Certified Robustness for Large Language Models with Self-Denoising

Anonymous ACL submission

Abstract

Although large language models (LLMs) have achieved great success in vast real-world applications, their vulnerabilities towards noisy inputs have significantly limited their uses, especially in high-stake environments. In these contexts, it is crucial to ensure that every prediction made by large language models is stable, *i.e.*, LLM predictions should be consistent given minor differences in the input. This largely falls into the study of certified robust LLMs, *i.e.*, all predictions of LLM are certified to be correct in a local region around the input. Randomized smoothing has demonstrated great potential in certifying the robustness and prediction stability of LLMs. However, randomized smoothing requires adding noise to the input before model prediction, and its certification performance depends largely on the model's performance on corrupted data. As a result, its direct application to LLMs remains challenging and often results in a small certification radius. To address this issue, we take advantage of the multitasking nature of LLMs and propose to denoise the corrupted inputs with LLMs in a self-denoising manner. Different from previous works like denoised smoothing, which requires training a separate model to robustify LLM, our method enjoys far better efficiency and flexibility. Our experiment results show that our method outperforms the existing certification methods under both certified robustness and empirical robustness.

1 Introduction

Large language models have shown exceptional performances in vast applications (Touvron et al., 2023; Wu et al., 2023; Taylor et al., 2022; Li et al., 2023; Yang et al., 2022; Nijkamp et al., 2023), even outperforming humans over multiple benchmarks (Chowdhery et al., 2022). However, unlike human intelligence, LLMs are vulnerable to noises and perturbations on the input which does not change the semantic meaning. For example, as shown in Figure 1, with minor changes in the input, the state-of-the-art ChatGPT model gives opposite predictions. Such vulnerability has impeded LLMs from being used in high-stake environments, like financial and medical applications, where prediction stability and reliability are crucial. To address the problem, it largely falls into the study of certified robustness (Cohen et al., 2019), which ensures that all predictions made by the model are correct within a local region around the input.

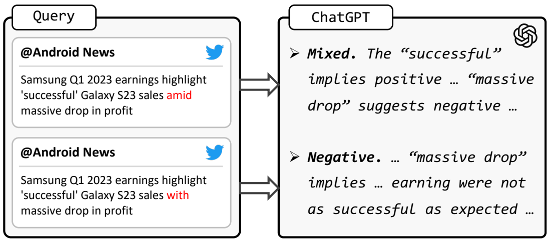

Figure 1: Prompting LLMs for Tweet sentiment analysis. The state-of-the-art ChatGPT language model shows vulnerabilities to minor changes in the input.

The enormous model size and limited access to parameters of LLMs have brought great obstacles to most certification techniques (Shi et al., 2020). As a result, as far as we know, the only potential way to provide a certified robustness guarantee for LLMs is randomized smoothing, which converts the original LLM into a smoothed model (Zeng et al., 2021a). However, the certification performances by directly applying randomized smoothing in LLMs are still far from satisfactory. The underlying reason is that, randomized smoothing requires adding noise to the input before model prediction, and its certification performance depends largely on the LLM's performance on corrupted data. Several previous works alleviate the problem by fine-tuning the model with noisy inputs for a certain task, while this is infeasible for LLMs due to the partial access to parameters and the huge computational costs for fine-tuning.

To address this issue, in this paper, we propose SELFDENOISE, a self-denoising LLM certification framework based on randomized smoothing. The proposed approach first generates multiple perturbed inputs by randomly masking words in the original input. Different from vanilla randomized smoothing which directly feeds these perturbed inputs to the model, we additionally denoise these perturbed inputs by using the LLM itself as a denoiser. Specifically, the perturbed inputs are fed to the LLM, and the LLM is asked to complete the sentences by filling in the masked parts. The resulting sentences are then forwarded to LLM for performing certain downstream tasks such as sentiment analysis. Such a denoising mechanism is inspired by denoised smoothing (Salman et al., 2020), where a separate model is trained to robustify the base model. Extensive experiments are conducted on two datasets using state-of-the-art LLM, Alpaca, and the results show our superiority over baselines on both certified and empirical robustness.

2 Related Work

Certifying the robustness of neural networks is challenging due to the non-convexity and the growing size of neural networks. The mainstream of existing works can be divided into two main categories: ① linearization-based verification that is often based on the branch and bound (BaB) technique (Zhang et al., 2019; Singh et al., 2019; Gehr et al., 2018; Bonaert et al., 2021; Mirman et al., 2018; Jia et al., 2019; Huang et al., 2019). ② certification with randomized smoothing (Cohen et al., 2019; Salman et al., 2020; Levine and Feizi, 2019; Zhao et al., 2022; Zeng et al., 2021a; Ye et al., 2020). Linearization-based method recursively splits the original verification problem into subdomains (*e.g.*, splitting a ReLU activation into positive/negative linear regions by adding split constraints). Then each sub-domain is verified with specialized incomplete verifiers. With the enormous model size and non-linear operations (*e.g.*, self-attention), it is very challenging to verify LLMs. The discrete nature of text data makes certification even more difficult as it poses extra challenges on optimization. Due to the difficulty of applying linearization-based methods on LLMs, we focus on randomized smoothing-based methods.

Several existing works have adopted randomized smoothing in the NLP domain, where noises are added to the input by uniformly sampling some positions in the input and then mask them (Zeng et al., 2021a) or replace them with their synonyms (Ye et al., 2020; Wang et al., 2021; Zhao et al., 2022). Among them, the methods that replace selected tokens with synonyms (*e.g.*, SAFER, Ye et al. (2020)) introduce additional assumptions on the perturbations. However, in realistic scenarios, we do not have full knowledge about the potential perturbations, making these methods less practical. Therefore, in this paper, we add noises by masking the selected tokens, *i.e.*, replacing them with *[MASK]*. Besides, the certification performance of randomized smoothing depends largely on the model's performance on masked inputs. Existing methods finetune the base model (Zeng et al., 2021a; Zhao et al., 2022) or train an additional denoiser (Salman et al., 2019), which requires access to the LLM parameter and huge computational costs. In contrast, we propose a self-denoising framework where LLM itself is used as the denoiser for free.

3 Preliminaries and Notation

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

Certified robustness The model $f(\cdot)$ is certified robust if it satisfies following condition for any $\boldsymbol{x}$,

$$f(\boldsymbol{x}') = y, \; ||\boldsymbol{x}' - \boldsymbol{x}||_0 \leq dL\,, \tag{1}$$

where we use $||\boldsymbol{x}' - \boldsymbol{x}||_0$ to denote the Hamming distance, *i.e.*, $\sum_{i=1}^{L} \mathbb{I}(x'_i \neq x_i)$ with $\mathbb{I}(\cdot)$ as the indicator function, and d refers to perturbation scale. A certified robust LLM is expected to generate the correct output y, given at max d percentage word perturbation on the input. Our definition for robustness differs from previous works (Ye et al., 2020) in that we do not assume a synonym candidate list for word replacement in $\boldsymbol{x}'$, *i.e.*, each position could be replaced to any word, to mimic the vast kinds of noisy inputs in real-world applications.

Randomized smoothing Randomized smoothing robustify the original LLM $f(\cdot)$ by turning it into a smoothed model $g(\cdot)$, which returns the most likely output predicted by $f(\cdot)$, *i.e.*,

$$g(\boldsymbol{x}) = \arg\max_{c \in \mathcal{Y}} \underbrace{P_{\boldsymbol{s} \sim \mathcal{U}(L,m)}(f(\mathcal{M}(\boldsymbol{x}, \boldsymbol{s})) = c)}_{p_c(\boldsymbol{x})}, \tag{2}$$

where we introduce $\boldsymbol{s}$ as a mask position selector, sampled from a uniform distribution $\mathcal{U}(L, m)$ over all possible sets of mL unique indices of $\{1, \ldots, L\}$. $\mathcal{M}$ refers to the masking operation, which masks the corresponding m percent words indicated by $\boldsymbol{s}$ with *[MASK]*. $p_c(\boldsymbol{x})$ refers to the probability that f returns class c after random masking. The smoothed classifier predictions are certified to be consistent with input perturbations,

Theorem 1. *For any $\boldsymbol{x}$, $\boldsymbol{x}'$, $||\boldsymbol{x} - \boldsymbol{x}'||_0 \leq dL$, if*

$$\underline{p_c(\boldsymbol{x})} - \beta\Delta > 0.5, \tag{3}$$

then with probability at least $(1 - \alpha)$, $g(\boldsymbol{x}') = c$.

where $\underline{p_c(\boldsymbol{x})}$ refers to a lower bound on $p_c(\boldsymbol{x})$. β is set to 1 in Levine and Feizi (2019) and approximated with $p_c(\boldsymbol{x})$ in Zeng et al. (2021b). $\Delta = 1 - \binom{L-dL}{L-mL}/\binom{L}{L-mL}$ is determined by the input length L, masked word percentage m and perturbation scale d. We refer the readers to Zeng et al. (2021b); Cohen et al. (2019) for detailed calculation of $\underline{p_c(\boldsymbol{x})}$, β and Δ, and the related proof.

In practice, for a certain $\boldsymbol{x}$ and scale d, one could try different values of masked word percentage m to calculate the corresponding $\underline{p_c(\boldsymbol{x})}$, Δ and β. The model $g(\cdot)$ is certified to be robust on $\boldsymbol{x}$ with scale d if the probability that f returns ground truth label $\underline{p_y(x)} - \beta\Delta > 0.5$, following Equation 3. We then use $r = \max_{(\underline{p_y(x)} - \beta\Delta > 0.5)} d$ as the certification radius on $\boldsymbol{x}$, *i.e.*, perturbations with at most d percent words cannot alter model prediction.

4 Methodology

The performance of randomized smoothing largely depends on $p_y(\boldsymbol{x})$, which is determined by the performances of the base model $f(\cdot)$ on the masked inputs $\mathcal{M}(\boldsymbol{x}, \boldsymbol{s})$. However, naively applying the randomized smoothing on the base LLM could give a small certification radius as the LLMs are not trained to be robust to random masks on the inputs for downstream tasks. As discussed, many previous works alleviate this problem by fine-tuning the base model (Zeng et al., 2021b; Ye et al., 2020) or training an external denoiser (Salman et al., 2020) to augment the base model with better performances on masked texts. Despite the effectiveness, these methods require access to the parameters of LLMs, which is often unavailable. In the following, we will show how to use LLM itself as a denoiser in a self-denoising manner.

Our objective is to improve the randomized smoothing certification radius on existing LLMs with no access to parameters and no further training. Specifically, we add an additional denoising step with a denoiser $D(\cdot)$, which processes the masked input before feeding it to the base LLM, *i.e.*,

$$g'(\boldsymbol{x}) = \arg\max_{c \in \mathcal{Y}} P_{\boldsymbol{s} \sim \mathcal{U}(L,k)}(f(D(\mathcal{M}(\boldsymbol{x}, \boldsymbol{s}))) = c). \tag{4}$$

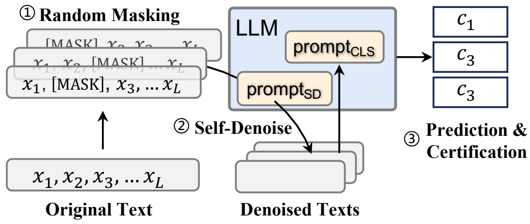

Figure 2: Prediction and certification process with our self-denoised smoothed classifier $g(\boldsymbol{x}')$.

The denoiser is expected to augment the base model to be more robust towards random masks on the inputs. Specifically, we consider two design choices for the denoiser, 1) instruct the LLM itself to recover the original input $\boldsymbol{x}$ given the masked input, and 2) directly remove the masks. To use the LLM as the denoiser, we use in-context learning to teach the LLM to fill in the masked positions so that the completed sentence is fluent and could preserve the original semantic. The prompt we used to instruct the LLM could be seen in Appendix A. On the other hand, we note that when mask rate m is high, such a filling-in-mask may fail to capture the original semantic due to limited remaining words and thus lead to undesired denoising results. Therefore, under such scenarios, we directly remove the *[Mask]* in the masked positions and use the remaining parts for the next step downstream prediction.

The prediction and certification pipeline of SELFDENOISE could be seen in Figure 2, where a Monte Carlo algorithm is used for estimating $g'(\boldsymbol{x})$. The input sentence is firstly perturbed with random masking multiple times. Different from the original randomized smoothing (with only step ① and ③ in the figure), we additionally add a denoising step, where the perturbed inputs are fed into the denoiser. The returned denoised results are fed into the LLM for downstream task prediction, and all predicting results are then integrated to get the final prediction following Equation 4. The certification process follows the original randomized smoothing[1] with our smoothed classifier $g'(\boldsymbol{x})$.

5 Experiment

5.1 Experiment Setup

Dataset and models We use the `SST-2` (Socher et al., 2013) and `Agnews` (Zhang et al., 2015) datasets in our experiments. We randomly divide the original testing set of `Agnews` into two parts equally as the new validation set and testing set and

[1]The detailed algorithm could be seen in Zeng et al. (2021a) Algorithm 2, Line 13-24.

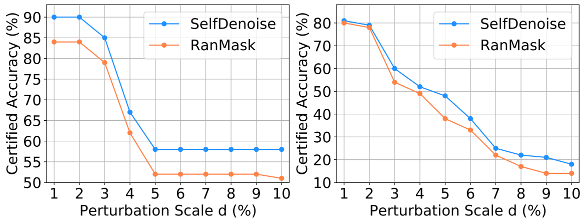

Figure 3: Certified accuracy under different perturbation scale d (%) on `SST-2` (*left*) and `Agnews` (*right*).

use the official split of the SST-2 dataset. We use the validation set for model selection and the testing set for evaluation. We consider Alpaca (Taori et al., 2023) as the base LLM to be robustified. We design prompts with in-context learning to instruct Alpaca to perform the corresponding tasks. See details in Appendix A.

Evaluation metrics Following Zeng et al. (2021a), we evaluate our methods together with all baselines with both certified accuracy and empirical robust accuracy. The certified accuracy is calculated for each perturbation scale d over 1% to 10%, *i.e.*, certified accuracy $= \frac{1}{n}\sum_{i=1}^{n}\mathbb{I}(r_i \geq d)$, where r_i is the certification radius for i-th input over in total n examples. The empirical robust accuracy is calculated using state-of-the-art adversarial attack methods DeepWordBug (Gao et al., 2018) and TextBugger (Li et al., 2018). Specifically, the attackers are adopted to attack the smoothed classifier with at most 10% words perturbation on each sentence, and the accuracy on the attacked adversarial examples are reported. We also report the clean accuracy on standard examples.

Baselines and implementation details We compare our method SELFDENOISE with the randomized smoothing-based certification method RANMASK for certified accuracy. Note that another similar certification method SAFER does not consider the same definition for certified robustness with us[2], so we only compare our method with them on empirical robust accuracy. The performances of the vanilla base model, termed ALPACA, are also reported. All baselines are evaluated with the same base model without any finetuning. The best hyper-parameters of each method are searched on the validation set. See details in Appendix A.

5.2 Experiment Results

Figure 3 shows the certification results of the proposed SELFDENOISE and baseline RANMASK on both `SST-2` and `Agnews`. We show that our

[2]See Section 2 for more explanations.

Dataset	Method	*Clean Acc. (%)*	*Empirical Robust Acc. (%)* DeepWordBug	*Empirical Robust Acc. (%)* 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.

method could effectively improve certified accuracy beyond RANMASK in both two 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`.

We further present the empirical robust accuracy (with at most 10% word perturbation) of the proposed SELFDENOISE and baselines in Table 3. Here are our key observations. First, our method achieves the best empirical robust accuracy in both two datasets under both attack methods. Specifically, SELFDENOISE improves the empirical robust accuracy by 13.2% in `SST-2` and 19.7% in `Agnews` compared with the second best method under DeepWordBug attack, with 2.8% and 24.5% improvements under TextBugger. Second, the proposed method demonstrates a better trade-off between robustness and standard accuracy. Our method achieves the best clean accuracy and empirical robust accuracy in `Agnews`. In `SST-2`, SELFDENOISE improves the empirical robust accuracy by 19.7% with only a 1.2% drop in clean accuracy, compared with the vanilla ALPACA.

6 Conclusion

In this paper, we proposed a randomized smoothing based LLM certification method, SELFDENOISE, which introduces a self-denoising framework to augment the original LLM by instructing the LLM to act as an additional denoiser, leading to larger certification radius of LLMs. The proposed could be used as a plug-in module for any LLM without any access to parameters, and no training is needed. Results from extensive experiments have demonstrated our superiority on both certified robustness and empirical robustness compared with existing works. For future works, we plan to replace our greedy self-denoising strategy with more plausible choices. We will investigate ways to find the optimal strategy by combining vast potential denoising transformations beyond mask filling.

7 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.

8 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 4. 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

Gregory Bonaert, Dimitar I Dimitrov, Maximilian Baader, and Martin Vechev. 2021. Fast and precise certification of transformers. In *PLDI*.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. 2022. Palm: Scaling language modeling with pathways.

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.

Timon Gehr, Matthew Mirman, Dana Drachsler-Cohen, Petar Tsankov, Swarat Chaudhuri, and Martin Vechev. 2018. Ai2: Safety and robustness certification of neural networks with abstract interpretation. In *IEEE symposium on security and privacy (SP)*.

Po-Sen Huang, Robert Stanforth, Johannes Welbl, Chris Dyer, Dani Yogatama, Sven Gowal, Krishnamurthy Dvijotham, and Pushmeet Kohli. 2019. Achieving verified robustness to symbol substitutions via interval bound propagation. *ArXiv*, abs/1909.01492.

Robin Jia, Aditi Raghunathan, Kerem Göksel, and Percy Liang. 2019. Certified robustness to adversarial word substitutions. In *Conference on Empirical Methods in Natural Language Processing*.

Alexander Levine and Soheil Feizi. 2019. Robustness certificates for sparse adversarial attacks by randomized ablation. In *AAAI Conference on Artificial Intelligence*.

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.

Matthew Mirman, Timon Gehr, and Martin Vechev. 2018. Differentiable abstract interpretation for provably robust neural networks. In *ICML*.

Erik Nijkamp, Bo Pang, Hiroaki Hayashi, Lifu Tu, Huan Wang, Yingbo Zhou, Silvio Savarese, and Caiming Xiong. 2023. Codegen: An open large language model for code with multi-turn program synthesis.

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

Hadi Salman, Greg Yang, Huan Zhang, Cho-Jui Hsieh, and Pengchuan Zhang. 2019. A convex relaxation barrier to tight robustness verification of neural networks. In *NeurIPS*.

Zhouxing Shi, Huan Zhang, Kai-Wei Chang, Minlie Huang, and Cho-Jui Hsieh. 2020. Robustness verification for transformers. In *ArXiv*, volume abs/2002.06622.

Gagandeep Singh, Timon Gehr, Markus Püschel, and Martin Vechev. 2019. An abstract domain for certifying neural networks. *PACMPL*.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew 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, Seattle, Washington, USA. Association for Computational Linguistics.

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, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

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

Shijie Wu, Ozan Irsoy, Steven Lu, Vadim Dabravolski, Mark Dredze, Sebastian Gehrmann, Prabhanjan Kambadur, David Rosenberg, and Gideon Mann. 2023. Bloomberggpt: A large language model for finance. *ArXiv*, abs/2303.17564.

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. 2021a. Certified robustness to text adversarial attacks by randomized [mask]. In *arXiv preprint arXiv:2105.03743*.

Jiehang Zeng, Xiaoqing Zheng, Jianhan Xu, Linyang Li, Liping Yuan, and Xuanjing Huang. 2021b. Certified robustness to text adversarial attacks by randomized [mask]. *Computational Linguistics*, 49:395–427.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Neurips*.

Yang Zhang, Shiyu Chang, Mo Yu, and Kaizhi Qian. 2019. An efficient and margin-approaching zero-confidence adversarial attack. *arXiv preprint arXiv:1910.00511*.

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*.

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.
```

A.2 Hyperparameter

We evaluate on 100 testing instances for certified accuracy in Figure 3 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 $p_c(\boldsymbol{x})$ with Monte Carlo in the certification process. In Figure 3, 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 2. 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 2: The best mask rate m (%) for each perturbation scale on SST-2 and Agnews for SELFDENOISE and RANMASK.