UC Irvine

Title

Improving Differentially Private Models with Active Learning

Permalink

https://escholarship.org/uc/item/37n4g2c6

Authors

Zhao, Zhengli
Papernot, Nicolas
Singh, Sameer
et al.

Publication Date

2019-10-02

Copyright Information

Peer reviewed

Improving Differentially Private Models with Active Learning

Zhengli Zhao[1,2], Nicolas Papernot[2], Sameer Singh[1], Neoklis Polyzotis[2], and Augustus Odena[2]

[1]UC Irvine, [2]Google Brain
zhengliz@uci.edu, augustusodena@google.com

Abstract

Broad adoption of machine learning techniques has increased privacy concerns for models trained on sensitive data such as medical records. Existing techniques for training differentially private (DP) models give rigorous privacy guarantees, but applying these techniques to neural networks can severely degrade model performance. This performance reduction is an obstacle to deploying private models in the real world. In this work, we improve the performance of DP models by fine-tuning them through active learning on public data. We introduce two new techniques — DiversePublic and NearPrivate — for doing this fine-tuning in a privacy-aware way. For the MNIST and SVHN datasets, these techniques improve state-of-the-art accuracy for DP models while retaining privacy guarantees.

1 Introduction

Privacy concerns surfaced with the increased adoption of machine learning in domains like healthcare (LeCun et al., 2015). One widely adopted framework for measuring privacy characteristics of randomized algorithms, such as machine learning techniques, is differential privacy (Dwork et al., 2006). Abadi et al. (2016) introduced an algorithm for differentially private stochastic gradient descent (DP-SGD), which made it feasible to scale differential privacy guarantees to neural networks. DP-SGD is now the de facto algorithm used for training neural networks with privacy guarantees.

There is, however, a crucial problem for models with a large number of parameters: it is difficult to achieve both non-trivial privacy guarantees and good accuracy. The reasons for this are numerous and involved, but the basic intuition is that DP-SGD involves clipping gradients and then adding noise to those gradients. However, gradient clipping has an increasing relative effect as the number of model parameters grows. This reduces the applicability of differential privacy to deep learning in practice, where strong performance tends to require a large number of model parameters.[1]

In this work, we propose a method to mitigate this problem, making DP learning feasible for modern image classifiers. The method is based on the observation that each DP-SGD update consumes privacy budget in proportion to the quotient of the batch size and the training-set size. Thus, by increasing the number of effective training examples, we can improve accuracy while maintaining the same privacy guarantees. Since obtaining more private training samples will generally be nontrivial, we focus on using 'public' data to augment the training set. This involves assuming that there will be unlabeled[2] public data that is sufficiently related to our private data to be useful, but we think this is a reasonable assumption to make, for two reasons: First, it is an assumption made by most of the semi-supervised learning literature (Oliver et al., 2018). Second, one of our techniques will explicitly address the situation where the 'sufficiently related' clause partially breaks down.

In summary, our contributions are:

---

[1]For example, the best performing CIFAR-10 classifier from Zagoruyko & Komodakis (2016) has 32.5 million parameters, while the private baseline we describe in Section 4.1 has about 26 thousand.

[2] If there is labeled public data, the situation is even better, but this is likely to be rare and is not the setting we consider here.

- We introduce an algorithm (DIVERSEPUBLIC) to select diverse representative samples from a public dataset to fine-tune a DP classifier.

- We then describe another algorithm (NEARPRIVATE) that pays a privacy cost to reference the private training data when querying the public dataset for representative samples.

- We establish new state-of-the-art results on the MNIST and SVHN datasets by fine-tuning DP models using simple active learning techniques, and then improve upon those results further using DIVERSEPUBLIC and NEARPRIVATE .

- We open source all of our experimental code.[3]

2 BACKGROUND

2.1 DIFFERENTIAL PRIVACY

We reason about privacy within the framework of differential privacy (Dwork et al., 2006). In this paper, the random algorithm we analyze is the training algorithm, and guarantees are measured with respect to the training data.

Informally, an algorithm is said to be differentially private if its behavior is indistinguishable on pairs of datasets that only differ by one point. That is, an observer cannot tell whether a particular point was included in the model's training set simply by observing the output of the training algorithm. Formally, for a training algorithm A to be (ε, δ)-differentially private, we require that it satisfies the following property for all pairs of datasets (d, d') differing in exactly one data point and all possible subsets $S \subset \text{Range}(A)$:

$$Pr[A(d) \in S] \leq e^{\varepsilon} Pr[A(d') \in S] + \delta$$

where δ is a (small) probability for which we tolerate the property not to be satisfied. The parameter ε measures the strength of the privacy guarantee: the smaller the value of ε is, the stronger the privacy guarantee is.

This guarantee is such that the output of a differentially private algorithm can be post-processed at no impact to the strength of the guarantee provided. In case privacy needs to be defined at a different granularity than invididual training points, the guarantee degrades by a factor which, naively, is the number of points that are included in the granularity considered. See Dwork et al. (2014a) for further information.

2.2 DIFFERENTIALLY PRIVATE STOCHASTIC GRADIENT DESCENT (DP-SGD)

Building on earlier work (Chaudhuri et al., 2011; Song et al., 2013), Abadi et al. (2016) introduce a variant of stochastic gradient descent to train deep neural networks with differential privacy guarantees. Two modifications are made to vanilla SGD. First, gradients are computed on a per-example basis and clipped to have a maximum known ℓ_2 norm. This bounds the sensitivity of the training procedure with respect to each individual training point. Second, noise calibrated to have a standard deviation proportional to this sensitivity is added to the average gradient. This results in a training algorithm known as differentially private SGD (DP-SGD). Unfortunately (as discussed in Section 1), DP-SGD does not perform well for models with large parameter counts, which motivates the improvements proposed in the next section.

3 IMPROVING DIFFERENTIALLY PRIVATE MODELS WITH ACTIVE LEARNING

This paper introduces the following high-level process to improve the performance of an existing DP classifier: First, find a public insensitive dataset containing relevant samples. Second, carefully select a subset of public samples that can improve the performance of the classifier. Third, fine-tune the existing classifier on the selected samples with labels. We want to perform the selection and the fine-tuning in a way which does not compromise the privacy guarantees of the existing classifier.

The first step can be done using domain knowledge about the classifier, e.g., utilizing relevant public genomic datasets for a DP classifier of genomics data. We assume standard fine-tuning techniques

---

[3]We will release the repository in the next update.

Algorithm 1 The DIVERSEPUBLIC Algorithm

Input: $\mathcal{M}_{\text{dp}}$ with privacy cost ε_{dp}, $\mathcal{D}_{\text{public}}$, N_{labeled}, $k > N_{\text{labeled}}$, p, N_{cluster}, N_{each}
Output: $\mathcal{M}_{\text{dp}}^*$ fine-tuned on selected public data with the same privacy cost ε_{dp}

$\quad E_{\text{public}} \leftarrow \mathcal{M}_{\text{dp}}(\mathcal{D}_{\text{public}})[-1]$ $\triangleright$ Compute 'embeddings' of public data
$\quad P \leftarrow \text{PCA}(E_{\text{public}}, p)$ $\triangleright$ Perform PCA on embeddings and get first p PCs
$\quad S_{\text{public}} \leftarrow P \times \text{SelectUncertain}(\mathcal{D}_{\text{public}}, k)$ $\triangleright$ Project k most 'uncertain' public data onto PCs
$\quad \text{ClusterCenters} \leftarrow \text{K-Means}(S_{\text{public}})$ $\triangleright$ Cluster projected embeddings to get N_{cluster} clusters
$\quad \mathcal{D}_{\text{labeled}} \leftarrow \emptyset$
$\quad \textbf{for } i = 1 \text{ to } N_{\text{cluster}} \textbf{ do}$ $\triangleright$ Label N_{each} data points from each cluster
$\quad\quad \mathcal{D}_{\text{labeled}} = \mathcal{D}_{\text{labeled}} \cup \text{TakeCenterPoints}(\text{ClusterCenters})$
$\quad \textbf{end for}$
$\quad \mathcal{M}_{\text{dp}}^* \leftarrow \text{FineTune}(\mathcal{M}_{\text{dp}}, \mathcal{D}_{\text{labeled}})$

---

for the last step. Therefore, the problem boils down to efficiently selecting samples from the public dataset while preserving privacy. We also assume a limit on the number of selected samples. This limit is relevant when the samples are unlabeled, in which case it controls the cost of labeling (e.g., hiring human annotators to process the selected samples).

We introduce two active learning algorithms, DIVERSEPUBLIC and NEARPRIVATE , for sample-selection. These algorithms make different assumptions about access to the private training data but are otherwise drop-in replacements in the end-to-end process.

3.1 PROBLEM STATEMENT AND BASELINE METHODS

We are given a differentially private model $\mathcal{M}_{\text{dp}}$ trained and tested on private sensitive data $\mathcal{D}_{\text{train}}$ and $\mathcal{D}_{\text{test}}$ respectively. The privacy cost of training $\mathcal{M}_{\text{dp}}$ on $\mathcal{D}_{\text{train}}$ is ε_{dp} (we omit δ for brevity in this paper, but it composes similarly). And we have an extra set of public insensitive unlabeled data $\mathcal{D}_{\text{public}}$ which can be utilized to further improve $\mathcal{M}_{\text{dp}}$. Given the number of extra data N_{labeled} that we are allowed to request labels for, and the total privacy budget $\varepsilon_{\text{limit}}$ of the improved model $\mathcal{M}_{\text{dp}}^*$, we want to efficiently pick $\mathcal{D}_{\text{labeled}} \subset \mathcal{D}_{\text{public}}$ where $|\mathcal{D}_{\text{labeled}}| = N_{\text{labeled}}$, using which we can fine-tune $\mathcal{M}_{\text{dp}}$ to $\mathcal{M}_{\text{dp}}^*$ of better performance on $\mathcal{D}_{\text{test}}$. The simplest baseline method we will use is to choose N_{labeled} 'random' samples out of $\mathcal{D}_{\text{public}}$ for fine-tuning. The other baseline that we use is Uncertain Sampling (Settles, 2009), according to either the 'entropy' of the logits or the 'difference' between the two largest logits. These are widely used active learning methods and serve as strong baselines.

3.2 THE DIVERSEPUBLIC METHOD

The baseline methods may select many redundant examples to label. To efficiently select diverse representative samples, we propose the DIVERSEPUBLIC method, adapted from clustering based active learning methods (Nguyen & Smeulders, 2004). Given a DP model $\mathcal{M}_{\text{dp}}$ of cost ε_{dp}, we obtain the 'embeddings' (the activations before the logits) E_{public} of all $\mathcal{D}_{\text{public}}$, and perform PCA on E_{public} as is done in Raghu et al. (2017). Then we select a number (more than N_{labeled}) of uncertain points (according to, e.g., the logit entropy of the private model) from $\mathcal{D}_{\text{public}}$, project their embeddings onto the top few principal components, and cluster those projections into groups. Finally, we pick a number of samples from each representative group up to N_{labeled} in total and fine-tune the DP model with these $\mathcal{D}_{\text{labeled}}$. Though this procedure accesses $\mathcal{M}_{\text{dp}}$, it adds nothing to ε_{dp}, since there is no private data referenced and the output of DP models can be post-processed at no additional privacy cost. *It can be applied even to models for which we cannot access the original training data.*[4] Algorithm 1 presents more details of the DIVERSEPUBLIC method.

3.3 THE NEARPRIVATE METHOD

The DIVERSEPUBLIC method works well under the assumption that $\mathcal{D}_{\text{public}}$ has the same distribution as $\mathcal{D}_{\text{train}}$. However, this may not be a reasonable assumption in general (Oliver et al., 2018). For instance, there may be a subset of $\mathcal{D}_{\text{public}}$ about which our pre-trained model has high uncertainty but

---

[4]e.g., models published to a repository like TensorFlow Hub: https://www.tensorflow.org/hub

---

Algorithm 2 The NEARPRIVATE Algorithm

Input: $\mathcal{M}_{dp}$ with privacy cost ε_{dp}, $\mathcal{D}_{public}$, $N_{labeled}$, $k > N_{labeled}$, p, $\mathcal{D}_{train}$, ε_{dpPCA} and $\varepsilon_{support}$
Output: $\mathcal{M}_{dp}^*$ fine-tuned on selected public data with privacy cost $\varepsilon_{dp} + \varepsilon_{dpPCA} + \varepsilon_{support}$

    $E_{train} \leftarrow \mathcal{M}_{dp}(\mathcal{D}_{train})[-1]$ ▷ Compute 'embeddings' of private data
    $P \leftarrow \text{DP-PCA}(E_{train}, \varepsilon_{dpPCA}, p)$ ▷ Perform DP-PCA on embeddings and get first p PCs
    $S_{train} \leftarrow P \times \text{SelectUncertain}(\mathcal{D}_{train}, k)$ ▷ Project k most 'uncertain' private data onto PCs
    $S_{public} \leftarrow P \times \text{SelectUncertain}(\mathcal{D}_{public}, k)$ ▷ Project k most 'uncertain' public data onto PCs
    NeighborCounts $\leftarrow \emptyset$
    for $a \in S_{train}$ **do**   ▷ In PC-space, assign each private point to exactly one public point
        Increment(NeighborCounts, NearestNeighbor(a, S_{public}))
    end for
    for $b \in S_{public}$ **do**   ▷ Compute support with Laplacian noise for each public data point
        $N_{support}(b) \leftarrow \text{NeighborCounts}(b) + \text{LaplaceNoise}(1/\varepsilon_{support})$
    end for
    $\mathcal{D}_{labeled} \leftarrow \text{TakeArgmaxPoints}(N_{support})$ ▷ Label $N_{labeled}$ data points of the highest $N_{support}$
    $\mathcal{M}_{dp}^* \leftarrow \text{FineTune}(\mathcal{M}_{dp}, \mathcal{D}_{labeled})$

---

which cannot improve performance if sampled. This may be because that subset contains corrupted data or it may be due simply to distribution shift. In order to mitigate this issue, we propose to check query points against our private data and decline to label query points that are too far from any training points (in projected-embedding-space). But doing this check while preserving privacy guarantees is nontrivial, since it involves processing the private training data itself in addition to $\mathcal{M}_{dp}$. Given a DP model of cost ε_{dp}, we obtain the embeddings E_{train} of all $\mathcal{D}_{train}$, and perform differentially private PCA (DP-PCA) (Dwork et al., 2014b) on E_{train} at a privacy cost of ε_{dpPCA}. We select a number of uncertain points from both $\mathcal{D}_{train}$ and $\mathcal{D}_{public}$. Then in the space of low dimensional DP-PCA projections, we assign each uncertain private example to exactly one uncertain public example according to Euclidean distance. This yields, for each uncertain public example, a count $N_{support}$ of 'nearby' uncertain private examples. Finally, we choose $N_{labeled}$ points with the largest values of $N_{support} + \text{LaplaceNoise}(1/\varepsilon_{support})$ from the uncertain public data. This sampling procedure is differentially private with cost $\varepsilon_{support}$ due to the Laplace mechanism (Dwork et al., 2014a). The total privacy cost of NEARPRIVATE is composed of the budgets expended to perform each of the three operations that depend on the private data: $\varepsilon_{dp} + \varepsilon_{dpPCA} + \varepsilon_{support}$. More details in Algorithm 2.

4 EXPERIMENTS

We conduct experiments on the MNIST (LeCun et al., 1998) and SVHN (Netzer et al., 2011) data sets. These may be seen as 'toy' data sets in the object recognition literature, but they are still challenging for DP object recognizers. In fact, at the time of this writing, there are no published examples of a differentially private SVHN classifier with both reasonable accuracy and non-trivial privacy guarantees. *The baseline we establish in Section 4.2 is thus a substantial contribution by itself.* For both datasets, we use the same model architecture as in the Tensorflow Privacy tutorials[5]. We obtain $\mathcal{M}_{dp}$ by training on $\mathcal{D}_{train}$ via DP-SGD (Abadi et al., 2016). Unless otherwise specified, we always aggregate results over 5 runs with different random seeds and use error bars to represent the standard deviation. We use the implementation of DP-SGD made available through the TensorFlow Privacy library (McMahan et al., 2018) with $\delta = 10^{-5}$.

4.1 EXPERIMENTS ON MNIST

We conduct our first set of experiments on the MNIST (LeCun et al., 1998) dataset. We use the Q-MNIST (Yadav & Bottou, 2019) dataset as our source of public data. In particular, we use the 50,000 examples from the Q-MNIST dataset that are reconstructions of the lost MNIST testing digits. We perform a hyperparameter optimization and find a baseline model with higher accuracy (97.3% at $\varepsilon_{dp} = 3.0$, and 97.5% at $\varepsilon_{dp} = 3.2$) than what is reported as the current state-of-the-art in the Tensorflow Privacy tutorial's README file (96.6% at $\varepsilon_{dp} = 3.0$).

---

[5]Tutorials for TensorFlow Privacy are found at: `https://github.com/tensorflow/privacy`

(a) DIVERSEPUBLIC vs baselines   (b) NEARPRIVATE vs DIVERSEPUBLIC
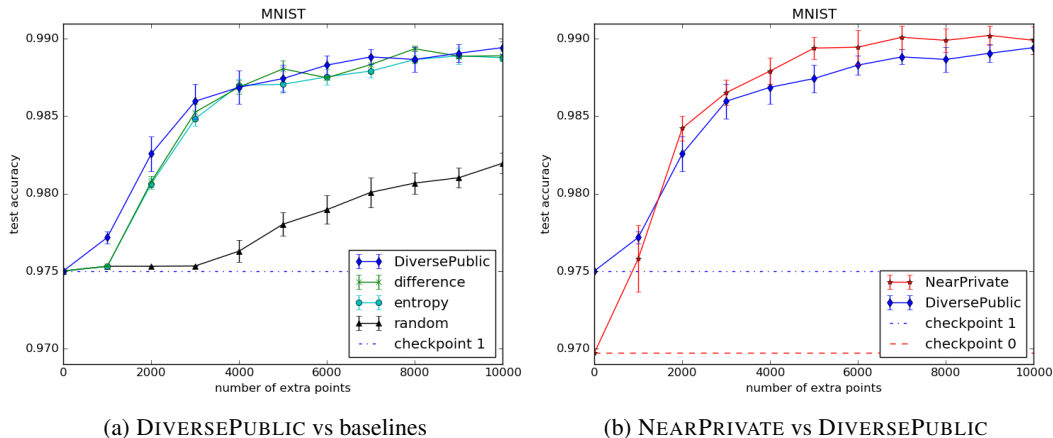
Figure 1: **MNIST experiments.** All plots show test-set-accuracy vs. the number of extra points labeled. **Left:** Comparison of our DIVERSEPUBLIC technique with the active learning baselines described above. All models are fine-tuned starting from the same baseline ('checkpoint 1': test accuracy 97.5% and $\varepsilon_{dp} = 3.2$). Active learning improves the performance of the DP-model to as much as 98.8% in the best case with no increase in privacy cost. **Right:** Comparison of our NEARPRIVATE technique with the DIVERSEPUBLIC technique. Since we spent $\varepsilon_{dpPCA} + \varepsilon_{support} = 1.0$ on the NEARPRIVATE technique, we fine-tune the NEARPRIVATE model from 'checkpoint 0' with privacy cost $\varepsilon_{dp} = 2.2$. Thus, both lines have the same privacy cost ($\varepsilon_{limit} = 3.2$), regardless of the number of extra points used.

Figure 1a shows results of our DIVERSEPUBLIC method compared with baselines. Starting from a checkpoint of test accuracy 97.5% ($\varepsilon_{dp} = 3.2$), our method can reach 98.8% accuracy with 7,000 extra labels. The DIVERSEPUBLIC method yields higher test accuracy than other active learning baselines in the low-query regime, and performs comparably in the high-query regime.

In Figure 1b, we compare NEARPRIVATE against DIVERSEPUBLIC . Since NEARPRIVATE adds extra privacy cost, we have to take special care when comparing it to DIVERSEPUBLIC . Therefore, we fine-tune from a starting checkpoint at test accuracy 97.0% with lower privacy cost ($\varepsilon_{dp} = 2.2$) and make sure its total privacy cost ($\varepsilon_{limit} = 3.2$) in the end is the same as the cost for the DIVERSEPUBLIC model. For this reason, NEARPRIVATE takes some number of labeled data points to 'catch up' to DIVERSEPUBLIC for the same DP cost — in this case 2,000. When $N_{labeled}$ is large enough, NEARPRIVATE outperforms all other methods. This shows that accessing the original training data in a privacy-aware way can substantially improve performance.

4.2 EXPERIMENTS ON SVHN

We conduct another set of experiments on the SVHN (Netzer et al., 2011) data. We use the set of '531,131 additional, somewhat less difficult samples' as our source of public data. Since a baseline model trained with DP-SGD on the SVHN training set performs quite poorly, we have opted to first pre-train the model on rotated images of $\mathcal{D}_{public}$ predicting only rotations as in Gidaris et al. (2018).

Broadly speaking, the results presented in Figure 2 are similar to MNIST results, but three differences stand out. First, the improvement given by active learning over the baseline private model is larger. Second, the improvement given by DIVERSEPUBLIC over the basic active learning techniques is also larger. Third, NEARPRIVATE actually underperforms DIVERSEPUBLIC in this case. We hypothesize that the first and second results are due to there being more 'headroom' in SVHN accuracy than for MNIST, and that the third result stems from the reported lower difficulty of the extra SVHN data. In the next section, we examine this phenomenon further.

4.3 EXPERIMENTS WITH DATASET POLLUTION

We were intrigued by the under-performance of NEARPRIVATE relative to DIVERSEPUBLIC on SVHN. We wondered whether it was due to the fact that SVHN and its extra data violate the
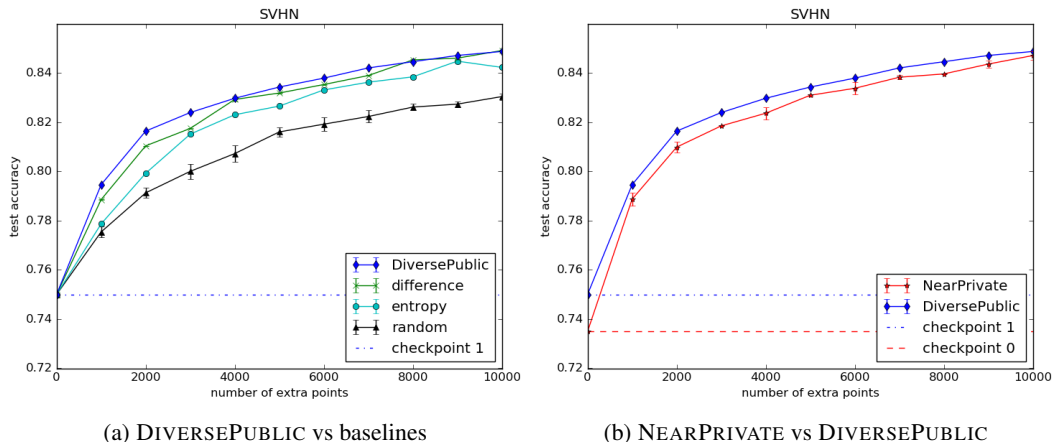
(a) DIVERSEPUBLIC vs baselines	(b) NEARPRIVATE vs DIVERSEPUBLIC

Figure 2: **SVHN experiments. Left:** Comparison of our DIVERSEPUBLIC technique with the active learning baselines. All models are fine-tuned starting from the same baseline ('checkpoint 1': test accuracy 75.0% and $\varepsilon_{\mathrm{dp}} = 6.0$). Active learning improves the performance of the DP-model to as much as 85% in the best case with no increase in privacy cost. Recall also that the 75.0% number is itself a baseline established in this paper. **Right:** Comparison of NEARPRIVATE with DIVERSEPUBLIC . The setup here is analogous to the one in the MNIST experiment, but DIVERSEPUBLIC performs better in this case. Since we spent $\varepsilon_{\mathrm{dpPCA}} + \varepsilon_{\mathrm{support}} = 1.0$ on NEARPRIVATE , we fine-tune from 'checkpoint 0' with test accuracy 73.5% and $\varepsilon_{\mathrm{dp}} = 5.0$.

assumption built into NEARPRIVATE – namely that we need to query the private data to 'throw out' unhelpful public data. Indeed, the SVHN website describes the extra set as 'somewhat less difficult' than the training data. To test this hypothesis, we designed a new experiment to check if NEARPRIVATE can actually select more helpful samples given a mixture of relevant data and irrelevant data as 'pollution'. We train the DP baseline with 30,000 of the SVHN training images, and treat a combination of another 40,000 SVHN training images and 10,000 CIFAR-10 (Krizhevsky et al., 2009) training images as the extra public dataset. These CIFAR-10 examples act as the unhelpful public data that we would hope NEARPRIVATE could learn to discard. As shown in Figure 3, all baselines perform worse than before with polluted public data. DIVERSEPUBLIC does somewhat better than random selection, but not much, achieving a peak performance improvement of around 1%. On the other hand, the difference between NEARPRIVATE and DIVERSEPUBLIC is more substantial, at over 2% accuracy in some cases. This is especially interesting considering that DIVERSEPUBLIC actually performed better in the results of Section 4.2. Broadly speaking, the results support our claim that NEARPRIVATE helps more relative to DIVERSEPUBLIC when there is 'unhelpful' data in the public dataset. This is good to know, since having some unhelpful public data and some helpful public data seems like a more realistic problem setting than the one in which all public data is useful.

5 ABLATION ANALYSES

In order to better understand how the performance of DIVERSEPUBLIC and NEARPRIVATE is affected by various hyper-parameters, we conduct several ablation studies.

5.1 HOW DO CLUSTERING HYPER-PARAMETERS AFFECT ACCURACY?

For DIVERSEPUBLIC , there are two parameters that affect the number of extra data points labeled for fine-tuning: the number of clusters we form (N_{cluster}) and the number of points we label per cluster (N_{each}). We write $N_{\mathrm{labeled}} = N_{\mathrm{cluster}} \times N_{\mathrm{each}}$. To study the relative effects of these, we conduct the experiment depicted in Figure 4a. In this experiment, we fine-tune the same DP MNIST model (test accuracy 97% at $\varepsilon_{\mathrm{dp}} = 2.2$) with varying values of N_{cluster} and N_{each}. We vary N_{cluster} from 100 to 500, which is depicted on the x-axis. We vary N_{each} from 5 to 20, with each value depicted as a different line. The general trend is one of diminishing returns on extra labeled data, as would be predicted by Hestness et al. (2017). We do not notice a strong correspondence between final test accuracy at a

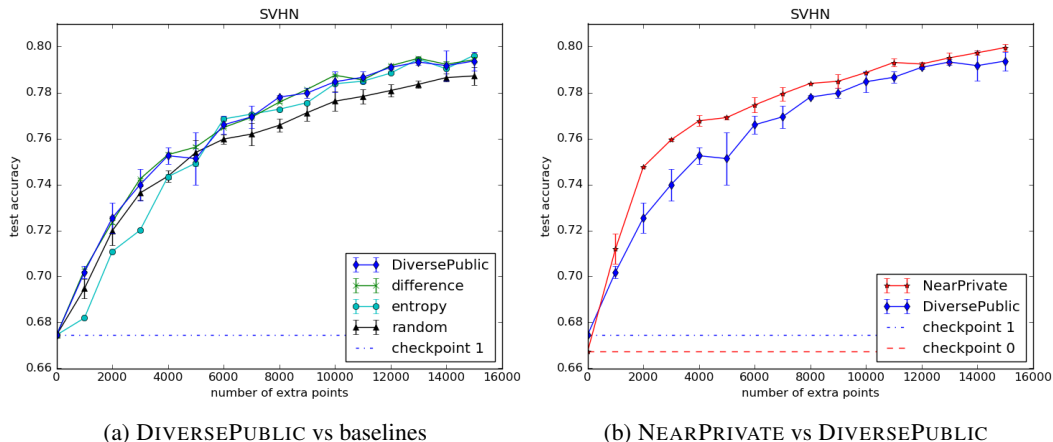(a) DIVERSEPUBLIC vs baselines

(b) NEARPRIVATE vs DIVERSEPUBLIC

Figure 3: **SVHN experiments with dataset pollution.** In this experiment, we train the DP baseline with 30,000 of the SVHN training images. The extra public dataset is a combination of 40,000 SVHN training images and 10,000 CIFAR-10 training images. **Left:** DIVERSEPUBLIC compared against other active learning baseline techniques as in Figure 2a. In this case, the active learning techniques do not outperform random selection by very much. Uncertainty by itself is not a sufficient predictor of whether extra data will be helpful here, since the baseline model is also uncertain about the CIFAR-10 images. **Right:** NEARPRIVATE compared against DIVERSEPUBLIC , but started from different checkpoints ('checkpoint 0' with $\varepsilon_{dp} = 5.0$, 'checkpoint 1' with $\varepsilon_{dp} = 6.0$) to keep the privacy cost constant as in Figure 2b. In this case, NEARPRIVATE substantially outperforms DIVERSEPUBLIC by selecting less of the CIFAR-10 images.
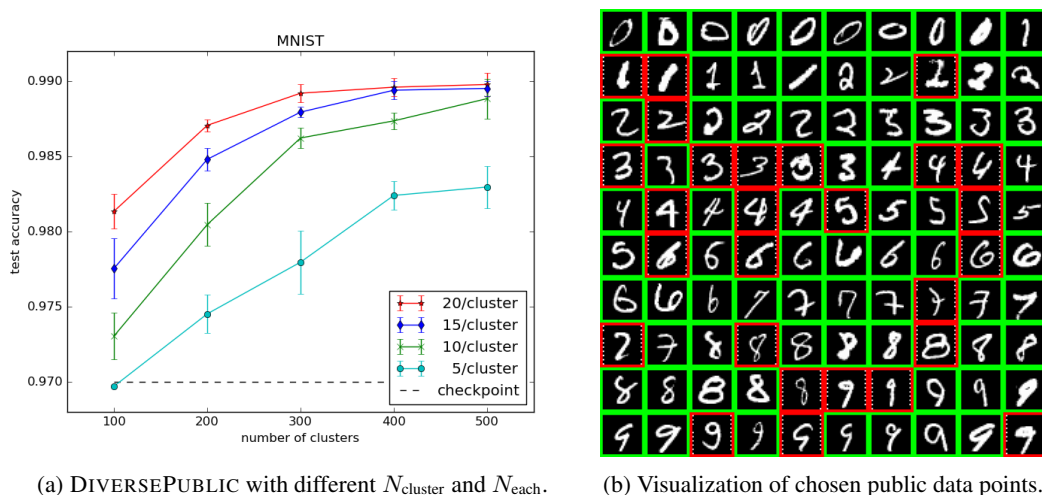

(a) DIVERSEPUBLIC with different $N_{cluster}$ and N_{each}.

(b) Visualization of chosen public data points.

Figure 4: DIVERSEPUBLIC **analysis. Left:** We apply DIVERSEPUBLIC to the same DP checkpoint (dashed horizontal line), varying the number of clusters (horizontal axis) and the number of chosen points from each cluster (lines with error bars). **Right:** For $N_{cluster} = 100, N_{each} = 20$, we visualize the most central example from each cluster. Since NEARPRIVATE does not have explicit clustering, we use DIVERSEPUBLIC for this visualization. Green borders mean that the initial checkpoint (dashed horizontal line) predicted correctly; while red bordered examples (originally predicted incorrectly) have dots on the left showing predictions and dots on the right showing true labels.

fixed number of extra labels and the values of $N_{cluster}$ and N_{each}. This is encouraging, as it suggests that practitioners can use our techniques without worrying too much about these hyper-parameters.

To address the question of which extra data points are being chosen for labeling, we create Figure 4b showing the most central example from each cluster, computed using $N_{cluster} = 100, N_{each} = 20$.
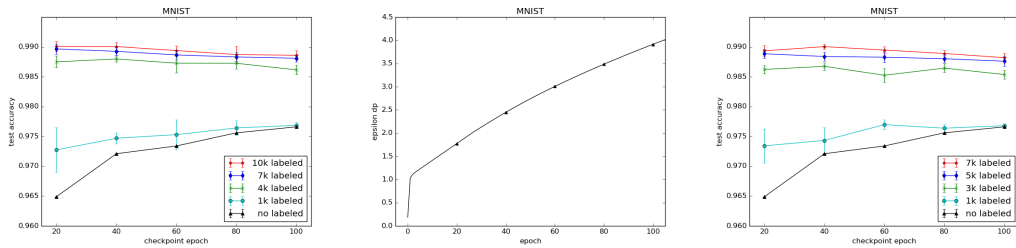
(a) Fixed extra privacy cost with $\varepsilon_{\text{dpPCA}} = 0.5$, $\varepsilon_{\text{support}} = 0.5$.

(b) The initial training privacy cost ε_{dp} for different checkpoints.

(c) Fixed total privacy cost with $\varepsilon_{\text{limit}} = 5.0$, $\varepsilon_{\text{dpPCA}} = 0.3$.

Figure 5: NEARPRIVATE **analysis.** We apply NEARPRIVATE to DP checkpoints (horizontal axis) of different privacy cost ε_{dp}. The black line with triangle represents those starting checkpoints. **Middle:** The initial training privacy cost ε_{dp} of DP checkpoints at different epochs of a single training run. **Left:** We fix the extra privacy cost $\varepsilon_{\text{dpPCA}} + \varepsilon_{\text{support}} = 1.0$, but vary the number of labeled public points for each line. We achieve large improvements from 1,000 to 4,000 labeled public points. With even larger N_{labeled}, the improvements are not significant and it may be better to start fine-tuning from a DP checkpoint of lower privacy cost. **Right:** With $\varepsilon_{\text{dpPCA}}$ set to 0.3, we fix the total privacy cost $\varepsilon_{\text{limit}} = 5.0$ and vary $\varepsilon_{\text{support}}$. For $N_{\text{labeled}} = 1000$, fine-tuning from the checkpoint at Epoch 60 is the best given a total privacy budget of 5.0.

The chosen examples are quite diverse, with a similar number of representatives from each class and variations in the thickness and shear of the digits. We can also inspect examples labeled incorrectly by the original checkpoint, such as the digit in Row 8, Col 1, which is a 7 that looks a lot like a '2'.

5.2   HOW DOES THE STARTING CHECKPOINT AFFECT RESULTS?

Recall that NEARPRIVATE accrues extra privacy cost by accessing the histogram of neighbor counts. This means that achieving a given accuracy under a constraint on the total privacy cost requires choosing how to allocate privacy between NEARPRIVATE and the initial DP-SGD procedure. Making this choice correctly requires a sense of how much benefit can be achieved from applying NEARPRIVATE to different starting checkpoints. Toward that end, we conduct (Figure 5) an ablation experiment on MNIST where we run NEARPRIVATE on many different checkpoints from the same training run. Figure 5a shows the test accuracies resulting from fine-tuning checkpoints at different epochs (represented on the x-axis) with fixed extra privacy cost of 1.0. Figure 5b shows the corresponding ε_{dp} for each checkpoint. Figure 5c varies other parameters given a fixed total privacy budget of 5.0.

In Figure 5a, the black line with triangle markers shows the initial test accuracies of the checkpoints. The other lines show results with different values of N_{labeled} from 1,000 to 10,000 respectively. With $N_{\text{labeled}} = 1000$, improvements are marginal for later checkpoints. In fact, the improvement from using $N_{\text{labeled}} = 1000$ at checkpoint 80 is not enough to compensate for the additional 1.0 privacy cost spent by NEARPRIVATE , because you could have had the same increase in accuracy by training the original model for 20 more epochs, which costs less than that. On the other hand, the improvements from using $N_{\text{labeled}} = 4000$ or higher are significant and cannot be mimicked by training for longer.

Given a total privacy budget $\varepsilon_{\text{limit}}$, how should we decide among ε_{dp}, $\varepsilon_{\text{dpPCA}}$, and $\varepsilon_{\text{support}}$? Empirically, we observe that $\varepsilon_{\text{dpPCA}}$ can be set to a small value (e.g., 0.3) without substantially affecting the results. Allocating between ε_{dp} and $\varepsilon_{\text{support}}$ is addressed in Figure 5c, which varies those parameters with $\varepsilon_{\text{limit}}$ fixed to 5.0. When the budget for gathering $\mathcal{D}_{\text{labeled}}$ is low, say N_{labeled} is around 1000, it is preferable to pick a later DP checkpoint, consuming a higher ε_{dp} and lower $\varepsilon_{\text{support}}$. On the other hand, when allowed to label more instances from the public data, we should use an earlier DP checkpoint (with a lower ε_{dp}) and choose better public samples with respect to the private data.

6   CONCLUSION

In addition to creating new baselines for DP image classifiers by fine-tuning on public data, we introduce two algorithms – DIVERSEPUBLIC and NEARPRIVATE – to perform fine-tuning in a privacy-

aware way. We conduct experiments showing that these algorithms bring DP object recognition closer to practicality, improving on the aforementioned benchmarks. We hope that this work will encourage further research into techniques for making differential privacy more useful in practice, and we hope that the techniques we propose here will be helpful to existing practitioners.

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. ACM, 2016.

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

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. *Foundations and Trends® in Theoretical Computer Science*, 9(3–4):211–407, 2014a.

Cynthia Dwork, Kunal Talwar, Abhradeep Thakurta, and Li Zhang. Analyze gauss: optimal bounds for privacy-preserving principal component analysis. In *Proceedings of the forty-sixth annual ACM symposium on Theory of computing*, pp. 11–20. ACM, 2014b.

Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. In *International Conference on Learning Representations*, 2018.

Joel Hestness, Sharan Narang, Newsha Ardalani, Gregory Diamos, Heewoo Jun, Hassan Kianinejad, Md Patwary, Mostofa Ali, Yang Yang, and Yanqi Zhou. Deep learning scaling is predictable, empirically. *arXiv preprint arXiv:1712.00409*, 2017.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. Technical report, Citeseer, 2009.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 1998.

Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436, 2015.

H. Brendan McMahan, Galen Andrew, Ulfar Erlingsson, Steve Chien, Ilya Mironov, Nicolas Papernot, and Peter Kairouz. A general approach to adding differential privacy to iterative training procedures. *arXiv preprint arXiv:1812.06210*, 2018.

Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. *NIPS Workshop on Deep Learning and Unsupervised Feature Learning*, 2011.

Hieu T Nguyen and Arnold Smeulders. Active learning using pre-clustering. In *Proceedings of the twenty-first international conference on Machine learning*, pp. 79. ACM, 2004.

Avital Oliver, Augustus Odena, Colin A Raffel, Ekin Dogus Cubuk, and Ian Goodfellow. Realistic evaluation of deep semi-supervised learning algorithms. In *Advances in Neural Information Processing Systems*, pp. 3235–3246, 2018.

Maithra Raghu, Justin Gilmer, Jason Yosinski, and Jascha Sohl-Dickstein. Svcca: Singular vector canonical correlation analysis for deep learning dynamics and interpretability. In *Advances in Neural Information Processing Systems*, pp. 6076–6085, 2017.

Burr Settles. Active learning literature survey. Technical report, University of Wisconsin-Madison Department of Computer Sciences, 2009.

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.

Chhavi Yadav and Léon Bottou. Cold case: The lost mnist digits. Technical report, arxiv:1905.10498, may 2019.

Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016.