Stealthy Poisoning Attack on Certified Robustness

Akshay Mehra[1], **Bhavya Kailkhura**[2], **Pin-Yu Chen**[3], **Jihun Hamm**[1]

[1]Tulane University, [2]Lawrence Livermore National Laboratory, [3]IBM Research
{amehra,jhamm3}@tulane.edu,kailkhura1@llnl.gov,pin-yu.chen@ibm.com

Abstract

Certifiably robust classifiers have a constant prediction around a neighborhood of a point, which makes them resilient to test-time attacks with a guarantee. In this work, we present a previously unrecognized threat to robust machine learning models. Specifically, we propose a data poisoning attack to degrade the robustness guarantees of certifiably robust classifiers. Unlike other data poisoning attacks that reduce the accuracy of the poisoned models on a set of target points, our attack can reduce the average certified radius of an entire target class in the dataset while ensuring high accuracy of the classifiers on clean data. Clean label poisoning points with imperceptible distortion and high accuracy of the poisoned models make our attack hard to detect. Moreover, the attack is effective even when the victim trains the models from scratch and uses Gaussian data augmentation. By poisoning MNIST and CIFAR10 datasets and training deep neural networks on them, we show the effectiveness of our attack in degrading the certified robustness guarantees obtained using randomized smoothing. Our results highlight the importance of data quality in achieving high certified robustness guarantees.

1 Introduction

Data poisoning ([3, 15, 28, 29, 34]) is a training-time attack where the attacker is assumed to have access to the data on which the victim will train the model. Since modern machine learning methods need large amounts of data, data poisoning becomes easy as an attacker can place the poisoned data online and wait for it to be scraped by victims looking to increase the size for their dataset. Another easy target for poisoning is data collection by crowd sourcing where malicious users can corrupt the data they contribute. In most cases an attacker can modify only certain parts of the training data. In this work, we assume the attacker wants to affect the performance of the victim's models on a target class and modifies the points of the class (without affecting the labels) by adding imperceptible perturbations. Several works have shown the effectiveness of using data poisoning in ([23, 28, 14, 17, 34, 7, 16, 30]) altering the training data to hurt the accuracy of the model trained on poisoned data compared to the accuracy achievable by the model trained on clean data.

In this work, we propose a new data poisoning attack which can reduce the certified robustness guarantees of models trained on poisoned data. Measuring the certified robustness of machine learning models has become important after several heuristic defenses claiming to provide robustness to adversarial attacks were broken by stronger adversaries ([5, 1, 31, 4]). However, many certification methods ([25, 12, 13, 32]) do not scale to deep neural networks or large datasets, due to their high complexity. Recently randomized smoothing (RS) based certification methods ([18, 19, 8]) have become popular due to their scalability to deep neural networks and high dimensional datasets. Thus, in this work we use data poisoning to reduce the certified robustness guarantees provided by RS. Unlike previous poisoning attacks, our attack affects the certified radius of all the points in the target class and maintains high accuracy of the poisoned models on the clean data. Moreover, the attack is effective when the victim trains the model scratch and uses Gaussian data augmentation based training. High accuracy of the poisoned model coupled with poisoning data having clean labels makes the attack stealthy. We formulate our attack as a constrained bilevel problem and theoretically analyze its solution for the case when the victim uses linear classifiers. Our theoretical analysis and empirical results suggest that the decision boundary of the smoothed classifiers (used for RS) learnt on the poisoned data is significantly different from the one learnt using clean data thus causing a reduction

NeurIPS 2020 Workshop on Dataset Curation and Security

in certified radius. We show the effectiveness of our attack on real world problems by poisoning MNIST and CIFAR10 datasets and training state-of-the-art deep neural networks on these poisoned datasets and certifying their robustness using RS ([8]). To the best of our knowledge, our attack is the first clean label poisoning attack which significantly reduces the certified robustness guarantees of the models trained on the poisoned dataset. This highlights the importance of training-data quality and curation for obtaining meaningful certified robustness guarantees to test time attacks, a factor not considered by current certification methods.

2 Background

Randomized Smoothing: RS ([8]) uses a smoothed version of the original classifier $f : \mathbb{R}^d \to \mathcal{Y}$ and certifies the robustness of the new classifier. The label of a point x under the smoothed classifier $g(x) = \arg\max_c \mathbb{P}_{\eta\sim\mathcal{N}(0,\sigma^2 I)}(f(x+\eta) = c)$, is the class whose decision region $\{x' \in \mathbb{R}^d : f(x') = c\}$ has the largest measure under the distribution $\mathcal{N}(x, \sigma^2 I)$, with σ used for smoothing. Suppose the base classifier f while classifying $\mathcal{N}(x, \sigma^2 I)$, returns the class c_A with probability $p_A = \mathbb{P}(f(x + \eta) = c_A)$, and returns the "runner-up" class c_B with probability $p_B = \max_{c\neq c_A} \mathbb{P}(f(x + \eta) = c)$, then the prediction of the smoothed classifier g is robust around x within the ℓ_2 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.

Bilevel Optimization: A bilevel optimization problem is of the form $\min_{u\in\mathcal{U}} \xi(u, v^*)$ s.t. $v^* = \arg\min_{v\in\mathcal{V}(u)} \zeta(u, v)$, where the upper-level problem is a minimization problem with v constrained to be the optimal solution to the lower-level problem (see [2]). Although general bilevel problems are difficult to solve, under some simplifying assumptions their solution can be obtained using gradient based methods. Several methods for solving bilevel problems in machine learning have been proposed previously ([9, 24, 10, 20, 27, 22]). We review these in Appendix B. Our attack is formulated as a constrained bilevel optimization problem and we use the method based of approximating the hypergradient by approximately solving a linear system (ApproxGrad Alg. 1) in this work.

3 Stealthy data poisoning attack for reducing certified radius

3.1 Attack Formulation

Here we present our attack on the certified robustness guarantees provided by RS using data poisoning. Let $(X^{\text{clean}}, Y^{\text{clean}})$ be the clean, unalterable portion of the training set. Let $u = \{u_1, ..., u_n\}$ denote the attacker's poisoning data which is added to the clean data: $X^{\text{clean}} \bigcup u$. For clean-label attack, we require that each poison example u_i has a limited amount of perturbation $\|u_i - X_i^{\text{base}}\| \leq \epsilon$ from the base data X_i^{base} and has the same label Y_i^{base}, for $i = 1, ..., n$. We use ℓ_∞-norm here but other norms can be used too. The goal of the attacker is to find u such that when the defender uses $X^{\text{clean}} \bigcup u$ to train the classifier f, the corresponding smooth classifier g has a small average certified radius on a clean dataset $(X^{\text{val}}, Y^{\text{val}})$. Additionally, to make the attack stealthier we require that the accuracy of f on clean validation data remains unaffected. Our attack is therefore the solution to the following bilevel optimization problem:

$$\begin{aligned} \min_{u} \;& \tilde{R}(\tilde{g}_{\theta^*}; X^{\text{val}}, Y^{\text{val}}, \sigma) + \lambda L^{\text{val}}(f_{\theta^*}; X^{\text{val}}, Y^{\text{val}}) \\ & \text{s.t. } \|u_i - X_i^{\text{base}}\|_\infty \leq \epsilon, \;\; i = 1, ..., n, \;\; \text{and} \\ & \quad \theta^* = \arg\min_{\theta} \; L^{\text{poison}}(f_\theta; X^{\text{clean}} \bigcup u, Y^{\text{clean}} \bigcup Y^{\text{base}}, \sigma). \end{aligned} \tag{1}$$

The lower-level solution θ^* is the best classifier found by the defender on the poisoned data using Gaussian augmentation-based training ($L^{\text{poison}} = \frac{1}{n^{\text{poison}}} \sum_{(x_i,y_i)\in(X^{\text{poison}},Y^{\text{poison}})} l(x_i + \eta, y_i)$), where $\eta \sim \mathcal{N}(0, \sigma^2 I)$). Note that using Gaussian data augmentation is favorable to the defender and we include it to make our attack effective even against such defenders. In the upper-level cost, the first term is the average certified radius and the second term is the loss of the base classifier f on the validation set: $L^{\text{val}} = \frac{1}{n^{\text{val}}} \sum_{(x_i,y_i)\in(X^{\text{val}},Y^{\text{val}})} l(x_i, y_i)$. Since the certified radius of the "hard" smooth classifier g is non-differentiable, we use the "soft" smooth classifier $\tilde{g}$ as an approximation ([26, 33]). Let $z_\theta : X \to \mathbb{P}(K)$ be a classifier whose last layer is softmax and $\sigma > 0$, then soft smoothed classifier $\tilde{g}_\theta$ of z_θ is defined as $\tilde{g}_\theta(x) = \arg\max_{c\in Y} \mathbb{E}_{\eta\sim\mathcal{N}(0,\sigma^2 I)}[z_c^\theta(x + \eta)]$. It was shown in [33] that if the ground truth of an input x is y and $\tilde{g}_\theta$ classifies x correctly, then $\tilde{g}_\theta$ is provably robust at x, with the certified radius given by $\tilde{r}(\tilde{g}_\theta; x, y, \sigma) = \frac{\sigma}{2}[\Phi^{-1}(\mathbb{E}_\eta[z_\theta^y(x + \eta)]) - \Phi^{-1}(\max_{y'\neq y} \mathbb{E}_\eta[z_\theta^{y'}(x + \eta)])]$. Letting $\tilde{r}(\tilde{g}_\theta; x, y, \sigma) = 0$ if x is misclassified, the average certified radius is $\tilde{R}(\tilde{g}_\theta; X, Y, \sigma) = \frac{1}{|X|} \sum_{(x,y)\in X} \tilde{r}(\tilde{g}_\theta; x, y, \sigma)$.

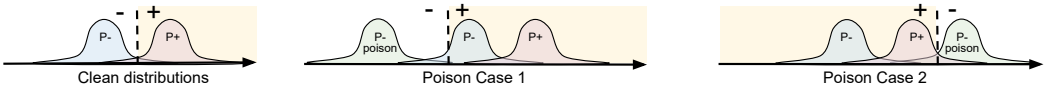


Figure 1: Analytical solutions of the bilevel problem (2) with linear classifiers. The poison distribution (P^-_{poison}) can change the decision boundary (broken line) and reduce the average certified radius of the clean distribution (P^-) in two ways (Cases 1 and 2). Perturbation is exaggerated for illustration.

3.2 Attack generation and evaluation

In this work, we focus on creating a poisoned set to reduce the certified adversarial robustness guarantees of all the points in a target class. We initialize the poisoning points from the clean points of the target class (i.e., base data) and optimize the perturbation to be added to each point by solving the bilevel problem in Eq. (1). We use a small value of ϵ to ensure the perturbations added are imperceptible and the poison points have clean labels when inspected visually (See Fig. 2 in Appendix). We solve the bilevel optimization using the ApproxGrad algorithm described in Alg. 1 in Appendix B. A full attack algorithm is shown Alg. 2 in Appendix C. We evaluate the effect of poisoning, by training models from scratch with Gaussian data augmentation on the poisoned set and report the average certified radius and certified accuracy on test points from the target class.

3.3 Analysis of a linear classifier

To gain a deeper insight into the effect of poisoning, we analyze the analytical solution of our bilevel problem for the case of linear classifiers. Suppose we have a one-dimensional balanced two-class problem and the attacker's goal is to poison the distribution of the *negative* class P^- so that the average certified radius $(\tilde{R})$ of the poisoned model for the test points of the *negative* class is reduced. Let the maximum permissible perturbation to the points of the class P^- be bounded by $|u_i - x_i^-| < \epsilon, \;\; i = 1, ..., n$. We do not assume any specific distributions P_+ and P_- here, but only that $\sum_i x_i^- < \sum_i x_i^+$ is true. Here x_i^+ and x_i^- refer to the training points of the positive and the negative class, respectively. A linear classifier in one-dimension is either $f(x) = 1$ iff $x \geq t$ or $f(x) = 1$ iff $x \leq t$ parameterized by the threshold t. For linear classifiers, it is known ([8]) that for any value of σ used for smoothing, the smoothed classifier g is the same as the unsmoothed classifier f and the certified radius for a point is the distance to the decision boundary. To make the problem analytically tractable, we use only the radius term without the accuracy term in the upper-level cost. Similarly, we use the squared-loss for the linear classifier at the lower level for tractability, i.e., $f(x) = wx + b$ and $l(x, y; w, b) = (wx + b - y)^2$. The bilevel formulation for the poisoning problem is

$$\begin{aligned} \min_{u} \;\; & \mathbb{E}_{P_-}[\max(\text{sign}(w)(-b/w - x), 0)] \\ & \text{s.t.} \;\; -\epsilon \leq u_i - x_i^- \leq \epsilon, \;\; \text{for i} = 1, ..., \text{n} \\ & w, b = \arg\min_{w,b} \; \frac{1}{2n}\Big[\sum_{i=1}^{n}(wx_i^+ + b - 1)^2 + \sum_{i=1}^{n}(wu_i + b + 1)^2\Big]. \end{aligned} \tag{2}$$

Theorem 1. *If the perturbation is large enough, i.e., $\epsilon \geq \frac{\sum_i x_i^+ - \sum_i x_i^-}{n}$ then there are two locally optimal solutions to (2) which are $u_i = x_i - \epsilon$ (Case 1) and $u_i = x_i + \epsilon$ (Case 2) for $i = 1, ..., n$. Otherwise, there is a unique globally optimal solution which is $u_i = x_i - \epsilon$ (Case 1) for $i = 1, ..., n$.*

The theorem states that optimal poisoning is achieved by shifting all training points of the negative class either towards left or right by the maximum amount ϵ (See Fig. 1 and Appendix D.2). In the linear case, reduction in the radius due to the change in the decision boundary also incurs the loss of accuracy in the target class (more so in Case 2 than Case 1). For the nonlinear case, direct analysis is intractable, but we empirically observe that poisoning on neural networks also moves the decision boundary closer to the target class as measured by the average distance of the test points of the target class to the decision boundary of the smoothed classifier. Furthermore, for nonlinear classifiers, it is feasible to reduce the radius without degrading the accuracy too much (See Tables 1 and 2 in Sec. 4).

4 Experiments

Here we present the results of poisoning on convolutional neural network models trained on the poisoned dataset (with Gaussian data augmentation) generated by our attack. The results are averaged over models trained starting from five random initializations. We report average certified radius

(ACR) using the certified radius obtained from RS ([8]) for correctly classified points and zero for misclassified and abstained points. The approximate certified accuracy (ACA) is the fraction of points correctly classified by the smoothed classifier. We use the same value of σ for smoothing during attack, retraining and evaluation. We also report the empirical robustness of the original (unsmoothed) and the smoothed classifiers using mean ℓ_2-distortion for successful attack using CW attack([6]) on the base classifier and PGD attack([26]) on the smoothed classifier. We use 200 and 100 randomly sampled points of the target class from the test sets to report certified and empirical robustness for MNIST and CIFAR10, respectively. We compare our results to watermarking ([28]) which has been used previously for clean label attacks (opacity 0.1 followed by clipping to make ℓ_∞ distortion equal ϵ), and show that solution to the bilevel optimization is significantly better at reducing the certified radius.

For the experiments with MNIST we randomly selected the digit 8 to be targeted by the attacker. To keep the attack clean label the maximum permissible ℓ_∞ distortion is bounded by $\epsilon \leq 0.1$ which is similar to the value used to generate imperceptible adversarial examples ([21, 11]). As the accuracy and ACR of models trained on clean data was high for large values of σ, we used $\sigma \in \{0.25, 0.5, 0.75\}$ for our experiments (Table 1). For all values of σ, we observe a significant reduction in certified radius of the target class on the model trained on the poisoned dataset with only minor changes in clean test accuracy. For the CIFAR10 experiments we randomly selected the "ship" class to be the target class. We used $\epsilon \leq 0.03$ as the maximum permissible distortion for the poisoned data. We only used $\sigma = 0.2$ here, as we found models trained on clean data had low certified radius for smaller values σ and low accuracy on the target class for higher values both of which are not preferable from a defender's perspective. Table 2 shows that our attack reduces the certified radius without degrading the clean accuracy much.

Table 1: Comparison of clean accuracy, certified adversarial robustness and empirical robustness before and after data poisoning attack on digit 8 of MNIST, with $\epsilon = 0.1$

σ	Dataset	Clean test accuracy of Base classifier (%)		Certified Robustness (Target class)		Empirical Robustness (Target class)	
		All	Target	ACR	ACA(%)	Base	Smoothed
0.25	Clean	99.29	99.20	0.899	99.30	1.531	3.749
	Watermarking	98.25	98.56	0.771	98.00	1.201	3.228
	Poisoned	98.35	98.33	**0.522**	89.30	1.366	1.952
0.5	Clean	99.18	98.97	1.459	99.30	1.684	3.855
	Watermarking	97.92	97.64	1.063	96.70	1.417	3.056
	Poisoned	97.78	97.45	**0.823**	92.00	1.422	2.269
0.75	Clean	98.72	98.62	1.581	98.40	1.742	4.008
	Watermarking	95.90	94.02	1.136	95.90	1.285	3.155
	Poisoned	98.69	97.97	**0.768**	87.50	1.700	2.169

Like the case for linear classifiers where data poisoning led to change in the decision boundary to decrease the average certified radius, we observe the similar behavior of poisoning in neural networks. The decrease in the mean distortion of successful attack against the smoothed classifier suggests the decision boundary of the smoothed classifier is closer to the test points of the target class after poisoning. The empirical robustness of the base model being relatively unchanged shows that the decision boundary of the smoothed classifier must be affected to reduce the certified radius. Results of the attack on other classes and attack examples (Fig. 2) are in Appendix D.

Table 2: Comparison of clean accuracy, certified adversarial robustness and empirical robustness before and after data poisoning attack on the class "Ship" from CIFAR10, with $\epsilon = 0.03$

σ	Dataset	Clean test accuracy of Base classifier(%)		Certified Robustness (Target class)		Empirical Robustness (Target class)	
		All	Target	ACR	ACA(%)	Base	Smoothed
0.2	Clean	64.26	79.64	0.405	84.60	0.453	1.645
	Watermarking	65.05	81.68	0.371	76.20	0.416	1.476
	Poisoned	64.90	77.12	**0.333**	77.60	0.388	1.202

5 Conclusion

Certified robustness has emerged as a way to gauge the susceptibility of machine learning models to test-time attacks. In this work we showed that these guarantees can be rendered ineffective by our poisoning attack. Our bilevel optimization based attack adds imperceptible perturbations to the points of the target class and ensures high accuracy of the poisoned models on clean data, making the attack difficult to detect. Unlike previous poisoning attacks, our attack can hurt the average certified radius of an entire class and is even effective against models trained using Gaussian data augmentation. Our results suggests the importance of data quality in achieving high certified robustness guarantees.

6 Acknowledgments

This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344, and was also supported by the NSF EPSCoR-Louisiana Materials Design Alliance (LAMDA) program #OIA-1946231.

References

[1] Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. *arXiv preprint arXiv:1802.00420*, 2018.

[2] Jonathan F Bard. *Practical bilevel optimization: algorithms and applications*, volume 30. Springer Science & Business Media, 2013.

[3] Battista Biggio, Blaine Nelson, and Pavel Laskov. Poisoning attacks against support vector machines. *arXiv preprint arXiv:1206.6389*, 2012.

[4] Saikiran Bulusu, Bhavya Kailkhura, Bo Li, Pramod K Varshney, and Dawn Song. Anomalous example detection in deep learning: A survey. *IEEE Access*, 8:132330–132347, 2020.

[5] Nicholas Carlini and David Wagner. Adversarial examples are not easily detected: Bypassing ten detection methods. In *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*, pages 3–14, 2017.

[6] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *2017 ieee symposium on security and privacy (sp)*, pages 39–57. IEEE, 2017.

[7] Xinyun Chen, Chang Liu, Bo Li, Kimberly Lu, and Dawn Song. Targeted backdoor attacks on deep learning systems using data poisoning. *arXiv preprint arXiv:1712.05526*, 2017.

[8] Jeremy M Cohen, Elan Rosenfeld, and J Zico Kolter. Certified adversarial robustness via randomized smoothing. *arXiv preprint arXiv:1902.02918*, 2019.

[9] Justin Domke. Generic methods for optimization-based modeling. In *Artificial Intelligence and Statistics*, pages 318–326, 2012.

[10] Luca Franceschi, Michele Donini, Paolo Frasconi, and Massimiliano Pontil. Forward and reverse gradient-based hyperparameter optimization. In *International Conference on Machine Learning*, pages 1165–1173, 2017.

[11] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

[12] Sven Gowal, Krishnamurthy Dvijotham, Robert Stanforth, Rudy Bunel, Chongli Qin, Jonathan Uesato, Relja Arandjelovic, Timothy Mann, and Pushmeet Kohli. On the effectiveness of interval bound propagation for training verifiably robust models. *arXiv preprint arXiv:1810.12715*, 2018.

[13] Po-Sen Huang, Robert Stanforth, Johannes Welbl, Chris Dyer, Dani Yogatama, Sven Gowal, Krishnamurthy Dvijotham, and Pushmeet Kohli. Achieving verified robustness to symbol substitutions via interval bound propagation. *arXiv preprint arXiv:1909.01492*, 2019.

[14] W Ronny Huang, Jonas Geiping, Liam Fowl, Gavin Taylor, and Tom Goldstein. Metapoison: Practical general-purpose clean-label data poisoning. *arXiv preprint arXiv:2004.00225*, 2020.

[15] Matthew Jagielski, Alina Oprea, Battista Biggio, Chang Liu, Cristina Nita-Rotaru, and Bo Li. Manipulating machine learning: Poisoning attacks and countermeasures for regression learning. In *2018 IEEE Symposium on Security and Privacy (SP)*, pages 19–35. IEEE, 2018.

[16] Yujie Ji, Xinyang Zhang, and Ting Wang. Backdoor attacks against learning systems. In *2017 IEEE Conference on Communications and Network Security (CNS)*, pages 1–9. IEEE, 2017.

[17] Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. *arXiv preprint arXiv:1703.04730*, 2017.

[18] Mathias Lecuyer, Vaggelis Atlidakis, Roxana Geambasu, Daniel Hsu, and Suman Jana. Certified robustness to adversarial examples with differential privacy. In *2019 IEEE Symposium on Security and Privacy (SP)*, pages 656–672. IEEE, 2019.

[19] Bai Li, Changyou Chen, Wenlin Wang, and Lawrence Carin. Certified adversarial robustness with additive noise. In *Advances in Neural Information Processing Systems*, pages 9464–9474, 2019.

[20] Dougal Maclaurin, David Duvenaud, and Ryan Adams. Gradient-based hyperparameter optimization through reversible learning. In *International Conference on Machine Learning*, pages 2113–2122, 2015.

[21] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017.

[22] Akshay Mehra and Jihun Hamm. Penalty method for inversion-free deep bilevel optimization. *arXiv preprint arXiv:1911.03432*, 2019.

[23] Luis Muñoz-González, Battista Biggio, Ambra Demontis, Andrea Paudice, Vasin Wongrassamee, Emil C Lupu, and Fabio Roli. Towards poisoning of deep learning algorithms with back-gradient optimization. In *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*, pages 27–38, 2017.

[24] Fabian Pedregosa. Hyperparameter optimization with approximate gradient. In *International conference on machine learning*, pages 737–746, 2016.

[25] Aditi Raghunathan, Jacob Steinhardt, and Percy S Liang. Semidefinite relaxations for certifying robustness to adversarial examples. In *Advances in Neural Information Processing Systems*, pages 10877–10887, 2018.

[26] Hadi Salman, Jerry Li, Ilya Razenshteyn, Pengchuan Zhang, Huan Zhang, Sebastien Bubeck, and Greg Yang. Provably robust deep learning via adversarially trained smoothed classifiers. In *Advances in Neural Information Processing Systems*, pages 11292–11303, 2019.

[27] Amirreza Shaban, Ching-An Cheng, Nathan Hatch, and Byron Boots. Truncated backpropagation for bilevel optimization. *arXiv preprint arXiv:1810.10667*, 2018.

[28] Ali Shafahi, W Ronny Huang, Mahyar Najibi, Octavian Suciu, Christoph Studer, Tudor Dumitras, and Tom Goldstein. Poison frogs! targeted clean-label poisoning attacks on neural networks. In *Advances in Neural Information Processing Systems*, pages 6103–6113, 2018.

[29] Jacob Steinhardt, Pang Wei W Koh, and Percy S Liang. Certified defenses for data poisoning attacks. In *Advances in neural information processing systems*, pages 3517–3529, 2017.

[30] Alexander Turner, Dimitris Tsipras, and Aleksander Madry. Clean-label backdoor attacks. 2018.

[31] Jonathan Uesato, Brendan O'Donoghue, Aaron van den Oord, and Pushmeet Kohli. Adversarial risk and the dangers of evaluating against weak attacks. *arXiv preprint arXiv:1802.05666*, 2018.

[32] Kaidi Xu, Zhouxing Shi, Huan Zhang, Yihan Wang, Kai-Wei Chang, Minlie Huang, Bhavya Kailkhura, Xue Lin, and Cho-Jui Hsieh. Automatic perturbation analysis for scalable certified robustness and beyond. *Advances in Neural Information Processing Systems*, 33, 2020.

[33] Runtian Zhai, Chen Dan, Di He, Huan Zhang, Boqing Gong, Pradeep Ravikumar, Cho-Jui Hsieh, and Liwei Wang. Macer: Attack-free and scalable robust training via maximizing certified radius. *arXiv preprint arXiv:2001.02378*, 2020.

[34] Chen Zhu, W Ronny Huang, Ali Shafahi, Hengduo Li, Gavin Taylor, Christoph Studer, and Tom Goldstein. Transferable clean-label poisoning attacks on deep neural nets. *arXiv preprint arXiv:1905.05897*, 2019.

Appendix

A Proofs

Theorem 1. *If the perturbation is large enough, i.e.,* $\epsilon \geq \frac{\sum_i x_i^+ - \sum_i x_i^-}{n}$ *then there are two locally optimal solutions to (2) which are* $u_i = x_i - \epsilon$ *(Case 1) and* $u_i = x_i + \epsilon$ *(Case 2) for* $i = 1, ..., n$*. Otherwise, the is a unique globally optimal solution which is* $u_i = x_i - \epsilon$ *(Case 1) for* $i = 1, ..., n$*.*

Proof. Let $t = -\frac{b}{w}$ be the threshold of the linear classifier. Also let $\Phi(t) := \int_{-\infty}^{t} P_-(x)\ dx$ and $\Psi(t) := \int_{-\infty}^{t} xP_-(x)\ dx$. There are two cases to consider.
Case 1 ($w > 0$): The upper-level cost function is

$$f(t) \quad = \quad \int_{-\infty}^{t} (t - x)P_-(x)\ dx = t\Phi(t) - \Psi(t)$$

Note that the range $[-\infty, t]$ is where classification is correct for the test data. (Certified radius is 0 for misclassified points by definition.)

The closed-from solution of the lower-level problem gives us $t = -\frac{b}{w} = \frac{\sum_i u_i + \sum_i x_i^+}{2n}$, and therefore the perturbation bound $|u_i - x_i^-| \leq \epsilon$ implies $\sum_i x_i^- - n\epsilon \leq \sum_i u_i \leq \sum_i x_i^- + n\epsilon$ and therefore

$$-\frac{\epsilon}{2} + \frac{\sum_i x_i^+ + \sum_i x_i^-}{2n} \leq t \leq \frac{\epsilon}{2} + \frac{\sum_i x_i^+ + \sum_i x_i^-}{2n}.$$

Also, the assumption $w > 0$ poses another constraint: $w \propto \sum_i x_i^+ - \sum_i u_i > 0$ and therefore $t = \frac{\sum_i u_i + \sum_i x_i^+}{2n} \leq \frac{\sum_i x_i^+}{n}$.

The upper-level problem is therefore

$$\min_t\ f(t) = t\Phi(t) - \Psi(t)\ \text{ s.t. }\ -\frac{\epsilon}{2} + \frac{\sum_i x_i^+ + \sum_i x_i^-}{2n} \leq t \leq \frac{\epsilon}{2} + \frac{\sum_i x_i^+ + \sum_i x_i^-}{2n}\ \text{ and }\ t \leq \frac{\sum_i x_i^+}{n}.$$

Since f is non-decreasing (i.e.,$f'(t) = \Phi(t) + tP_-(t) - tP_-(t) \geq 0$), the minimum is achieved at the left-most boundary $t = -\frac{\epsilon}{2} + \frac{\sum_i x_i^+ + \sum_i x_i^-}{2n}$ which corresponds to $u_i = -\epsilon,\ \ i = 1, ..., n$.
Case 2 ($w < 0$): The upper-level cost function is now

$$f(t) = \int_{t}^{\infty} (-t + x)P_-(x)\ dx = -t(1 - \Phi(t)) + (1 - \Psi(t)),$$

which is non-increasing $(i.e., f'(t) = -(1 - \Phi) + tP_- - tP_- \leq 0)$ and has the constraints:

$$-\frac{\epsilon}{2} + \frac{\sum_i x_i^+ + \sum_i x_i^-}{2n} \leq t \leq \frac{\epsilon}{2} + \frac{\sum_i x_i^+ + \sum_i x_i^-}{2n}.$$

and

$$t = \frac{\sum_i u_i + \sum_i x_i^+}{2n} \geq \frac{\sum_i x_i^+}{n}.$$

For the solution to be feasible, it is required that $\frac{\sum_i x_i^+}{n} \leq \frac{\epsilon}{2} + \frac{\sum_i x_i^+ + \sum_i x_i^-}{2n}$, that is $\frac{\epsilon}{2} \geq \frac{\sum_i x_i^+ - \sum_i x_i^-}{2n}$ (remember the assumption $\frac{\sum_i x_i^-}{n} \leq \frac{\sum_i x_i^+}{n}$). Therefore if the perturbation is large enough, i.e., $\epsilon \geq \frac{\sum_i x_i^+ - \sum_i x_i^-}{n}$ holds, then the minimum is achieved at the right-most boundary $t = \frac{\epsilon}{2} + \frac{\sum_i x_i^+ + \sum_i x_i^-}{2n}$ which corresponds to $u_i = \epsilon,\ \ i = 1, ..., n$. □

B Review of bilevel optimization

A bilevel optimization problem is of the form $\min_{u \in \mathcal{U}} \xi(u, v^*)$ s.t. $v^* = \arg\min_{v \in \mathcal{V}(u)}\ \zeta(u, v)$, where the upper-level problem is a minimization problem with v constrained to be the optimal solution to the lower-level problem. General bilevel problems are difficult to solve but if the solution

to the lower-level problem can be computed in closed form then we can replace the lower-level problem with its solution, reducing the bilevel problem into a single level problem. We can then use the gradient-based approaches to solve the single level problem. The total derivative $\frac{d\xi}{du}(u, v^*(u))$ (hypergradient) using the chain rule is

$$\frac{d\xi}{du} = \nabla_u \xi + \frac{dv}{du} \cdot \nabla_v \xi.$$

Since $\nabla_v \zeta = 0$ at $v = v^*(u)$ and assuming $\nabla^2_{vv}\zeta$ is invertible we can compute $\frac{dv}{du}$ using the implicit function theorem (this can be done even if the solution to lower-level problem can't be found in closed form) which gives

$$\frac{dv}{du} = -\nabla^2_{uv}\zeta(\nabla^2_{vv}\zeta)^{-1}.$$

Thus the hypergradient is

$$\frac{d\xi}{du} = \nabla_u \xi - \nabla^2_{uv}\zeta(\nabla^2_{vv}\zeta)^{-1}\nabla_v \xi \text{ at } (u, v^*(u)).$$

Since computation of $(\nabla^2_{vv}\zeta)^{-1}$ is difficult, [9, 24] proposed to approximate the solution to $q = (\nabla^2_{vv}\zeta)^{-1}\nabla_v\xi$ by solving the linear system of equations $\nabla^2_{vv} \cdot q \approx \nabla_v \xi$ by minimizing $\|\nabla^2_{vv}\zeta \cdot q - \nabla_v \xi\|$ using any iterative solver. Other methods for solving the bilevel optimization problems include using forward/reverse mode differentiation [10, 20, 27] to approximate the inverse and penalty method [22] to solve the single level problem as a constrained minimization problem.

Algorithm 1 Algorithm for ApproxGrad

Input: $\xi, \zeta, K, T_1, T_2, \{\tau_k\}, \{\rho_{k,t_1}\}, \{\beta_{k,t_2}\}, \epsilon, u_{base}$
Output: (u_K)
Initialize u_0, v_0 randomly
Begin
for $k = 0, \cdots, K$-1 **do**

# Approximately solve the lower-level problem
for $t = 0, \cdots, T_1$-1 **do**
$v_{t+1} \leftarrow v_t - \rho_{k,t_1}\nabla_v \zeta$
end for

# Approximately solve the linear system $\nabla^2_{vv}\zeta \cdot q_k = \nabla_v \xi$
for $t = 0, \cdots, T_2$-1 **do**
$q_{t+1} \leftarrow q_t - \beta_{k,t_2}\nabla_q(\|\nabla^2_{vv}\zeta \cdot q_k - \nabla_v \xi\|)$
end for

Compute the approximate Hypergradient
$p_k = \nabla_u \xi - \nabla^2_{uv}\zeta \cdot q_{T_2}$

Update u_k and use projection for the upper-level constraint
$u_{k+1} = P(u_k - \tau_k p_k, \epsilon, u_{base})$

end for

C Attack algorithm

Alg. 2 shows the complete algorithm used to generate the poisoning attack.

C.1 ApproxGrad

For an unconstrained bilevel problem of the form $\min_u \xi(u, v^*)$ s.t. $v^* = \arg\min_v \ \zeta(u, v)$ if $\zeta(u, v)$ is strongly convex then we can replace the lower-level problem with its necessary condition and write the bilevel problem as the following single level problem $\min_u \xi(u, v^*)$ s.t. $\nabla_v \zeta(u, v) = 0$.

Assuming $\nabla^2_{vv}\zeta$ is invertible everywhere we can compute the hypergradient at the point $(u, v^*(u))$ as $\frac{d\xi}{du} = \nabla_u \xi - \nabla^2_{uv}\zeta(\nabla^2_{vv}\zeta)^{-1}\nabla_v \xi$.

The ApproxGrad algorithm approximates the Hessian-inverse vector product by approximately solving a system of linear equation using an iterative solver such as gradient descent or conjugate gradient method. In this work we use Adam optimizer to solve this system. Since our problem for data poisoning in Eq. 1 involves a constraint in the upper-level we use projection to enforce the constraint. The full algorithm for solving the unconstrained bilevel optimization problem using ApproxGrad is present in Alg. 1. For our attack the lower-level problem involves a deep neural network, which can have multiple local minima and thus optimizing against a single local minima in the bilevel problem is not ideal. To overcome this problem we reinitialize the lower-level variable v after few upper-level iterations to prevent the poisoning points from overfitting to a particular local minima. Empirically, this helps us find poisoning points that remain effective even after the model is retrained from scratch making them generalize to different initialization of the neural network.

Algorithm 2 Full attack algorithm

Input: $(X^{\text{clean}}, Y^{\text{clean}}), (X^{\text{base}}, Y^{\text{base}}), (X^{\text{val}}, Y^{\text{val}}), \epsilon, \sigma, \lambda, M, \alpha, P, Loss,$
$T_1, T_2, \{\tau_k\}, \{\rho_{k,t_1}\}, \{\beta_{k,t_2}\}$
Output: $(X^{\text{poison}}, Y^{\text{poison}})$
Begin
  $X^{\text{poison}} := X^{\text{base}}$
  $Y^{\text{poison}} := Y^{\text{base}}$
  for $p = 0, \cdots, P$-1 **do**
    Sample a mini-batch $(x^{\text{clean}}, y^{\text{clean}}) \sim (X^{\text{clean}}, Y^{\text{clean}})$
    Sample a mini-batch of n samples$(x^{\text{val}}, y^{\text{val}}) \sim (X^{\text{val}}, Y^{\text{val}})$
    Sample a mini-batch $(x^{\text{poison}}, y^{\text{poison}}) \sim (X^{\text{poison}}, Y^{\text{poison}})$
    Pick the corresponding base samples for poison data $(x^{\text{base}}, y^{\text{base}})$
    For each x_i^{val}, sample M i.i.d. Gaussian samples $x_{i_1}^{\text{val}}, \cdots, x_{i_n}^{\text{val}} \sim \mathcal{N}(x_i^{\text{val}}, \sigma^2 I)$
    Compute $\tilde{z}_\theta(x_i^{\text{val}}) \leftarrow \sum_{j=1}^{M} \alpha z_\theta(x_{ij}^{\text{val}})/M$ for $i = 1, \cdots, n$
    Compute $\mathbb{G}_\theta = \{(x_i^{\text{val}}, y_i^{\text{val}}) : y_i^{\text{val}} = \arg\max_{c \in \mathcal{Y}} \tilde{z}_\theta^c(x_i^{\text{val}})\}$
    For each $(x_i, y_i) \in \mathbb{G}_\theta$, compute $\tilde{y}_i : \tilde{y}_i \leftarrow \arg\max_{c \in \mathcal{Y} \setminus \{y_i\}} \tilde{z}_\theta^c(x_i)$
    For each $(x_i, y_i) \in \mathbb{G}_\theta$, compute $CR(x_i, y_i) : \frac{\sigma}{2}(\Phi^{-1}(\tilde{z}_\theta^{y_i}(x_i)) - \Phi^{-1}(\tilde{z}_\theta^{\tilde{y}_i}(x_i)))$
    $\zeta := Loss((x^{\text{poison}}, y^{\text{poison}}) \bigcup (x^{\text{clean}}, y^{\text{clean}}), \sigma)$
    $\xi := Loss(x^{\text{val}}, y^{\text{val}}) + \frac{\lambda}{n} \sum_{(x_i, y_i) \in \mathbb{G}_\theta} CR(x_i, y_i)$
    $(x^{\text{poison}}, y^{\text{poison}}) = \text{ApproxGrad}(\xi, \zeta, 1, T_1, T_2, \{\tau_k\}, \{\rho_{k,t_1}\}, \{\beta_{k,t_2}\})$
  end for

D Additional experiments

D.1 Targeting other classes

In this section we present the results of our poisoning attack on different target classes. Since MNIST and CIFAR10 both have 10 classes we create 10 poisoning sets each targeting a particular class. The results of retraining models from five random initializations on each of the 10 poisoning sets are summarized in Table 3 and Table 4. The reduction in certified radius while maintaining accuracy high accuracy on clean data suggests that our attack is general purpose and does not depend on the choice of the target class.

D.2 Isotropic Gaussians

Here we validate the solution found by solving the bilevel optimization against the analytical solution of a simple problem. Consider a two-dimensional dataset comprising of points drawn from two isotropic Gaussian distributions. Let $\mathbb{P}(x|y=-1)) = \mathcal{N}(\mu_1, \sigma^2 I)$ and $\mathbb{P}(x|y=1) = \mathcal{N}(\mu_2, \sigma^2 I)$ and equal prior $\mathbb{P}(y=1) = \mathbb{P}(y=-1)$. For a point x, the Bayes optimal classifier predicts $y=1$ if $\mathbb{P}(y=1|x) >= \mathbb{P}(y=-1|x)$ and predicts $y=-1$ otherwise. The decision boundary of the Bayes optimal classifier is given by $(\mathbf{x} - \mu_\mathbf{1})^\mathbf{T}(\mathbf{x} - \mu_\mathbf{1}) = (\mathbf{x} - \mu_\mathbf{2})^\mathbf{T}(\mathbf{x} - \mu_\mathbf{2})$. This is also the

Table 3: Comparison of clean accuracy, certified adversarial robustness and empirical robustness before and after data poisoning attack on MNIST. The results are averaged over poisoning attacks generated by considering each of the 10 MNIST digits as targets, one at a time ($\epsilon = 0.1$).

σ	Dataset	Clean test accuracy of Base classifier(%)		Certified Robustness (Target class)		Empirical Robustness (Target class)	
		All	Target	ACR	ACA(%)	Base	Smoothed
0.5	Clean	99.17±0.01	99.18±0.01	1.555±0.11	99.35±0.01	1.981±0.27	3.681±0.44
	Poisoned	98.20±0.01	98.50±0.01	1.072±0.33	94.25±0.07	1.750±0.22	2.709±0.65

Table 4: Comparison of clean accuracy, certified adversarial robustness and empirical robustness before and after data poisoning attack on CIFAR10. The results are averaged over poisoning attacks generated by considering each of the 10 classes in CIFAR10 as targets, one at a time ($\epsilon = 0.03$).

σ	Dataset	Clean test accuracy of Base classifier(%)		Certified Robustness (Target class)		Empirical Robustness (Target class)	
		All	Target	ACR	ACA(%)	Base	Smoothed
0.2	Clean	64.76±0.01	64.41±0.11	0.298±0.11	67.13±0.15	0.331±0.13	1.524±0.42
	Poisoned	64.47±0.01	60.18±0.11	0.253±0.08	60.75±0.13	0.295±0.12	1.397±0.45

decision boundary of the smoothed classifier. Assuming the attacker is poisoning the class with label -1 and maximum permissible distortion is ϵ, our analysis showed that maximum reduction in radius occurs if the entire distribution shifts by ϵ i.e. the new mean of the class with label -1 is $\mu_1 - \epsilon$ and the decision boundary is $(\mathbf{x} - (\mu_\mathbf{1} - \epsilon))^\mathbf{T}(\mathbf{x} - (\mu_\mathbf{1} - \epsilon)) = (\mathbf{x} - \mu_\mathbf{2})^\mathbf{T}(\mathbf{x} - \mu_\mathbf{2})$. Since the test distribution is unchanged, the average certified radius for the test points with labels -1 is reduced by $\frac{\epsilon}{\sqrt{2}}$. Using $\mu_1 = 0.2, \mu_2 = 0.8, \sigma_1 = \sigma_2 = 0.3, \epsilon = 0.1$ and using logistic regression in the lower-level, analytically, certified radius must decrease from 0.4243 to 0.3546. The solution by solving the bilevel optimization numerically (Table 5) matches the analytic solution.

E Details of experiments

All codes are written in Python using Tensorflow/Keras, and were run on Intel Xeon(R) W-2123 CPU with 64 GB of RAM and dual NVIDIA TITAN RTX. Implementation and hyperparameters are described below.

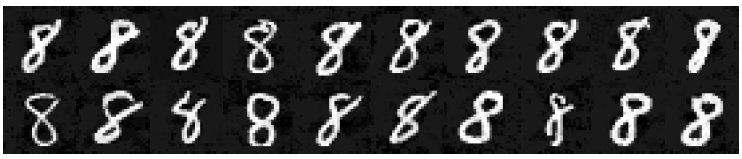
E.1 Data splits

For MNIST, we use 55000 points as the training data and 5000 points for validation data. We have roughly 500 points belonging to the target class in the validation set which is used in the upper-level problem of Eq. (1). The test set comprises of 10000 points. We use 200 randomly sampled points of the target class from the test set to report certified and empirical robustness of the retrained models.

For CIFAR10, we use 45000 points as the training data and 5000 points for validation data. Similar to MNIST we have roughly 500 points belonging to the target class in the validation set which is used in the upper-level problem of Eq. (1). The test set comprises of 10000 points. We use 100 randomly

Table 5: Comparison of clean accuracy, certified adversarial robustness and empirical robustness before and after data poisoning attack on class -1 using isotropic Gaussians, with $\epsilon = 0.1$

σ	Dataset	Clean test accuracy of Base classifier(%)		Certified Robustness (Target class)		Empirical Robustness (Target class)	
		All	Target	ACR	ACA(%)	Base	Smoothed
0.25	Clean	91.00	90.40	0.4047	90.00	0.4326	0.4292
	Poisoned	90.80	88.00	0.3585	88.00	0.3861	0.3798
0.5	Clean	90.80	90.40	0.4139	90.00	0.4290	0.4308
	Poisoned	91.00	88.00	0.3587	87.60	0.3747	0.3751
0.75	Clean	90.80	90.40	0.4123	90.00	0.4303	0.4315
	Poisoned	91.00	88.00	0.3544	87.60	0.3728	0.3736



(a) Poisoning points for digit 8 of MNIST

(b) Poisoning points for the class ship of CIFAR10

Figure 2: Poisoning points generated by our attack. The poisoned data has very little distortion showing that our attack points will have clean labels when inspected by an expert.

sampled points of the target class from the test set to report certified and empirical robustness of the retrained models.

The accuracy of the base is measured on the entire test set and also on all the points belonging to the target class.

E.2 Model Architecture

For the experiments on the MNIST dataset, our network consists of a convolution layer with kernel size of 5x5, 20 filters and ReLU activation, followed by a max pooling layer of size 2x2. This is followed by another convolution layer with 5x5 kernel, 50 filters and ReLU activation followed by similar max pooling and dropout layers. Then we have a fully connected layers with ReLU activation of size 500. Lastly, we have a softmax layer with 10 classes. The accuracy of the model on clean data when optimized with the Adam optimizer using a learning rate of 0.001 for 100 epochs with batch size of 200 is 99.3%, without Gaussian data augmentation.

For the experiments on the CIFAR10 dataset, our network consists of 3 convolution blocks with filter sizes of 48, 96, and 192. Each convolution block consists of two convolution layers, each with kernel size of 3x3 and ReLU activation. This is followed by a max pooling layer of size 2x2 and a drop out layer with drop rate of 0.25. After these 3 blocks we have 2 dense layers with ReLU activation of size 512 and 256 respectively, each followed by a dropout layer with rate 0.5. Finally we have a softmax layer with 10 classes. The accuracy of the model on clean data when optimized with the Adam optimizer using a learning rate of 0.001 for 100 epochs with batch size of 200 is 81%, without Gaussian data augmentation.

We use the same parameters for training the models with Gaussian data augmentation on clean and poisoned data.

E.3 Hyperparameters

For experiments with MNIST we used $\epsilon = 0.1, \lambda = 0.5, M = 20, \alpha = 16$. The batch size used for lower-level training was 1000, of which 100 points belonged to the poisoned set (target class). The batch size for validation set was 100 which only consisted of points from the target class. The lower-level was trained with Gaussian augmentations of the clean and poisoned data.

For experiments with CIFAR10 we used $\epsilon = 0.03, \lambda = 0.06, M = 20, \alpha = 16$. The batch size used for lower-level training was 200, of which 20 points belonged to the poisoned set (target class). The batch size for validation set was 20 which only consisted of points from the target class. The lower-level was trained with Gaussian augmentations of the clean and poisoned data.

In all the experiments used $P = 100, T_1 = T_2 = 10, \tau = 0.1, \rho = 0.001, \beta = 0.01$ for ApproxGrad. For certification we used the CERTIFY procedure of [8], with $n_0 = 1000, n = 100000, \alpha = 0.001$. For measuring empirical robustness of the smoothed classifier, we used the mean ℓ_2 distortion

required by PGD attack to generate an adversarial example as done in [26]. The attack is optimized for 100 iterations for different values of ℓ_2 distortion between (0.01, 7). We use 20 augmentations for each test points. We used CW attack [6] optimized for 100 steps with 10 binary search steps to find the adversarial examples for the base classifier. For both attacks the minimum distortion for a successful attack is recorded for each test point and is used to report the empirical robustness of smoothed and base classifiers.

For the watermarking baseline, we randomly selected an image ($other$) from the classes other than the target class and overlayed them on top of the target class images ($base$) with an opacity of $\gamma = 0.1$ i.e. ($poison\_image = \gamma \cdot other + (1 - \gamma) \cdot base$). We then clip the images to have ℓ_∞ distortion of ϵ to make our bilevel attack comparable in terms of maximum distortion.