Practical Adversarial Training with Differential Privacy for Deep Learning

Anonymous authors
Paper under double-blind review

Abstract

Deep learning models are often vulnerable to privacy risks and adversarial attacks, rendering them un-trustworthy on crowd-sourced tasks. However, these risks are rarely resolved jointly, despite the fact that there are separate solutions in the security community and the privacy community. In this work, we propose the practical adversarial training with differential privacy (DP-Adv), to combine the backbones from both communities and deliver robust and private models with high accuracy. Our algorithm is concise by design and capable of taming technical advances from both communities into our framework. For example, DP-Adv works with all existing DP optimizers and attacking methods off-the-shelf. In particular, DP-Adv is as private as non-robust DP training, and as efficient as non-DP adversarial training. Our experiments on multiple datasets show that DP-Adv outperforms state-of-the-art methods that preserve robustness and privacy simultaneously. Furthermore, we observe that adversarial training and DP can notably worsen the calibration, but the mis-calibration can be mitigated by pre-training.

1 Introduction

Deep learning models have demonstrated amazing performance in classification problems, especially on the computer vision and natural language processing tasks. However, these neural networks are known to be vulnerable to privacy risks and adversarial attacks: an adversary may extract sensitive information from the training data and/or use small perturbations to fool the model's prediction. These urgent risks have prohibited people to truly trust deep learning models in the sense of both privacy and security.

On one hand, many privacy attacks have been studied to demonstrate the risk of leaking private information. For example, an adversary can conduct membership inference attack (Shokri et al., 2017; Carlini et al., 2019) to guess whether a data point belongs to the training data, e.g. a dataset collecting information from patients with a specific disease. Therefore, knowing whether a person's medical record belongs to this dataset implies whether this person has such disease. Another concerning attack is the extraction attack (Carlini et al., 2020), including re-identifying anonymized users in a dataset (e.g. the anonymized MovieLens dataset is re-identified and leads to the cancellation of the second NetFlix Prize competition (Narayanan & Shmatikov, 2006)) and extracting the training features (e.g. name, address, phone number from the generative model GPT-2 (Radford et al., 2019)).

On the other hand, various adversarial attacks have posed significant threat on the robustness of deep learning models. For example, the projected gradient descent (PGD) attack by Madry et al. (2017) can worsen an advanced network, ResNet50, from 95% accuracy to around 8% using a small perturbation of L_∞ magnitude 0.25 on CIFAR10, as well as from 76% accuracy to around 3% using a perturbation of L_2 magnitude 0.5 on ImageNet. In fact, Su et al. (2019) shows that sometimes the adversary only needs to perturb one pixel out of the hundreds in an image to make the neural network predict wrongly.

To protect the privacy with a theoretical guarantee, one line of work proposes to use **differential privacy** (DP) (Dwork et al., 2006), a gold standard in the privacy regime. In the seminal work by Abadi et al. (2016), differential privacy has been applied to deep learning by

leveraging DP optimizers, e.g. DP-SGD or DP-Adam, and achieved strong performance in later work that refines the DP learning (Bu et al., 2020; Papernot et al., 2020). As an example, non-DP models can achieve test accuracy over 95% on CIFAR10 and DP models achieve around 60% without pre-training (or $> 70\%$ with pre-training on CIFAR100). In particular, DP deep learning uses the same training procedure and neural network architecture, and the only difference is the use of DP optimizers instead of regular optimizers. Algorithmically speaking, DP optimizers are randomized by adding independent Gaussian noises to clipped gradients (see Algorithm 1 for details). Therefore, DP deep learning is different from the regular learning in terms of the *optimizer* of the same optimization problem.

To protect the *adversarial robustness*, another line of work studies the **adversarial training** (Kurakin et al., 2016; Madry et al., 2017) with the adversarial examples, which is nowadays the workhorse in the robustness community. The intuition is straight-forward: instead of training on the original but vulnerable data, one can train the neural network on the adversarial examples that are known to cause robustness issues, thus expecting the neural network to become immune to the adversarial attacks. Therefore, we can view the adversarial examples as a type of data *pre-processing*. In fact, adversarial training is different from the regular learning in terms of the *objective function* in the optimization problem.

Contributions To develop truly safe deep learning models, it is necessary to preserve both privacy and security simultaneously. Yet, most existing researches work on the two problems separately. We take one step further towards this goal (see Table 1 and Figure 1):

- We propose a unified, flexible and practical framework — the `DP-Adv` training in Algorithm 1. Our approach can incorporate any advances in both the adversarial robustness community (e.g. new defense methods or faster adversarial training) and the differential privacy community (e.g. new privacy accountant or DP optimizers).
- Our `DP-Adv` method is compatible with existing optimizers for both the outer minimization and inner maximization in (3.1). In addition, `DP-Adv` can be efficiently trained, adding little overhead to speed or memory in comparison to adversarial training and DP learning. This covers many deep learning tasks such as multi-label classification, data generation and federated learning.
- We conduct a rigorous DP analysis for our `DP-Adv` method and show it is exactly as private as traditional DP learning.
- Numerous experiments demonstrate that `DP-Adv` outperforms state-of-the-art methods that preserve both privacy and robustness. Especially, we give the first empirical study of calibration issue in the private and robust regime.

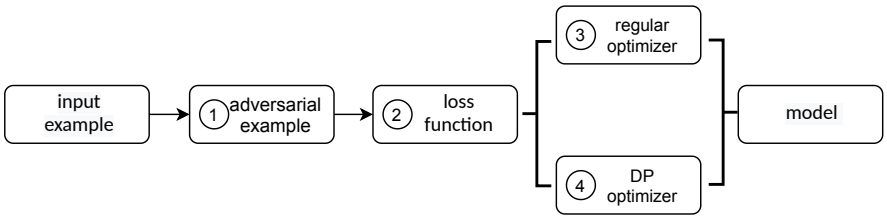

Figure 1: Flow charts of regular training (②③), differentially private training (②④), adversarial training (①②③) and `DP-Adv` training (①②④).

Trade-off between differential privacy and adversarial robustness Previous research has focused on either testing the privacy risk of adversarially robust but non-DP models (Song et al., 2019) or testing the robustness of DP but non-robust models (Boenisch et al., 2021; Tursynbek et al., 2020; Han et al., 2021). A trade-off between privacy and robustness has been empirically observed. In the non-DP regime, Song et al. (2019) demonstrates that non-robust models can have an empirically low privacy risk while adversarially robust models are even much more vulnerable to privacy attacks, increasing a privacy attacker's accuracy by 25% on CIFAR10 than that of non-robust models. From the other end,

Boenisch et al. (2021) demonstrates empirically via MNIST dataset that DP models can be more vulnerable to adversarial attacks, when using large noise and large clipping norm: the attack success rate being $\approx 90\%$ for DP models and $\approx 40\%$ for non-DP models.

Nevertheless, it remains unclear how combining DP and adversarial robustness affects this trade-off. Especially, since it is well-known that the cost of preserving DP or robustness in deep learning is the accuracy, it is significant to understand whether such combination will lead to an accuracy too bad to be useful.

Connection between differential privacy and adversarial robustness To clear any possible confusion, we point out that a series of work including (Lecuyer et al., 2019; Li et al., 2019; Cohen et al., 2019) connects DP and adversarial robustness in a totally different manner than ours. To be specific, these work proposes the *randomized smoothing* algorithm to preserve certified adversarial robustness but such algorithm is not privacy-preserving. In fact, DP itself is not connected to adversarial robustness directly, but it motivates another notion 'pixelDP' (see Lecuyer et al. (2019), which does not protect the privacy of data) that leads to the certified robustness.

Models	Vanilla training	Adversarial training	DP training	`DP-Adv` training
Clean accuracy	High	High	High	High
Robust accuracy	Low	High	Low	High
Privacy protection	Low	Low	High	High
Computation Efficiency	High	Low	High	Low
Memory Efficiency	High	High	Low	Low

Table 1: Performance of different training procedures.

Concurrent but different work We acknowledge the `StoBatch` algorithm (Phan et al., 2020, restated in Algorithm 2) which simultaneously preserve DP and adversarial robustness, through adversarial training and DP training. `StoBatch` (1) adds Laplacian noise to the input as pre-processing (2) uses an auto-encoder to learn DP benign examples; (3) attacks the benign examples with a random perturbation bound γ to obtain DP adversarial examples; (4) privately trains on both DP benign and DP adversarial examples. The authors achieve mediocre robust accuracy under weak attacks: around 85% robust accuracy on MNIST with Fast Gradient Sign Method (FGSM, $\gamma = 0.2$). However, the clean accuracy is low (10% lower than our `DP-Adv`) and the performance degrades severely on CIFAR10 or strong attacks. See the comparison in Section 4. Another related work is `SecureSGD` (Phan et al., 2019), which provides adversarial robustness via the randomized smoothing of a DP model. I.e., `SecureSGD` is a non-adversarial-training-based method. This approach is empirically less competitive than `StoBatch`. In contrast to the state-of-the-art `StoBatch`, our `DP-Adv` training is fundamentally different by removing all unnecessary randomness to avoid harming the accuracy and to provide straight-forward privacy analysis: e.g. we need no auto-encoder and we only train on the adversarial examples, same as the modern adversarial training.

2 Background and Notations

In this section, we introduce the background knowledge and notations used throughout this paper. Consider n samples $x_i, i \in [n]$, and an arbitrary classifier $f(x;\theta) : \mathbb{R}^p \to \mathbb{R}^m$, where p is feature size (e.g. number of pixels in one image for computer vision) and m is the number of classes. This classifier is parameterized by its weights and biases θ and outputs the continuous logits from its input x. The actual classifier which outputs discrete classes is $F(x;\theta) \equiv \text{argmax}_k[f(x;\theta)]_k \in [m]$. True labels are denoted as y and the loss function as $\mathcal{L}$.

2.1 Adversarial Robustness

Definition 2.1. Given a model f, the adversarial example of $\boldsymbol{x}$ is $\boldsymbol{x} + \Delta^*$, where Δ^* is the optimal perturbation in the perturbation set U (e.g. $U = \{u \in \mathbb{R}^p : \|u\|_p \leq \gamma\}$ for $p = 0, 1, 2$

or ∞):

$$\Delta^*(\boldsymbol{x}) := \text{argmax}_{\Delta \in U} \mathcal{L}(f(\boldsymbol{x}+\Delta;\theta), y) \tag{2.1}$$

A model is adversarially robust against an attack on U if such attack fails as $F(\boldsymbol{x} + \Delta^*(\boldsymbol{x});\theta) = F(\boldsymbol{x};\theta)$. The robust (or adversarial) accuracy is $\sum_i \mathbb{I}(F(\boldsymbol{x}_i+\Delta^*(\boldsymbol{x}_i);\theta) = y_i)/n$ and the clean accuracy is $\sum_i \mathbb{I}(F(\boldsymbol{x}_i;\theta) = y_i)/n$.

Adversarial training, depicted in Algorithm 1 by removing step 6 and 8, is the backbone of learning robust deep neural networks (Kurakin et al., 2016; Tramèr et al., 2017; Madry et al., 2017; Athalye et al., 2018). We highlight that some sub-areas in adversarial training, including new attackers and faster computation (Shafahi et al., 2019), are compatible with our `DP-Adv` framework.

2.2 Differential Privacy

Differential privacy (Dwork et al., 2006) is a strong and mathematically rigorous guarantee against privacy risks via randomized mechanisms.

Definition 2.2. A randomized algorithm M is (ϵ, δ)-DP, if for any two neighboring datasets S, S' that differ in an arbitrary single sample and for any event E,

$$\mathbb{P}[M(S) \in E] \leq \exp(\epsilon)\mathbb{P}\left[M\left(S'\right) \in E\right] + \delta.$$

In words, adding or removal an arbitrary sample from the dataset S has indistinguishable effect on the final parameters of the neural networks. Needless to say, smaller (ϵ, δ) is preferred. Algorithmically, DP optimizers add independent Gaussian noise to the gradients in order to achieve DP. This is known as the Gaussian mechanism.

Lemma 2.3 (Theorem A.1 Dwork et al. (2014))**.** *The ℓ_2 **sensitivity** of g is defined as $\Delta g = \sup_{S,S'} \|g(S) - g(S')\|_2$ over all pairs of neighboring (S, S'). The **Gaussian mechanism** $\hat{g}(S) = g(S) + \sigma\Delta g \cdot \mathcal{N}(0, \mathbf{I})$ is $(\epsilon(\sigma, n, \delta), \delta)$-DP, depending on the privacy accountant.*

DP training, depicted in Algorithm 1 by removing step 4, has been widely applied in the deep learning community, including image datasets (Abadi et al., 2016; Zhang et al., 2021), NLP tasks, and federated learning (McMahan et al., 2017). We highlight that new techniques in DP can be seamlessly merged into `DP-Adv`, e.g. new privacy accountants (Dong et al., 2021; Gopi et al., 2021; Zhu et al., 2021) and new DP optimizers (Bu et al., 2021b;a).

3 Differentially private adversarial training

3.1 DP-Adv optimization problem

Since pre-processing of data (or data augmentation) does not affect the privacy of models, `DP-Adv` in fact solves exactly the same optimization problem as the traditional adversarial training (Madry et al., 2017, Equation (2.1)), training only on adversarial examples:

$$\min_{\theta} \max_{\Delta:\|\Delta\| \leq \gamma} \mathcal{L}(f(x+\Delta;\theta), y). \tag{3.1}$$

For the inner maximization problem, we still utilize the non-DP regular optimizers, also known as the **attackers**, to learn Δ. Some examples include FGSM (Goodfellow et al., 2014), PGD (Madry et al., 2017) and OnePixel attack (Su et al., 2019),

$$\texttt{attacker}(\boldsymbol{x}_i, y_i, f) := \boldsymbol{x}_i + \Delta^*$$

where Δ^* is the optimal perturbation defined in (2.1) and $\boldsymbol{x}_i + \Delta^*$ is the adversarial example for the benign example $\boldsymbol{x}_i$. For the outer minimization problem, we apply the DP optimizers such as DP-SGD or DP-Adam to learn θ. For instance, DP-SGD can be written as

$$\theta_{t+1} = \theta_t - \frac{\eta_t}{n}\left(\sum_i \frac{\nabla_\theta \mathcal{L}(f(\boldsymbol{x}_i, \theta_t), y_i)}{\max\{1, \|\nabla_\theta \mathcal{L}(f(\boldsymbol{x}_i, \theta_t), y_i)\|_2 / R\}} + \sigma R \cdot \mathcal{N}(0, \mathbf{I})\right)$$

where the hyperparameters can be found in Algorithm 1. Interestingly, our `DP-Adv` problem is similar but different to DP GAN which learns $\min_G \max_D \mathcal{L}(G, D)$ where G is the generator and D is the discriminator. Note that only D is learned privately. At high level, `DP-Adv`

is learning $\min_D \max_G \mathcal{L}(G, D)$ and also only D is learned privately. The difference is that the output of G in data-dependent in `DP-Adv` and D is learning data-label pairs instead of pairs of examples (hence D is a classifier but not a discriminator).

Remark 3.1. The combination of DP and adversarial training requires extra caution. For example, no batch normalization can be used in the network as it violates DP. Another line of adversarial training (Kurakin et al., 2016) uses both benign and adversarial examples for training: for some constant $\xi \geq 1$, the loss function is

$$\widehat{\mathcal{L}}(B, B^{\text{adv}}, \theta) = \frac{1}{|B| + \xi|B|^{\text{adv}}} \Big(\sum_{x_i \in B} \mathcal{L}\big(f(x_i, \theta), y_i\big) + \xi \sum_{x_j^{\text{adv}} \in B^{\text{adv}}} \mathcal{L}\big(f(x_j^{\text{adv}}, \theta), y_j\big) \Big).$$

This makes the privacy accounting difficult due to the complicated sensitivity analysis.

3.2 DP-Adv training algorithm

Combining the two optimization procedures above, we propose the complete `DP-Adv` training algorithm. Here we only present the special case of DP-SGD in Algorithm 1. We leave the more general case (e.g. with DP-Adam) and a fully-detailed version with PGD attack in Appendix A.

Algorithm 1 Differentially Private Adversarial Training [`DP-Adv`]

Parameters: initial weights θ_0, learning rate η_t, subsampling probability q, number of iterations T, perturbation bound γ, noise scale σ, gradient norm bound R.

1: **for** $t = 0, \ldots, T-1$ **do**
2: Subsample a batch $B_t \subseteq \{1, \ldots, n\}$ with subsampling probability q
3: **for** $i \in B_t$ **do**
4: $\boldsymbol{x}_i \leftarrow \texttt{attacker}(\boldsymbol{x}_i, y_i, f; \gamma)$ ▷ Generate adversarial example
5: $g_i \leftarrow \nabla_\theta \mathcal{L}(f(\boldsymbol{x}_i, \theta_t), y_i)$
6: $g_i \leftarrow g_i \cdot \min\big\{1, R/\|g_i\|_2\big\}$ ▷ Clip the per-sample gradient
7: $g_t \leftarrow \sum_{i \in B_t} g_i$
8: $g_t \leftarrow g_t + \sigma R \cdot \mathcal{N}(0, \mathbf{I})$ ▷ Apply Gaussian mechanism
9: $\theta_{t+1} \leftarrow \theta_t - \frac{\eta_t}{|B_t|} g_t$

The choices of adversarial attack and DP optimizer are flexible. In particular, one only needs to change line 5-9 in Algorithm 1 to use another DP optimizer, or line 4 to use another attacker. In comparison, we present `StoBatch` in Phan et al. (2020).

Algorithm 2 StoBatch

Parameters: initial weights θ_0 and W_0, learning rate η_t, number of iterations T, neural network f with randomized first hidden layer (adding Noise_2 to the forward propagation).

1: **for** $t = 0, \ldots, T-1$ **do**
2: Subsample a batch $B_t \subseteq \{1, \ldots, n\}$ and a sub-batch $b_t \subseteq B_t$
3: **for** $i \in B_t$ **do**
4: $\bar{\boldsymbol{x}}_i \leftarrow \boldsymbol{x}_i + \text{Noise}_1$
5: $g_i \leftarrow \nabla_\theta \mathcal{L}(f(\bar{\boldsymbol{x}}_i, \theta_t), y_i)$
6: $h_i \leftarrow w_t^\top \bar{\boldsymbol{x}}_i, \bar{h}_i = h_i + 2 \cdot \text{Noise}_1$ ▷ Perturbed hidden layer representation
7: **for** $i \in b_t$ **do**
8: Draw random perturbation bound $\gamma \in (0, 1]$.
9: $\bar{\boldsymbol{x}}_i^{adv} \leftarrow \texttt{attacker}(\bar{\boldsymbol{x}}_i, y_i, f; \gamma)$ ▷ Generate adversarial example
10: $g_i^{\text{adv}} \leftarrow \nabla_\theta \mathcal{L}(f(\bar{\boldsymbol{x}}_i^{\text{adv}}, \theta_t), y_i)$
11: $g_t \leftarrow \sum_{i \in B_t} g_i$
12: $g_t^{\text{adv}} \leftarrow \sum_{i \in b_t} g_i^{\text{adv}}$
13: $\theta_{t+1} \leftarrow \theta_t - \frac{\eta_t}{|B_t| + \xi|b_t|}(g_t + \xi g_t^{\text{adv}}) + \text{Noise}_3$ ▷ Descend of neural network
14: $W_{t+1} \leftarrow W_t - \eta_t \sum_{x \in \{\bar{\boldsymbol{x}}_i, \bar{\boldsymbol{x}}_i^{\text{adv}}\}} \frac{\partial \mathcal{R}(x; \bar{h}, W_t)}{\partial W_t}$ ▷ Descend of linear auto-encoder

where $\mathcal{R}(\boldsymbol{x}_i; \bar{h}_i, W) := \sum_{j \in [p]} \frac{1}{2} W_j \bar{h}_i - \bar{\boldsymbol{x}}_i W \bar{h}_i$ is the reconstruction error of the linear auto-encoder and $W_t h_i$ is the reconstruction of $\boldsymbol{x}_i$.

3.3 Privacy Guarantee

Theorem 1. *DP adversarial training (`DP-Adv`) is as (ϵ, δ)-DP as regular DP training.*

Proof of Theorem 1. By Lemma 2.3, it suffices to show that the ℓ_2 sensitivity of $\sum_i g_t^{(i)}$ is R, the same as in the regular DP. The rest of the proof follows since the noise magnitude σR is proportional to R and thus (ϵ, δ) does not depend on R. The sensitivity of adversarial examples is indeed guaranteed by the per-sample gradient clipping. □

In other words, line 4 (the existence of adversarial example) does not affect DP: each benign example is replaced by exactly one adversarial example and the sensitivity is bounded by the gradient clipping. In contrast, `Stobatch` (Phan et al., 2020) also uses the benign examples in training. Additional noises are required to make them private at the cost of lower accuracy.

To be specific, we consider two commonly applied privacy accountants: moments accountant (Abadi et al., 2016) and Gaussian DP (Dong et al., 2021; Bu et al., 2020).

Corollary 3.2 (adapted from Theorem 1, Abadi et al., 2016)**.** *There exist constants c_1 and c_2 so that given the sampling probability $q = |B_t|/n$ and the number of iterations T, for any $\epsilon < c_1 q^2 T$, `DP-Adv` is (ϵ, δ)-DP, for any $\delta > 0$ if we choose $\sigma \geq c_2 q\sqrt{T}\log(1/\delta)/\epsilon$.*

Corollary 3.3 (adapted from Bu et al., 2020)**.** *Given the sampling probability $q = |B_t|/n$ and the number of iterations T, `DP-Adv` is asymptotically μ-GDP with $\mu = q\sqrt{T(e^{1/\sigma^2} - 1)}$. Equivalently, `DP-Adv` is (ϵ, δ)-DP for any $\delta > 0$ and $\delta = \Phi(-\frac{\epsilon}{\mu} + \frac{\mu}{2}) + e^{\epsilon}\Phi(-\frac{\epsilon}{\mu} - \frac{\mu}{2})$.*[1]

4 Experiments

We conduct experiments on MNIST, CIFAR10 and CelebA datasets to demonstrate the superior performance of `DP-Adv`. We emphasize that our framework works flexibly with other DP optimizers such as DP-Adam, DP-SGD-JL (Bu et al., 2021a), DP-SGD with global clipping (Bu et al., 2021b), DP-FedSGD (McMahan et al., 2017), as well as other attack methods, among which we cover FGSM and PGD (l_2 and l_∞, denoted in subscript).

4.1 Comparison with previous arts

We compare our `DP-Adv` training with the existing methods that guarantee both DP and adversarial robustness on the MNIST dataset[2]: `StoBatch` by Phan et al. (2020) and `SecureSGD` by Phan et al. (2019). We use FGSM attack for defense as reported in Phan et al. (2020) with $l_\infty(0.2)$, i.e. the perturbation set is $\|\Delta\|_\infty \leq 0.2$. We highlight that in all four DP methods, the DP optimizer is the DP-SGD from Abadi et al. (2016). Both `SecureSGD` and `StoBatch` uses a two-layer CNN and hyperparameters from Phan et al. (2020) but other methods use a different two-layer CNN (Abadi et al., 2016; Papernot et al., 2020) for best performance. Here `Adv` means regular adversarial training. Experiment details are in Appendix B.

Method	Clean accuracy	Robust accuracy	ϵ
SecureSGD	38.5%	39.1%	0.2
StoBatch	83.4%	82.7%	0.2
DP-SGD	94.5%	55.7%	0.2
DP-SGD	97.2%	63.0%	1
DP-SGD	97.6%	64.3%	2
DP-Adv	94.0%	74.0%	0.2
DP-Adv	97.3%	86.0%	1
DP-Adv	97.8%	89.1%	2
SGD	99.1%	67.8%	∞
Adv	99.2%	95.3%	∞

Table 2: Comparison of robustness and privacy by different methods on MNIST, under FGSM $l_\infty(0.2)$.

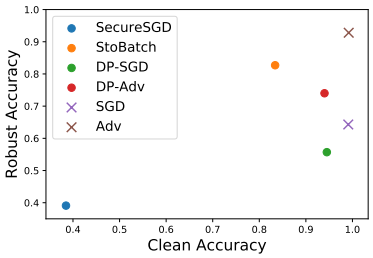

Figure 2: Clean and robust accuracy when $\epsilon = 0.2$. The cross means not DP.

[1] Here Φ is the cumulative distribution function of standard normal distribution.

[2] MNIST is a black-white image dataset of digits with 60000 training and 10000 testing examples, each of dimension 28×28.

Further experiments on MNIST demonstrate that `SecureSGD` achieves 62% and 77% robust accuracy at $\epsilon = 1$ and 2, respectively; `StoBatch` achieves 84% and 89% robust accuracy at the same privacy levels. From Table 2, we see from the viewpoint of robust accuracy that, our `DP-Adv` improves on `StoBatch` and much more on `SecureSGD`, while being slightly lower than non-DP adversarial training. In particular, `DP-Adv` significantly outperforms the state-of-the-art `StoBatch` on the clean accuracy by $> 10\%$, due to our concise design without adding 3 types of noises as in `StoBatch`, Algorithm 2.

4.2 MNIST WITH FURTHER EXPERIMENTS

We further evaluate `DP-Adv` with various attacks on MNIST. We consider two most popular $l_\infty(0.2)$ attacks, FGSM and PGD. We also test $l_2(1)$ attacks such as PGD-L2 and CW, but the latter is not effective even on neural networks without adversarial training. Omitted experiments can be found in Appendix C. From Figure 3, non-robust training (SGD or DP-SGD) has high clean accuracy above 96% but low robust accuracy, below 40%. In contrast, `DP-Adv` preserves high clean accuracy and 80% robust accuracy with strong privacy $\epsilon = 1$.

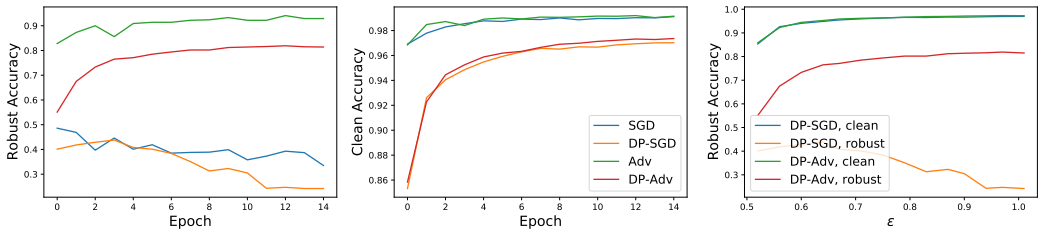

Figure 3: Training against $\text{PGD}_\infty(0.2)$ attack on MNIST, with $\epsilon = 1$.

4.3 TRANSFERABILITY

We investigate the transferability and calibration of DP and/or adversarially robust algorithms. Transferability is a measure of the robustness of models under various attacks, especially those which the models have not been trained to defend against.

From Table 3, we observe that `DP-Adv` and `Adv` both enjoy transferable defense: for instance, when adversarially trained with PGD_∞, the models automatically learn to defend against unseen attacks like BIM (Kurakin et al., 2016), APGD (Croce & Hein, 2020) and AutoAttack (Croce & Hein, 2020). We only present l_∞ attacks here and leave l_2 attacks on MNIST in Appendix C, where transferability of `DP-Adv` is consistently observed.

Defense/Attack	Clean	FGSM_∞	PGD_∞	BIM_∞	APGD_∞	AutoAttack_∞
SGD	99.1%	67.8%	33.0%	40.6%	28.9%	27.7%
DP-SGD	97.2%	63.0%	28.0%	36.9%	20.6%	18.7%
Adv+FGSM_∞	99.2%	95.3%	92.4%	93.1%	92.2%	91.9%
$\text{DP-Adv+FGSM}_\infty$	97.3%	86.0%	80.0%	81.5%	78.4%	78.3%
Adv+PGD_∞	99.2%	93.8%	92.3%	92.6%	91.8%	91.7%
DP-Adv+PGD_∞	97.4%	85.5%	81.4%	82.2%	80.6%	80.4%
Adv+PGD_2	99.3%	92.5%	90.0%	90.6%	89.2%	89.0%
DP-Adv+PGD_2	97.3%	83.9%	76.3%	78.1%	74.5%	74.1%

Table 3: Accuracy from different defense and $l_\infty(0.2)$ attacks on MNIST, with $\epsilon = 1$.

4.4 CIFAR10 WITH PRE-TRAINING

We experiment with CIFAR10[3], a more challenging dataset on which deep learning models deteriorate their robust accuracy close to zero, and the state-of-the-art accuracy with DP is less than 60%. In fact, it is a common practice to pre-train DP models, thus boosting about 10% DP accuracy on CIFAR10. In our experiments, we pre-train a two-layer CNN from Abadi et al. (2016), see Appendix B.2, on CIFAR100 for 10 epochs before privately train on CIFAR10.

[3]CIFAR10 is a colored image dataset of objects from 10 classes, with 50000 training examples and 10000 testing ones, each of dimension 32×32.

From Table 4, `DP-Adv` achieves moderately worse clean accuracy than other methods in order to achieve the strong robustness and privacy guarantees. As a tradeoff, `DP-Adv`'s robust accuracy is about 10% lower than non-DP `Adv` training. On the bright side, we again observe the transferability of `DP-Adv` in Appendix C, preserving adversarial robustness against PGD, APGD and AutoAttack.

Method	SGD	DP-SGD	Adv (FGSM)	DP-Adv (FGSM)	Adv (PGD_∞)	DP-Adv (PGD_∞)	Adv (PGD_2)	DP-Adv (PGD_2)
Clean accuracy	69.0%	64.0%	64.3%	51.4%	66.9%	55.7%	63.3%	54.3%
Robust accuracy	17.7%	18.1%	41.5%	32.7%	40.0%	30.0%	44.5%	37.6%

Table 4: Accuracy by different methods on CIFAR10, under $l_\infty(4/255)$ or $l_2(100/255)$.

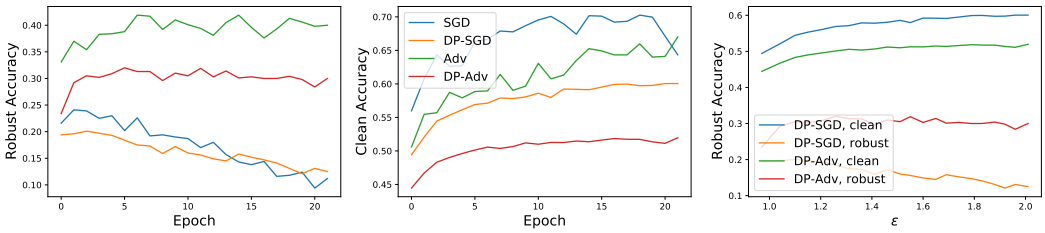

Figure 4: Training against $\text{PGD}_\infty(4/255)$ attack on CIFAR10, with $\epsilon = 1$.

4.5 Calibration

We also look into the calibration issue of modern neural networks (Guo et al., 2017), which are known to suffer from over-confidence: predicting with high probability but only low accuracy. Popular measures of calibration include negative log-likelihood (NLL)[4], expected calibration error (ECE) and maximum calibration error (MCE). It is empirically observed that (1) DP models tend to be less calibrated (Bu et al., 2021b) and (2) pre-training mitigates mis-calibration (Hendrycks et al., 2019). Our experiments support these observations in the DP and robust regime. In addition, we find that adversarial training also worsens the calibration in Figure 5, which can be significantly mitigated by pre-training in Figure 6.

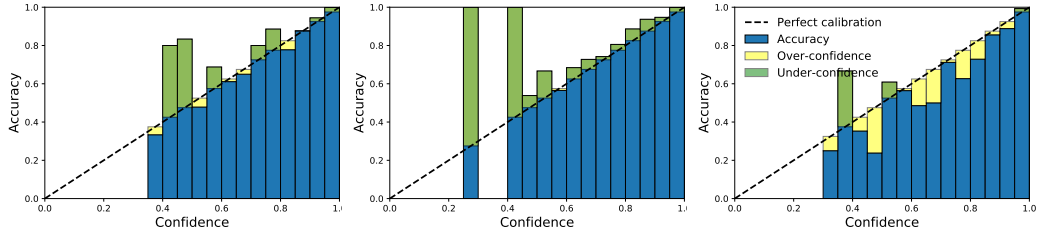

Figure 5: Reliability diagrams on MNIST (without pre-training). Left: non-DP SGD. Mid: non-DP Adv by PGD_∞. Right: DP-Adv by PGD_∞.

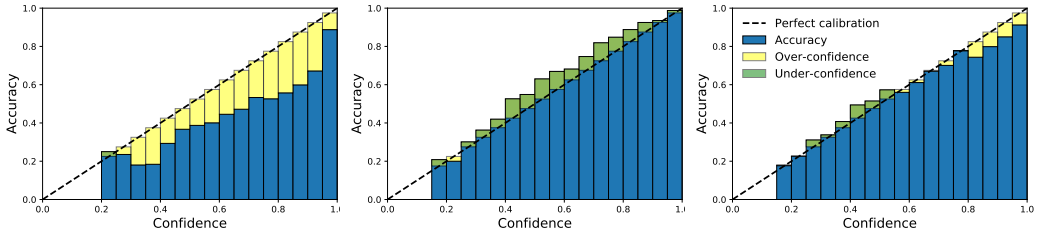

Figure 6: Reliability diagrams on CIFAR10 (with pre-training). Left: non-DP SGD. Mid: non-DP Adv by PGD_∞. Right: DP-Adv by PGD_∞. For further comparison, see Figure 10.

In Table 6 and Table 5, we observe notable increase of NLL, ECE and MCE on MNIST by DP and/or adversarial training, which is signicantly mitigated or even improved upon by methods using pre-training on CIFAR10.

[4]In the optimization literature, NLL can be viewed as the loss function, e.g. cross-entropy or mean squared error.

Method	NLL	ECE	MCE
SGD	1.17	16.6%	27.6%
DP-SGD	1.39	16.5%	26.9%
Adv+FGSM	1.06	6.5%	13.5%
DP-Adv+FGSM	1.36	4.4%	9.8%
Adv+PGD$_\infty$	0.96	5.7%	10.5%
DP-Adv+PGD$_\infty$	1.33	3.5%	8.1%
Adv+PGD$_2$	1.07	2.6%	9.2%
DP-Adv+PGD$_2$	1.31	4.4%	11.8%

Table 5: Calibration metrics of different methods on CIFAR10, with $\epsilon = 1$.

Method	NLL	ECE	MCE
SGD	0.026	9.6%	37.5%
DP-SGD	0.027	6.4%	17.5%
Adv+FGSM	0.092	13.7%	67.5%
DP-Adv+FGSM	0.087	13.0%	67.5%
Adv+PGD$_\infty$	0.099	14.2%	72.5%
DP-Adv+PGD$_\infty$	0.091	10.1%	29.6%
Adv+PGD$_2$	0.098	9.0%	29.6%
DP-Adv+PGD$_2$	0.091	8.7%	17.0%

Table 6: Calibration metrics of different methods on MNIST, with $\epsilon = 1$.

4.6 CelebA: Multi-label classification on Real Face dataset

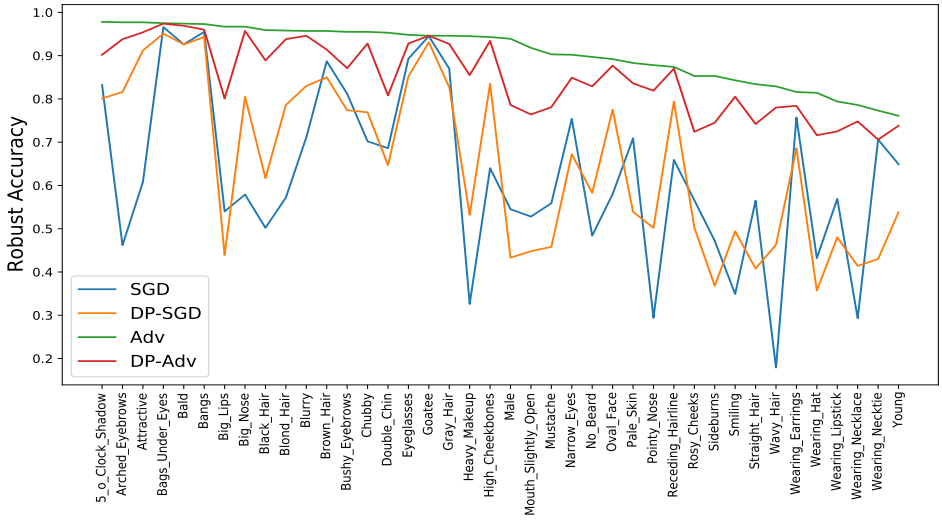

Figure 7: Robust accuracy of training against FGSM(0.1) attack on CelebA, with $\epsilon = 1$.

To test our `DP-Adv` on real-world scenario, we experiment on the CelebA dataset[5] containing real human faces. Each face image is tagged with 40 labels describing the face: e.g. male or female, smiling or not, young or old. Our model is two-layer CNN (see Appendix B). We pre-process by resizing and center-crop to 64×64 and normalizing the image. As expected, non-adversarially-trained methods (SGD and DP-SGD) are not robust, for instance, giving 40-50% accuracy on predicting smiling or not under FGSM attack. `DP-Adv` achieves about 80-90% robust accuracy across all labels. We highlight that, unlike on CIFAR10, where `DP-Adv` is usually 10% less in robust accuracy than `Adv`, the gap is much smaller on CelebA (about 5% gap with high accuracy). This result implies the promising application of `DP-Adv` in practice when there is a sufficient amount of data.

5 Conclusion

In this paper, we lay down the `DP-Adv` framework to simultaneously guarantee the privacy and adversarial robustness in deep learning. Our framework is flexible to absorb other attacking strategies (e.g. attacking optimizer, targeted or untargeted attack) and DP methods (e.g. new optimizer or privacy accountant). Especially, our method is practical in three ways: `DP-Adv` is concise in design and outperforms existing methods in performance as well as complexity; `DP-Adv` is as private as DP training and, in terms of wall-clock time, as fast as adversarial training; `DP-Adv` demonstrates strong performance on large-scale real datasets while being compatible to calibration techniques (e.g. pre-training and temperature scaling), thus delivering accurate, private, robust and calibrated models.

[5]CelebA is a colored image dataset, with 200000 examples which we split into 180000 training and 20000 testing examples. Each image is associated with 40 binary labels and of dimension 218×178.

References

Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, pp. 308–318, 2016.

Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *International conference on machine learning*, pp. 274–283. PMLR, 2018.

Franziska Boenisch, Philip Sperl, and Konstantin Böttinger. Gradient masking and the underestimated robustness threats of differential privacy in deep learning. *arXiv preprint arXiv:2105.07985*, 2021.

Zhiqi Bu, Jinshuo Dong, Qi Long, and Weijie J Su. Deep learning with gaussian differential privacy. *Harvard data science review*, 2020(23), 2020.

Zhiqi Bu, Sivakanth Gopi, Janardhan Kulkarni, Yin Tat Lee, Judy Hanwen Shen, and Uthaipon Tantipongpipat. Fast and memory efficient differentially private-sgd via jl projections. *arXiv preprint arXiv:2102.03013*, 2021a.

Zhiqi Bu, Hua Wang, Qi Long, and Weijie J Su. On the convergence of deep learning with differential privacy. *arXiv preprint arXiv:2106.07830*, 2021b.

Nicholas Carlini, Chang Liu, Úlfar Erlingsson, Jernej Kos, and Dawn Song. The secret sharer: Evaluating and testing unintended memorization in neural networks. In *28th {USENIX} Security Symposium ({USENIX} Security 19)*, pp. 267–284, 2019.

Nicholas Carlini, Florian Tramer, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Ulfar Erlingsson, et al. Extracting training data from large language models. *arXiv preprint arXiv:2012.07805*, 2020.

Jeremy Cohen, Elan Rosenfeld, and Zico Kolter. Certified adversarial robustness via randomized smoothing. In *International Conference on Machine Learning*, pp. 1310–1320. PMLR, 2019.

Francesco Croce and Matthias Hein. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In *International conference on machine learning*, pp. 2206–2216. PMLR, 2020.

Jinshuo Dong, Aaron Roth, and Weijie Su. Gaussian differential privacy. *Journal of the Royal Statistical Society*, 2021.

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.

Cynthia Dwork, Aaron Roth, et al. The algorithmic foundations of differential privacy. *Found. Trends Theor. Comput. Sci.*, 9(3-4):211–407, 2014.

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

Sivakanth Gopi, Yin Tat Lee, and Lukas Wutschitz. Numerical composition of differential privacy. *arXiv preprint arXiv:2106.02848*, 2021.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *International Conference on Machine Learning*, pp. 1321–1330. PMLR, 2017.

Yaowei Han, Yang Cao, and Masatoshi Yoshikawa. Understanding the interplay between privacy and robustness in federated learning. *arXiv preprint arXiv:2106.07033*, 2021.

Dan Hendrycks, Kimin Lee, and Mantas Mazeika. Using pre-training can improve model robustness and uncertainty. In *International Conference on Machine Learning*, pp. 2712–2721. PMLR, 2019.

Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial machine learning at scale. *arXiv preprint arXiv:1611.01236*, 2016.

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)*, pp. 656–672. IEEE, 2019.

Bai Li, Changyou Chen, Wenlin Wang, and Lawrence Carin. Certified adversarial robustness with additive noise. *Advances in Neural Information Processing Systems*, 32:9464–9474, 2019.

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.

H Brendan McMahan, Daniel Ramage, Kunal Talwar, and Li Zhang. Learning differentially private recurrent language models. *arXiv preprint arXiv:1710.06963*, 2017.

Arvind Narayanan and Vitaly Shmatikov. How to break anonymity of the netflix prize dataset. *arXiv preprint cs/0610105*, 2006.

Nicolas Papernot, Abhradeep Thakurta, Shuang Song, Steve Chien, and Ulfar Erlingsson. Tempered sigmoid activations for deep learning with differential privacy. *arXiv preprint arXiv:2007.14191*, 2020.

Hai Phan, My T Thai, Han Hu, Ruoming Jin, Tong Sun, and Dejing Dou. Scalable differential privacy with certified robustness in adversarial learning. In *International Conference on Machine Learning*, pp. 7683–7694. PMLR, 2020.

NhatHai Phan, Minh Vu, Yang Liu, Ruoming Jin, Dejing Dou, Xintao Wu, and My T Thai. Heterogeneous gaussian mechanism: Preserving differential privacy in deep learning with provable robustness. *arXiv preprint arXiv:1906.01444*, 2019.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

Ali Shafahi, Mahyar Najibi, Amin Ghiasi, Zheng Xu, John Dickerson, Christoph Studer, Larry S Davis, Gavin Taylor, and Tom Goldstein. Adversarial training for free! *arXiv preprint arXiv:1904.12843*, 2019.

Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. In *2017 IEEE Symposium on Security and Privacy (SP)*, pp. 3–18. IEEE, 2017.

Liwei Song, Reza Shokri, and Prateek Mittal. Privacy risks of securing machine learning models against adversarial examples. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, pp. 241–257, 2019.

Jiawei Su, Danilo Vasconcellos Vargas, and Kouichi Sakurai. One pixel attack for fooling deep neural networks. *IEEE Transactions on Evolutionary Computation*, 23(5):828–841, 2019.

Florian Tramèr, Alexey Kurakin, Nicolas Papernot, Ian Goodfellow, Dan Boneh, and Patrick McDaniel. Ensemble adversarial training: Attacks and defenses. *arXiv preprint arXiv:1705.07204*, 2017.

Nurislam Tursynbek, Aleksandr Petiushko, and Ivan Oseledets. Robustness threats of differential privacy. *arXiv preprint arXiv:2012.07828*, 2020.

Qiyiwen Zhang, Zhiqi Bu, Kan Chen, and Qi Long. Differentially private bayesian neural networks on accuracy, privacy and reliability. *arXiv preprint arXiv:2107.08461*, 2021.

Yuqing Zhu, Jinshuo Dong, and Yu-Xiang Wang. Optimal accounting of differential privacy via characteristic function. *arXiv preprint arXiv:2106.08567*, 2021.