DOES DIFFERENTIAL PRIVACY DEFEAT DATA POISONING?

Matthew Jagielski
Northeastern University
jagielski.m@northeastern.edu

Alina Oprea
Northeastern University
a.oprea@northeastern.edu

ABSTRACT

Data poisoning attacks have attracted considerable interest, both from the practical and theoretical machine learning communities. Recently, following widespread adoption for its privacy properties, differential privacy has been proposed as a defense from data poisoning attacks. In this paper, we show that the connection between poisoning and differential privacy is more complicated than it would appear. We argue that differential privacy itself does not serve as a defense, but that differential privacy benefits from robust machine learning algorithms, explaining much of differential privacy's success against poisoning.

1 INTRODUCTION

Modern machine learning applications involve large scale data collection, where a large amount of data is collected from a variety of sources. For example, Google trains word prediction models by allowing Android phones to send updates produced with local user data (Yang et al., 2018). The large scale of the data collection makes verifying the data's trustworthiness an impossible task. As a result, adversaries can add maliciously crafted training data to corrupt the model learned from the training data, in what is known as a poisoning attack. Poisoning attacks have been studied for a variety of models and applications, including linear regression, support vector machines, and logistic regression, and significant effort has been made in developing robust machine learning algorithms to defend against poisoning attacks. Indeed, recently, poisoning attacks were identified as the most worrisome threat among a survey of industry machine learning professionals (Kumar et al., 2020).

Independently, interest in protecting the privacy of users who appear in training data has spurned an interest in privacy-preserving data analysis in general, and privacy preserving machine learning in particular. Differential privacy (Dwork et al., 2006) has risen as the primary approach for ensuring users' privacy, having been used in the US Census (Haney et al., 2017), at Google (Erlingsson et al., 2014; Bittau et al., 2017), and at Apple (Thakurta et al., 2017). An algorithm satisfying differential privacy, informally, means that no adversary can learn more about a single data point than could have been learned if it was not contained in the dataset. Differential privacy's guarantee has natural ramifications for data poisoning—if the data poisoning attack is reasonably small, then differential privacy cannot reveal that the poisoning attack has occurred. This connection has motivated recent work to explore the extent to which differential privacy can defend against poisoning attacks (Ma et al., 2019; Hong et al., 2020). The goal of our work is to understand this connection more deeply.

To explore the connection, we introduce multiple metrics that a defense from poisoning should satisfy. The first metric, vulnerability difference, measures how vulnerable the defense is compared to an undefended model. The second metric, friendly fire, measures how much the defense compromises the performance of the model outside of the attack's target. We show that differential privacy provides surprisingly little benefit for these two goals: on MNIST, logistic regression trained with objective perturbation (Chaudhuri et al., 2011) becomes more vulnerable to poisoning and has larger friendly fire as the model becomes more private. However, to reconcile the robustness observed by prior work with our negative results, we notice that gradient clipping offers some robustness. We also discuss the implications of these observations for both poisoning and privacy research.

2 Background

Differential Privacy. Differential privacy is a formal guarantee of data privacy, defined as follows:

Definition 2.1. (Dwork et al., 2006). An algorithm $\mathcal{A} : \mathcal{D} \mapsto \mathcal{R}$ is (ε, δ)-*differentially private* if for any two datasets D_0, D_1 which differ on at most one row, and every set of outputs $\mathcal{O} \subseteq \mathcal{R}$:

$$\Pr[\mathcal{A}(D_0) \in \mathcal{O}] \leq e^{\varepsilon} \Pr[\mathcal{A}(D_1) \in \mathcal{O}] + \delta, \tag{1}$$

where probabilities are taken only over the randomness of $\mathcal{A}$.

Essentially, modifying one row of the dataset will not heavily modify the output of the algorithm, meaning that no one row heavily.

Lemma 1 (Group Privacy)**.** Let D_0, D_1 be two datasets differing on at most k rows, $\mathcal{A}$ is an (ε, δ)-differentially private algorithm, and $\mathcal{O}$ an arbitrary output set. Then

$$\Pr[\mathcal{A}(D_0) \in \mathcal{O}] \leq e^{k\varepsilon} \Pr[\mathcal{A}(D_1) \in \mathcal{O}] + \tfrac{e^{k\varepsilon}-1}{e^{\varepsilon}-1} \cdot \delta. \tag{2}$$

Group privacy provides a qualitatively similar guarantee to the differential privacy definition which holds for larger groups. This property has been used to justify differential privacy as a defense for poisoning attacks in Ma et al. (2019), which we discuss in the related work in the Appendix.

Poisoning Attacks. Poisoning attacks on machine learning insert or modify rows of a dataset, to manipulate the model produced from that dataset. For this work, we consider an adversary who has the ability to modify k rows of the dataset arbitrarily. This allows compatibility with the definition of differential privacy. The adversary replaces points to satisfy their *poisoning objective*, a measurement of how successful the attack is, while maintaining a low *collateral damage*, i.e. the model's performance should not change significantly outside of those points relevant to the poisoning objective. There is a wide range of objectives considered in the poisoning literature (we refer the interested reader to Jagielski et al. (2020a) for a more thorough description of poisoning objectives) - for this paper, we focus on an adversary who wishes to misclassify a single test point, which suffices for our preliminary investigation. This is often referred to as a *targeted* poisoning attack. We consider a simple adversary, who, in order to misclassify a given target point, adds k mislabeled examples of that point (for our experiments, $k = 5$). That is, to target some sample (x, y), we add $k = 5$ copies of $(x, 1 - y)$.

3 What's in a defense?

The remainder of this paper considers the ability of differential privacy to defend against poisoning. Before we go further, we must specify what it means to "defend against" a poisoning attack, and what it means to use "differential privacy" as a defense.

For the former, we assert that a defense from a poisoning attack should satisfy two criteria. First, it should reduce the poisoning objective, which we refer to as vulnerability. Second, it should not cause friendly fire, a metric we introduce. Friendly fire happens either by damaging the classifier when no poisoning is happening, or harming the performance of the classifier outside of the target of the attack. Friendly fire can be measured on any region of input space—it is possible the defense harms specific regions of input space disproportionately. For example, a defense which ignores high loss points may perform similarly overall but damage poorly supported regions of the training distribution. The success of a defense is the extent to which both vulnerability and friendly fire are low. We define these metrics concretely, where A is the original learning algorithm, A^D is the proposed defensive learning algorithm, D_0 is the unpoisoned dataset, D_1 is a poisoned dataset, $\textsc{Obj}$ is a measurement of the effectiveness of the poisoning attack, ℓ is a loss function, T is a constant, and D^* is a specific subset of data which may be impacted by the defense:

$$\mathrm{V}_{\textsc{Obj}}(D_0, D_1; A^D, A) = \frac{\textsc{Obj}(A^D(D_1)) - \textsc{Obj}(A(D_0))}{\textsc{Obj}(A(D_1)) - \textsc{Obj}(A(D_0))} = 1 + \frac{\textsc{Obj}(A^D(D_1)) - \textsc{Obj}(A(D_1))}{\textsc{Obj}(A(D_1)) - \textsc{Obj}(A(D_0))} \tag{3}$$

$$\mathrm{FF}(D^*, T; D_b, A^D, A) = \Pr[\ell(D^*; A^D(D_b)) - \ell(D^*; A(D_b)) > T] \tag{4}$$

We allow $D_b \in \{D_0, D_1\}$, as friendly fire is important both when poisoned and unpoisoned. We use T to understand the probability an attack causes significant damage. Friendly fire should ideally be low for any meaningful setting of D^* and T. Because learning algorithms are randomized, we measure the expected value of each term in V_{OBJ}. Notice that V_{OBJ} is defined such that it is 0 when the defense, trained on poisoned data, recovers the unpoisoned objective value of the undefended model, and is 1 when the defense is equally vulnerable to the attack as when no defense is applied. The ideal (albeit unrealistic) defense, which perfectly identifies and trains without poisoning, would achieve a robustness of 0 and friendly fire of 0 for all D^*, $T > 0$ and both $D_b \in \{D_0, D_1\}$.

Differential privacy as a defense? By analyzing multiple differentially private algorithms, we show that simply applying differential privacy does not provide protection from poisoning. We do so by showing that the objective perturbation algorithm for private logistic regression actually is worse than nonprivate logistic regression in its robustness and friendly fire. However, this does not preclude private algorithms from being robust for other reasons, as in Hong et al. (2020).

4 No: Objective Perturbation

In this section, we will show that private logistic regression trained using objective perturbation does not perform well as a defense, using the metrics introduced in Section 3. We use objective perturbation because $(\varepsilon, 0)$-differentially private objective perturbation recovers standard training as $\varepsilon \to \infty$, allowing us to isolate the impact of differential privacy.

We use two datasets - the synthetic two-dimensional dataset used in Ma et al. (2019) and MNIST. We modify the synthetic dataset so all data points have an ℓ_2 norm of 1, making the dataset uniformly distributed on the unit circle. We use the classes 3 and 5 from the MNIST dataset, also normalized so data has an ℓ_2 norm of 1. For both datasets, we randomly select 100 data points x_t, y_t from the test set to poison, converting the unpoisoned D into a poisoned D_p by replacing 5 rows from the training set with $x_t, 1 - y_t$. We attack one point at a time to ensure poisoning attacks do not interfere with each other. Over the 100 data points, we report the 25th, 50th, and 75th percentile of the robustness and friendly fire, using 100 trials of Monte Carlo simulation to compute these metrics. We use cross-entropy loss to compute both robustness and vulnerability. As a result, the robustness measures the factor by which cross entropy increases on a target point, while friendly fire computes the probability that a given point will be have its loss increased due to differential privacy.

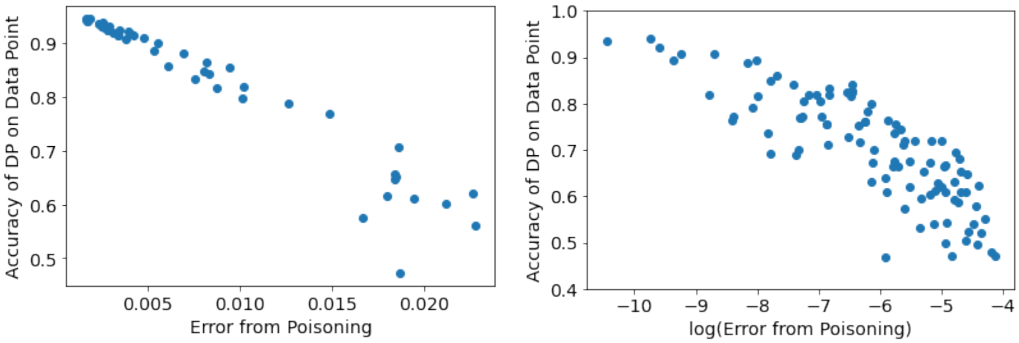
Dataset	ε	Acc	ε-DP Acc	XE V_{OBJ}			XE FF - $T = 0.1$		
				25th	50th	75th	25th	50th	75th
Synth	0.5	1.0	0.920	3.81	4.32	5.30	0.24	0.31	0.39
Synth	1.0		0.957	1.74	2.00	2.34	0.08	0.11	0.23
Synth	2.0		0.975	1.06	1.16	1.27	0.00	0.03	0.1
MNIST	0.5	1.0	0.914	10.41	27.28	88.02	0.31	0.37	0.41
MNIST	1.0		0.949	9.14	16.52	57.76	0.23	0.28	0.345
MNIST	2.0		0.972	7.07	9.23	41.87	0.15	0.19	0.30

Table 1: Accuracy (Acc), robustness, and friendly fire for ε-differentially private objective perturbation. XE=cross-entropy loss, FF is measured with $D_b = D_0$. Lower V_{OBJ} is better - a perfect defense achieves 0 V_{OBJ} and not applying any defense achieves 1 V_{OBJ}. Lower FF is better - both a perfect defense and standard training achieve 0 FF.

In Table 1, we document the accuracy of nonprivate and private models, and the robustness and friendly fire of differential privacy. For both datasets, differentially private models are less robust than nonprivate models; nonprivate models achieve $\mathrm{V}_{\text{OBJ}} = 1$. Furthermore, vulnerability and friendly fire both increase as training becomes more private. Notice, too, that the accuracy of the models decrease significantly as ε decreases. While it is well known that differential privacy harms utility, this is important to be careful of for poisoning robustness—if a point is misclassified, it doesn't matter whether it was due to poisoning or privacy.

To reinforce this point, we will now show that, with differential privacy, those points which are most easily targeted by poisoning attacks are also those points which are most harmed by differential privacy, casting further doubt on the ability to simply apply differential privacy as a poisoning countermeasure. To do so, we measure the error from poisoning (the denominator of Equation 3) and the accuracy of objective perturbation on a data point (the fraction of private models which correctly classify the point, equivalent to FF with the 0-1 loss) for 100 poisoning points. We run objective perturbation 250 times to compute the private accuracy. We show in Figure 1 that these values are highly correlated. The Pearson's R measures the correlation as -0.96 for the synthetic dataset and -0.83 on MNIST, which is very strong correlation in both cases. Very reliably, the points most heavily impacted by poisoning are also those points which are most frequently misclassified by privacy, giving indication that differential privacy should not itself be used as a defense.

Notice that this our results here do not contradict results of Ma et al. (2019), which shows that there is little difference between poisoned and unpoisoned models both trained with differential privacy. This is because they do not compare against models trained nonprivately, which better informs the decision to use differential privacy in practice for robustness.



(a) Synthetic (Ma et al., 2019), $\varepsilon = .5$

(b) MNIST, $\varepsilon = 1$

Figure 1: Correlation between error induced by privacy and one poisoning point with objective perturbation on two datasets. Those points which are more impacted by data poisoning are also more impacted by differential privacy, with a Pearson's R of -0.96 on synthetic data, and -0.83 on MNIST. Due to MNIST's larger dataset size, single point poisoning attacks are less effective, so we plot and measure Pearson's R of the poisoning error in log scale.

5 Yes: Robust Machine Learning and Differential Privacy

Despite the negative statement of the previous section, it is still possible that private algorithms can defend against poisoning. The connection, though, must be more subtle: it cannot come directly from differential privacy. We observe that a long line of work in differential privacy has leveraged robust algorithms to reduce sensitivity, as we note in the Appendix. Results such as Hong et al. (2020), indicating DP-SGD's robustness to attack, may simply come from the robustness of the underlying gradient clipping.

5.1 Clipped Gradient Descent

We experiment here with only clipped gradient descent, without adding noise required to impose differential privacy. We present the full algorithm for DP-SGD and our hyperparameter settings in the Appendix, but here, it suffices to note that the clipping norm parameter, C, governs how much each data point contributes to the model—a smaller clipping norm diminishes the influence of any one point. We show that clipped gradient descent offers some protection from data poisoning[1]. We do this by evaluating the vulnerability and friendly fire of clipped gradient descent with multiple values of the clipping norm C, following a similar experimental setup to Section 4, using only MNIST.

We present the results of this experiment in Table 2, varying the clipping norm C. Notice that we use $T = 0.01$ rather than $T = 0.1$ in this section due to the much smaller friendly fire when clipping

[1] This cannot be universally true, as the clipping norm can be set large enough to be irrelevant. However, in practice, this parameter setting is unlikely, as it would result in too much noise being added during training.

Dataset	C	No Clip Acc	Clip Acc	XE $\mathrm{V_{OBJ}}$			XE FF - $T = 0.01$		
				25th	50th	75th	25th	50th	75th
MNIST	0.05	0.957	0.880	1.09	5.67	21.06	0	0	0
MNIST	0.1		0.930	0.01	1.95	5.00	0	0	0.11
MNIST	0.2		0.947	-0.50	0.24	2.08	0	0	0.84
MNIST	0.4		0.953	-0.57	0.94	3.50	0	0.03	0.36

Table 2: Accuracy of models trained without clipping (No Clip Acc), with clipping (Clip Acc), and robustness and friendly fire for clipped gradient descent. XE=cross-entropy loss, FF is measured with $D_b = D_0$. Lower $\mathrm{V_{OBJ}}$ is better - a perfect defense achieves 0 $\mathrm{V_{OBJ}}$ and standard training achieves 1 $\mathrm{V_{OBJ}}$. Lower FF is better - both a perfect defense and standard training achieve 0 FF.

is applied. Even then, the vast majority of points do not see this decrease in loss. Clipped gradient descent performs significantly better in terms of friendly fire compared to differential privacy. Notice that, while clipped gradient descent is not robust in every case, it is more robust (that is, $\mathrm{V_{OBJ}} < 1$) than standard SGD in 45% of cases with $C = 0.1$, 60% of cases with $C = 0.2$, and 50% of cases with $C = 0.4$. While clipped gradient descent is not successful all of the time, it is often successful and comparing with Table 1 shows that it is significantly more successful than differential privacy, which always had $\mathrm{V_{OBJ}} > 1$.

6 Conclusion

In this paper, we argued that differential privacy itself is not a defense from poisoning attacks. While differential privacy has strong formal guarantees, it frequently makes models more vulnerable to attack, induces damaging friendly fire, and its damage to utility directly mirrors the damage done by poisoning attacks. Recent successes in defending against attacks with private training likely come from the underlying training algorithm being robust. Our results would likely be magnified in a federated learning context, where clipping is done more aggressively, per user rather than per data point. More work is necessary to understand clipped SGD's robustness, as it is unclear how the clipping interacts with stronger attacks and attacks with different goals, such as the clipping-aware attacks proposed by Jagielski et al. (2020b).

The overall statement of this paper, however, is not that differential privacy should *not* be used as a defense for poisoning. After all, as discussed, private mechanisms can leverage robustness to improve utility, and the underlying robustness can be advantageous to preventing poisoning attacks. And of course, many settings require both privacy and security, in which any robustness benefits of private algorithms comes for free. It is also possible that DP-SGD exhibits other properties that cannot be explained by the guarantees of differential privacy alone, such as the clipping or noise impacting optimization (Song et al., 2020; Neelakantan et al., 2015), which we briefly discuss in the related work found in the Appendix.

Our work highlights the importance of poisoning research to attempt stealthy, defense-aware attacks. Additionally, defenses must measure all relevant metrics: robustness and friendly fire should both be low, and if they are not, the tradeoffs between robustness and friendly fire must be well documented.

Acknowledgments

This research was sponsored by the U.S. Army Combat Capabilities Development Command Army Research Laboratory under Cooperative Agreement Number W911NF-13-2-0045 (ARL Cyber Security CRA). The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Combat Capabilities Development Command Army Research Laboratory or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation.

REFERENCES

Daniel Alabi, Audra McMillan, Jayshree Sarathy, Adam Smith, and Salil Vadhan. Differentially private simple linear regression. *arXiv preprint arXiv:2007.05157*, 2020.

Raef Bassily, Adam Smith, and Abhradeep Thakurta. Private empirical risk minimization: Efficient algorithms and tight error bounds. In *2014 IEEE 55th Annual Symposium on Foundations of Computer Science*, pp. 464–473. IEEE, 2014.

Sourav Biswas, Yihe Dong, Gautam Kamath, and Jonathan Ullman. Coinpress: Practical private mean and covariance estimation. *arXiv preprint arXiv:2006.06618*, 2020.

Andrea Bittau, Úlfar Erlingsson, Petros Maniatis, Ilya Mironov, Ananth Raghunathan, David Lie, Mitch Rudominer, Ushasree Kode, Julien Tinnes, and Bernhard Seefeld. Prochlo: Strong privacy for analytics in the crowd. In *Proceedings of the 26th Symposium on Operating Systems Principles*, pp. 441–459, 2017.

Mark Bun and Thomas Steinke. Average-case averages: Private algorithms for smooth sensitivity and mean estimation. In *Advances in Neural Information Processing Systems*, pp. 181–191, 2019.

Kamalika Chaudhuri, Claire Monteleoni, and Anand D Sarwate. Differentially private empirical risk minimization. *Journal of Machine Learning Research*, 12(Mar):1069–1109, 2011.

Min Du, Ruoxi Jia, and Dawn Song. Robust anomaly detection and backdoor attack detection via differential privacy. *arXiv preprint arXiv:1911.07116*, 2019.

Cynthia Dwork and Jing Lei. Differential privacy and robust statistics. In *Proceedings of the forty-first annual ACM symposium on Theory of computing*, pp. 371–380, 2009.

Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of cryptography conference*, pp. 265–284. Springer, 2006.

Úlfar Erlingsson, Vasyl Pihur, and Aleksandra Korolova. Rappor: Randomized aggregatable privacy-preserving ordinal response. In *Proceedings of the 2014 ACM SIGSAC conference on computer and communications security*, pp. 1054–1067, 2014.

Samuel Haney, Ashwin Machanavajjhala, John M Abowd, Matthew Graham, Mark Kutzbach, and Lars Vilhuber. Utility cost of formal privacy for releasing national employer-employee statistics. In *Proceedings of the 2017 ACM International Conference on Management of Data*, pp. 1339–1354, 2017.

Naoise Holohan, Stefano Braghin, Pól Mac Aonghusa, and Killian Levacher. Diffprivlib: the ibm differential privacy library. *arXiv preprint arXiv:1907.02444*, 2019.

Sanghyun Hong, Varun Chandrasekaran, Yiğitcan Kaya, Tudor Dumitraş, and Nicolas Papernot. On the effectiveness of mitigating data poisoning attacks with gradient shaping. *arXiv preprint arXiv:2002.11497*, 2020.

Matthew Jagielski, Giorgio Severi, Niklas Pousette Harger, and Alina Oprea. Subpopulation data poisoning attacks. *arXiv preprint arXiv:2006.14026*, 2020a.

Matthew Jagielski, Jonathan Ullman, and Alina Oprea. Auditing differentially private machine learning: How private is private sgd? *arXiv preprint arXiv:2006.07709*, 2020b.

Ram Shankar Siva Kumar, Magnus Nyström, John Lambert, Andrew Marshall, Mario Goertzel, Andi Comissoneru, Matt Swann, and Sharon Xia. Adversarial machine learning-industry perspectives. In *2020 IEEE Security and Privacy Workshops (SPW)*, pp. 69–75. IEEE, 2020.

Yuzhe Ma, Xiaojin Zhu, and Justin Hsu. Data poisoning against differentially-private learners: Attacks and defenses. *arXiv preprint arXiv:1903.09860*, 2019.

Arvind Neelakantan, Luke Vilnis, Quoc V Le, Ilya Sutskever, Lukasz Kaiser, Karol Kurach, and James Martens. Adding gradient noise improves learning for very deep networks. *arXiv preprint arXiv:1511.06807*, 2015.

Kobbi Nissim, Sofya Raskhodnikova, and Adam Smith. Smooth sensitivity and sampling in private data analysis. In *Proceedings of the thirty-ninth annual ACM symposium on Theory of computing*, pp. 75–84, 2007.

Shuang Song, Kamalika Chaudhuri, and Anand D Sarwate. Stochastic gradient descent with differentially private updates. In *2013 IEEE Global Conference on Signal and Information Processing*, pp. 245–248. IEEE, 2013.

Shuang Song, Om Thakkar, and Abhradeep Thakurta. Characterizing private clipped gradient descent on convex generalized linear problems. *arXiv preprint arXiv:2006.06783*, 2020.

Ziteng Sun, Peter Kairouz, Ananda Theertha Suresh, and H Brendan McMahan. Can you really backdoor federated learning? *arXiv preprint arXiv:1911.07963*, 2019.

Abhradeep Guha Thakurta, Andrew H Vyrros, Umesh S Vaishampayan, Gaurav Kapoor, Julien Freudiger, Vivek Rangarajan Sridhar, and Doug Davidson. Learning new words, March 14 2017. US Patent 9,594,741.

Timothy Yang, Galen Andrew, Hubert Eichner, Haicheng Sun, Wei Li, Nicholas Kong, Daniel Ramage, and Françoise Beaufays. Applied federated learning: Improving google keyboard query suggestions. *arXiv preprint arXiv:1812.02903*, 2018.

A Related Work

A.1 Poisoning and Privacy

Ma et al. (2019) show the following result bounding the impact of poisoning on private algorithms:

Theorem 2. (Ma et al., 2019) Let D be an unpoisoned dataset and D_p a poisoned dataset differing from D on at most k rows. If C is a positive cost function representing the effectiveness of the poisoning attack and $\mathcal{A}_\varepsilon$ is an ε-differentially private algorithm, then

$$\mathbb{E}[C(\mathcal{A}_\varepsilon)(D)] \geq \exp(-k\varepsilon)\mathbb{E}[C(\mathcal{A}_\varepsilon)(D_p)].$$

Our work argues that this theorem is ineffective at deciding whether or not to apply differential privacy to protect against poisoning, as it does not compare a private model to a nonprivate model. As such, it cannot capture friendly fire or robustness. The theorem is more useful in situations when differential privacy *must* be applied, perhaps for regulatory reasons, and one seeks to understand how robust it will be.

Hong et al. (2020) show that DP-SGD can be used as a defense from some poisoning attacks, which we corroborate in Section 5. They also find that it is ineffective against availability poisoning attacks (where the poisoning attacks are large), and can lead to decreased accuracy. It is worth noting that the theoretical results of Ma et al. Ma et al. (2019) are very weak for availability attacks. We show that the utility decrease of differential privacy is connected to poisoning, by showing that easy to poison examples are also most heavily impacted by differential privacy's utility cost.

DP-SGD has also been considered as a defense from poisoning in federated learning (Sun et al., 2019). Interestingly, the experimental results of Sun et al. (2019) show that differential privacy provides robustness beyond that which is offered by clipping. While it is difficult to draw conclusions about randomized algorithms from small numbers of trials, it is possible that the observed effect is the result of an undocumented impact of DP-SGD on the optimization process. As we show, this effect does not appear with logistic regression trained with objective perturbation.

Differential privacy has also been proposed to aid in anomaly detection by Du et al. (2019), which is similar to poisoning robustness. Interestingly, they find that models trained privately do offer benefit over nonprivately trained models, when used for backdoor and anomaly detection. Our paper argues that more private models should not provide this benefit, a phenomenon that should be untangled by future work. The most notable difference between the papers is that Du et al. (2019) uses much larger models. It is possible that training with privacy interacts with neural network training and linear model training differently. Future work should investigate this disparity.

Poisoning attacks have been proposed to audit differentially private models by Jagielski et al. (2020b). Essentially, a successfully poisoned model reveals the presence of poisoning in the dataset, causing privacy leakage. This strategy would not be possible if differential privacy was perfectly robust to poisoning.

A.2 Robust Algorithms and Differential Privacy

That DP-SGD is robust to poisoning attacks is reminiscent of the large body of work connecting differential privacy to robust statistics. Early work in this direction has shown how to compute robust statistics privately. This includes Nissim et al. (2007), who introduce smooth sensitivity framework and the Sample-and-Aggregate algorithm. These approaches avoid the requirement for noise to grow proportionally with global sensitivity. Dwork & Lei (2009) show how to use Propose-Test-Release to privately compute general robust estimators. Several works show that private algorithms can benefit from using robust algorithms. For single dimension means, for example, Bun & Steinke (2019) use the trimmed mean to avoid dependence on the global sensitivity. Biswas et al. (2020) also use the trimmed mean to achieve practical high dimensional mean estimation. For linear regression, Alabi et al. (2020) empirically demonstrate that robust linear regression algorithms outperform global sensitivity-based algorithms.

B Supplement for Section 4

In Algorithm 1, we present the objective perturbation algorithm from Chaudhuri et al. (2011).

Algorithm 1: Objective Perturbation (Chaudhuri et al., 2011)

```
Data: Dataset $X, Y$, privacy $\varepsilon$, regularization value $\lambda$, loss $\ell$, constant $c$
Function ObjPert($X, Y, \varepsilon, \lambda, \ell, c$):
    $\varepsilon' = \varepsilon - \log(1 + \frac{2c}{n\lambda} + \frac{c^2}{n^2\lambda^2})$
    If $\varepsilon' > 0$
        $\Delta = 0$
    Else
        $\Delta = \frac{c}{n(\exp(\varepsilon/4)-1)} - \lambda$
        $\varepsilon' = \varepsilon/2$
    $b \sim \exp(-b\varepsilon'/2)$
    $L(w) = \frac{1}{n}\sum_i \ell(x_i, y_i; w) + \lambda||w||_2^2 + \frac{1}{n}\mathbf{b}^T w$
    return $\arg\min_w(L(w) + \frac{1}{2}\Delta||w||_2^2)$
```

We use the implementation provided by diffprivlib (Holohan et al., 2019). We use 20 poisoning points and 100 trials to compute V_{OBJ} and FF for both datasets. We use 5 poisoning points, which are constructed by simply taking the target point x_t, y_t and flipping its label to $x_t, 1 - y_t$ before adding 5 copies of it to the training set.

C Supplement for Section 5

In Algorithm 2, we present the differentially private stochastic gradient descent algorithm from Song et al. (2013). We use a noise multiplier of 0 in our experiments, as we attempt to understand the impact of clipping alone. We use a batch size of 250, learning rate of 0.5, momentum of 0.9, and train for $T = 20$ epochs. We use the implementation found in pytorch-dp[2]. Because training with DP-SGD is more computationally expensive than objective perturbation, we use only 25 trials to compute V_{OBJ} and FF, and continue to use 5 poisoning points and 20 different target points.

Algorithm 2: Differentially Private Stochastic Gradient Descent (Song et al., 2013; Bassily et al., 2014)

```
Data: Dataset $X, Y$, loss $\ell$, inital parameters $w_0$, learning rate $\eta$, batch size $b$, iterations $T$,
      noise magnitude $\sigma$, clipping norm $C$
Function DPSGD($X, Y, \ell, w_0, \eta, b, T, \sigma, C$):
    For $i \in [T]$
        $G = 0$
        For $(x_i, y_i) \in$ random batch of $b$ elements from $X, Y$
            $g = \nabla\ell(w_0; x_i, y_i)$
            $G = G + \frac{\min(C,||g||_2)}{b||g||_2}g$
        $w_i = w_{i-1} - \eta(G + \mathcal{N}(0, (C\sigma)^2\mathbb{I}))$
    return $w_T$
```

[2] https://github.com/pytorch/opacus