Certifiably-Robust Federated Adversarial Learning via Randomized Smoothing

Cheng Chen
Department of ECE
University of Utah
Salt Lake City, US
u0952128@utah.edu

Bhavya Kailkhura
LLNL National Lab
Livermore, US
kailkhura1@llnl.gov

Ryan Goldhahn
LLNL National Lab
Livermore, US
goldhahn1@llnl.gov

Yi Zhou
Department of ECE
University of Utah
Salt Lake City, US
yi.zhou@utah.edu

***Abstract*—Federated learning is an emerging data-private distributed learning framework, which, however, is vulnerable to adversarial attacks. Although several heuristic defenses are proposed to enhance the robustness of federated learning, they do not provide certifiable robustness guarantees. In this paper, we incorporate randomized smoothing techniques into federated adversarial training to enable data-private distributed learning with certifiable robustness to test-time adversarial perturbations. Through comprehensive experiments, we show that such an advanced federated adversarial learning framework can deliver models as robust as those trained by the centralized training. Further, this enables training provably-robust classifiers to ℓ_2-bounded adversarial perturbations in a distributed setup. We also show that the one-point gradient estimation-based training approach is $2-3\times$ faster than the popular stochastic estimator-based approach without any noticeable certified robustness differences.**

***Index Terms*—federated learning , certifiable robustness, randomized smoothing, adversarial training, gradient estimation**

I. Introduction

Federated learning is an emerging distributed learning framework that enables edge computing at a large scale [1]–[4], and has been successfully applied to various areas such as Internet of Things (IoT), autonomous driving, health care [2], etc. In particular, federated learning aims to exploit the distributed computation and heterogeneous data of a large number of edge devices to perform distributed learning while preserving full data privacy. The original federated learning framework proposed the federated averaging (FedAvg) algorithm [3]. In each learning round, a subset of edge devices are selected to download a global model from the cloud server, based on which the selected devices train their local models using local data for multiple stochastic gradient descent (SGD) iterations. Then, these devices upload the trained local models to the server, where the local models are aggregated and averaged to obtain an updated global model that will be used in the next learning round. Throughout the federated learning process, all data are kept privately on the local devices.

However, as modern federated learning often adopts over-parameterized models (e.g., deep neural networks) that have been proven to be vulnerable to adversarial perturbations to the test data [5]–[7], there is a rising concern about the adversarial robustness of the federated learning models used by massive number of edge devices. As an example, if a federated-trained model is vulnerable to adversarial examples, then its performance on edge devices solving safety-critical tasks can be significantly degraded in turn having serious consequences. To defend such adversarial attacks in federated learning, many studies propose to include standard adversarial training in the local training steps of federated learning [8]–[11]. However, these approaches may not be able to defend strong adversaries and do not have certifiable adversarial robustness guarantee. To address these issues, some studies proposed the randomized smoothing technique that can train certifiably robust models at scale [12]–[14].

Specifically, randomized smoothing procedure uses a smoothed version of the original classifier f and certifies the adversarial robustness of the new classifier. The smoothed classifier is defined as $g(x) = \arg\max_c \mathbb{P}_{\delta\sim\mathcal{N}(0,\sigma^2 I)}(f(x+\delta)=c)$, meaning the label of a data sample x corresponds to the class whose decision region $\{x' \in \mathbb{R}^d : f(x')=c\}$ has the largest measure under the distribution $\mathcal{N}(x,\sigma^2 I)$, where σ is used for smoothing. Suppose that while classifying a point $\mathcal{N}(x,\sigma^2 I)$, the original classifier f returns the top class c_A with probability $p_A = \mathbb{P}(f(x+\delta)=c_A)$, and the "runner-up" class c_B is returned with probability $p_B = \max_{c\neq c_A}\mathbb{P}(f(x+\delta)=c)$, then the prediction of the point x under the smoothed classifier g is guaranteed to be robust within the radius $r(g;\sigma) = \frac{\sigma}{2}(\Phi^{-1}(p_A) - \Phi^{-1}(p_B))$, where Φ^{-1} is the inverse CDF of the standard Normal distribution. In practice, Monte Carlo sampling is used to estimate a lower bound on p_A and an upper bound on p_B as its difficult to estimate the actual values for p_A and p_B. Since standard training of the base classifier does not achieve high robustness guarantees, [13] proposed to use Gaussian data augmentation-based training in which the base classifier is trained on Gaussian noise corruptions of the clean data. Such a smoothed model has been shown to outperform other existing certifiably robust models [13] and the randomized smoothing scheme is applicable to deep networks and large datasets. To further enhance certifiable robustness of deep models, the authors in [15] combined standard adversarial training approach with the randomized smoothing technique to obtain significantly improved certification guarantees. In particular,

978-1-7281-6251-5/20/$31.00 ©2021 IEEE

[15] demonstrated that such an adversarial training approach can substantially improve the robustness of smoothed models. However, these certifiably-robust training approaches are only applied to centralized learning setup, and similar provably-robust approaches in a federated learning setup is virtually non-existent. To bridge this gap, in this paper, we incorporate the randomized smoothing (with adversarial training) approach into the paradigm of federated learning to develop certifiably robust federated learning models.

A. Our Contributions

We apply the randomized smoothing (with adversarial training) approach to enable the certifiable robustness of federated learning to adversarial perturbations. Specifically, in the local training phase, each device applies adversarial training to train a robust smoothed local model to defend ℓ_2 adversarial attacks. These local models are further aggregated by the central server to obtain a robust global model. To the best of our knowledge, this is the first work in the direction of enabling certifiable robust federated learning.

We also conduct comprehensive deep learning experiments to validate the effectiveness of our proposed approach. Our experiments show that such an advanced federated adversarial learning framework can deliver models as robust as those trained by the centralized training. Further, this enables training provably-robust classifiers to ℓ_2-bounded adversarial perturbations in a distributed setup. We also show that one-point gradient estimation-based training approach is $2-3\times$ faster than the popular stochastic estimator-based approach without any noticeable certified robustness differences.

II. Adversarial Learning with Randomized Smoothing

Consider a standard soft classifier F_θ that is parameterized by θ and maps an input data $x \in \mathbb{R}^d$ to a probability mass of class labels $\mathcal{Y}$. Then, its corresponding smoothed soft classifier G_θ is defined as

$$G_\theta(x) := \mathbb{E}_{\delta\sim\mathcal{N}(0,\sigma^2 I)}[F_\theta(x+\delta)]. \tag{1}$$

Intuitively, the smoothed classifier G_θ perturbs the input sample with Gaussian noises and averages the predicted class distributions of all corrupted samples. In particular, the standard deviation σ of the Gaussian noise controls the level of certifiable robustness of the smoothed classifier.

To improve the performance, in [15], the authors proposed to leverage adversarial examples of the input data against the smoothed classifier G_θ (instead of F_θ). Specifically, [15] proposed the following adversarial training problem, where the training uses the adversarial data $\widehat{x}$ that is found within an ℓ_2 ball of the original data x by attacking the smoothed soft classifier G_θ.

$$\textbf{(SmoothAdv)}: \ \min_{\theta} \max_{\|\widehat{x}-x\|_2\leq\epsilon} J_\theta(\widehat{x}) := -\log[G_\theta(\widehat{x})]_y, \tag{2}$$

where $[G_\theta(\widehat{x})]_y$ denotes the y-th entry of the predicted classification probability mass. This approach is referred to as **SmoothAdv** and the objective function is highly stochastic and non-convex. To solve the above adversarial optimization problem, two approaches were proposed in [15]. For the first approach, the authors approximate the gradient of the above objective function using stochastic samples as follows

(Stochastic estimator)

$$\nabla_x J(\widehat{x}) \approx -\nabla_x \log\left(\frac{1}{m}\sum_{i=1}^{m}[F_\theta\left(\widehat{x}+\delta_i\right)]_y\right), \tag{3}$$

where $\delta_i, i=1,...,m$ are drawn i.i.d from $\mathcal{N}\left(0,\sigma^2 I\right)$. Then, standard projected gradient ascent is applied to find adversarial samples. While the above stochastic gradient estimator provides an accurate gradient estimation, it is computational expensive as for every sample x we need to perform back-propagation on a mini-batch of m corrupted samples.

To avoid performing back-propagation, [15] discussed another gradient-free [16] approach. Specifically, note that the adversarial optimization problem is equivalent to $\widehat{x} = \arg\min_{\|\widehat{x}-x\|_2\leq\epsilon}\left[G_\theta(\widehat{x})\right]_y$. In particular, the gradient of $[G(\widehat{x})]_y$ can be conveniently characterized using the following one-point gradient-free estimator using Stein's lemma.

(One-point estimator)

$$\nabla_x\left[G_\theta\left(\widehat{x}\right)\right]_y \approx \frac{1}{m}\sum_{i=1}^{m}\left[\frac{\delta_i}{\sigma^2}\cdot[F_\theta\left(\widehat{x}+\delta_i\right)]_y\right]. \tag{4}$$

The above estimator only involves function values that can be efficiently computed via forward-propagation. In particular, each gradient estimate $\frac{\delta_i}{\sigma^2}\cdot[F_\theta\left(\widehat{x}+\delta_i\right)]_y$ only needs to evaluate the function value at a single point $\widehat{x}+\delta_i$. Compared to the gradient-based stochastic estimator, this one-point estimator is computation lighter but induces a higher estimation variance. In [15], the performance of the one-point estimator was not evaluated for **SmoothAdv**, and its comparison with the stochastic estimator was not comprehensive.

III. Federated Adversarial Learning

In this section, we incorporate the **SmoothAdv** method into the federated learning framework. Our proposed algorithm is referred to as **Fed-SmoothAdv** and is shown in Algorithm 1.

To elaborate, first note that the hierarchical structure of **Fed-SmoothAdv** is the same as that of standard federated learning, i.e., a subset of edge devices is sampled in every round to perform local training, and then their local models are aggregated by the cloud server through a standard model-averaging scheme (see the Central-server pseudo codes). However, in our federated adversarial learning, each client uses **SmoothAdv** to perform local adversarial training instead of the local SGD training in standard federated learning (see the LocalTrain and SmoothAdv pseudo codes). In particular, the local **SmoothAdv** adversarial training will generate strong adversarial samples by attacking the smoothed local model, and use these strong adversarial samples to significantly enhance the adversarial-robustness of the local model. Finally, the robust local models are aggregated in the central server to produce a globally robust federated model.

Algorithm 1: Federated Adversarial Learning (**Fed-SmoothAdv**)

Central-server executes: *# Run on the central server*

for learning round $t = 1, 2, \ldots$ **do**
- Sample a subset S_t of clients
- **for** each client $k \in S_t$ in parallel **do**
  - $\theta_{t+1}^k \leftarrow$ **LocalTrain** (k, θ_t)
  - Send θ_{t+1}^k to the server
- Server aggregates $\theta_{t+1} \leftarrow \sum_{k \in S_t} \frac{n_k}{n} \theta_{t+1}^k$

LocalTrain (k, θ) : *# Local training of client k*

for local iteration $i = 1, 2, ..., E$ **do**
- Sample a minibatch of data b
- $\theta \leftarrow$ **SmoothAdv** (θ, b) *# Use one of the two gradient estimators*

SmoothAdv (θ, b) : *# Adversarial training with randomized smoothing*

Data samples $\left(x^{(1)}, y^{(1)}\right), \left(x^{(2)}, y^{(2)}\right), \ldots, \left(x^{(b)}, y^{(b)}\right)$

Generate noises $\{\delta_i^{(j)}\}_{i=1}^m \sim \mathcal{N}\left(0, \sigma^2 I\right)$ for any $x^{(j)}, j = 1, ..., b$

$L \leftarrow []$ *# List of adversarial examples*

for $1 \leq j \leq b$ **do**
- Generate adversarial sample $\widehat{x}^{(j)}$ for $x^{(j)}$ by attacking the smoothed classifier using one of the gradient estimators in eqs.(3,4) and noises $\{\delta_i^{(j)}\}_{i=1}^m$.
- Append $\{(\widehat{x}^{(j)} + \delta_1^{(j)}, y^{(j)}), \ldots, (\widehat{x}^{(j)} + \delta_m^{(j)}, y^{(j)})\}$ to list L.

Train model θ using adversarial samples in L for multiple SGD steps.

IV. Experiments

A. Experiment Setup and hyperparameters

We compare the standard certified robustness of **Fed-SmoothAdv** with the baseline method **SmoothAdv** in training an AlexNet [17] on CIFAR-10 [18]. Here, certified robustness is defined as the fraction of the test samples that are correctly classified (without abstaining) by G_θ and are certified within an ℓ_2 radius of r. We set the smoothing parameter $\sigma = \{0.12, 0.25, 0.5\}$ and the perturbation bound $\epsilon = \{64, 128, 256\}$, and use the same σ for certification as that used in the training. For both methods, we apply both the stochastic estimator and the one-point estimator. Moreover, we test **Fed-SmoothAdv** under different levels of device data heterogeneity γ_{device} (the higher the more heterogeneous).

For **Fed-SmoothAdv**, we simulate 1000 edge devices and only 10% of them are sampled in each learning round. Each device holds 500 data samples. To control the data heterogeneity of each device, we define a data heterogeneity ratio γ_{device} in $(0, 1)$. Specifically, we randomly assign one class label as the major class of each device. Then, for each device, γ_{device} portion of samples are sampled from the major class, and the rest $(1 - \gamma_{\text{device}})$ portion of samples are drawn from the remaining classes uniformly at random. In the experiments , we set $\gamma_{\text{device}} = 0.1, 0.5$ that correspond to homogeneous data and heterogeneous data, respectively.

In the experiments, we set the number of Gaussian noise samples to be $m = 2$, and use 2 projected gradient descent steps for generating the adversarial samples. We set the inner-learning-rate for generating adversarial samples to 0.01, and the outer-learning-rate for updating the model parameters to 0.01. We set batch-size to 30 for each activated device of **Fed-SmoothAdv** and 60 for **SmoothAdv**. Moreover, each activated device of **Fed-SmoothAdv** uses 20 batches of data in the local training of a learning round, and this is equivalent to 1000 batches of data used by the centralized **SmoothAdv**. The total number of learning rounds is 150. In the certification phase, we set $\alpha = 0.001$, which means that there is at most 0.1% chance that the certification falsely certifies a non-robust input.

B. Comparison of Certified Accuracy

In Figure 1, we plot the certified accuracy of both **SmoothAdv** and **Fed-SmoothAdv** (with heterogeneity $\gamma_{\text{device}} = 0.1, 0.5$) with $\sigma = 0.25, \epsilon = 128$. It can be seen that the certified accuracy of **Fed-SmoothAdv** is slightly lower than that of **SmoothAdv**, but is reasonably close. Also, the data heterogeneity γ_{device} does not affect the certified accuracy of **Fed-SmoothAdv**, which implies that **SmoothAdv** can be effectively applied to enhance the adversarial robustness of heterogeneous federated learning. Moreover, we note that while the performance of the one-point estimator is almost the same as that of the stochastic estimator, the training time is significantly reduced by 2-3 times due to avoidance of backpropagation. In Figure 2, we plot the certified accuracy results under $\sigma = 0.5$ and $\epsilon = 128$. One can observe a similar comparison between the two methods as that in Figure 1. In particular, with a larger σ, the certified accuracy is lower but spans over a wider range of ℓ_2 radius. The results corresponding to all other choices of σ, ϵ can be found in Appendix A, where one can make very similar observations and conclusions.

C. Ablation Study

In this subsection, we further explore the certified accuracy of **Fed-SmoothAdv** under the following ablation settings: (1) Without adv & smooth, i.e., standard training of original classifier; (2) With adv only, i.e., adversarial training of original classifier (using stochastic estimator); and (3) With adv & smooth, i.e., adversarial training of smoothed classifier (using stochastic estimator). All the trained models in a given figure use the same σ value to be certified so that we can evaluate their accuracy within the same radius.

Figure 3 plots the results under $\sigma = 0.25, \epsilon = 128$ with homogeneous data ($\gamma_{\text{device}} = 0.1$) and heterogeneous data ($\gamma_{\text{device}} = 0.5$), respectively. First, the certified accuracy of **Fed-SmoothAdv** is much higher than that of standard and adversarial training of original (non-smoothed) classifier, which indicates that randomized smoothing is very helpful

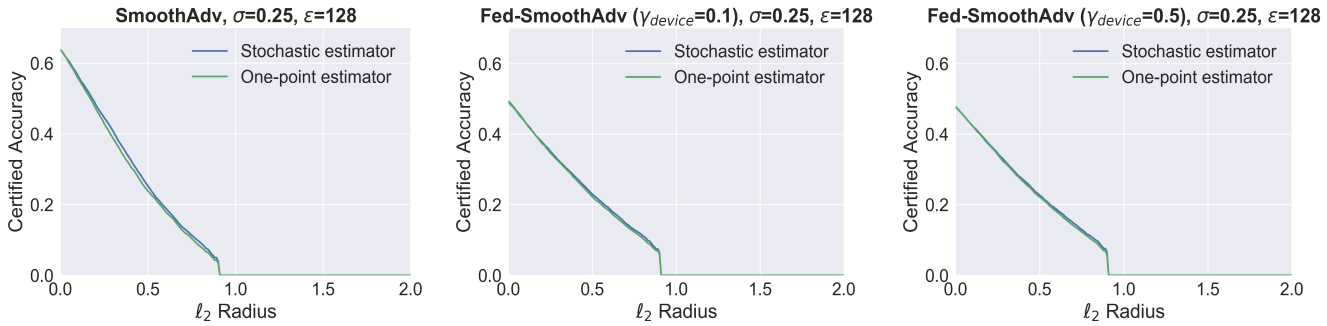

Fig. 1: Certified accuracy of **SmoothAdv** and **Fed-SmoothAdv** with $\sigma = 0.25$ and $\epsilon = 128$.

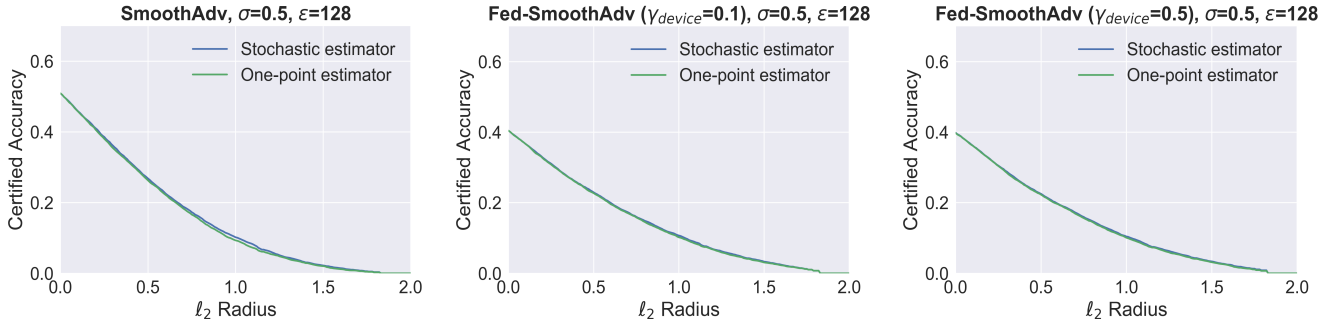

Fig. 2: Certified accuracy of **SmoothAdv** and **Fed-SmoothAdv** with $\sigma = 0.5$ and $\epsilon = 128$.

to improve the performance of **Fed-SmoothAdv**. Second, adversarial training does not achieve significantly higher certified accuracy than standard training, which again indicates that importance of having a smoothed classifier. Moreover, Figure 4 plots the results under $\sigma = 0.5, \epsilon = 128$ with homogeneous data ($\gamma_{\text{device}} = 0.1$) and heterogeneous data ($\gamma_{\text{device}} = 0.5$), respectively, where one can make similar observations and conclusions. In particular, by comparing Figure 3 with Figure 4, one can see that as σ increases, the certified accuracy is lower but spans over a wider range of ℓ_2 radius. In summary, adversarial training is a little helpful to improve **Fed-SmoothAdv**'s certified accuracy, while randomized smoothing is much more helpful to do so. This demonstrates the necessity to add randomized smoothing to certifiably-robust federated adversarial learning, which is what we proposed in this paper.

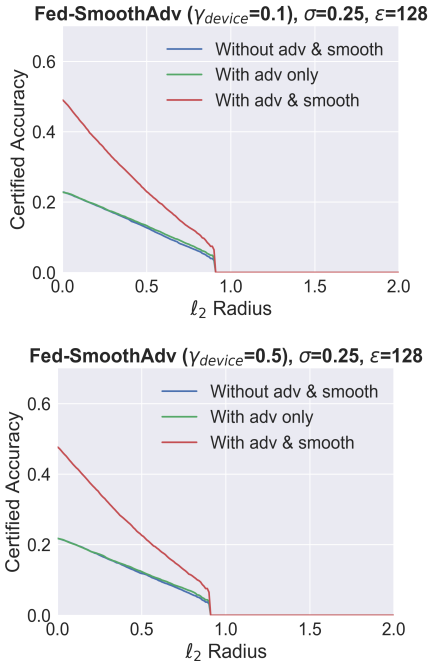

Fig. 3: Ablation study of **Fed-SmoothAdv** with $\sigma = 0.25$ and $\epsilon = 128$.

V. Conclusion

In this paper, we incorporated the randomized smoothing techniques into the federated adversarial learning framework to enable certifiable robustness to test-time adversarial perturbations. We demonstrated through extensive experiments that our adversarially smooth federated learning models could successfully achieve similar certified robustness as the centralized models. Meanwhile, we empirically proved that the device data heterogeneity and type of gradient estimator did not affect the performance much. The attempt in this paper is crucial for the applications of federated learning because of the adversarial attacks on its user's devices and the resulting strong demand for user's data privacy and security in the real world. In the future, we will apply randomized smoothing to more complex federated learning frameworks [4] and theoretically study its performance.

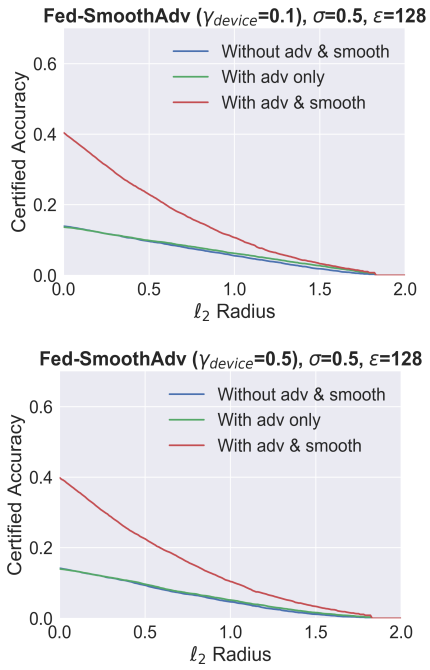

Fig. 4: Ablation study of **Fed-SmoothAdv** with $\sigma = 0.5$ and $\epsilon = 128$.

ACKNOWLEDGMENT

This work was performed under the auspices of the U.S. Department of Energy by the Lawrence Livermore National Laboratory under Contract No. DE-AC52-07NA27344, Lawrence Livermore National Security, LLC. This document was prepared as an account of the work sponsored by an agency of the United States Government. Neither the United States Government nor Lawrence Livermore National Security, LLC, nor any of their employees makes any warranty, expressed or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or Lawrence Livermore National Security, LLC. The views and opinions of the authors expressed herein do not necessarily state or reflect those of the United States Government or Lawrence Livermore National Security, LLC, and shall not be used for advertising or product endorsement purposes. This work was supported by LLNL Laboratory Directed Research and Development project 20-SI-005 and released with LLNL tracking number LLNL-CONF-820514.

REFERENCES

[1] J. Konečnỳ, H. B. McMahan, D. Ramage, and P. Richtárik, "Federated optimization: Distributed machine learning for on-device intelligence," *ArXiv:1610.02527*, 2016.

[2] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, "Federated learning: Challenges, methods, and future directions," *IEEE Signal Processing Magazine*, vol. 37, no. 3, pp. 50–60, 2020.

[3] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proc. International Conference on Artificial Intelligence and Statistics (AISTATS)*, vol. 54, 20–22 Apr 2017, pp. 1273–1282.

[4] C. Chen, Z. Chen, Y. Zhou, and B. Kailkhura, "Fedcluster: Boosting the convergence of federated learning via cluster-cycling," *arXiv preprint arXiv:2009.10748*, 2020.

[5] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," in *International Conference on Learning Representations*, 2014.

[6] I. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," in *International Conference on Learning Representations*, 2015.

[7] S. Bulusu, B. Kailkhura, B. Li, P. K. Varshney, and D. Song, "Anomalous example detection in deep learning: A survey," *IEEE Access*, vol. 8, pp. 132 330–132 347, 2020.

[8] Y. Zhou, J. Wu, and J. He, "Adversarially robust federated learning for neural networks," 2021. [Online]. Available: https://openreview.net/forum?id=5xaInvrGWp

[9] G. Zizzo, A. Rawat, M. Sinn, and B. Buesser, "Fat: Federated adversarial training," *arXiv:2012.01791*, 2020.

[10] R. Kerkouche, G. Ács, and C. Castelluccia, "Federated learning in adversarial settings," *arXiv:2010.07808*, 2020.

[11] A. N. Bhagoji, S. Chakraborty, P. Mittal, and S. Calo, "Analyzing federated learning through an adversarial lens," in *Proc. International Conference on Machine Learning*, vol. 97, 2019, pp. 634–643.

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

[13] J. Cohen, E. Rosenfeld, and Z. Kolter, "Certified adversarial robustness via randomized smoothing," in *Proc. International Conference on Machine Learning*, vol. 97, 09–15 Jun 2019, pp. 1310–1320.

[14] L. Li, M. Weber, X. Xu, L. Rimanic, T. Xie, C. Zhang, and B. Li, "Provable robust learning based on transformation-specific smoothing," *arXiv preprint arXiv:2002.12398*, 2020.

[15] H. Salman, J. Li, I. Razenshteyn, P. Zhang, H. Zhang, S. Bubeck, and G. Yang, "Provably robust deep learning via adversarially trained smoothed classifiers," in *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[16] S. Liu, P.-Y. Chen, B. Kailkhura, G. Zhang, A. O. Hero III, and P. K. Varshney, "A primer on zeroth-order optimization in signal processing and machine learning: Principals, recent advances, and applications," *IEEE Signal Processing Magazine*, vol. 37, no. 5, pp. 43–54, 2020.

[17] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems*, vol. 25, 2012.

[18] A. Krizhevsky, "Learning multiple layers of features from tiny images," Tech. Rep., 2009.

APPENDIX A
ADDITIONAL EXPERIMENTAL RESULTS

In this section, we present the certified accuracy results of both **SmoothAdv** and **Fed-SmoothAdv** (with heterogeneity $\gamma_{\text{device}} = 0.1, 0.5$) under some other choices of σ and ϵ. From Figure 5-Figure 11, we can observe the same comparison and make the same conclusions as those in Section IV.

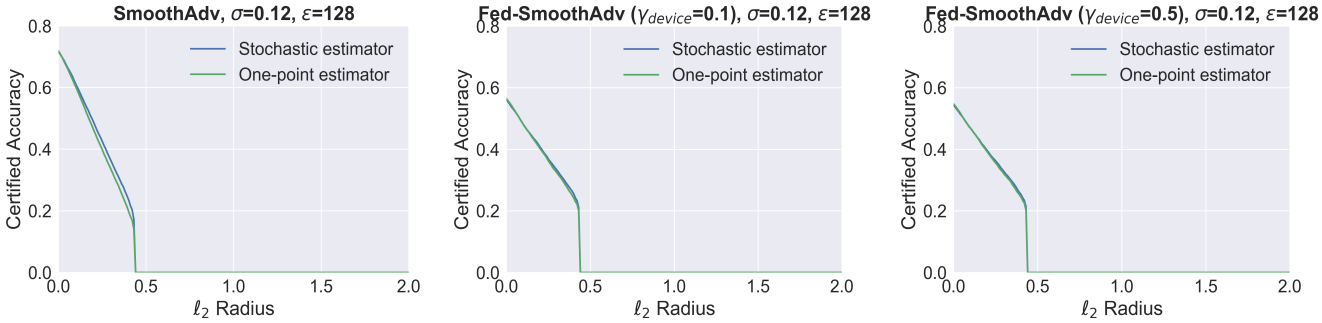

Fig. 5: Certified accuracy of **SmoothAdv** and **Fed-SmoothAdv** with $\sigma = 0.12$ and $\epsilon = 128$.

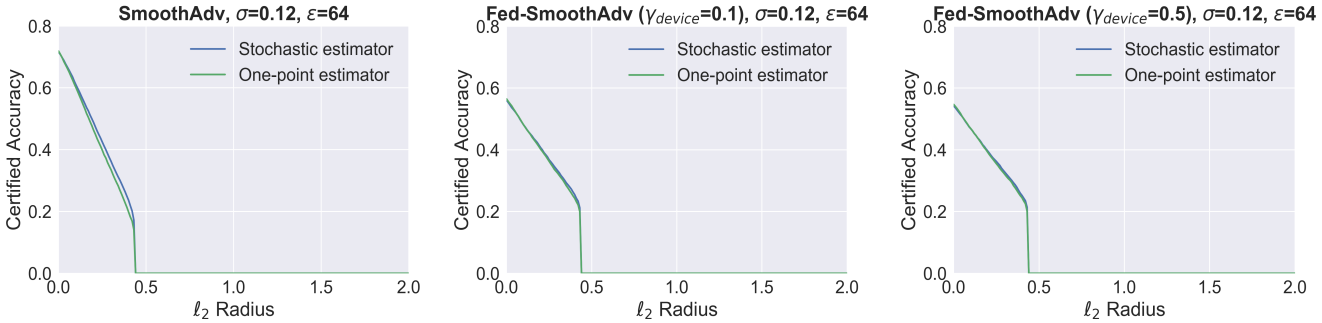

Fig. 6: Certified accuracy of **SmoothAdv** and **Fed-SmoothAdv** with $\sigma = 0.12$ and $\epsilon = 64$.

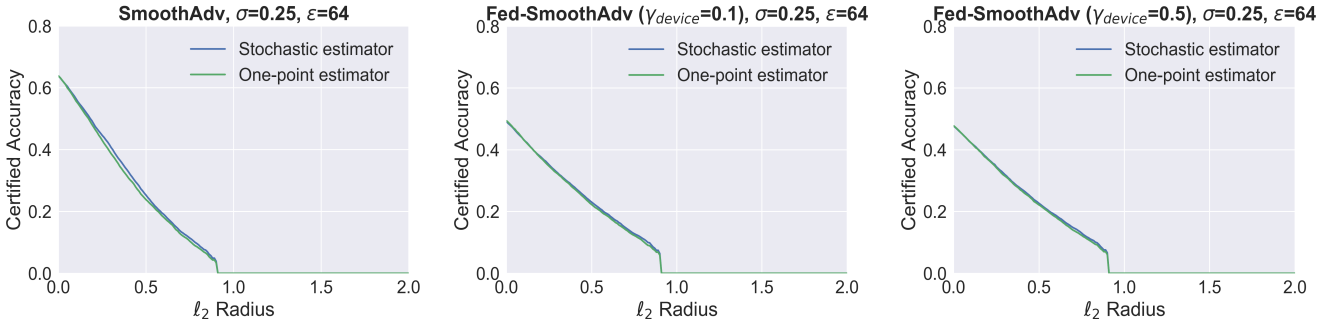

Fig. 7: Certified accuracy of **SmoothAdv** and **Fed-SmoothAdv** with $\sigma = 0.25$ and $\epsilon = 64$.

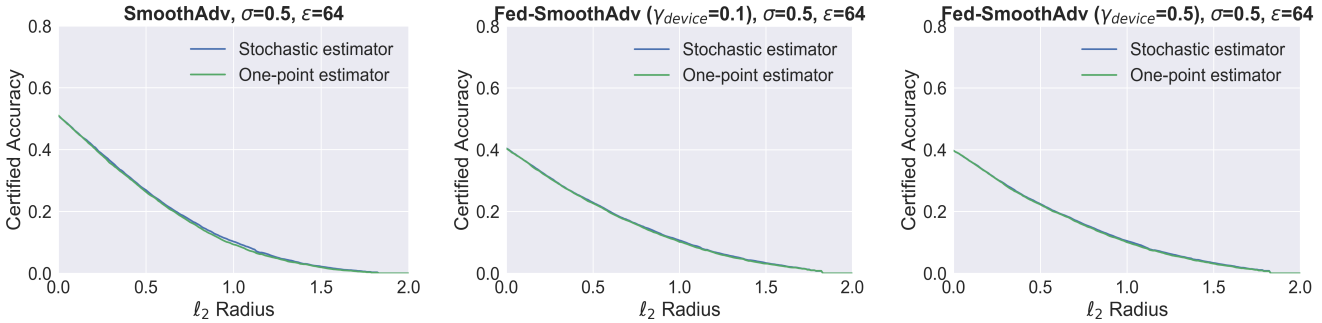

Fig. 8: Certified accuracy of **SmoothAdv** and **Fed-SmoothAdv** with $\sigma = 0.5$ and $\epsilon = 64$.

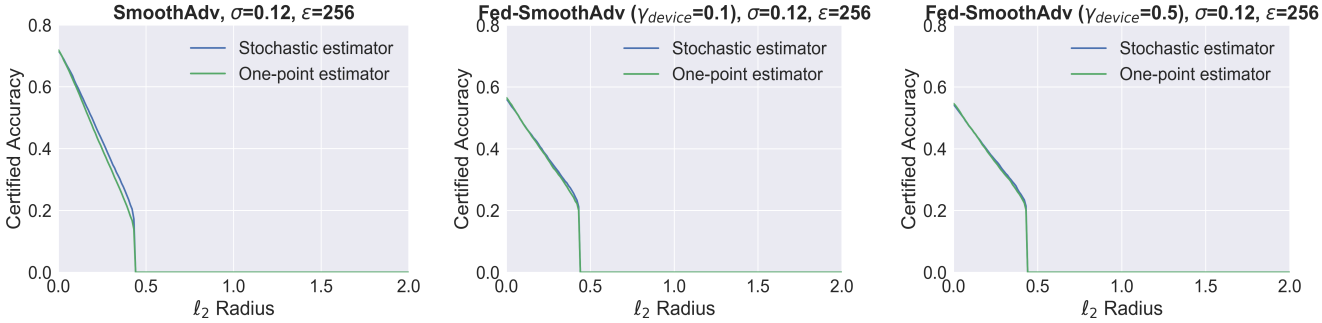

Fig. 9: Certified accuracy of **SmoothAdv** and **Fed-SmoothAdv** with $\sigma = 0.12$ and $\epsilon = 256$.

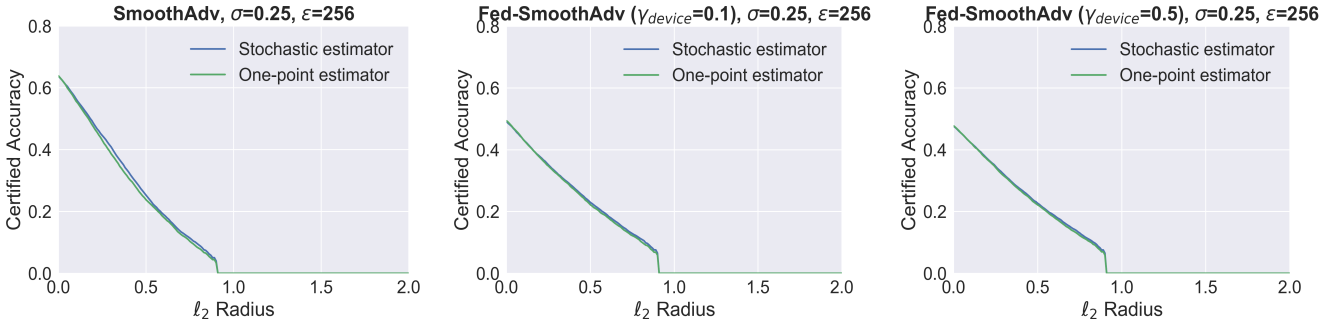

Fig. 10: Certified accuracy of **SmoothAdv** and **Fed-SmoothAdv** with $\sigma = 0.25$ and $\epsilon = 256$.

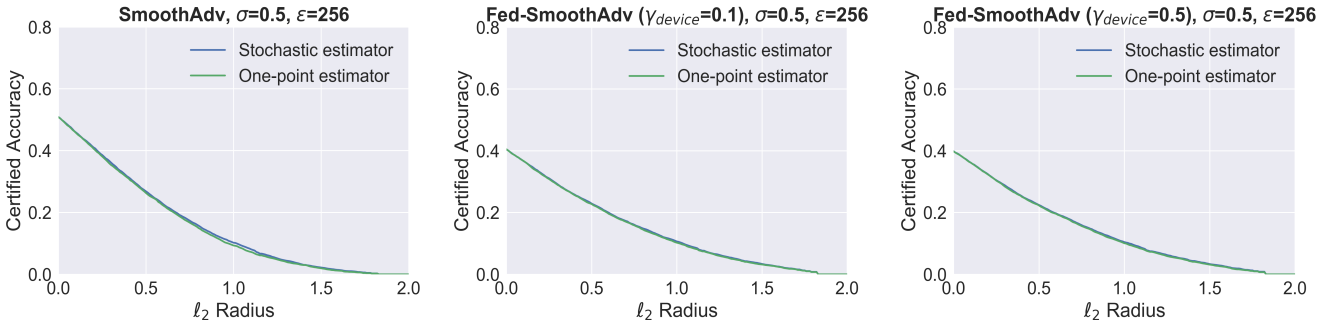

Fig. 11: Certified accuracy of **SmoothAdv** and **Fed-SmoothAdv** with $\sigma = 0.5$ and $\epsilon = 256$.