Limitations Of Gradient Shaping as a Defense Against Data Poisoning

Himanshu Jahagirdar
Department of Electrical and Computer Engineering
Virginia Tech
Blacksburg, VA 24060
himanshugj@vt.edu

Abstract

Gradient Shaping uses Differentially Private Stochastic Gradient Descent (DP-SGD) as an off-the-shelf defense against training-time data poisoning attacks. Intuitively, by clipping individual gradients and addition of noise, no single point can overly influence model behavior. (1) proved this to be effective for smaller poison ratios. This project attempts to use attacks that do not necessarily enhance the gradient size of poisoned samples. Further, this project tries to test whether this method also applies to larger poisoning ratios. Additionally this project also attempts to use more recent state of the art attacks to prove that with better attacks we can attack a model with small poison ratios as well.

1 Introduction

Last week, OpenAI released ChatGPT -a language model capable of writing your resume, writing a top-shelf codebase, a research paper among other exciting applications. It's safe to say that AI is increasingly becoming a bigger part of our lives. Thousands of research papers in the domain push the state of the art in AI every month. With this speed, research in AI security has become a rising concern.

First and foremost, AI systems are increasingly being used in critical applications, such as healthcare and finance, where a failure or security breach could have serious consequences. Ensuring the security of these systems is crucial in order to protect the safety and well-being of individuals. Additionally, AI systems often process large amounts of sensitive data, such as personal information or financial records. Protecting this data from unauthorized access or misuse is essential in order to prevent identity theft, financial fraud, and other forms of cybercrime.

Furthermore, as AI systems become more advanced and integrated into our daily lives, there is a growing concern about the potential for malicious actors to use AI for nefarious purposes, such as creating fake news or spreading misinformation. Ensuring the security of AI systems is therefore important for protecting against these types of threats. Overall, AI security is crucial for protecting individuals, organizations, and society as a whole from the potential negative consequences of AI.

1.1 Gradient Shaping

Differentially private SGD (DP-SGD) (2) is a variant of the stochastic gradient descent algorithm that is designed to protect the privacy of the training data. This is achieved by adding noise to the gradients of the model in order to prevent an attacker from inferring any information about the training data from the model's parameters. First, the gradients are computed sample-wise. Then, the gradients for each sample are clipped to a fixed maximum norm. Finally the above-mentioned noise is added to the aggregated gradients.

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

While the primary objective of DP-SGD is ensuring and quantifying privacy guarantees, (1) hypothesized that it could be used off-the-shelf as a defense against data poisoning attacks. The reasoning behind this intuition is that since poisoned samples have a larger gradient magnitude than a clean sample, the clipped norm and added noise can minimize the effect of a poisoned sample during training. Secondly, they found that the gradients from poisoned samples also show an orientation difference from clean gradients. Similar to the magnitude, DP-SGD also naturally mitigates the orientation difference.

1.2 Can this defense be broken?

The question asked through this project is if this defense can be broken. The biggest drawback of DP-SGD is utility-degradation - we cannot achieve state of the art classification performance. In a talk by DeepMind, the best accuracy on CIFAR-10 using DP-SGD was 78% and 74% without using random augmentation. Meanwhile SGD can achieve SOTA accuracy well above 95%. While we do not reach SOTA accuracy in this project, we do achieve near-SOTA performance, especially through DP-SGD. The second drawback, especially discussed in the Gradient Shaping paper (1) is that we lose all privacy guarantees if this defense is deployed. The primary objective of DP-SGD is to ensure privacy and we lose that benefit. The reasoning behind this phenomenon is that we require smaller clipping norm and larger noise to achieve this defense, opposite to the requirements of privacy. Hence, the paper accepts that this model may be susceptible to privacy attacks like Membership Inference (3). For the purpose of this project we assume the same.

The *contributions* of this project are the following:

- Study the **limitations of Gradient Shaping** as a defense, if any, and try to launch a successful attack against this defense.
- Evaluate a baseline with a naive trigger and large poison ratio (1%-10%) to **test if Gradient Shaping only applies to small poison ratios**
- Evaluate a stronger clean-label attack (4) against this defense and **test if DP-SGD can be exploited even at small poison ratios** (0.1% - 1%)

2 Related Work

Gradient Shaping: (2) first introduced Differentially Private SGD as an optimizer that binds the influence of a single gradient to the model. The aim of this work was to ensure privacy guarantees in a ML model by ensuring that if no single sample has superior influence over the model, then even if that sample is removed, the model utility does not decrease. (1) used DP-SGD as an off-the-shelf optimizer to defend against data poisoning attacks. The intuition is that at the cost of sacrificing privacy, clipping the gradients and adding noise will limit the contribution of a single sample. Hence, small poison ratios in a dataset can be defended using DP-SGD.

However, DP-SGD does come with it's own challenges. By aggregating clipped gradients, and the additional noise, model utility is degraded. State of the art (SOTA) accuracies cannot be achieved. Also, (1) accept that by sacrificing privacy, their model is vulnerable to privacy attacks such as Membership Inference (3). (5) claimed that DP-SGD will fail for larger poison ratios since more number of poisoned samples can potentially influence the model. More importantly, for poisoned samples that do not differ from

Backdoor Attacks and Defenses Backdoor attacks are training-time attacks that manipulate a model behavior to cause a misclassification to target class when a trigger is injected into a test sample. Attacker assumes that poisoned samples with the trigger can be injected into the training pipeline. Attacks can be indiscriminate or targeted, and this project deals with the latter. Target backdoor attacks can be dirty-label (labels of poisoned samples are flipped), clean-label or label-only (only the label is changed, no trigger is injected).

(6) first emphasized that samples with high influence can be used to increase classification error towards a target class. (7) were the first to inject backdoors into training data as an attack towardsa ML model. (8) showed that a simple image can be patched on another image to serve as a trigger-based attack. (9) showed that these simple triggers can be visibly perceptible and easy to detect. (10) further showed that a clean-label attack is possible via feature collision, which can be stealthier than the previous attacks. This attack was improved by (**?**) More recently, (11) Narcissus Backdoor attack is

a clean-label attack that can achieve high ASR on CIFAR-10 with just 25 samples out of 50K being poisoned. Most importantly, the trigger is imperceptible to the naked eye.

With advancements in research on Backdoor Attacks, there has been an equally motivatived research on defenses against these attacks. (12) use a simple outlier-detection strategy to filter out bad data. Among other popular trigger inversion based defenses, (13) introduced Neural Cleanse as a trigger synthesis based method to detect triggers and mitigate data poisoning. (14) introduce STRIP use benign image patches to detect which samples contain a trigger. (15) present a trigger-agnostic method to detect backdoored samples based on their response to carefully crafted inputs.(16) present an adversarial unlearning based method to remove a potential trigger from a sample during a forward pass. (17) interestingly outline an approach that has some shared principles with DP-SGD, since they use gradient matching as one of their key components. However, with 1% poison ratio, even their attack achieves low ASR against DP-SGD based defense.

3 Methodology

Threat Model: The original paper (1) assumes an interesting threat model - pre-train a model on clean data for 100 epochs and then fine-tune for 50 epochs on the poisoned data. Additionally, the poison ratios discussed in the paper are 0.1% to 1%. For the scope of this project, we only consider the CIFAR-10 dataset as it is covered in the original paper. We compare every result with similar settings for SGD against DP-SGD to demonstrate both utility degradation and defense performance. The attacker assumes access to a private training set. Trigger is generated using a surrogate model. The threat models are discussed in further detail separately for the baseline attack and the clean-label attack:

Blend Attack (8): Blend is a simple patch-based backdoor attack where a Hello Kitty image is patched on some samples as the trigger. This naive attack is easily identifiable to the naked eye and does not require a large number of epochs for training. Hence, we run experiments with Blend for a maximum of 40 epochs. Two models are used to test the Blend attack - SmallCNN (a simple model with 2 Convolutional layers and 2 Dense Layers) and ResNet-18 (18). The project also tests out two configurations - (i) pretraining on clean data for 40 epochs and fine-tuning on poisoned data for 20 epochs, (ii) Direct training on poisoned data (more realistic setting). A low learning rate of 1e-3 is used for the SmallCNN and a learning rate of 0.1 is used for the ResNet-18 model.

Narcissus Attack (11) : Narcissus is a clean-label backdoor attack that has a high ASR for poison ratios as low as 0.0005% or 25 samples in CIFAR-10. The Narcissus trigger is generated by optimizing the following equation for a particular target class. Delta here refers to the perturbation for a particular target class. t refers to the target class. The gradient from the sample(s) with minimum loss is used to sample the gradient at each stage.

$$\delta^* = \delta \in \Delta \arg\min \sum_{(x,t)\in D_t} \mathcal{L}\left(f_{\theta_{sur}}(x+\delta), t\right),$$

Based on results for the baseline, we directly train the model on poisoned data for 200 epochs. Learning rate of 0.1 is used for our only model in this setting -ResNet-18. DP-SGD specifications: The project use Opacus (19), a tool by Meta, to integrate DP-SGD with pytorch. The tradeoff between clipping norm and noise multiplier were studied in the original paper and the preliminary results achieved for this project also motivated the use of a low value of clipping norm (0.1) and higher value of noise multiplier (4.0). While we do not focus on privacy, a delta of 1e-5 was chosen.

Metrics : This project utilizes a fewer number of metrics for the baseline, as some only apply to Narcissus' backdoor attack. All the metrics used in this project are discussed below:

- **Clean Train loss:** The training Loss under a clean-label attack after model has converged. This metric is crucial and a low value is necessary to fool a defender.
- **ASR:** Attack success rate is defined as the percentage of poisoned test samples mis-classified into the target class.
- **Clean test accuracy:** This metric is simply the validation accuracy of the model under clean data. We need a high value to fool the defender, however as seen from experiments - it is hard to achieve a high value with DP-SGD.

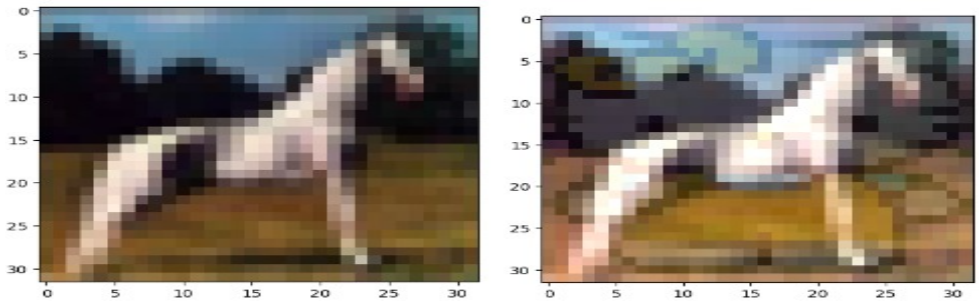

Figure 1: **Visualization of the Blend Trigger:** Left image is clean while the right image is poisoned

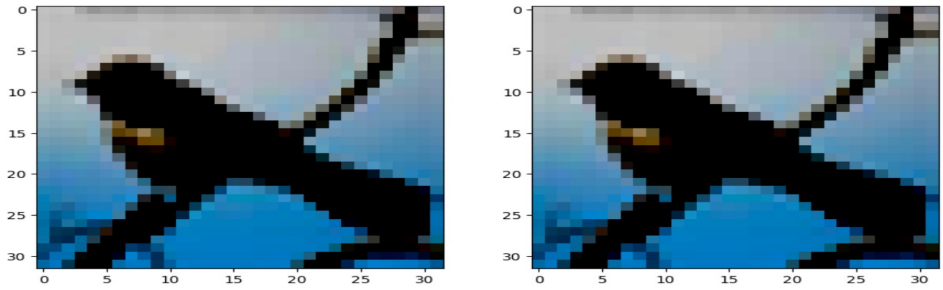

Figure 2: **Visualization of Narcissus Trigger:** Left image is clean while the right image is poisoned

- **Clean target test accuracy:** The validation accuracy of clean data belonging to target class. A higher value of this metric will show that the target class has not overfit to the trigger at the cost of underfitting the clean distribution of target class.

4 Experiments and Results

Blend Backdoor Attack: The project injects a naive trigger (an image patch) on to a random image and the label of the image is changed to the target class. This forces the class to learn spurious correlations between the patched image and the target class. The trigger can be visualized in Fig. 1.

Table 4 shows the results from our experiments on Blend attack using SGD vs DP-SGD, which are consistent across both SmallCNN and ResNet-18. As we can see, SmallCNN and ResNet-18 (18)both show an accuracy drop on SGD vs DP-SGD. We find that Validation accuracy for DP-SGD is unable to exceed 69%, which is a serious utility degradation.

First, we consider the pre-training scenario. We find that for 1% poison ratio, DP-SGD is indeed a good defense and the ASR is only 29% for SmallCNN. However, both SmallCNN and ResNet-18 models acheive high ASR in DP-SGD settings (80% and 85% respectively) for a poison ratio of 10%. This proves that DP-SGD on a pre-trained model is **not a suitable defense for larger poison ratios**. Secondly, we also compute the ASR for direct training on poisoned data (last column in Table 4) and we find that the ASR is higher for both models. The '*' result is for a model poisoned with 1% poison ratio. We observe a diverging loss and hard-to-tune hyperparameters. Hence, Blend Trigger fails to get high ASR for small poison ratios. We also declare that since the ASR on direct training is higher by only around 5% in both cases, we do not consider pre-training in the Narcissus Backdoor Attack setting. It is possible that pre-training offers some robustness but it is surely not enough to justify a successful defense. Additionally we also test the effect of DP-SGD hyperparameters on the ASR and find that lower value of clipping norm gives a higher ASR.

Narcissus Backdoor Attack: Narcissus Backdoor Attack, unlike Blend, poisons images from target class with the trigger and there is no label-flipping involved. Hence, it is expected that the gradients from poisoned samples do not differ from their clean alternatives. Moreover, this attack is imperceptible to a human user (Fig. 2). Narcissus (11) claims to achieve a poison ratio of 85%

Table 1: **Baseline:** Naive Trigger - Blend Attack

Model	SGD Validation acc 40 epochs	DP-SGD Validation acc 40 epochs	Poison Ratio	ASR -Blend SGD 20 epochs	ASR -Blend DP-SGD 40-60 epochs	ASR- Direct training on poisoned data 20 epochs
SmallCNN	74%	58%	10%	93%	**80%**	90%
			1%	30 %	**29%**	∼**20%***
ResNet-18	83%	69%	10%	93%	85%	90%

Table 2: **Narcissus Backdoor Attack on ResNet-18 with DP-SGD**

Model ResNet with DP-SGD	Clean train loss	Clean test accuracy	ASR	Target clean test accuracy
Poison Ratio 0.0005%	0.13	68.3%	13.13%	70.6%
Poison Ratio 1%	0.21	65%	75%	72.8%
Poison Ratio 2%	0.3	67%	99%	77.5%

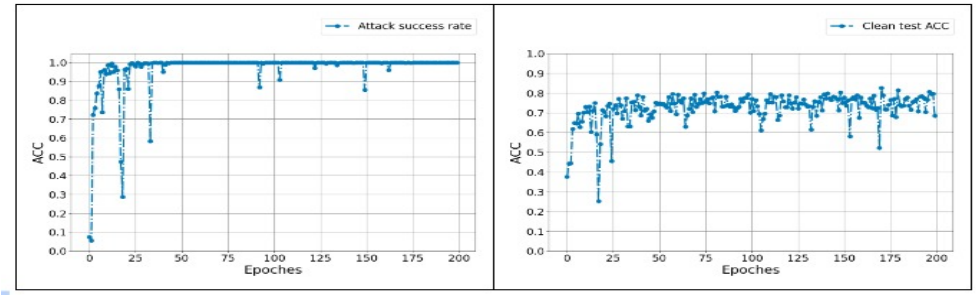

Figure 3: Empirical Convergence Study.

poisoning only 25 samples i.e. 0.0005% of CIFAR-10 (50K training samples) while successfully fooling the classifier with high validation accuracy.

Table 2 shows results for ResNet-18 model trained using a DP-SGD for poison ratios 0.0005%, 1% and 2%. Interestingly, with DP-SGD and a really small poison ratio of 0.0005%, the ASR drops to 13.3%. It is important to note that the decrease in model utility is likely from the utility-degradation usually accompanying DP-SGD. However, for a poison ratio of 1%, we see that the ASR has increaseed to 75% and eventually with a 2% poison ratio, 99% ASR can be achieved. More importantly, the clean test accuracy has not decreased a lot (only 2-3%). In Table 4, training a ResNet 18 on clean data resulted in a validation accuracy of 69% while our best result with 2% poison ratio results in 67% validation accuracy. The clean test accuracy drop with poison ratio 1% may be attributed to additional fine-tuning required or training for more than 200 epochs to achieve convergence. The target clean test accuracy is generally higher than the clean test accuracy which does highlight that the model has learnt target class better than the average case. An intuitive justification can be that since the model needs to additionally learn the class features that do not contain the trigger, the model training involved extra emphasis on learning them.

We also study the empirical convergence in Fig. 3 for the last column of Table. 2 - ResNet-18 poisoned (2%) with the Narcissus Backdoor Attack. This proves that our model is converging (ASR and clean test accuracy) and the results are stable.

5 Discussion and Conclusion

The main takeaway from the project is that for larger poison ratios, DP-SGD fails to act as a successful defense. It is obvious that a model trained with DP-SGD has poor utility in comparison with the SOTA. However, we do find that the original paper's experiments stand true and lower poison ratios are defended by Gradient Shaping. But as we can see from the results in Narcissus Backdoor, a stronger attack and a more recent attack can achieve higher ASR than any example in the original

paper with a low poison ratio (eg. 1%). It is fair to conclude that as attacks increasingly get better, we may find that an even smaller poison ratio can break the Gradient Shaping defense.

A secondary takeaway from this project is that breaking the Gradient Shaping defense is helped by the clean-label nature of attack. Since the trigger does not change the semantic similarity of its gradient with that of a clean sample's gradients, a clean-label attack is harder to defend using Gradient Shaping.

While it is hard to recommend DP-SGD as a generally robust solution, it does help performance with smaller poison ratios which is a more realistic scenario. If we can achieve better model performance using DP-SGD, it has the potential to become a light-weight defense in the future.

References

[1] S. Hong, V. Chandrasekaran, Y. Kaya, T. Dumitraş, and N. Papernot, "On the effectiveness of mitigating data poisoning attacks with gradient shaping," *arXiv preprint arXiv:2002.11497*, 2020.

[2] M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang, "Deep learning with differential privacy," in *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, 2016, pp. 308–318.

[3] R. Shokri, M. Stronati, C. Song, and V. Shmatikov, "Membership inference attacks against machine learning models," in *2017 IEEE S&P*. IEEE, 2017, pp. 3–18.

[4] A. Turner, D. Tsipras, and A. Madry, "Clean-label backdoor attacks," 2018.

[5] N. Carlini and A. Terzis, "Poisoning and backdooring contrastive learning," in *ICLR*, 2021.

[6] P. W. Koh and P. Liang, "Understanding black-box predictions via influence functions," in *ICML*, 2017, pp. 1885–1894.

[7] T. Gu, K. Liu, B. Dolan-Gavitt, and S. Garg, "Badnets: Evaluating backdooring attacks on deep neural networks," *IEEE Access*, vol. 7, pp. 47 230–47 244, 2019.

[8] X. Chen, C. Liu, B. Li, K. Lu, and D. Song, "Targeted backdoor attacks on deep learning systems using data poisoning," *arXiv:1712.05526*, 2017.

[9] K. Doan, Y. Lao, W. Zhao, and P. Li, "Lira: Learnable, imperceptible and robust backdoor attacks," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 11 966–11 976.

[10] A. Shafahi, W. R. Huang, M. Najibi, O. Suciu, C. Studer, T. Dumitras, and T. Goldstein, "Poison frogs! targeted clean-label poisoning attacks on neural networks," in *NeurIPS*, vol. 31, 2018.

[11] Y. Zeng, M. Pan, H. A. Just, L. Lyu, M. Qiu, and R. Jia, "Narcissus: A practical clean-label backdoor attack with limited information," *arXiv:2204.05255*, 2022.

[12] A. Paudice, L. Muñoz-González, A. Gyorgy, and E. C. Lupu, "Detection of adversarial training examples in poisoning attacks through anomaly detection," 2018.

[13] B. Wang, Y. Yao, S. Shan, H. Li, B. Viswanath, H. Zheng, and B. Y. Zhao, "Neural cleanse: Identifying and mitigating backdoor attacks in neural networks," in *2019 IEEE S&P*, 2019, pp. 707–723.

[14] Y. Gao, C. Xu, D. Wang, S. Chen, D. C. Ranasinghe, and S. Nepal, "Strip: A defence against trojan attacks on deep neural networks," in *ACSAC '19*, 2019, p. 113–125.

[15] X. Xu, Q. Wang, H. Li, N. Borisov, C. A. Gunter, and B. Li, "Detecting ai trojans using meta neural analysis," in *2021 IEEE S&P*, 2021, pp. 103–120.

[16] Y. Zeng, S. Chen, W. Park, Z. Mao, M. Jin, and R. Jia, "Adversarial unlearning of backdoors via implicit hypergradient," in *ICLR*, 2022.

[17] H. Souri, M. Goldblum, L. Fowl, R. Chellappa, and T. Goldstein, "Sleeper agent: Scalable hidden trigger backdoors for neural networks trained from scratch," *arXiv:2106.08970*, 2021.

[18] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, 2016, pp. 770–778.

[19] A. Yousefpour, I. Shilov, A. Sablayrolles, D. Testuggine, K. Prasad, M. Malek, J. Nguyen, S. Ghosh, A. Bharadwaj, J. Zhao *et al.*, "Opacus: User-friendly differential privacy library in pytorch," *arXiv preprint arXiv:2109.12298*, 2021.