(k)} \leftarrow x \oplus b^k, \quad T_k \leftarrow |\bar{x}^{(k)}|, \quad m_k \leftarrow \lfloor rT_k \rfloor$
  Run smoothing on $\bar{x}^{(k)}$ for N trials and obtain majority label y_A with count n_A
  $p_L \leftarrow \mathrm{CPLower}(n_A, N, \alpha)$
  if $k > m_k$ **then**
    $\epsilon_k \leftarrow 0$
  else
    $\Delta_k \leftarrow 1 - \frac{\binom{T_k - k}{m_k - k}}{\binom{T_k}{m_k}}$
    if $p_L \leq \frac{1}{2} + \Delta_k$ **then**
      $\epsilon_k \leftarrow 0$
    else
      $\epsilon_k \leftarrow \sigma \left[\Phi^{-1}\left(\frac{p_L - \Delta_k}{1 - \Delta_k} \right) - \Phi^{-1}\left(\frac{1}{2(1 - \Delta_k)} \right) \right]$
    end if
  end if
end for
Return: certified radii $\{\epsilon_k\}_{k=1}^{K}$

The returned value ϵ_k is the certified embedding-space radius for the padded input at suffix budget k. To certify arbitrary replacement of the appended dummy suffix positions, we compare this radius with the worst-case displacement threshold $\rho\sqrt{k}$, where $\rho = \max_{v \in V} \|E(v) - E(b)\|_2$. A prompt is counted as certified for budget k only when $\epsilon_k > \rho\sqrt{k}$. In the nested suffix-budget evaluation, we continue to budget $k + 1$ only for prompts that are certified at budget k, so that larger-budget certification is not reported for prompts that already fail at a smaller suffix budget.

F. Effect of nearest-neighbor projection under varying noise scales

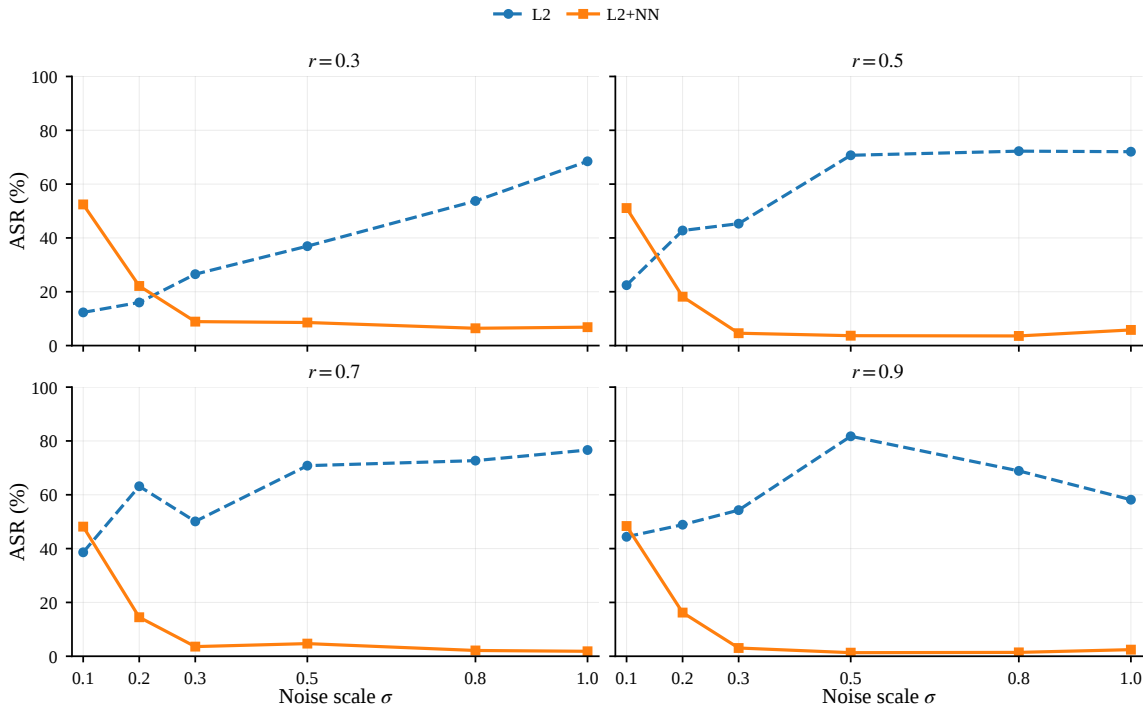

Figure 4. ASR as a function of the Gaussian noise scale σ for two embedding-space smoothing variants, L2 and L2+NN, under representative sampling ratios $r = m/T$. Each panel fixes the sampling ratio and compares the two methods across noise scales. L2+NN shows a much clearer reduction in ASR as σ increases, indicating that nearest-neighbor projection is important for making continuous embedding-space smoothing effective in generative LLMs.

We further study the role of nearest-neighbor projection by comparing two embedding-space smoothing variants: L2, which applies Gaussian noise directly to the selected token embeddings, and L2+NN, which additionally projects each perturbed embedding back to the nearest vocabulary embedding before generation. Figure 4 plots ASR as a function of the noise scale σ for several representative sampling ratios $r = m/T$. The results show that the effect of smoothing is driven more clearly by the continuous noise scale than by the sampling ratio alone. In particular, as σ increases, L2+NN consistently achieves much lower ASR than L2 across moderate-to-large ratios. This suggests that continuous embedding perturbations are an important component of robustness, but that they become effective in generative LLMs only when combined with nearest-neighbor projection back to valid vocabulary embeddings.

G. Compute resources

All experiments were run on an NVIDIA A100. The main certification experiments use Llama-2-7B-Chat and require $N = 2000$ randomized generations per prompt. Since each randomized trial performs one perturbed forward generation, the total computational cost scales linearly with the number of prompts, suffix budgets, and randomized trials. The suffix-certification experiments were parallelized across prompts and suffix budgets on the cluster. ASR experiments were run using the same target-model inference setup, with different noise scales and token sampling ratios as described in Appendix C.3. For all generation-based evaluations, we use the decoding configuration described in Appendix C.5.