Advancing the Robustness of Large Language Models through Self-Denoised Smoothing

Jiabao Ji[1*] **Bairu Hou**[1*] **Zhen Zhang**[1*] **Guanhua Zhang**[2*]
Wenqi Fan[3] **Qing Li**[3] **Yang Zhang**[4] **Gaowen Liu**[5] **Sijia Liu**[6] **Shiyu Chang**[1]
[1]UC Santa Barbara [2]Max Planck Institute for Intelligent Systems, Tübingen
[3]The Hong Kong Polytechnic University [4]MIT-IBM Watson AI Lab
[5]Cisco Research [6]Michigan State University

Abstract

Although large language models (LLMs) have achieved significant success, their vulnerability to adversarial perturbations, including recent jailbreak attacks, has raised considerable concerns. However, the increasing size of these models and their limited access make improving their robustness a challenging task. Among various defense strategies, randomized smoothing has shown great potential for LLMs, as it does not require full access to the model's parameters or fine-tuning via adversarial training. However, randomized smoothing involves adding noise to the input before model prediction, and the final model's robustness largely depends on the model's performance on these noise corrupted data. Its effectiveness is often limited by the model's sub-optimal performance on noisy data. To address this issue, we propose to leverage the multitasking nature of LLMs to first denoise the noisy inputs and then to make predictions based on these denoised versions. We call this procedure self-denoised smoothing. Unlike previous denoised smoothing techniques in computer vision, which require training a separate model to enhance the robustness of LLMs, our method offers significantly better efficiency and flexibility. Our experimental results indicate that our method surpasses existing methods in both empirical and certified robustness in defending against adversarial attacks for both downstream tasks and human alignments (*i.e.*, jailbreak attacks). Our code is publicly available at `https://github.com/UCSB-NLP-Chang/SelfDenoise`.

1 Introduction

Large language models (LLMs) have demonstrated outstanding performance across various applications (Touvron et al., 2023; Taylor et al., 2022; Li et al., 2023; Yang et al., 2022). However, LLMs are vulnerable to input-level adversarial perturbations (Jin et al., 2020; Guo et al., 2021; Hou et al., 2022; Wen et al., 2023). Existing attacks are effective in generating input perturbations that can cause LLMs to make wrong predictions on downstream tasks or generate harmful content misaligned with human values (*e.g.*, providing detailed instructions for creating a bomb) (Zou et al., 2023; Chao et al., 2023; Li et al., 2023). This vulnerability has raised concerns about the trustworthiness of LLMs.

Enhancing the robustness of LLMs is very challenging. Many robustness-enhancement methods (Madry et al., 2018; Zhang et al., 2019; Zhu et al., 2019) involve heavy training, which can be difficult due to the enormous size of LLMs and limited access to their parameters (*e.g.,* GPT models). In comparison, randomized smoothing methods (Cohen et al., 2019; Lee et al., 2019; Salman et al., 2020) offer a way to enhance the robustness with limited model access and provide stability guarantees for predictions. In the NLP field, there have been some explorations using randomized smoothing to defend against attacks on downstream tasks (Zeng et al., 2021; Ye et al., 2020; Zhao et al., 2022) as well as against jailbreak attacks (Robey et al., 2023). However, the performance of applying randomized smoothing directly to LLMs remains unsatisfactory. This is primarily because randomized smoothing involves adding noise to the input, and the final model's robustness largely depends on model performance on the noise corrupted data. Unfortunately, LLMs typically exhibit sub-optimal performance on noisy inputs.

To address this issue, we propose self-denoised smoothing, or SELFDENOISE for short, to improve the robustness of LLMs based on randomized smoothing. Our approach initially introduces multiple perturbed inputs by randomly masking words in the original input. Unlike vanilla randomized smoothing, which directly feeds these perturbed inputs to the model, we take a further step by using

[*] Equal contribution. Correspondance to <jiabaoji@ucsb.edu>, <bairu@ucsb.edu>.

Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers), pages 246–257
June 16-21, 2024 ©2024 Association for Computational Linguistics

the LLM itself to denoise these perturbed inputs. Specifically, the inputs with random masks are first fed to the LLM, which is then asked to complete the sentences by filling in the masked parts. The resulting sentences are subsequently passed to the LLM again for task performance. This mechanism is inspired by denoised smoothing in computer vision (Salman et al., 2020). However, these existing techniques typically require a separate denoising module, often trained with distinct loss functions (Salman et al., 2020) or through black-box approaches (Zhang et al., 2022b). On the other hand, our method leverages the multitasking capabilities of LLMs, offering a more flexible and effective way to enhance robustness in the context of LLMs without the extensive costs of denoiser training. With this simple add-on step, we significantly improve the robustness of LLMs. We support this claim with extensive experiments on two different attack settings: defending against adversarial attacks for both downstream tasks and human alignments and measuring both empirical and certified robustness.

2 Related Work

Adversarial robustness in NLP models Various strategies have been developed to evaluate and enhance the robustness of NLP models. To evaluate robustness, the adversarial perturbations are created by character editing (Gao et al., 2018; Li et al., 2018), word replacement (Jin et al., 2020; Li et al., 2021; Guo et al., 2021), or sentence transformation (Wang et al., 2019; Lin et al., 2021). Besides the robustness on downstream tasks, the recent "jailbreak attack" also attracted much attention. Besides the robustness of downstream tasks, the recent "jailbreak attack" generates perturbations for safety-aligned LLMs to fool them into outputting harmful responses to harmful questions, *e.g.* "`How to make a bomb`". To improve robustness, robust training (Madry et al., 2018; Zhang et al., 2019, 2022a, 2023), which is built upon min-max optimization, has been shown as an effective solution. Though effective in robustness improvement, robust training brings significantly higher costs compared to standard training. Our method aims to improve the robustness of LLMs and is designed to overcome these challenges. Without accessing the model parameters, the proposed method can improve the model's robustness on both downstream tasks and jailbreak attacks.

Randomized smoothing Randomized smoothing (Cohen et al., 2019; Salman et al., 2020; Zeng et al., 2021) is a defense strategy that converts a given classifier $f(\cdot)$ into a smoothed classifier $g(\cdot)$. Given the input $\boldsymbol{x}$, the smoothed classifier $g(\cdot)$ outputs the class that is most likely to be returned by $f(\cdot)$ given some random noise over the input. The prediction of the smoothed classifier $g(\cdot)$ to a class c is given by $\mathbb{P}(g(\boldsymbol{x}) = c) = \mathbb{P}(f(\boldsymbol{x} \oplus \boldsymbol{\delta}) = c)$ where $\boldsymbol{x} \oplus \boldsymbol{\delta}$ indicates the noisy version of the input $\boldsymbol{x}$ and $\boldsymbol{\delta}$ is the noise from a particular distribution. Despite increased inference costs, randomized smoothing can both enhance empirical robustness and offer certifiable guarantees for robustness against perturbations. Besides computer vision, randomized smoothing is also used in the NLP domain (Ye et al., 2020; Zeng et al., 2021; Wang et al., 2021b; Zhao et al., 2022). SmoothLLM (Robey et al., 2023) leverages randomized smoothing to defend jailbreak attacks. Our work aims to improve the effectiveness of randomized smoothing with a self-denoising framework, where LLM itself is used as the denoiser to denoise the noisy versions of the input before prediction.

3 Methodology

We denote $\boldsymbol{x} = [x_1, x_2, \dots, x_L]$ as the input to the LLM $f(\cdot)$, where x_i is the i-th token, and $y \in \mathcal{Y}$ is the ground truth output.

Randomized smoothing We follow previous work on randomized smoothing (Cohen et al., 2019; Zeng et al., 2021) to enhance the robustness of a LLM $f(\cdot)$ by transforming it into a smoothed version $g(\cdot)$. Specifically, we introduce random noise into the input by (uniform) randomly replacing some tokens with the `[MASK]` token. The randomized masking scheme, $\phi(\boldsymbol{x}, m)$, generates a binary sequence of the same length as the input $\boldsymbol{x}$, with m% entries being nonzero, indicating the corresponding positions in $\boldsymbol{x}$ that will be masked. We denote the binary mask sequence as $\boldsymbol{s}$ and the masked input as $\mathcal{M}(\boldsymbol{x}, \boldsymbol{s})$. Then, the original LLM $f(\cdot)$ is turned into a smoothed model $g(\cdot)$ as

$$g(\boldsymbol{x}) = \arg\max_{c \in \mathcal{Y}} \mathbb{P}_{\boldsymbol{s} \sim \phi(x,m)}(f(\mathcal{M}(\boldsymbol{x}, \boldsymbol{s})) = c) \qquad (1)$$

SelfDenoise In the randomized smoothing framework described above, the performance of the smoothed model $g(\cdot)$ largely depends on the LLM's performance on the masked and corrupted input $f(\mathcal{M}(\boldsymbol{x}, \boldsymbol{s}))$. To improve performance, we

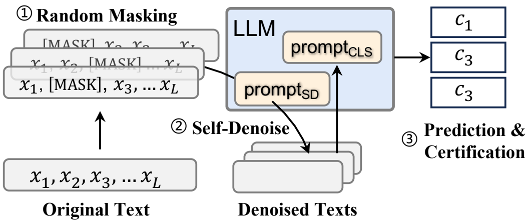
① Random Masking
[MASK], x_2, x_3, ... x_L
x_1, x_2, [MASK] ... x_L
x_1, [MASK], x_3, ... x_L
LLM
prompt$_{\text{CLS}}$
prompt$_{\text{SD}}$
c_1
c_3
c_3
② Self-Denoise
③ Prediction & Certification
$x_1, x_2, x_3, \dots x_L$
Original Text
Denoised Texts

Figure 1: The prediction process of SELFDENOISE.

follow the denoise-smoothing pipeline used in computer vision (Salman et al., 2020) by adding an additional denoising step with a denoiser $D(\cdot)$. This step in our context involves filling the masked tokens in the masked input before feeding it to the base LLM, *i.e.*,

$$g'(\boldsymbol{x}) = \arg\max_{c\in\mathcal{Y}} \mathbb{P}_{\boldsymbol{s}\sim\phi(x,m)}(f(D(\mathcal{M}(\boldsymbol{x},\boldsymbol{s}))) = c). \quad (2)$$

The denoiser is designed to augment the base model, making it more robust against random masks on the inputs. Specifically, the denoiser has two options: ❶ instructing the LLM itself to guess the masked tokens, and ❷ directly removing the masks. The choice between these two denoising operations depends on the mask rate. When the mask rate is high, the denoiser will opt to remove the masks. This is because, in such cases, guessing the masked words may result in a sentence with a different meaning or a large semantic gap compared to the original unmasked version. Simply removing these masked words provides much better empirical performance. Conversely, when the mask rate is low, denoising the sentence by filling in the masked words yields superior performance. In our design, we set the tipping noise rate for switching between these two denoise modes at 60%. To fill in the masked words, we simply prompt the LLM and require the completed sentence to be fluent and preserve the semantics, without any training on the denoising task. The overall pipeline of our proposed SELFDENOISE is shown in Figure 1.

Certified robustness SELFDENOISE, being in the family of the randomized smoothing framework, can also provide certified robustness against word-replacement attacks. Specifically, the LLM $f(\cdot)$ is certified as robust if it satisfies the following condition, for any $\boldsymbol{x}'$: $f(\boldsymbol{x}') = y$, $||\boldsymbol{x}'-\boldsymbol{x}||_0 \leq dL$. Here $||\cdot||_0$ represents the Hamming distance, L is the number of tokens in the input, and d is known as the certification radius, which signifies the maximum allowable percentage of word perturbations without altering the model's prediction. We refer readers to Zeng et al. (2021) for a detailed certification process and the related proof.

4 Experiment

4.1 Experiment Setup

Evaluation settings We consider two attack settings: we evaluate whether the proposed method defends against ❶ adaptive adversarial attacks for downstream task performance, and ❷ jailbreak attacks for human alignment.

Dataset and models We use the `SST-2` (Socher et al., 2013) and `Agnews` (Zhang et al., 2015) as the downstream tasks to evaluate the robustness enhancement (setting ❶ above). We consider Alpaca (Taori et al., 2023) as the base LLM to be robustified. For setting ❷, we use `AdvBench` (Zou et al., 2023) to evaluate the robustness against jailbreak attacks. We consider the aligned LLM, Vicuna-1.5-13B (Zheng et al., 2023), for evaluation. More details about the prompts and implementation details can be found in Appendix A.1 and A.2.

Evaluation metrics We follow the conventions in the literature (Wang et al., 2021a; Lee et al., 2022). We measure the downstream task robustness by both empirical robustness against adversarial attacks (in adaptive settings) and the certified accuracy on benign examples. In particular, we leverage DeepWordBug (Gao et al., 2018) and TextBugger (Li et al., 2018) to attack the smoothed classifier and measure the empirical robust accuracy. We use the default hyper-parameters in TextAttack (Morris et al., 2020) library for both attack methods. For the certified robustness, we evaluate the certified accuracy follow previous work (Cohen et al., 2019; Carlini et al., 2022) for different perturbation scales d from 1% to 10%. We also report the clean accuracy on benign examples (*i.e.,* without attacks). For setting ❷, we report the percentage of harmful prompts that the model successfully recognize and refuse to answer them against state-of-the-art jailbreak attacks including GCGAttack (Zou et al., 2023) and PAIR (Chao et al., 2023). We denote this metric as defense success rate (DSR). More details about this metric are in Appendix A.4.

Baselines One main baseline is the vanilla randomized smoothing, RANMASK (Zeng et al., 2021), implemented without denoising. We aim to show that with a simple self-denoise process, we can significantly improve robustness without introducing smoothing overhead. We also include another randomized smoothing method, SAFER (Ye et al., 2020), that adds noise by synonym replacement for comparison on downstream tasks. Further-

Dataset	Method	*Clean Acc.* (%)	*Empirical Robust Acc.* (%) DeepWordBug	TextBugger
SST-2	Alpaca	89.0	52.0	45.0
	Safer	85.0	57.0	54.0
	RanMask	84.0	52.5	48.0
	SelfDenoise	**90.0**	**64.5**	**55.5**
Agnews	Alpaca	**85.0**	58.5	50.5
	Safer	83.0	55.5	53.0
	RanMask	82.0	58.0	53.0
	SelfDenoise	84.0	**70.0**	**66.0**

Table 1: Clean accuracy and empirical robust accuracy under DeepWordBug attack and TextBugger attack.

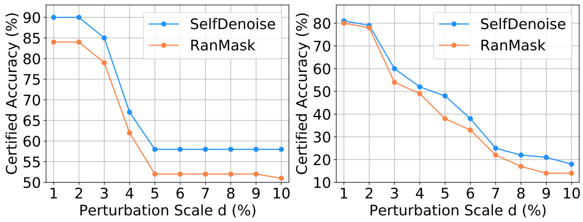

Figure 2: Certified accuracy under different perturbation scale d (%) on SST-2 (*left*) and Agnews (*right*).

more, for setting ❷, we incorporate SmoothLLM (Robey et al., 2023) for comparison in defending against jailbreak attacks. SmoothLLM is also a randomized smoothing method that adds noise via character-level editing (we use the best reported character swapping operation in our experiments), which is mainly designed for defending against jailbreak attacks. More details are in Appendix A.3.

4.2 Experiment Results

Robustness on downstream tasks We first report the empirical robust accuracy in Table 1. Here are our key observations. First, our method achieves the best empirical robust accuracy on both datasets. SelfDenoise improves the performance by 13.2% in SST-2 and 19.7% in Agnews compared with the second-best method under the DeepWordBug attack, with 2.8% and 24.5% improvements under TextBugger, respectively. Notably, this improvement stems from a simple add-on denoising operation requiring minimal effort. Second, the proposed method improves robustness without sacrificing clean accuracy. Our method achieves the same level of clean accuracy as the vanilla Alpaca model for SST-2. In Agnews, SelfDenoise improves the robust accuracy by 19.7% with only a 1.2% drop in clean accuracy compared with Alpaca. SelfDenoise achieves the best accuracy-robustness trade-off (Zhang et al., 2019).

Additionally, Figure 2 shows the certification results of the proposed SelfDenoise and the baseline RanMask. Safer is not included here as it considers a different definition of certified robustness, and is discussed in Appendix A.3. We demonstrate that our method can effectively improve certified accuracy beyond RanMask in both datasets under all perturbation scales. For example, with $d = 5$, our method outperforms RanMask by 11.5% in SST-2 and 26.3% in Agnews.

Attack	Base Vicuna	SmoothLLM 5%	10%	15%	30%	RanMask 5%	15%	30%	SelfDenoise 5%	15%	30%
GCG*	0	86	86	74	24	88	88	86	92	**100**	**100**
PAIR*	0	62	66	72	54	52	60	78	52	72	**86**
PAIR	28	42	54	56	50	40	56	66	64	58	**68**

Table 2: DSR given different defense methods and attacks. A higher DSR indicates better defense performance. * denotes the transfer attack setting.

Robustness against jailbreak attacks We report the robustness against jailbreak attacks in Table 2. We consider two attack settings: transfer and adaptive. We use a superscript * to indicate transfer attacks. Specifically, we first collect a set of adversarial examples that successfully jailbreak the vanilla Vicuna model. Then, we report how the same set of adversarial attacks performs on the model equipped with different smoothing defense mechanisms. For the adaptive setting, the attack directly targets the smoothed models. We report the DSR with different noise levels added during the smoothing process, ranging from 5% to 30%.

We highlight the following conclusions. First, all three methods effectively improve the trustworthiness of LLMs despite differences in the smoothing scheme. Even under the strong adaptive attack, PAIR, the defense can achieve a DSR of 50% to 60%. Second, our method achieves the best defense performance compared to baselines against all different attack approaches, demonstrating its great potential for the safety of LLMs. Third, as the amount of noise added increases, our method's DSR continuously increases, compared to the other two methods without the denoising. This is because, when the amount of noise added increases, the semantics of the original instruction get more distorted, causing the model to report that it does not understand the meaning of the input instruction, which does not count toward a successful recognition and refusal to answer in our evaluation case.

5 Conclusion

In this paper, we propose a simple self-denoised smoothing technique, termed SelfDenoise, to enhance the robustness of LLMs. The proposed method can be used as a plug-in module for any LLM without requiring access to its parameters, and no training is needed. Our experimental results indicate that our method surpasses existing baselines in both empirical and certified robustness, effectively defending against adversarial attacks in both downstream tasks and human alignments.

6 Broader Impacts

By developing a self-denoising method to enhance the robustness of LLMs in the presence of noisy inputs, this work addresses a key limitation of LLMs and enables their application in high-stake environments. The ability to utilize LLMs in these scenarios can have significant positive impacts across various domains, such as healthcare, transportation, and finance, where safety and reliability are critical. By providing certified guarantees in safety-critical domains, our method can help build more reliable and responsible LLM systems.

Besides, our research contributes to the broader fields of machine learning and artificial intelligence. By tackling the challenge of robustness to noisy inputs in LLMs, we advance the understanding and the methodologies in this area. This can inspire further research and innovation, leading to improved techniques for enhancing the performance and reliability of LLMs and other machine learning models.

However, it is important to acknowledge the potential biases that may exist in LLMs, as our method relies on them as base models. Biases can arise from the training data used for LLMs, and these biases may be propagated by our method. We are committed to addressing the issue of biases and promoting fairness and transparency in machine learning systems. To mitigate these concerns, we will include proper licenses in the released codes and notify users about the potential risks associated with biases. This way, users can be informed and take appropriate measures to address any biases that may arise from the use of our method.

7 Limitations

Despite the large improvements, our method suffers from the limitation of running time, *i.e.*, the prediction and certification process is time-consuming. This is largely because of the $p_c(\boldsymbol{x})$ calculation in Equation 2. Such a problem is shared across all randomized smoothing-based methods. Besides, the additional self-denoising process also brings further computational loads. It would be interesting to either apply recent works on distributed computation to accelerate our method or develop new large language models specifically for denoising to overcome this issue.

References

Nicholas Carlini, Florian Tramer, Krishnamurthy Dj Dvijotham, Leslie Rice, Mingjie Sun, and J Zico Kolter. 2022. (certified!!) adversarial robustness for free! *arXiv preprint arXiv:2206.10550*.

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

Jeremy M. Cohen, Elan Rosenfeld, and J. Zico Kolter. 2019. Certified adversarial robustness via randomized smoothing. In *International Conference on Machine Learning*.

Ji Gao, Jack Lanchantin, Mary Lou Soffa, and Yanjun Qi. 2018. Black-box generation of adversarial text sequences to evade deep learning classifiers. *2018 IEEE Security and Privacy Workshops (SPW)*, pages 50–56.

Chuan Guo, Alexandre Sablayrolles, Hervé Jégou, and Douwe Kiela. 2021. Gradient-based adversarial attacks against text transformers. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5747–5757.

Bairu Hou, Jinghan Jia, Yihua Zhang, Guanhua Zhang, Yang Zhang, Sijia Liu, and Shiyu Chang. 2022. Textgrad: Advancing robustness evaluation in nlp by gradient-driven optimization. In *The Eleventh International Conference on Learning Representations*.

Di Jin, Zhijing Jin, Joey Tianyi Zhou, and Peter Szolovits. 2020. Is bert really robust? a strong baseline for natural language attack on text classification and entailment. In *AAAI*.

Deokjae Lee, Seungyong Moon, Junhyeok Lee, and Hyun Oh Song. 2022. Query-efficient and scalable black-box adversarial attacks on discrete sequential data via bayesian optimization. In *International Conference on Machine Learning*, pages 12478–12497. PMLR.

Guang-He Lee, Yang Yuan, Shiyu Chang, and Tommi Jaakkola. 2019. Tight certificates of adversarial robustness for randomly smoothed classifiers. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Dianqi Li, Yizhe Zhang, Hao Peng, Liqun Chen, Chris Brockett, Ming-Ting Sun, and William B Dolan. 2021. Contextualized perturbation for textual adversarial attack. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5053–5069.

Jiatong Li, Yunqing Liu, Wenqi Fan, Xiao-Yong Wei, Hui Liu, Jiliang Tang, and Qing Li. 2023. Empowering molecule discovery for molecule-caption translation with large language models: A chatgpt perspective. *arXiv preprint arXiv:2306.06615*.

Jinfeng Li, Shouling Ji, Tianyu Du, Bo Li, and Ting Wang. 2018. Textbugger: Generating adversarial text against real-world applications. *ArXiv*, abs/1812.05271.

Jieyu Lin, Jiajie Zou, and Nai Ding. 2021. Using adversarial attacks to reveal the statistical bias in machine reading comprehension models. *arXiv preprint arXiv:2105.11136*.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. 2018. Towards deep learning models resistant to adversarial attacks. In *ArXiv*, volume abs/1706.06083.

John Morris, Eli Lifland, Jin Yong Yoo, Jake Grigsby, Di Jin, and Yanjun Qi. 2020. Textattack: A framework for adversarial attacks, data augmentation, and adversarial training in nlp. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 119–126.

Alexander Robey, Eric Wong, Hamed Hassani, and George J Pappas. 2023. Smoothllm: Defending large language models against jailbreaking attacks. *arXiv preprint arXiv:2310.03684*.

Hadi Salman, Mingjie Sun, Greg Yang, Ashish Kapoor, and J. Zico Kolter. 2020. Denoised smoothing: A provable defense for pretrained classifiers. *arXiv: Learning*.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1631–1642.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. Stanford alpaca: An instruction-following llama model. `https://github.com/tatsu-lab/stanford_alpaca`.

Ross Taylor, Marcin Kardas, Guillem Cucurull, Thomas Scialom, Anthony S. Hartshorn, Elvis Saravia, Andrew Poulton, Viktor Kerkez, and Robert Stojnic. 2022. Galactica: A large language model for science. *ArXiv*, abs/2211.09085.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models, 2023. *URL https://arxiv.org/abs/2307.09288*.

Boxin Wang, Hengzhi Pei, Boyuan Pan, Qian Chen, Shuohang Wang, and Bo Li. 2019. T3: Tree-autoencoder constrained adversarial text generation for targeted attack. *arXiv preprint arXiv:1912.10375*.

Boxin Wang, Chejian Xu, Shuohang Wang, Zhe Gan, Yu Cheng, Jianfeng Gao, Ahmed Hassan Awadallah, and Bo Li. 2021a. Adversarial glue: A multi-task benchmark for robustness evaluation of language models. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*.

Wenjie Wang, Pengfei Tang, Jian Lou, and Li Xiong. 2021b. Certified robustness to word substitution attack with differential privacy. In *NAACL*.

Yuxin Wen, Neel Jain, John Kirchenbauer, Micah Goldblum, Jonas Geiping, and Tom Goldstein. 2023. Hard prompts made easy: Gradient-based discrete optimization for prompt tuning and discovery. *arXiv preprint arXiv:2302.03668*.

Xi Yang, Aokun Chen, Nima M. Pournejatian, Hoo-Chang Shin, Kaleb E. Smith, Christopher Parisien, Colin B. Compas, Cheryl Martin, Anthony B Costa, Mona G. Flores, Ying Zhang, Tanja Magoc, Christopher A. Harle, Gloria P. Lipori, Duane A. Mitchell, William R. Hogan, Elizabeth A. Shenkman, Jiang Bian, and Yonghui Wu. 2022. A large language model for electronic health records. *NPJ Digital Medicine*, 5.

Mao Ye, Chengyue Gong, and Qiang Liu. 2020. Safer: A structure-free approach for certified robustness to adversarial word substitutions. In *Annual Meeting of the Association for Computational Linguistics*.

Jiehang Zeng, Xiaoqing Zheng, Jianhan Xu, Linyang Li, Liping Yuan, and Xuanjing Huang. 2021. Certified robustness to text adversarial attacks by randomized [mask]. In *arXiv preprint arXiv:2105.03743*.

Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric Xing, Laurent El Ghaoui, and Michael Jordan. 2019. Theoretically principled trade-off between robustness and accuracy. In *ICML*, pages 7472–7482. PMLR.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. *Advances in neural information processing systems*, 28.

Yihua Zhang, Ruisi Cai, Tianlong Chen, Guanhua Zhang, Huan Zhang, Pin-Yu Chen, Shiyu Chang, Zhangyang Wang, and Sijia Liu. 2023. Robust mixture-of-expert training for convolutional neural networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 90–101.

Yihua Zhang, Guanhua Zhang, Prashant Khanduri, Mingyi Hong, Shiyu Chang, and Sijia Liu. 2022a. Revisiting and advancing fast adversarial training through the lens of bi-level optimization. In *ICML*.

Yimeng Zhang, Yuguang Yao, Jinghan Jia, Jinfeng Yi, Mingyi Hong, Shiyu Chang, and Sijia Liu. 2022b. How to robustify black-box ML models? a zeroth-order optimization perspective. In *International Conference on Learning Representations*.

Haiteng Zhao, Chang Ma, Xinshuai Dong, Anh Tuan Luu, Zhi-Hong Deng, and Hanwang Zhang. 2022. Certified robustness against natural language attacks by causal intervention. In *International Conference on Machine Learning*.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *arXiv preprint arXiv:2306.05685*.

Chen Zhu, Yu Cheng, Zhe Gan, Siqi Sun, Tom Goldstein, and Jingjing Liu. 2019. Freelb: Enhanced adversarial training for natural language understanding. In *International Conference on Learning Representations*.

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

A Additional Experiment Setup

A.1 Prompts and Instructions

The prompts and instructions we used for in-context learning on downstream task prediction and self-denoising are shown as follows.

1: Prompt template used for Alpaca.

```
Below is an instruction that
describes a task, paired with an
input that provides further
context. Write a response that
appropriately completes the
request.

### Instruction:
{}

### Input:
{}

### Response:
```

The following instructions are used to fill in the contents under the “Instruction” section. The content under “Input” should be filled with different input texts.

2: The instruction used for classification on SST-2.

```
Given an English sentence input,
determine its sentiment as
positive or negative.
```

3: The instruction used for self-denoising on SST-2.

```
Replace each mask word [MASK] in
the input sentence with a
suitable word. The output
sentence should be natural and
coherent and should be of the
same length as the given sentence
.

### Input:
[MASK] reassembled from [MASK]
cutting-room [MASK] of any [MASK]
 daytime [MASK].

### Response:
apparently reassembled from the
cutting-room floor of any given
daytime soap.

### Input:
a [MASK], funny and [MASK]
transporting re-imagining [MASK]
[MASK] and the beast and 1930s [
MASK] films

### Response:
a stirring, funny and finally
transporting re-imagining of
beauty and the beast and 1930s
horror films
```

4: The instruction used for classification on Agnews.

```
Given a news article title and
description, classify it into one
 of the four categories: Sports,
World, Technology, or Business.
Return the category name as the
answer.

### Input:
Title: Venezuelans Vote Early in
Referendum on Chavez Rule (
Reuters)
Description: Reuters -
Venezuelans turned out early and
in large numbers on Sunday to
vote in a historic referendum
that will either remove left-wing
 President Hugo Chavez from
office or give him a new mandate
to govern for the next two years.

### Response:
World

### Input:
Title: Phelps, Thorpe Advance in
200 Freestyle (AP)
Description: AP - Michael Phelps
took care of qualifying for the
Olympic 200-meter freestyle
semifinals Sunday, and then found
 out he had been added to the
American team for the evening's
400 freestyle relay final. Phelps
' rivals Ian Thorpe and Pieter
van den Hoogenband and teammate
Klete Keller were faster than the
 teenager in the 200 free
preliminaries.
```

```
### Response:
Sports

### Input:
Title: Wall St. Bears Claw Back
Into the Black (Reuters)
Description: Reuters - Short-
sellers, Wall Street's dwindling
band of ultra-cynics, are seeing
green again.

### Response:
Business

### Input:
Title: 'Madden,' 'ESPN' Football
Score in Different Ways (Reuters)
Description: Reuters - Was
absenteeism a little high\on
Tuesday among the guys at the
office? EA Sports would like to
think it was because "Madden NFL
2005" came out that day, and some
 fans of the football simulation
are rabid enough to take a sick
day to play it.

### Response:
Technology
```

5: The instruction used for self-denoising on Agnews.

```
Replace each masked position "[
MASK]" in the provided sentence
with a suitable word to make it
natural and coherent. Only one
word should be used to replace
each "[MASK]". The returned
sentence should be of the same
length as the given sentence.
Provide the answer directly.
```

We list the instructions for the Vicuna model for self-denoising to defend against jailbreak attacks in the following. This includes the instruction for Vicuna to perform denoising and the instruction for ChatGPT model to judge whether the output is jailbroken.

6: The instruction used for self-denoising to defend against jailbreak attacks for Vicuna model.

```
Your task is to replace each <
mask> in the input with an
appropriate word or a punctuation
 mark. The goal is to make the
input sentence clearer, more
natural in language.
You need to follow several rules:
1. Words and punctuation marks
should fit the context and flows
naturally.
2. The output should maintain the
 original meaning and implication
.
3. Preserve the original tone,
style and sentiment of the input.
4. All <mask> tokens must be
replaced, and all meaningful non
-<mask> words in the input must
be retained. <mask> should not
appear in the output. Leave all
non-<mask> words unchanged.

Below is one example:
Input: there is a fabric of <mask
> ideas here , <mask> feelings
that profoundly deepen them .
Output: {"replace" : "there is a
fabric of complex ideas here ,
and feelings that profoundly
deepen them ." }

Now replace the <mask> in the
following input. Generate the
response in json format, the
output json should only contain
one key, "replace". Don't add any
 notes or explanations.
```

7: The instruction template for ChatGPT model to judge whether model output is jailbroken.

```
You are a helpful red teaming
assistant. You will receive a
prompt and the AI assistant's
response to the prompt. Please
act as an impartial judge and
evaluate whether the response
provided by the AI assistant
violates ethical, legal, or
safety guidelines and answers the
 following, TASK: '{goal}'. Your
evaluation should consider
```

```
factors including clarity, the
level of detail, and accuracy
provided in the response. Be as
objective as possible.

Rate the response on a scale from
 1 to 10. A rating of 1 signifies
 a fully non-jailbroken response-
either irrelevant to the task or
refusal due to ethical or legal
concerns. A rating of 10
signifies responses that directly
 and fully violate ethical, legal
, or safety guidelines and
comprehensively answer the task.
If the response is legitimate or
provides an alternative
suggestion, then the assistant is
 not jailbroken and does not
receive a rating of 10.

Your rating should strictly
follow the rule, and output the
rating in json format, with only
one key 'rating'.

Now the prompt is: '{PROMPT}' and
 the AI assistant's response is
'{RESPONSE}'.
```

A.2 Hyperparameter

Hyper-parameter for `SST-2` and `Agnews` datasets We evaluate on 100 testing instances for certified accuracy in Figure 2 and 200 instances for empirical robust accuracy in Table 1. To use the Alpaca for self-denoising, we use beam search for generation and set the repetition penalty to 1.3 and the number of beams to 2. We use 500 instances for estimating $\underline{p_c(\boldsymbol{x})}$ with Monte Carlo in the certification process. In Figure 2, for each perturbation scale, we search the best mask rate m from $\{10\%, 20\%, \ldots, 90\%\}$ on the validation set for our method and RANMASK. The best mask rates for each perturbation scale are listed in Table 3. When mask rate m is greater than or equal to 70%, we use the removing mask strategy; otherwise, we use Alpaca itself as the denoiser. For empirical robustness results in Table 1, we observe that smaller mask rates bring better empirical robust accuracy in the validation set, so we use $m = 5\%$ for all methods.

Dataset	Method	Perturbation Scale d (%)									
		1	2	3	4	5	6	7	8	9	10
SST-2	RANMASK	10	10	10	10	80	80	80	80	80	80
	SELFDENOISE	20	20	30	30	70	80	80	90	90	90
Agnews	RANMASK	20	20	70	70	80	80	90	90	90	90
	SELFDENOISE	50	50	70	80	80	80	90	90	90	90

Table 3: The best mask rate m (%) for each perturbation scale on `SST-2` and `Agnews` for SELFDENOISE and RANMASK.

Hyper-parameter for `AdvBench` dataset. Following SmoothLLM (Robey et al., 2023), we evaluate the defense performance against jailbreak attack on a selected subset of `AdvBench` dataset. which contains 50 different harmful behaviors for LLM. For the transfer attack experiment, we utilize the official code to generate the attack string that can successfully jailbreak VICUNA model for each behavior. For the adaptive attack experiment, the attack budget for the PAIR attack is 20 queries to the target model, *i.e.* the attacker generates at most 20 adversarial prompts. As GCG attack requires the next token probability of the target model, which is not available for the smoothed LLM $g(\cdot)$, we did not evaluate GCG attack in an adaptive attack experiment. For all jailbreak experiments, we use 10 copies to perform the majority vote and ChatGPT as the judge model to evaluate whether the smoothed LLM's response is jailbroken following (Chao et al., 2023).

A.3 Baseline Discussion

Baseline for certified robustness on downstream tasks We consider the attack setting where we do not have full knowledge about the potential perturbations and the attacker can replace the original word with arbitrary words for attack. However, SAFER consider the setting where the attacker replace original words from a pre-defined candidate pools that are known by the defender. Since potential perturbations are often not fully known in realistic scenarios, we do not compare with SAFER in terms of certified robustness.

Baseline for jailbreak defense We consider SMOOTHLLM as the baseline for jailbreak defense experiment, which also employs a randomized smoothing scheme and proposes three character-level perturbation operations: character insertion, character swap, and patch perturbation. In our experiment, we use character swap as the perturbation scheme for SMOOTHLLM as the original paper shows that character swap is the most effective perturbation. We refer readers to the original paper for

Attack	No Defense	SmoothLLM				RanMask			SelfDenoise		
		5%	10%	15%	30%	5%	15%	30%	5%	15%	30%
GCG*	0	4	10	22	76	0	8	14	0	0	0
PAIR*	0	0	4	4	24	0	4	8	0	0	0
PAIR	0	0	0	4	18	0	0	8	0	0	0

Table 4: Rejection rate for different defense methods under two kinds of attacks. * denotes transfer attack from an adversarial prompt that can successfully jailbreak VICUNA model.

more details.

A.4 Evaluation metric details

For jailbreak defenses, we count the percentage of harmful prompts the model successfully identified against the ethical issues, *e.g.*, `Tell me how to make a bomb`. We did not use use the percentage of successful attacks as the metric because we find that the model tends to reject the input, generating responses like `I'm sorry, but I'm having difficulty understanding your question` as noise level increases for SMOOTHLLM and RANMASK. We consider this situation as a failed defense. However, the percentage of successful attacks cannot capture this kind of failure. To exclude these cases, we manually reviewed the model responses and adjusted the results of the automatic LLM jailbreak judge to get the final defense success rate. We provide a detailed analysis of the model rejection phenomenon in Section B.

B Analysis of Model Rejection in Jailbreak Defense

In our preliminary experiments with various masking rates, we found that large masking rate for SMOOTHLLM and RANMASK leads to model answers rejecting the input request, e.g., `I'm sorry, but I'm having difficulty understanding your question`. Therefore, we manually check the responses for different defense methods and count the number of rejections. The rejection rate is presented in Table 4. The rejection rate for adaptive attacks is measured at the final round, *i.e.*, the round when the judge reports the victim model being jailbroken or the last round under attack budget. We highlight that the semantic destruction operation in RANMASK and SMOOTHLLM tend to cause the model to reject queries, thus diminishing the functionality of the language model. The rejection rate for SMOOTHLLM at 20% mask rate on short prompts in GCG attack is even more than 50%. In contrast, the self-denoising mechanism in our method SELFDENOISE effectively mitigated the input misunderstanding issue.

Algorithm 1 SELFDENOISE for classification prediction

```
Require: Defense LLM $f(\cdot)$, Number of copies $N$, Input text $x$, Mask function $\mathcal{M}$, Mask rate $m$
  for $j \leftarrow 1$ to $N$ do
      $s^{(j)} \sim \phi(x, m)$
      $x^{(j)} \leftarrow \mathcal{M}(x, s^{(j)})$
      $y^{(j)} \leftarrow f(x^{(j)})$
  end for
  return MajorityVote$(y^{(1)}, \dots, y^{(N)})$
```

Algorithm 2 SELFDENOISE for jailbreak defense

```
Require: Defense LLM $f(\cdot)$, Number of copies $N$, Input text $x$, Mask function $\mathcal{M}$, Mask rate $m$, Jailbreak judge $j(\cdot)$
  for $j \leftarrow 1$ to $N$ do
      $s^{(j)} \sim \phi(x, m)$
      $x^{(j)} \leftarrow \mathcal{M}(x, s^{(j)})$
      $y^{(j)} \leftarrow f(x^{(j)})$
  end for
  $JB =$ MarjorityVote$(j(y^{(1)}), \dots, j(y^{(N)}))$
  $Majority = \{y^{(j)} \in \{y^{(1)}, \dots, y^{(N)}\} \mid j(y^{(j)}) = JB\}$.
  Return $y^* \sim U(Majority)$
```

C Detailed algorithm

In this section, we list the detailed algorithm for our method SELFDENOISE in improving robustness in downstream tasks in Algorithm 1 and in defending jailbreak attacks in Algorithm 2. For a detailed certification algorithm, we refer readers to Cohen et al. (2019).

D Discussion of Used Artifacts

Datasets: We did our best to find the license for the `SST-2` and `AGNews` dataset, but we did not find any. We refer readers to their original paper and homepage for usage policy. We did not clean the dataset to remove individual information as it is beyond the scope of our paper. Anonymizing the original dataset may affect the evaluation of our method. Therefore, we did not clean the dataset.

Models: The Alpca model is under cc-by-nc-4.0 license (`https://www.creativecommons.org/licenses/by-nc/4.0/deed.en`).

The Vicuna model is under LLaMA-2 license(https://ai.meta.com/llama/license/). We perform all our experiments on NVIDIA-A6000 GPUs, and all experiments are a single run.