RECYCLING SCRAPS: IMPROVING PRIVATE LEARNING BY LEVERAGING INTERMEDIATE CHECKPOINTS

Anonymous authors
Paper under double-blind review

ABSTRACT

All state-of-the-art (SOTA) differentially private machine learning (DP ML) methods are iterative in nature, and their privacy analyses allow publicly releasing the intermediate training checkpoints. However, DP ML benchmarks, and even practical deployments, typically use only the final training checkpoint to make predictions. In this work, for the first time, we comprehensively explore various methods that aggregate intermediate checkpoints to improve the utility of DP training. Empirically, we demonstrate that checkpoint aggregations provide significant gains in the prediction accuracy over the existing SOTA for CIFAR10 and StackOverflow datasets, and that these gains get magnified in settings with periodically varying training data distributions. For instance, we improve SOTA StackOverflow accuracies to 22.7% (+0.43% absolute) for $\varepsilon = 8.2$, and 23.84% (+0.43%) for $\varepsilon = 18.9$. Theoretically, we show that uniform tail averaging of checkpoints improves the empirical risk minimization bound compared to the last checkpoint of DP-SGD. Lastly, we initiate an exploration into estimating the uncertainty that DP noise adds in the predictions of DP ML models. We prove that, under standard assumptions on the loss function, the sample variance from last few checkpoints provides a good approximation of the variance of the final model of a DP run. Empirically, we show that the last few checkpoints can provide a reasonable lower bound for the variance of a converged DP model.

1 INTRODUCTION

Machine learning models can unintentionally memorize sensitive information about the data they were trained on, which has led to numerous attacks that extract private information about the training data (Ateniese et al., 2013; Fredrikson et al., 2014; 2015; Carlini et al., 2019; Shejwalkar et al., 2021; Carlini et al., 2021; 2022). For instance, membership inference attacks (Shokri et al., 2017) can infer whether a target sample was used to train a given ML model, while property inference attacks (Melis et al., 2019; Mahloujifar et al., 2022) can infer certain sensitive properties of the training data. To address such privacy risks, literature has introduced various approaches to privacy-preserving ML (Nasr et al., 2018; Shejwalkar & Houmansadr, 2021; Tang et al., 2022). In particular, iterative techniques like differentially private stochastic gradient decent (DP-SGD) (Song et al., 2013; Bassily et al., 2014; Abadi et al., 2016b; McMahan et al., 2017) and DP Follow The Regularized Leader (DP-FTRL) (Kairouz et al., 2021) have become the state-of-the-art for training DP neural networks.

For establishing benchmarks, prior works in DP ML (Abadi et al., 2016b; McMahan et al., 2017; 2018; Thakkar et al., 2019; Erlingsson et al., 2019; Wang et al., 2019b; Zhu & Wang, 2019; Balle et al., 2020; Erlingsson et al., 2020; Papernot et al., 2020; Tramer & Boneh, 2020; Andrew et al., 2021; Kairouz et al., 2021; Amid et al., 2022; De et al., 2022; Feldman et al., 2022) use only the final model output by the DP algorithm. This is also how DP models are deployed in practice (Ramaswamy et al., 2020; McMahan et al., 2022). However, the privacy analyses for the techniques used allow releasing/using all of the intermediate training checkpoints. In this work, we comprehensively study various methods that leverage intermediate checkpoints to 1) improve the utility of DP training, and 2) quantify the uncertainty in DP ML models that is due to the DP noise.

Accuracy improvement using checkpoints: We propose two classes of aggregation methods based on aggregating the *parameters* of checkpoints, or their *outputs*. We provide both theoretical and empirical analyses for our aggregation methods. Theoretically, we show that excess empirical risk of the final checkpoint of DP-SGD is $\log(n)$ times more than that of the weighted average of the past k checkpoints. Here, n is the size of dataset. Empirically, we demonstrate significant top-1 accuracy gains due to our aggregations for image classification (CIFAR10) and a next word prediction (StackOverflow) tasks. Specifically, we show that our checkpoints aggregations achieve absolute (relative) prediction accuracy improvements of 3.79% (7.2%) at $\varepsilon = 1$ for CIFAR10 (DP-SGD), and 0.43% (1.9%) at $\varepsilon = 8.2$ for the StackOverflow (DP-FTRLM) SOTA baselines, respectively. We also show that our aggregations significantly reduce the variance in the performance of DP models over training. Finally, we show that these benefits further magnify in more practical settings with periodically varying training data distributions. For instance, we note absolute (relative) accuracy gains of 17.4% (28.6%) at $\varepsilon = 8$ for CIFAR10 over DP-SGD baseline in such a setting.

Uncertainty quantification using checkpoints: There are various sources of randomness in a ML training pipeline (Abdar et al., 2021), e.g., choice of initial parameters, dataset, batching, etc. This randomness induces uncertainty in the predictions made using such ML models. In critical domains, e.g., medical diagnosis, self-driving cars and financial market analysis, failing to capture the uncertainty in such predictions can have undesirable repercussions. DP learning adds an additional source of randomness by injecting noise at every training round. Hence, it is paramount to quantify reliability of the DP models, e.g., by quantifying the uncertainty in their predictions.

To this end, we take the first steps towards *quantifying the uncertainty that DP noise adds* to DP ML training. As prior work, Karwa & Vadhan (2017) develop finite sample confidence intervals but for the simpler Gaussian mean estimation problem. Various methods exist for uncertainty quantification in ML-based systems (Mitchell, 1980; Roy et al., 2018; Begoli et al., 2019; Hubschneider et al., 2019; McDermott & Wikle, 2019; Tagasovska & Lopez-Paz, 2019; Wang et al., 2019a; Nair et al., 2020; Ferrando et al., 2022). However, these methods either use specialized (or simpler) model architectures to facilitate uncertainty quantification, or are not directly applicable to quantify the uncertainty in DP ML due to DP noise. For e.g., a common way of uncertainty quantification (Barrientos et al., 2019; Nissim et al., 2007; Brawner & Honaker, 2018; Evans et al., 2020) that we call the *independent runs* method, needs k independent (bootstrap) runs of the ML algorithm. However, repeating a DP ML algorithm multiple times can incur significant privacy and computation costs.

To address the above issue, we propose to use the last k checkpoints of a single run of a DP ML algorithm as a proxy for the k final checkpoints from independent runs. This does not incur any additional privacy cost to the DP ML algorithm. Furthermore, it is useful in practice as it does not incur additional training compute, and can work with any algorithm having intermediate checkpoints.

Theoretically, we consider using the sample variance of a statistic $f(\theta)$ at checkpoints $\theta_{t_1}, \ldots, \theta_{t_k}$ as an estimator of the variance of $f(\theta_{t_k})$, i.e., the statistic at the final checkpoint, and give a bound on the bias of this estimator. As expected, our bound on the bias decreases as the "*burn-in*" time t_1 as well as the time between checkpoints both increase. Intuitively, our proof shows that (i) as the burn-in time increases, the marginal distribution of each θ_{t_i} approaches the distribution of θ_{t_k}, and (ii) as the time between checkpoints increases, any pair $\theta_{t_i}, \theta_{t_j}$ approaches pairwise independence. Both (i) and (ii) are proven via a mixing time bound, which shows that starting from any point distribution θ_0, the Markov chain given by DP-SGD approaches its stationary distribution at a certain rate. Empirically, we show our method provides reasonable lower bounds on the uncertainty quantified using the more accurate (but privacy and computation intensive) method that uses independent runs.

Related work on Checkpoint aggregations: (Chen et al., 2017; Izmailov et al., 2018) explore checkpoint aggregation methods to improve performance in (non-DP) ML settings, but observe negligible performance gains. To our knowledge, De et al. (2022) is the only work in the DP ML literature that uses intermediate checkpoints post training. They apply an exponential moving average (EMA) over the checkpoints of DP-SGD, and note non-trivial gains in performance. However, we propose various aggregation methods that outperform EMA on standard benchmarks.

2 Improving accuracy by aggregating DP training checkpoints

In this section, we describe our checkpoint aggregation methods, followed by the experimental setup we use for evaluation. Next, we detail our experimental results that demonstrate the significant gains in accuracy of DP ML models due to checkpoints aggregations.

DP Preliminaries: Differential Privacy (DP) (Dwork et al., 2006) is a notion to quantify the privacy leakage from the outputs of a data analysis procedure. A randomized algorithm $M : \mathcal{D}^* \to \mathcal{Y}$ is (ε, δ)-DP if, for all neighbouring datasets $D, D' \in \mathcal{D}^*$ (i.e., datasets that differ in one data sample) and all measurable sets of outputs $S \subseteq \mathcal{Y}$, we have $\mathbb{P}\left[M(D) \in S\right] \leq e^{\varepsilon} \cdot \mathbb{P}\left[M(D') \in S\right] + \delta$. We add Gaussian noise to functions of *bounded sensitivity* to ensure DP. We also use *DP's post-processing property*, i.e., any analysis on the outputs of a DP algorithm does not worsen its DP guarantees.

We consider two state-of-the-art DP ML algorithms: 1) *DP-SGD* for the *central learning* setting, i.e., when all data is pooled at a central location, e.g., a server, and 2) *DP-FTRL* for the *federated learning* (FL) setting, i.e., when all private data is distributed across remote collaborative devices. The privacy analyses for both of these techniques involves *composition* across training rounds, allowing the release of all intermediate checkpoints computed during training.

2.1 Checkpoints aggregation methods

We propose two types of aggregation methods: *parameter aggregation*, and *output aggregation*. Parameter aggregations compute a function of the parameters of intermediate checkpoints from a DP run, and then use the resulting aggregate parameters for inference. On the other hand, output aggregations compute a function of the outputs of the intermediate checkpoints, and use it for making predictions. These classes can encompass a vast majority of possible aggregation algorithms, but for brevity, we experiment with two algorithms in each of the classes: *exponential moving average* and *uniform past_k average* that aggregate parameters, and *predictions average* and *labels majority vote* that aggregate outputs. Note that all of our aggregation algorithms *post-process* the checkpoints, and hence, do not incur any additional privacy cost. Additionally, our aggregation methods are general, i.e., they are applicable to any training algorithm that computes intermediate checkpoints.

2.1.1 Parameter aggregation methods

Exponential moving average (EMA): EMA has been previously used (Tan & Le, 2019; Brock et al., 2021) to improve the performance of ML models at inference time. Starting from the last checkpoint of the run, EMA assigns exponentially decaying weights to each of previous checkpoints; the weights are a function of the EMA *coefficient* β_t at step t. During training, at each step t, EMA maintains a moving average θ^t_{ema} that is a weighted average of θ^{t-1}_{ema} and the t^{th} checkpoint, θ^t. This is formalized as follows: $\theta^t_{ema} = (1 - \beta_t) \cdot \theta^{t-1}_{ema} + \beta_t \cdot \theta^t$. Following (Tan & Le, 2019; De et al., 2022), we use a warm-up schedule for the EMA coefficient as: $\beta_t = \min\left(\beta, \frac{1+t}{10+t}\right)$.

Uniform past_k average (UPA): For step t of training, UPA computes the mean of the past k checkpoints, i.e., checkpoints from steps $[t-(k-1), t]$. We formalize this as: $\theta^{t,k}_{upa} = \frac{1}{k}\sum_{i=t-(k-1)}^{t} \theta^i$.

2.1.2 Output aggregation methods

Output predictions averaging (OPA): For a given test sample $\mathbf{x}$, OPA first computes prediction vectors $f_{\theta^i}(\mathbf{x})$ of the past k checkpoints, i.e., checkpoints from steps $\in [t-(k-1), t]$, averages the prediction vectors, and computes argmax of the average vector as the final output label. We formalize OPA as $\hat{y}_{opa}(\mathbf{x}) = \text{argmax}\left(\frac{1}{k}\sum_{i=t-(k-1)}^{t} f_{\theta^i}(\mathbf{x})\right)$.

Output labels majority vote (OMV): For a given test sample $\mathbf{x}$, OMV computes output prediction vectors for $\mathbf{x}$ and the corresponding labels, i.e., argmax $f_{\theta^i}(\mathbf{x})$. Finally, it uses the majority label among the k labels (breaking ties arbitrarily) as its final output label. We formalize OPA as $\hat{y}_{omv}(\mathbf{x}) = \text{Majority}\big(\text{argmax}(f_{\theta^i}(\mathbf{x}))^{t}_{i=t-(k-1)}\big)$.

2.2 Experiments

2.2.1 Experimental setup

Datasets: We evaluate our checkpoints aggregation algorithms on three benchmark datasets in two different settings: CIFAR10 (Krizhevsky et al., 2009) and CIFAR100 for image classification, and simulated federated learning over StackOverflow (Kaggle, 2018) for next word prediction. Privacy guarantees are sample-level for CIFAR tasks and user-level for StackOverflow (as it is keyed by users). For experiments with DP, we fix the privacy parameter δ to 10^{-5} on CIFAR-10/CIFAR-100, and 10^{-6} on StackOverflow, ensuring that $\delta < n^{-1}$, where n is the number of examples in CIFAR10/CIFAR-100 and the number of users in StackOverflow.

Model architectures, and training details: Following the setup of the state-of-the-art (SOTA) in (De et al., 2022), we train a WideResNet-16-4 with depth 16 and width 4 using DP-SGD (Abadi et al., 2016b) in JAXline (Babuschkin et al., 2020) for $\varepsilon \in \{1, 8\}$. For CIFAR100, we follow (De et al., 2022) and *fine-tune the last, classifier layer of* a WideResNet-28-10 pre-trained on ImageNet data. For StackOverflow, we follow SOTA in (Kairouz et al., 2021; Denisov et al., 2022) and train a one-layer LSTM using DP-FTRL in TFF (Abadi et al., 2016a) for $\varepsilon \in \{8.2, 18.9\}$. For UPA, OPA and OMV, we treat k, the number of checkpoints to aggregate, as a hyperparameter and tune it using validation data. All our results are computed over 5 runs of each setting. We provide additional details of our experimental setup in Section B.1.

2.2.2 Experimental results

First, we discuss the results for original datasets and then for more practical periodic distribution shifting datasets. Below, the tables present results for the final training round/step, i.e., for the aggregates computed *until* the last step/round, while plots show results over the last k rounds for some k much smaller than total number of training steps/rounds. For StackOverflow, due to large size of its test data, we provide plots for accuracy on validation data and tables with test accuracy.

CIFAR10 results: Table 1 and the left-most two plots in Figure 1 present the accuracy gains in CIFAR10 for $\varepsilon \in \{1, 8\}$. For $\varepsilon = 1$, OPA provides the maximum accuracy gain of 3.79%, while for $\varepsilon = 8$, EMA provides maximum gain of 2.23%.[1] We note from Figure 1 that all checkpoints aggregations improve accuracy for all the training steps of DP-SGD for both ε's. Next, note from Figure 1 that the accuracy of baseline DP-SGD has a high variance across training steps, i.e., based on the hyperparameters, DP-SGD can produce bad/good models which can be undesirable in practice. However, checkpoints aggregations significantly reduce the variance in accuracy of models, and therefore, increase the confidence in the final DP-SGD model.

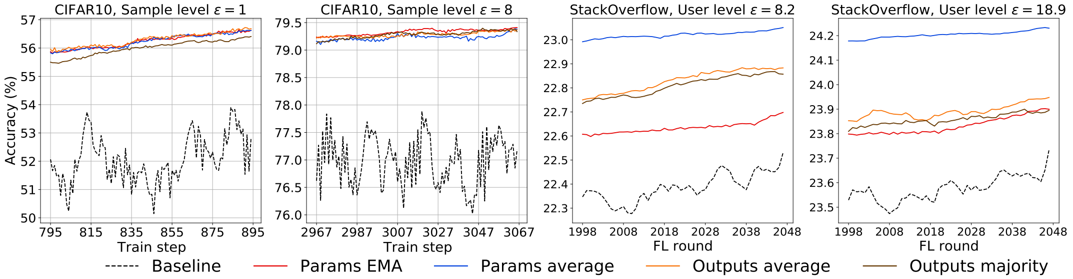

Figure 1: Accuracy improvements due to checkpoints aggregation methods for DP-SGD trained CIFAR10 and DP-FTRL trained StackOverflow.

StackOverflow results: Table 1 and the rightmost two plots in Figure 1 present the accuracy gains in StackOverflow for $\varepsilon \in \{18.9, 8.2\}$; Figure 5 present the results for $\varepsilon = \infty$. The maximum accuracy gains are 0.57%, 0.43%, and 0.42% for ε of ∞, 18.9, and 8.2, respectively. We observe that UPA aggregation always provides the best accuracy. Note that these improvements are very significant, because there are 10,004 classes in StackOverflow data. Similarly for CIFAR10, Figures 1 and 5 show that checkpoints aggregations improve accuracy for all training rounds by large margins and also significantly reduce the variance of model accuracy across training rounds. Note that *these improvements are on top of the momentum optimizer* used during training.

CIFAR100 results: Due to space constraints, we defer the results to Table 7 in Appendix B, and discuss the major observations here. First, we improve the SOTA baseline of De et al. (2022)[2] from

[1] Note that EMA (with default EMA coefficients from non-private settings) has also been used in De et al. (2022) to improve DP-SGD's performance. We observe that even a coarse-grained tuning of the EMA coefficient provides significant accuracy gains. For instance, for CIFAR10, our tuned EMA coefficients outperform those of De et al. (2022) by $\sim 0.6\%$ and 0.35% for ε of 1 and 8, respectively.

[2] For CIFAR100, Bu et al. (2022) achieve SOTA accuracy of 83% at $\varepsilon = 1$ using 303M parameters vision transformers pre-trained on ImageNet21k. However, due to computational constraints, we use the 36M pa-

Table 1: Test accuracy gains for original CIFAR10 and StackOverflow.

Privacy level	Aggregation algorithms				
	None (Baseline)	Parameters aggregation		Outputs aggregation	
		EMA	UPA	OPA	OMV
CIFAR10; DP-SGD; sample-level privacy					
$\varepsilon = 8$	77.18 ± 1.46	**79.41 ± 0.51**	79.39 ± 0.52	79.4 ± 0.59	79.34 ± 0.54
$\varepsilon = 1$	52.83 ± 2.17	56.61 ± 0.91	56.62 ± 0.89	**56.68 ± 0.89**	56.4 ± 0.69
StackOverflow; DP-FTRL; user-level privacy					
$\varepsilon = \infty$	25.24 ± 0.16	25.72 ± 0.02	**25.81 ± 0.02**	25.79 ± 0.01	25.78 ± 0.01
$\varepsilon = 18.9$	23.41 ± 0.08	23.56 ± 0.02	**23.84 ± 0.01**	23.6 ± 0.02	23.57 ± 0.02
$\varepsilon = 8.2$	22.28 ± 0.08	22.43 ± 0.04	**22.7 ± 0.03**	22.57 ± 0.04	22.52 ± 0.04

70.6% to 75.51% at $\varepsilon = 1$, and from 77.6% to 80.8% at $\varepsilon = 8$. To achieve these gains, *we perform fine-tuning over the EMA checkpoint of ImageNet-trained WRN-28-10 instead of the final checkpoint* as in (De et al., 2022). Subsequently, we observe that for fine-tuning using CIFAR100, checkpoint aggregations provide small accuracy gains: for both ε of 1 and 8, we observe maximum gains are 0.11% due to UPA and OPA, respectively.

2.2.3 Accuracy improvements in periodic distribution shifting settings

In many real-world settings, for instance, in federated learning (FL) settings, training data distribution may vary over time. Zhu et al. (2021) demonstrate the adverse impacts of distribution shifts in training data on the performances of resulting FL models. Considering the practical significance of such distribution shifts, we consider settings where the training data distribution has *diurnal* variations, i.e., it is a function of two oscillating distributions, $\mathcal{D}_1$ and $\mathcal{D}_2$; Figure 2 shows these distributions. Such a scenario commonly occurs in FL training, e.g., when a model is trained using FL with client devices participating from two significantly different time zones.

Experimental setup: Following Zhu et al. (2021), we consider a setting where training data is a combination of clients/samples drawn from two disjoint data distributions, which oscillate over time as Figure 2 shows. Here the probability of sampling from, e.g., $\mathcal{D}_1$, at time t is $p(\mathcal{D}_1, t) = \left|2\frac{t \bmod T}{T} - 1\right|$, where T is the period of oscillation of $\mathcal{D}_{\{1,2\}}$. To simulate such diurnal distribution, for CIFAR datasets, we design $\mathcal{D}_{\{1,2\}}$ such that $\mathcal{D}_1$ and $\mathcal{D}_2$ respectively contain the data from even and odd classes of the original data. For StackOverflow, we design $\mathcal{D}_{\{1,2\}}$ such that $\mathcal{D}_1$ only has the questions from each user, while $\mathcal{D}_2$ only has answers from them. Then, we draw clients from $\mathcal{D}_{\{1,2\}}$. Apart from data distribution, the rest of experimental setup is the same as before. We use test and validation data same as for the original StackOverflow setting.

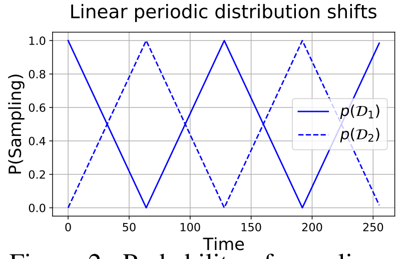

Figure 2: Probability of sampling data from distributions $\mathcal{D}_1$ and $\mathcal{D}_2$.

CIFAR10 results: Table 2 and Figure 3 (left two plots) show accuracy gains for diurnal CIFAR10. We note very large accuracy gains for both $\varepsilon \in \{1, 8\}$; the absolute accuracy gains are 7.45% and 17.37%, respectively, and are due to OPA. Observe that, such diurnal settings have large variances in accuracy across training steps, but our aggregation methods almost completely eliminate the variances. We note that the improvements in diurnal settings are significantly more than that in original CIFAR10. This is because in diurnal settings, the variances in model accuracy over training steps is very large, and hence, the benefits of checkpoints aggregations magnify in these settings.

StackOverflow results: Table 2 and Figure 3 (rightmost two plots) present accuracy gains for diurnal StackOverflow. The maximum accuracy gains are 0.09%, 0.44%, and 0.51% for ε of ∞, 18.9, and 8.2, respectively. In contrast to the diurnal CIFAR10 case, improvements in diurnal StackOverflow and StackOverflow are similar. This is because the two distributions in diurnal CIFAR10 (completely different images from even/odd classes) are significantly farther apart compared to the two distributions in diurnal StackOverflow (text from questions/answers).

rameter WRN pre-trained on downsampled ImageNet1k from De et al. (2022). Moreover, similar to De et al. (2022), we observe that in small ε regimes, fine-tuning only the classifier layer provides better accuracy.

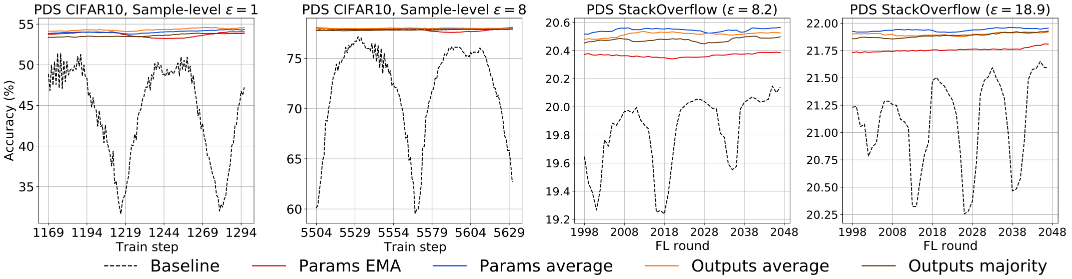

Figure 3: Accuracy gains due to checkpoints aggregations for DP-SGD trained *periodic distribution shifting* (PDS) CIFAR10 (test data) and DP-FTRL trained PDS StackOverflow (validation data).

Table 2: Test accuracy gains for *periodic distribution shifting* (PDS) CIFAR10 and StackOverflow.

Privacy level	Aggregation algorithms				
	None (Baseline)	Parameters aggregation		Outputs aggregation	
		EMA	UPA	OPA	OMV
PDS CIFAR10; DP-SGD; sample-level privacy					
$\varepsilon = 8$	60.74 ± 1.75	78.24 ± 0.92	77.92 ± 0.89	**78.27 ± 0.84**	77.99 ± 0.94
$\varepsilon = 1$	47.13 ± 1.76	54.04 ± 0.81	54.35 ± 0.9	**54.58 ± 0.82**	54.03 ± 1.08
PDS StackOverflow; DP-FTRL; user-level privacy					
$\varepsilon = \infty$	23.89 ± 0.14	23.92 ± 0.12	**23.98 ± 0.02**	23.87 ± 0.01	23.91 ± 0.07
$\varepsilon = 18.9$	21.6 ± 0.13	21.82 ± 0.07	**22.04 ± 0.11**	21.99 ± 0.13	21.95 ± 0.16
$\varepsilon = 8.2$	20.24 ± 0.29	20.36 ± 0.06	**20.75 ± 0.05**	20.67 ± 0.03	20.72 ± 0.16

CIFAR100 results: Due to space constraints, we defer full results in Table 8 in Appendix B. For CIFAR100 also, we observe significant gains due to checkpoints aggregations in PDS setting: for ε of 1 and 8, we note maximum gains of 4.96% (due to UPA) and 3.37% (due to EMA), respectively. Similar to CIFAR10, the gains due to checkpoints aggregation are significantly higher for PDS-CIFAR100 than for non-PDS CIFAR100.

Table 3: Accuracy gains on test data due to *data-dependent* checkpoints aggregations for StackOverflow (SO) and periodic distribution shifting StackOverflow (PDS-SO) trained using DP-FTRL.

Privacy level	Dataset	Baseline	Best of data-independent	Best of data-dependent
$\varepsilon = \infty$	SO	25.24 ± 0.16	25.81 ± 0.02 (UPA)	**25.97 ± 0.01** (UPA)
	PDS-SO	23.89 ± 0.14	23.98 ± 0.02 (UPA)	**24.18 ± 0.02** (EMA)
$\varepsilon = 18.9$	SO	23.41 ± 0.08	23.84 ± 0.01 (UPA)	**23.92 ± 0.03** (UPA)
	PDS-SO	21.60 ± 0.13	22.04 ± 0.11 (UPA)	**22.18 ± 0.07** (EMA)
$\varepsilon = 8.2$	SO	22.28 ± 0.08	22.70 ± 0.03 (UPA)	**22.80 ± 0.05** (UPA)
	PDS-SO	20.24 ± 0.29	20.75 ± 0.05 (UPA)	**20.90 ± 0.06** (UPA)

2.3 Data-dependent checkpoint selection for aggregration

So far, our aggregated methods operated over a fixed sequence of checkpoints, e.g., past k. However, when some public data similar to the private training data is available, we propose a *data-dependent* checkpoint selection method for aggregation: aggregate the checkpoints that perform the best on the (held-out) public data. We expect this to improve over data-independent aggregation schemes. We validate this hypothesis for StackOverflow dataset. We use 10,000 random samples (less than 0.01% of the training data) from the StackOverflow validation data as the held-out public data. Note that this is disjoint from the validation data that we use for hyperparameter tuning. We omit the CIFAR datasets from this evaluation due to the lack of availability of additional similar data.

We compute accuracy of all training checkpoints on the held-out data, find the best k checkpoints, and aggregate them as detailed above. We tune the hyperparameter k using disjoint validation data. Table 3 presents the gains due to data-dependent aggregations over the baseline and data-independent aggregations; Tables 5 and 6 in Appendix B present full results.[3] We see that data-

[3] Note that only the data-dependent aggregations use public data, so this comparison is just for illustration.

dependent aggregations outperform data-independent aggregations for all ε's. For instance, at ε of 8.2, accuracies of baseline, data-independent and data-dependent aggregations are 22.28%, 22.7% and 22.8%, respectively. For ε's ∞ and 18.9, aggregating best past k checkpoints outperform aggregating past k checkpoints (baseline) by 0.16% (0.73%) and 0.08% (0.51%), respectively. We make the same observation for diurnal PDS StackOverflow dataset—the maximum accuracy gains due to data-dependent aggregations are 0.29%, 0.58%, and 0.66% for ε of ∞, 18.9, and 8.2, respectively.

2.4 Improved Excess Risk via Tail Averaging

In this section, we formally show that checkpoint aggregations like uniform tail averaging provably improves the privacy/utility trade-offs, compared to that of the last checkpoint of DP-SGD. To formalize the problem, we define the following notation: Consider a data set $D = \{d_1, \ldots, d_n\}$ and a loss function $\mathcal{L}(\theta; D) = \frac{1}{n}\sum\limits_{i=1}^{n} \ell(\theta; d_i)$, where each of the loss function ℓ is convex and L-Lipschitz in the first parameter, and $\theta \in \mathcal{C}$ with $\mathcal{C} \subseteq \mathbb{R}^p$ being a convex constraint set. We analyze the following variant of DP-SGD, which is guaranteed to be ρ-zCDP[4].

1. $\theta_0 \leftarrow \mathbf{0}^p$.
2. For $t \in [T]$, $\theta_{t+1} \leftarrow \Pi_{\mathcal{C}}\left(\theta_t - \eta_t\left(\nabla\mathcal{L}(\theta_t; D) + b_t\right)\right)$, where $b_t \sim \mathcal{N}\left(0, \frac{L^2 T}{2n\rho}\mathbb{I}_{p\times p}\right)$ and $\Pi_{\mathcal{C}}(\cdot)$ being the ℓ_2-projection onto the set $\mathcal{C}$.

We will provide the utility guarantee for this algorithm by directly appealing to the result of Shamir & Zhang (2013). For a given $\alpha \in (0, 1)$, UPA (Section 2.1.1) corresponds to the average of the last αT models, i.e., $\theta^{\mathsf{priv}}_{\mathsf{upa}} = \frac{1}{\alpha T}\sum\limits_{t=(1-\alpha)T+1}^{T} \theta_t$. One can also consider *polynomial-decay averaging* (PDA) with parameter $\gamma \geq 0$, defined as follows: $\theta^{\mathsf{priv}}_{\mathsf{pda}}[t] = \left(1 - \frac{\gamma+1}{t+\gamma}\right)\theta^{\mathsf{priv}}_{\mathsf{pda}}[t-1] + \frac{\gamma+1}{t+\gamma}\cdot\theta_t$.

For $\gamma = 0$, PDA matches UPA over all iterates. As γ increases, PDA places more weight on later iterates; in particular, if $\gamma = cT$, the averaging scheme is very similar to EMA (Section 2.1.1), since as $t \rightarrow T$ the decay parameter $\frac{\gamma+1}{t+\gamma}$ approaches a constant $\frac{c}{c+1}$. In that sense, PDA can be viewed as a method interpolating between UPA and EMA.

Theorem 2.1 (Adapted from Theorems 2 and 4 of Shamir & Zhang (2013))**.** *Consider the DP-SGD algorithm above, and the associated parameters. Then there exists choice of learning rate η_t and the number of time steps T s.t. the following are true for $\alpha = \Theta(1)$:*

$$\mathbb{E}\left[\mathcal{L}\left(\theta^{\mathsf{priv}}_{\mathsf{upa}}; D\right)\right] - \min_{\theta\in\mathcal{C}}\mathcal{L}(\theta; D) = \mathcal{O}\left(\frac{L\|\mathcal{C}\|_2\sqrt{p}}{n\rho}\right), \textit{ and } \mathbb{E}\left[\mathcal{L}(\theta_T; D)\right] - \min_{\theta\in\mathcal{C}}\mathcal{L}(\theta; D) = \mathcal{O}\left(\frac{L\|\mathcal{C}\|_2\sqrt{p}\log(n)}{n\rho}\right).$$

Furthermore, for $\gamma = \Theta(1)$, we have, $\mathbb{E}\left[\mathcal{L}\left(\theta^{\mathsf{priv}}_{\mathsf{pda}}[T]; D\right)\right] - \min\limits_{\theta\in\mathcal{C}}\mathcal{L}(\theta; D) = \mathcal{O}\left(\frac{L\|\mathcal{C}\|_2\sqrt{p}}{n\rho}\right)$.

Proof. If we choose $T = \lceil n\rho \rceil$ and set η_t appropriately, the proof of Theorem 2 (Shamir & Zhang, 2013) implies the following for $\theta^{\mathsf{priv}}_{\mathsf{upa}}$: $\mathbb{E}\left[\mathcal{L}\left(\theta^{\mathsf{priv}}_{\mathsf{upa}}; D\right)\right] - \min\limits_{\theta\in\mathcal{C}}\mathcal{L}(\theta; D) = O\left(\frac{L\|\mathcal{C}\|_2\sqrt{p}}{n\rho}\log\left(\frac{1}{\alpha}\right)\right)$.

Setting $\alpha = \Theta(1)$ gives the theorem's first part, and $\alpha T = 1$, i.e., $1/\alpha = T = \lceil n\rho \rceil$ gives the second. The third follows from modifying Theorem 4 of Shamir & Zhang (2013) for the convex case (see the end of Section 4 of Shamir & Zhang (2013) for details). □

The excess empirical risk for θ_T is higher by factor of $\log(n)$ in comparison to $\theta^{\mathsf{priv}}_{\mathsf{upa}}$ and $\theta^{\mathsf{priv}}_{\mathsf{pda}}[T]$. For the step size selections typically used in practice (e.g., fixed or inverse polynomial step sizes), the last iterate will suffer from the extra $\log(n)$ factor, and we do not know how to avoid it. Furthermore, Harvey et al. (2019) showed that this is unavoidable in the non-private, high probability regime.[5]

[4] Using Bun & Steinke (2016), it is easy to convert the privacy guarantee to an (ε, δ)-DP guarantee.

[5] Jain et al. (2021) show that for carefully chosen step sizes, the logarithmic factor can be removed, and Feldman et al. (2020) extend this analysis to a DP-SGD variant with varying batch sizes. Unlike those methods, averaging can be done as post-processing of DP-SGD outputs, rather than a modification of the algorithm.

3 Quantifying uncertainty in predictions due to DP noise

In this section, we discuss our proposal of how to quantify the uncertainty that the differential privacy (DP) noise adds to the outputs of ML algorithms, without additional privacy cost or computation. First, we discuss the issues with the current established approach of uncertainty quantification in ML when used in DP setting, and then discuss our proposal and theoretical results. Finally, we provide experimental results that demonstrate the practical utility of our approach.

(Naive) uncertainty quantification, and its hurdles: As discussed in Section 1, although uncertainty quantification has a long history of research, prior methods, including the most common independent runs method, are not applicable in DP settings. The two major issues with the independent runs method in DP settings are: First, the additional runs of a DP-ML algorithm can incur a significant increase in the privacy cost. Second, the additional computation required in independent runs method can be prohibitive, e.g., in production FL. Hence, it is not feasible to use the naive method for uncertainty quantification in DP settings.

3.1 Two birds, one stone: Our uncertainty quantification proposal

To address the two hurdles discussed above, we propose a simple yet efficient method that leverages intermediate checkpoints computed during a DP run. Specifically, we substitute the k output models from the independent runs method with k checkpoints, e.g., the last k checkpoints, from a single DP run. The rest of the confidence interval computation is the same for both the methods. Specifically, we compute the most likely classification labels of the k models/checkpoints on a test input. Use the set of k output labels as a sample from the *student's t-distribution* and compute a confidence interval of the mean of the distribution. Finally, we use the average of multiple such intervals over multiple test inputs as a proxy for uncertainty of outputs of the DP ML algorithm.

3.1.1 Theory

In this section, we give a theoretical motivation for using checkpoints to estimate the variance of a statistic. We show that under certain assumptions, if we take the sample variance of a statistic $f(\theta)$ computed at the checkpoints $\theta_{t_1}, \ldots, \theta_{t_k}$, in expectation it is a good approximation of the sample variance of the limiting distribution of DP-SGD. As expected, the quality of the approximation improves by increasing the length of the "burn-in" time as well as the time between checkpoints. To simplify the proof, we actually prove the theorem for DP-LD, a continuous-time version of DP-SGD. DP-LD can be defined as follows. We first reformulate (unconstrained) DP-SGD with step size η as: $\widetilde{\theta}_{(t+1)\eta} \leftarrow \widetilde{\theta}_{t\eta} - \eta \nabla \mathcal{L}(\widetilde{\theta}_{t\eta}; D) + b_t, b_t \sim \mathcal{N}(0, 2\eta\sigma^2 \mathbb{I}_{p \times p})$.

Notice that we have reparameterized $\widetilde{\theta}$ so that its subscript refers to the sum of all step-sizes so far, i.e. after t iterations we have $\widetilde{\theta}_{t\eta}$ and not $\widetilde{\theta}_t$. Also notice that the variance of the noise we added is proportional to the step size η. In turn, for any η that divides t, after t/η iterations with step size η, the sum of variances of noises added is $2t\sigma^2$. Now, taking the limit as η goes to 0 of the sequence of random variables $\{\widetilde{\theta}_{t\eta}\}_{t \in \mathbb{Z}_{\geq 0}}$ defined by DP-SGD, we get a continuous sequence $\{\theta_t\}_{t \in \mathbb{R}_{\geq 0}}$. In particular, if we fix some t, then θ_t is the limit as η goes to 0 of $\widetilde{\theta}_t$ defined by DP-SGD with step size η. This sequence is exactly the sequence defined by DP-LD, which is more formally given by the following stochastic differential equation:

$$d\theta_t = -\nabla \mathcal{L}(\theta_t; D)dt + \sigma\sqrt{2}dW_t. \quad (1)$$

Here, W_t is a standard Brownian motion and σ^2 is analogous to the variance of noise added in DP-SGD. In Appendix A we show the following:

Theorem 3.1 (Simplified version of Theorem A.1)**.** *Suppose $\mathcal{L}$ is 1-strongly convex and M-smooth, and $\sigma = 1$ in* (1)*. Let $0 < t_1 < t_2 < \ldots < t_k$ be such that $t_{i+1} \geq t_i + \gamma, \forall i > 0$ and some γ. Let $\{\theta_{t_i} : i \in [k]\}$ be the checkpoints, and $f : \Theta \to [-1, 1]$ be a statistic whose variance we wish to estimate. Let S be the sample variance of f over the checkpoints, and V be the variance of $f(\theta_{t_k})$, i.e. the statistic at the final checkpoint. Then, for some "burn-in" time B that is a function of θ_0, M, p, we have: $|\mathbb{E}[S] - V| = \exp(-\Omega(\min\{t_1, \gamma\} - B))$.*

Intuitively, Theorem A.1 and its proof say the following: (i) As we increase t_1, the time before the first checkpoint, each of the checkpoints' marginal distributions approaches the distribution of θ_{t_k}, and (ii) As we increase γ, the time between checkpoints, the checkpoints' distributions approach pairwise independence. So increasing both t_1 and γ causes our checkpoints to approach k pairwise independent samples from the distribution of θ_{t_k}, i.e., our variance estimator approaches the true variance in expectation. To show both (i) and (ii), we build upon past results from the sampling literature to show a mixing bound of the following form: running DP-LD from any point initialization[6] θ_0, the Rényi divergence between θ_t and the limit as $t \to \infty$ of DP-LD, θ_∞, decays exponentially in t. This mixing bound shows (i) since if t_1 is sufficiently large, then the distributions of all of $\theta_{t_1}, \theta_{t_2}, \ldots, \theta_{t_k}$ are close to θ_∞, and thus close to each other. This also shows (ii) since DP-LD is a Markov chain, i.e. the distribution of θ_{t_j} conditioned on θ_{t_i} is equivalent to the distribution of $\theta_{t_j - t_i}$ if we run DP-LD starting from θ_{t_i} instead of θ_0. So our mixing bound shows that even after conditioning on θ_{t_i}, θ_{t_j} has distribution close to θ_∞. Since θ_{t_j} is close to θ_∞ conditioned on any value of θ_{t_i}, then θ_{t_j} is almost independent of θ_{t_i}.

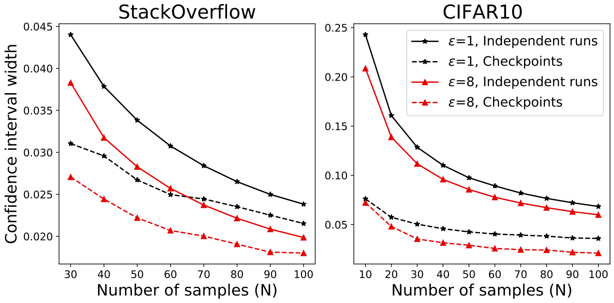

Figure 4: Uncertainty due to DP noise measured using confidence interval widths. We compute the intervals using N bootstrap (independent) runs, and using the last N checkpoints of a single run.

Empirical analysis: We compare the uncertainty quantified using the independent runs method and using our method; experimental setup is the same as in Section 2.2.1. First, for both StackOverflow and CIFAR10 datasets, we do 100 independent training runs. Then to compute uncertainty using the independent runs method, we take the final model from k of these runs (chosen randomly), compute prediction labels for a given input, compute confidence interval widths for the input, and finally, use the average of confidence interval widths of a large number of inputs as the final uncertainty estimate. To compute uncertainty using our checkpoints based method, we instead select our k models to be the last k checkpoints of a random training run, and obtain average confidence intervals as before.

Figure 4 shows the results for StackOverflow and CIFAR10. Plots show intervals averaged over 5 runs, i.e., by sampling k final models 5 times or by using the last k models of 5 random runs. We observe that uncertainty computed using the intermediate checkpoints consistently give a reasonable lower bound on the uncertainty computed using the independent runs.

4 Conclusions

In this work, we explore methods for aggregating intermediate checkpoints to improve the utility of DP training. Using checkpoints aggregation methods, we show significant improvements in prediction accuracy over the SOTA for CIFAR and StackOverflow datasets. We also show that uniform tail averaging of checkpoints improves the ERM bound compared to the last checkpoint of DP-SGD. Lastly, we prove that for some standard loss functions, the sample variance from last few checkpoints provides a good approximation of the variance of the final model of a DP run. We also conduct experiments demonstrating that the last few checkpoints can provide a reasonable lower bound for the variance of a converged DP model. For future work, using checkpoint aggregates during DP training could be an interesting direction to further improve its utility. Leveraging intermediate checkpoints to provide variance estimates for checkpoint aggregates could also be a promising direction.

References

Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, et al. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467*, 2016a.

[6] To our knowledge, all prior mixing results for DP-LD require that the α-Rényi divergence between θ_0 and θ_∞ is finite for some $\alpha > 1$ or some stronger assumption, which cannot be satisfied by a point distribution for θ_0. Thus, our mixing result for DP-LD starting from a point distribution may be of independent interest.

Martín Abadi, Andy Chu, Ian J. Goodfellow, H. Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proc. of the 2016 ACM SIGSAC Conf. on Computer and Communications Security (CCS'16)*, pp. 308–318, 2016b.

Moloud Abdar, Farhad Pourpanah, Sadiq Hussain, Dana Rezazadegan, Li Liu, Mohammad Ghavamzadeh, Paul Fieguth, Xiaochun Cao, Abbas Khosravi, U Rajendra Acharya, et al. A review of uncertainty quantification in deep learning: Techniques, applications and challenges. *Information Fusion*, 76:243–297, 2021.

Ehsan Amid, Arun Ganesh, Rajiv Mathews, Swaroop Ramaswamy, Shuang Song, Thomas Steinke, Vinith M Suriyakumar, Om Thakkar, and Abhradeep Thakurta. Public data-assisted mirror descent for private model training. In *International Conference on Machine Learning*, pp. 517–535. PMLR, 2022.

Galen Andrew, Om Thakkar, Brendan McMahan, and Swaroop Ramaswamy. Differentially private learning with adaptive clipping. In *Advances in Neural Information Processing Systems*, volume 34, pp. 17455–17466, 2021.

Giuseppe Ateniese, Giovanni Felici, Luigi V Mancini, Angelo Spognardi, Antonio Villani, and Domenico Vitali. Hacking smart machines with smarter ones: How to extract meaningful data from machine learning classifiers. *arXiv preprint arXiv:1306.4447*, 2013.

Igor Babuschkin, Kate Baumli, Alison Bell, Surya Bhupatiraju, Jake Bruce, Peter Buchlovsky, David Budden, Trevor Cai, Aidan Clark, Ivo Danihelka, Claudio Fantacci, Jonathan Godwin, Chris Jones, Ross Hemsley, Tom Hennigan, Matteo Hessel, Shaobo Hou, Steven Kapturowski, Thomas Keck, Iurii Kemaev, Michael King, Markus Kunesch, Lena Martens, Hamza Merzic, Vladimir Mikulik, Tamara Norman, John Quan, George Papamakarios, Roman Ring, Francisco Ruiz, Alvaro Sanchez, Rosalia Schneider, Eren Sezener, Stephen Spencer, Srivatsan Srinivasan, Luyu Wang, Wojciech Stokowiec, and Fabio Viola. The DeepMind JAX Ecosystem, 2020. URL `http://github.com/deepmind`.

Borja Balle, Peter Kairouz, Brendan McMahan, Om Thakkar, and Abhradeep Guha Thakurta. Privacy amplification via random check-ins. *Advances in Neural Information Processing Systems*, 33:4623–4634, 2020.

Andrés F. Barrientos, Jerome P. Reiter, Ashwin Machanavajjhala, and Yan Chen. Differentially private significance tests for regression coefficients. *Journal of Computational and Graphical Statistics*, 28(2):440–453, 2019. doi: 10.1080/10618600.2018.1538881. URL `https://doi.org/10.1080/10618600.2018.1538881`.

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

Edmon Begoli, Tanmoy Bhattacharya, and Dimitri Kusnezov. The need for uncertainty quantification in machine-assisted medical decision making. *Nature Machine Intelligence*, 1(1):20–23, 2019.

Leonard Berrada, Soham De, Jamie Hayes, Samuel L Smith, and Borja Balle. JAX-Privacy: Algorithms for Privacy-Preserving Machine Learning in JAX. `https://github.com/deepmind/jax_privacy`, 2022. [Online; accessed 15-September-2022].

James Bradbury, Roy Frostig, Peter Hawkins, Matthew James Johnson, Chris Leary, Dougal Maclaurin, and Skye Wanderman-Milne. JAX: composable transformations of Python+NumPy programs, 2018. URL `http://github.com/google/jax`.

Thomas Brawner and James Honaker. Bootstrap inference and differential privacy: Standard errors for free. *Unpublished Manuscript*, 2018.

Andy Brock, Soham De, Samuel L. Smith, and Karen Simonyan. High-performance large-scale image recognition without normalization. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pp. 1059–1071. PMLR, 2021. URL `http://proceedings.mlr.press/v139/brock21a.html`.

Zhiqi Bu, Jialin Mao, and Shiyun Xu. Scalable and efficient training of large convolutional neural networks with differential privacy. *arXiv preprint arXiv:2205.10683*, 2022.

Mark Bun and Thomas Steinke. Concentrated differential privacy: Simplifications, extensions, and lower bounds. In *Theory of Cryptography Conference*, pp. 635–658. Springer, 2016.

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. In *30th USENIX Security Symposium (USENIX Security 21)*, pp. 2633–2650, 2021.

Nicholas Carlini, Steve Chien, Milad Nasr, Shuang Song, Andreas Terzis, and Florian Tramer. Membership inference attacks from first principles. In *2022 IEEE Symposium on Security and Privacy (SP)*, pp. 1897–1914. IEEE, 2022.

Hugh Chen, Scott Lundberg, and Su-In Lee. Checkpoint ensembles: Ensemble methods from a single training process. *arXiv preprint arXiv:1710.03282*, 2017.

Soham De, Leonard Berrada, Jamie Hayes, Samuel L Smith, and Borja Balle. Unlocking high-accuracy differentially private image classification through scale. *arXiv preprint arXiv:2204.13650*, 2022.

Sergey Denisov, Brendan McMahan, Keith Rush, Adam Smith, and Abhradeep Guha Thakurta. Improved differential privacy for sgd via optimal private linear operators on adaptive streams. *arXiv preprint arXiv:2202.08312*, 2022.

Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Proc. of the Third Conf. on Theory of Cryptography (TCC)*, pp. 265–284, 2006. URL http://dx.doi.org/10.1007/11681878_14.

Úlfar Erlingsson, Vitaly Feldman, Ilya Mironov, Ananth Raghunathan, Kunal Talwar, and Abhradeep Thakurta. Amplification by shuffling: From local to central differential privacy via anonymity. In Timothy M. Chan (ed.), *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2019, San Diego, California, USA, January 6-9, 2019*, pp. 2468–2479. SIAM, 2019. doi: 10.1137/1.9781611975482.151. URL https://doi.org/10.1137/1.9781611975482.151.

Úlfar Erlingsson, Vitaly Feldman, Ilya Mironov, Ananth Raghunathan, Shuang Song, Kunal Talwar, and Abhradeep Thakurta. Encode, shuffle, analyze privacy revisited: Formalizations and empirical evaluation. *CoRR*, abs/2001.03618, 2020. URL https://arxiv.org/abs/2001.03618.

Georgina Evans, Gary King, Margaret Schwenzfeier, and Abhradeep Thakurta. Statistically valid inferences from privacy protected data. *American Political Science Review*, 2020.

Vitaly Feldman, Tomer Koren, and Kunal Talwar. Private stochastic convex optimization: Optimal rates in linear time. In *Proc. of the Fifty-Second ACM Symp. on Theory of Computing (STOC'20)*, 2020.

Vitaly Feldman, Audra McMillan, and Kunal Talwar. Hiding among the clones: A simple and nearly optimal analysis of privacy amplification by shuffling. In *2021 IEEE 62nd Annual Symposium on Foundations of Computer Science (FOCS)*, pp. 954–964. IEEE, 2022.

Cecilia Ferrando, Shufan Wang, and Daniel Sheldon. Parametric bootstrap for differentially private confidence intervals. In *International Conference on Artificial Intelligence and Statistics*, pp. 1598–1618. PMLR, 2022.

Matt Fredrikson, Somesh Jha, and Thomas Ristenpart. Model inversion attacks that exploit confidence information and basic countermeasures. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2015.

Matthew Fredrikson, Eric Lantz, Somesh Jha, Simon Lin, David Page, and Thomas Ristenpart. Privacy in pharmacogenetics: An end-to-end case study of personalized warfarin dosing. In *USENIX Security Symposium*, 2014.

Arun Ganesh and Kunal Talwar. Faster differentially private samplers via rényi divergence analysis of discretized langevin MCMC. *CoRR*, abs/2010.14658, 2020. URL `https://arxiv.org/abs/2010.14658`.

Nicholas J. A. Harvey, Christopher Liaw, Yaniv Plan, and Sikander Randhawa. Tight analyses for non-smooth stochastic gradient descent. In *COLT*, 2019.

Christian Hubschneider, Robin Hutmacher, and J Marius Zöllner. Calibrating uncertainty models for steering angle estimation. In *2019 IEEE intelligent transportation systems conference (ITSC)*, pp. 1511–1518. IEEE, 2019.

Pavel Izmailov, Dmitrii Podoprikhin, Timur Garipov, Dmitry Vetrov, and Andrew Gordon Wilson. Averaging weights leads to wider optima and better generalization. In *34th Conference on Uncertainty in Artificial Intelligence 2018, UAI 2018*, pp. 876–885. Association For Uncertainty in Artificial Intelligence (AUAI), 2018.

Prateek Jain, Dheeraj M. Nagaraj, and Praneeth Netrapalli. Making the last iterate of sgd information theoretically optimal. *SIAM Journal on Optimization*, 31(2):1108–1130, 2021. doi: 10.1137/19M128908X. URL `https://doi.org/10.1137/19M128908X`.

Kaggle. The StackOverflow data. `https://www.kaggle.com/datasets/stackoverflow/stackoverflow`, 2018. [Online; accessed 15-September-2022].

Peter Kairouz, Brendan McMahan, Shuang Song, Om Thakkar, Abhradeep Thakurta, and Zheng Xu. Practical and private (deep) learning without sampling or shuffling. In *International Conference on Machine Learning*, pp. 5213–5225. PMLR, 2021.

Vishesh Karwa and Salil Vadhan. Finite sample differentially private confidence intervals. *arXiv preprint arXiv:1711.03908*, 2017.

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

Saeed Mahloujifar, Esha Ghosh, and Melissa Chase. Property inference from poisoning. In *2022 IEEE Symposium on Security and Privacy (SP)*, pp. 1569–1569. IEEE Computer Society, 2022.

Patrick L McDermott and Christopher K Wikle. Deep echo state networks with uncertainty quantification for spatio-temporal forecasting. *Environmetrics*, 30(3):e2553, 2019.

Brendan McMahan, Abhradeep Thakurta, Galen Andrew, Borja Balle, Peter Kairouz, Daniel Ramage, Shuang Song, Thomas Steinke, Andreas Terzis, Om Thakkar, and Zheng Xu. Federated learning with formal differential privacy guarantees. `https://ai.googleblog.com/2022/02/federated-learning-with-formal.html`, 2022. [Online; accessed 15-September-2022].

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

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.

Luca Melis, Congzheng Song, Emiliano De Cristofaro, and Vitaly Shmatikov. Exploiting unintended feature leakage in collaborative learning. In *2019 IEEE symposium on security and privacy (SP)*, pp. 691–706. IEEE, 2019.

Ilya Mironov. Rényi differential privacy. In *2017 IEEE 30th Computer Security Foundations Symposium (CSF)*, pp. 263–275. IEEE, 2017.

Tom M Mitchell. *The need for biases in learning generalizations*. Department of Computer Science, Laboratory for Computer Science Research …, 1980.

Tanya Nair, Doina Precup, Douglas L Arnold, and Tal Arbel. Exploring uncertainty measures in deep networks for multiple sclerosis lesion detection and segmentation. *Medical image analysis*, 59:101557, 2020.

Milad Nasr, Reza Shokri, and Amir Houmansadr. Machine learning with membership privacy using adversarial regularization. In *Proceedings of the 2018 ACM SIGSAC conference on computer and communications security*, pp. 634–646, 2018.

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.

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

Swaroop Ramaswamy, Om Thakkar, Rajiv Mathews, Galen Andrew, H Brendan McMahan, and Françoise Beaufays. Training production language models without memorizing user data. *arXiv preprint arXiv:2009.10031*, 2020.

Sashank Reddi, Zachary Charles, Manzil Zaheer, Zachary Garrett, Keith Rush, Jakub Konečnỳ, Sanjiv Kumar, and H Brendan McMahan. Adaptive federated optimization. *arXiv preprint arXiv:2003.00295*, 2020.

Abhijit Guha Roy, Sailesh Conjeti, Nassir Navab, and Christian Wachinger. Inherent brain segmentation quality control from fully convnet monte carlo sampling. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 664–672. Springer, 2018.

Ohad Shamir and Tong Zhang. Stochastic gradient descent for non-smooth optimization: Convergence results and optimal averaging schemes. In *International Conference on Machine Learning*, pp. 71–79, 2013.

Virat Shejwalkar and Amir Houmansadr. Membership privacy for machine learning models through knowledge transfer. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 9549–9557, 2021.

Virat Shejwalkar, Huseyin A Inan, Amir Houmansadr, and Robert Sim. Membership inference attacks against nlp classification models. In *NeurIPS 2021 Workshop Privacy in Machine Learning*, 2021.

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, 2017.

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.

Natasa Tagasovska and David Lopez-Paz. Single-model uncertainties for deep learning. *Advances in Neural Information Processing Systems*, 32, 2019.

Mingxing Tan and Quoc V. Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pp. 6105–6114. PMLR, 2019. URL `http://proceedings.mlr.press/v97/tan19a.html`.

Xinyu Tang, Saeed Mahloujifar, Liwei Song, Virat Shejwalkar, Milad Nasr, Amir Houmansadr, and Prateek Mittal. Mitigating membership inference attacks by {Self-Distillation} through a novel ensemble architecture. In *31st USENIX Security Symposium (USENIX Security 22)*, pp. 1433–1450, 2022.

Om Thakkar, Galen Andrew, and H Brendan McMahan. Differentially private learning with adaptive clipping. *arXiv preprint arXiv:1905.03871*, 2019.

Florian Tramer and Dan Boneh. Differentially private learning needs better features (or much more data). In *International Conference on Learning Representations*, 2020.

Tim van Erven and Peter Harremos. Rényi divergence and kullback-leibler divergence. *IEEE Transactions on Information Theory*, 60(7):3797–3820, 2014. doi: 10.1109/TIT.2014.2320500.

Santosh Vempala and Andre Wibisono. Rapid convergence of the unadjusted langevin algorithm: Isoperimetry suffices. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL `https://proceedings.neurips.cc/paper/2019/file/65a99bb7a3115fdede20da98b08a370f-Paper.pdf`.

Guotai Wang, Wenqi Li, Michael Aertsen, Jan Deprest, Sébastien Ourselin, and Tom Vercauteren. Aleatoric uncertainty estimation with test-time augmentation for medical image segmentation with convolutional neural networks. *Neurocomputing*, 338:34–45, 2019a.

Yu-Xiang Wang, Borja Balle, and Shiva Prasad Kasiviswanathan. Subsampled renyi differential privacy and analytical moments accountant. In *The 22nd International Conference on Artificial Intelligence and Statistics, AISTATS 2019, 16-18 April 2019, Naha, Okinawa, Japan*, pp. 1226–1235, 2019b.

Chen Zhu, Zheng Xu, Mingqing Chen, Jakub Konečnỳ, Andrew Hard, and Tom Goldstein. Diurnal or nocturnal? federated learning of multi-branch networks from periodically shifting distributions. In *International Conference on Learning Representations*, 2021.

Yuqing Zhu and Yu-Xiang Wang. Poission subsampled rényi differential privacy. In *International Conference on Machine Learning*, pp. 7634–7642. PMLR, 2019.

A Proof of Theorem A.1

For convenience/formality, we review the setup for the theorem we wish to prove. Recall that given a loss function $\mathcal{L}$, by taking the limit as η goes to 0 in DP-SGD, we recover the stochastic differential equation (1), restated here for convenience:

$$d\theta_t = -\nabla\mathcal{L}(\theta_t; D)dt + \sigma\sqrt{2}dW_t.$$

Note that the solutions θ_t to this equation are random variables. A key property of (1) is that the stationary distribution (equivalently, the limiting distribution as $t \to \infty$) has pdf proportional to $\exp(-\mathcal{L}(\theta; D)/\sigma)$ under mild assumptions on $\mathcal{L}$ (which are satisfied by strongly convex and smooth functions).

To simplify proofs and presentation in the section, we will assume that (a) θ_0 is a point distribution, (b) we are looking at unconstrained optimization over $\mathbb{R}^p$, i.e., there is no need for a projection operator in DP-SGD and (1), (c) the loss $\mathcal{L}$ is 1-strongly convex and M-smooth, and (d) $\sigma = 1$. We note that (a) can be replaced with θ_0 being sampled from a random initialization without too much work, and (c) can be enforced for Lipschitz, smooth functions by adding a quadratic regularizer. We let θ^* refer to the (unique) minimizer of $\mathcal{L}$ throughout the section.

Now, we consider the following setup: We obtain a single sample of the trajectory $\{\theta_t : t \in [0, T]\}$. We have some statistic $f : \Theta \to [-1, 1]$, and we wish to estimate the variance of the statistic for the final value θ_T, i.e. the variance $V := \mathbf{Var}\left(f(\theta_T)\right)$. To do so, we use the sample variance of the checkpoints at times $0 < t_1 < t_2 < t_3 < \ldots < t_k = T$. That is, our estimator is defined as $S = \frac{1}{k-1}\sum_{i=1}^{k}(f(\theta_{t_i}) - \widehat{\mu})^2$ where $\widehat{\mu} = \frac{1}{k}\sum_{i=1}^{k} f(\theta_{t_i})$.

Theorem A.1. *Under the preceding assumptions/setup, for some sufficiently large constant c, let $\gamma = \frac{1}{2M} + \ln(cM(p + \ln(1/\Delta) + \|\theta_0 - \theta^*\|_2^2)) + c\ln(1/\Delta)$ (recall that p is the dimensionality of the space). Then, if $t_1 > \gamma$ and $t_{i+1} > t_i + \gamma$ for all $i > 0$, for S, V as defined above:*

$$|\mathbb{E}[S] - V| = O(\Delta).$$

Before proving this theorem, we need a few helper lemmas about Rényi divergences:

Definition A.2. *The Rényi divergence of order $\alpha > 1$ between two distributions $\mathcal{P}$ and $\mathcal{Q}$ (with support $\mathbb{R}^d$), $D_\alpha(\mathcal{P}, \mathcal{Q})$, is defined as follows:*

$$D_\alpha(P, Q) := \int_{\theta\in\mathbb{R}^d} \frac{P(\theta)^\alpha}{Q(\theta)^{\alpha-1}} d\theta$$

We refer the reader to e.g. van Erven & Harremos (2014); Mironov (2017) for properties of the Rényi divergence. The following property shows that for any two random variables close in Rényi divergence, functions of them are close in expectation:

Lemma A.3. *[Adapted from Lemma C.2 of Bun & Steinke (2016)] Let $\mathcal{P}$ and $\mathcal{Q}$ be two distributions on Ω and $g : \Omega \to [-1, 1]$. Then,*

$$\left|\mathbb{E}_{x\sim\mathcal{P}}\left[g(x)\right] - \mathbb{E}_{x\sim\mathcal{Q}}\left[g(x)\right]\right| \leq \sqrt{e^{D_2(\mathcal{P}||\mathcal{Q})} - 1}.$$

Here, $D_2(\mathcal{P}||\mathcal{Q})$ corresponds to Rényi divergence of order two between the distributions $\mathcal{P}$ and $\mathcal{Q}$.

The next lemma shows that the solution to (1) approaches θ_∞ exponentially quickly in Rényi divergence.

Lemma A.4. *Fix some point θ_0. Assume $\mathcal{L}$ is 1-strongly convex, and M-smooth. Let $\mathcal{P}$ be the distribution of θ_t according to* (1) *for $\sigma = 1$ and:*

$$t := 1/2M + \ln(c(M\left\|\theta_0 - \theta^*\right\|_2^2 + p\ln(M))) + c\ln(1/\Delta).$$

Where c is a sufficiently large constant. Let Q be the stationary distribution of (1). *Then:*

$$D_2(\mathcal{P}, \mathcal{Q}) = O(\Delta^2).$$

The proof of this lemma builds upon techniques in Ganesh & Talwar (2020), and we defer it to the end of the section. Our final helper lemma shows that θ_∞ is close to θ^* with high probability:

Lemma A.5. *Let θ_∞ be the random variable given by the stationary distribution of* (1) *for* $\sigma = 1$. *If $\mathcal{L}$ is 1-strongly convex, then:*

$$\mathbf{Pr}[\|\theta_\infty - \theta^*\|_2 > \sqrt{p} + x] \leq \exp(-x^2/2).$$

Proof. We know the stationary distribution has pdf proportional to $\exp(-\mathcal{L}(\theta_t; D))$. In particular, since $\mathcal{L}$ is 1-strongly convex, this means θ_∞ is a sub-Gaussian random vector (i.e., its dot product with any unit vector is a sub-Gaussian random variable), and thus the above tail bound applies to it. □

We now will show that under the assumptions in Theorem A.1, every checkpoint is close to the stationary distribution, and that every pair of checkpoints is nearly pairwise independent.

Lemma A.6. *Under the assumptions/setup of Theorem A.1, we have:*

(E1) $\forall i : |\mathbb{E}[(f(\theta_{t_i}))] - \mathbb{E}[(f(\theta_{t_k}))]| = O(\Delta)$,

(E2) $\forall i : |\mathbb{E}[(f(\theta_{t_i})^2)] - \mathbb{E}[f(\theta_{t_k})^2]| = O(\Delta)$,

(E3) $\forall i < j : |\mathbf{Cov}\left(f(\theta_{t_i}), f(\theta_{t_j})\right)| = O(\Delta)$.

Proof. We assume without loss of generality Δ is at most a sufficiently small constant; otherwise, since f has range $[-1, 1]$, all of the above quantities can easily be bounded by 2, so a bound of $O(\Delta)$ holds for any distributions on $\{\theta_{t_i}\}$.

For (E1), by triangle inequality, it suffices to prove a bound of $O(\Delta)$ on $|\mathbb{E}[f(\theta_{t_i})] - \mathbb{E}[f(\theta_\infty)]|$. We abuse notation by letting θ_t denote both the random variable and its distribution. Then:

$$|\mathbb{E}[f(\theta_{t_i})] - \mathbb{E}[f(\theta_\infty)]| \overset{\text{Lemma A.3}}{\leq} \sqrt{e^{D_2(f(\theta_{t_i}), f(\theta_\infty))} - 1} \overset{(*_1)}{\leq} \sqrt{e^{D_2(\theta_{t_i}, \theta_\infty)} - 1}$$
$$\overset{\text{Lemma A.4}}{=} \sqrt{e^{O(\Delta^2)} - 1} \overset{(*_2)}{=} O(\Delta).$$

In $(*_1)$ we use the data-processing inequality (Theorem 9 of van Erven & Harremos (2014)), and in $(*_2)$ we use the fact $e^x - 1 \leq 2x, x \in [0, 1]$ and our assumption on Δ.

(E2) follows from (E1) by just using f^2 (which is still bounded in $[-1, 1]$) instead of f.

For (E3), note that since (1) is a (continuous) Markov chain, the distribution of θ_{t_j} conditioned on θ_{t_i} is the same as the distribution of $\theta_{t_j - t_i}$ according to (1) if we start from θ_{t_i} instead of θ_0. Let $\mathcal{P}$ be the joint distribution of $\theta_{t_i}, \theta_{t_j}$. Let $\mathcal{Q}$ be the joint distribution of $\theta_{t_i}, \theta_\infty$ (since (1) has the same stationary distribution regardless of its initialization, this is a pair of independent variables). Let $\mathcal{P}'$, $\mathcal{Q}'$ be defined identically to $\mathcal{P}$, $\mathcal{Q}$, except when sampling θ_{t_i}, if $\|\theta_{t_i} - \theta^*\|_2 > \sqrt{p} + \sqrt{2\ln(1/\Delta)}$ we instead set $\theta_{t_i} = \theta^*$ (and in the case of $\mathcal{P}'$, we instead sample θ_{t_j} from $\theta_{t_j} | \theta_{t_i} = \theta^*$ when this happens). Let $\mathcal{R}$ denote this distribution over θ_{t_i}. Then similarly to the proof of (E1) we have:

$$|\mathbb{E}_{\mathcal{P}'}[f(\theta_{t_i}) f(\theta_{t_j})] - \mathbb{E}_{\mathcal{Q}'}[f(\theta_{t_i})]\mathbb{E}[f(\theta_\infty)]| \overset{\text{Lemma A.3}}{\leq} \sqrt{e^{D_2(\mathcal{P}', \mathcal{Q}')} - 1}$$
$$\overset{(*_3)}{\leq} \sqrt{e^{\max_{\theta_{t_i} \in \text{supp}(\mathcal{R})} \{D_2(\theta_{t_j} | \theta_{t_i}, \theta_\infty)\}} - 1}.$$
$$\overset{\text{Lemma A.4}}{=} \sqrt{e^{O(\Delta^2)} - 1} = O(\Delta).$$

Here $(*_3)$ follows from the convexity of Rényi divergence, and in our application of A.4, we are using the fact that for all $\theta_{t_i} \in \text{supp}(\mathcal{R})$, $\|\theta_{t_i} - \theta^*\|_2 \leq \sqrt{p} + \sqrt{2\ln(1/\Delta)}$. Furthermore, by Lemma A.5, we know $\mathcal{P}$ and $\mathcal{P}'$ (resp. $\mathcal{Q}$ and $\mathcal{Q}'$) differ by at most Δ in total variation distance. So, since f is bounded in $[-1, 1]$, we have:

$$|\mathbb{E}_{\mathcal{P}}[f(\theta_{t_i}) f(\theta_{t_j})] - \mathbb{E}_{\mathcal{P}'}[f(\theta_{t_i}) f(\theta_{t_j})]| \leq \Delta,$$

$$|\mathbb{E}_{\mathcal{Q}}[f(\theta_{t_i})]\mathbb{E}[f(\theta_\infty)] - \mathbb{E}_{\mathcal{Q}'}[f(\theta_{t_i})]\mathbb{E}[f(\theta_\infty)]| \le \Delta.$$

Then by applying triangle inequality twice:

$$|\mathbb{E}_{\mathcal{P}}[f(\theta_{t_i})f(\theta_{t_j})] - \mathbb{E}_{\mathcal{Q}}[f(\theta_{t_i})]\mathbb{E}[f(\theta_\infty)]| = O(\Delta)$$

Now we can prove (E3) as follows:

$$\begin{aligned}
|\mathbf{Cov}\left(f(\theta_{t_i}), f(\theta_{t_j})\right)| &= |\mathbb{E}[(f(\theta_{t_i}) - \mathbb{E}[f(\theta_{t_i})])(f(\theta_{t_j}) - \mathbb{E}[f(\theta_{t_j})])]| \\
&= |\mathbb{E}[f(\theta_{t_i})f(\theta_{t_j})] - \mathbb{E}[f(\theta_{t_i})]\mathbb{E}[f(\theta_{t_j})]| \\
&\le |\mathbb{E}[f(\theta_{t_i})f(\theta_{t_j})] - \mathbb{E}[f(\theta_{t_i})]\mathbb{E}[f(\theta_\infty)]| + |\mathbb{E}[f(\theta_{t_i})]\mathbb{E}[f(\theta_\infty)] - \mathbb{E}[f(\theta_{t_i})]\mathbb{E}[f(\theta_{t_j})]| \\
&\le O(\Delta) + |\mathbb{E}[f(\theta_\infty)] - \mathbb{E}[f(\theta_{t_j})]| = O(\Delta).
\end{aligned}$$

□

Proof of Theorem A.1. We again assume without loss of generality Δ is at most a sufficiently small constant. The proof strategy will be to express $\mathbb{E}[S]$ in terms of individual variances $\mathbf{Var}\left(f(\theta_{t_i})\right)$, which can be bounded using Lemma A.6.

We have the following:

$$\mathbb{E}[S] = \frac{1}{k-1}\sum_{i=1}^{k}\mathbb{E}\left[(f(\theta_{t_i}) - \widehat{\mu})^2\right] = \frac{1}{k-1}\sum_{i=1}^{k}\mathbb{E}\left[\left(\frac{k-1}{k}\right)^2\left(\underbrace{f(\theta_{t_i})}_{x_i} - \underbrace{\frac{1}{k-1}\sum_{j\in[k],j\neq i} f(\theta_{t_j})}_{y_i}\right)^2\right]. \tag{2}$$

From (2), we have the following:

$$\begin{aligned}
\mathbb{E}\left[(x_i - y_i)^2\right] &= \mathbb{E}[x_i^2] - 2\mathbb{E}[x_i y_i] + \mathbb{E}[y_i^2] \\
&= \left(\mathbb{E}[x_i^2] - (\mathbb{E}[x_i])^2\right) + \left(\mathbb{E}[y_i^2] - (\mathbb{E}[y_i])^2\right) + \left((\mathbb{E}[x_i])^2 + (\mathbb{E}[y_i])^2 - 2\mathbb{E}\left[x_i y_i\right]\right) \\
&= \underbrace{\mathbf{Var}\left(x_i\right)}_{A} + \underbrace{\mathbf{Var}\left(y_i\right)}_{B} + \underbrace{\left((\mathbb{E}[x_i])^2 + (\mathbb{E}[y_i])^2 - 2\mathbb{E}\left[x_i y_i\right]\right)}_{C}.
\end{aligned} \tag{3}$$

In the following, we bound each of the terms A, B, and C individually. First, let us consider the term B. We have the following:

$$B = \mathbf{Var}\left(y_i\right) = \frac{1}{(k-1)^2}\left(\sum_{j\in[k],j\neq i}\mathbf{Var}\left(f(\theta_{t_j})\right) + 2\sum_{\substack{1\le j<\ell\le k\\ j\neq i,\ell\neq i}}\mathbf{Cov}\left(f(\theta_{t_j}), f(\theta_{t_\ell})\right)\right). \tag{4}$$

Plugging Lemma A.6, (E3) into (4) we bound the variance of y_i as follows:

$$B = \mathbf{Var}\left(y_i\right) = \frac{1}{(k-1)^2}\left(\sum_{j\in[k],j\neq i}\mathbf{Var}\left(f(\theta_{t_j})\right)\right) \pm O(\Delta). \tag{5}$$

We now focus on bounding the term C in (3). Lemma A.6, (E1) and (E3) implies the following:

$$(\mathbb{E}[x_i])^2 = (\mathbb{E}[f(\theta_{t_k})])^2 \pm O(\Delta), \tag{6}$$

$$(\mathbb{E}[y_i])^2 = (\mathbb{E}[f(\theta_{t_k})])^2 \pm O(\Delta), \tag{7}$$

$$\mathbb{E}[x_i y_i] = (\mathbb{E}[f(\theta_{t_k})])^2 + O(\Delta). \tag{8}$$

Plugging (6),(7), and (8) into (3), we have

$$\mathbb{E}\left[(x_i - y_i)^2\right] = \mathbf{Var}\left(f(\theta_{t_i})\right) + \frac{1}{(k-1)^2}\left(\sum_{j\in[k], j\neq i} \mathbf{Var}\left(f(\theta_{t_j})\right)\right) \pm O(\Delta). \tag{9}$$

Now, Lemma A.6, (E1) and (E2) implies $\forall i : |\mathbf{Var}\left(f(\theta_{t_i})\right) - \mathbf{Var}\left(f(\theta_{t_k})\right)| = O(\Delta)$. So from (9) we have the following:

$$\mathbb{E}\left[(x_i - y_i)^2\right] = \mathbf{Var}\left(f(\theta_{t_k})\right) \cdot \frac{k}{k-1} \pm O(\Delta). \tag{10}$$

Plugging this bound back in (2), we have the following:

$$\mathbb{E}[S] = \frac{1}{k-1} \cdot \left(\frac{k-1}{k}\right)^2 \cdot k \cdot \left(\mathbf{Var}\left(f(\theta_{t_k})\right) \cdot \frac{k}{k-1} \pm O(\Delta)\right) = \mathbf{Var}\left(f(\theta_{t_k})\right) \pm O(\Delta). \tag{11}$$

Which completes the proof. □

A.1 Optimizing the Number of Checkpoints

In Theorem A.1, we fixed the number of checkpoints and gave lower bounds on the burn-in time and separation between checkpoints needed for the sample variance bound to have bias at most Δ. We could instead consider the problem where T, the time of the final checkpoint, is fixed, and we want to choose k which minimizes the (upper bound on) mean squared error of the sample variance of $\{f(\theta_{iT/k})\}_{i\in[k]}$. Here, we sketch a solution to this problem using the bound from this section.

The mean squared error of the sample variance is the sum of the bias and variance of this estimator. We will use the following reparameterization of Theorem A.1:

Theorem A.7 (Reparameterized version of Theorem A.1)**.** *Let* $c_1 := \frac{1}{2M} + \ln(c_2 M(p + \|\theta_0 - \theta^*\|_2^2))$*, where* c_2 *is a sufficiently large constant. Then if* S *is the sample variance of* $\{f(X_{iT/k})\}_{i\in[k]}$*,* V *is the true variance of* X_T*, and* $T/k > c_1$*:*

$$|\mathbb{E}[S] - V|^2 \leq \exp\left(-\frac{T/k - c_1}{c_2}\right).$$

One can also show the variance of S is close to the true sample variance:

Lemma A.8. *If* $\bar{S}$ *is the sample variance of* $k > 1$ *i.i.d. samples of* θ_T*, then if* c_2 *is a sufficiently large constant, for* c_1 *as defined in theorem A.7:*

$$\mathbf{Var}\left(\bar{S}\right) \leq \frac{1}{k}, |\mathbf{Var}\left(S\right) - \mathbf{Var}\left(\bar{S}\right)| \leq 2\exp\left(-\frac{T/k - c_1}{c_2}\right).$$

Proof. Let $x_1, \ldots, x_k$ be k i.i.d. samples of $f(\theta_T)$, then since each x_i is in the interval $[-1, 1]$:

$$\mathbf{Var}\left(\bar{S}\right) = \frac{\mathbb{E}[x_1^4]}{k} - \frac{\mathbf{Var}\left(x_1\right)(k-3)}{k(k-1)} \leq \frac{1}{k}.$$

Giving the first part of the lemma. For the second part, let x_i be the sampled value of $f(\theta_{iT/k})$. Then:

$$\mathbb{E}[S^2] = \mathbb{E}\left[\left(\frac{1}{k-1}\sum_{i\in[k]}\left(x_i - \frac{1}{k}\sum_{j\in[k]} x_j\right)^2\right)^2\right].$$

For some coefficients $c_{i,j,\ell,m}$, this can be written as $\sum_{i\leq j\leq \ell\leq m} c_{i,j,\ell,m}\mathbb{E}[x_i x_j x_\ell x_m]$ where $\sum_{i\leq j\leq \ell\leq m} |c_{i,j,\ell,m}| \leq 2$. By a similar argument to Theorem A.1, the change in this expectation if we instead use x_i that are i.i.d. is then at most $\exp\left(-\frac{T/k - c_1}{c_2}\right)$ as long as c_2 is a sufficiently large constant. In other words, $|\mathbb{E}[S^2] - \mathbb{E}[\bar{S}^2]| \leq \exp\left(-\frac{T/k-c_1}{c_2}\right)$. A similar argument applies to $E[S]^2$, giving the second part of the lemma. □

Putting it all together, we have an upper bound on the mean squared error of the sample variance of:

$$\frac{1}{k} + 3\exp\left(-\frac{T/k - c_1}{c_2}\right),$$

Assuming $k > 1, T/k > c_1$. Minimizing this expression with respect to k gives

$$k = \frac{T}{c_1 + c_2 \ln(3T/c_2)},$$

which we can then round to the nearest integer larger than 1 to determine the number of checkpoints to use that minimizes our upper bound on the mean squared error. Of course, if $T < 2c_1$ then Theorem A.1 cannot be applied to give a meaningful bias bound for any number of checkpoints, so this choice of k is not meaningful in that case.

A.2 Proof of Lemma A.4

We will bound the divergences $D_\alpha(P_1, P_2), D_\alpha(P_2, P_3), D_\alpha(P_3, P_4)$ where P_1 is the distribution θ_η that is the solution to (1), P_2 is a Gaussian centered at the point $\theta_0 - \eta\nabla\mathcal{L}(\theta_0; D)$, P_3 is a Gaussian centered at θ^*, and P_4 is the stationary distribution of (1). Then, we can use the approximate triangle inequality for Rényi divergences to convert these pairwise bounds into the desired bound.

Lemma A.9. *Fix some θ_0. Let P_1 be the distribution of θ_η that is the solution to* (1)*, and let P_2 be the distribution $N(\theta_0 - \eta\nabla\mathcal{L}(\theta_0; D), 2\eta)$. Then:*

$$D_\alpha(P_1, P_2) = O\left(M^2 \ln(\alpha) \cdot \max\{p\eta^2, \|\theta_0 - \theta^*\|_2^2 \eta^3\}\right)$$

Proof. Let θ_t be the solution trajectory of (1) starting from θ_0, and let θ'_t be the solution trajectory if we replace $\nabla\mathcal{L}(\theta_t; D)$ with $\nabla\mathcal{L}(\theta_0; D)$. Then θ_η is distributed according to P_1 and θ'_η is distributed according to P_2.

By a tail bound on Brownian motion (see e.g. Fact 32 in Ganesh & Talwar (2020)), we have that $\max_{t\in[0,\eta]} \left\|\int_0^t dW_s ds\right\|_2 \leq \sqrt{\eta(p + 2\ln(2/\delta))}$ w.p. $1-\delta$. Then following the proof of Lemma 13 in Ganesh & Talwar (2020), w.p. $1-\delta$,

$$\max_{t\in[0,\eta]} \|\theta_t - \theta_0\|_2 \leq cM(\sqrt{p} + \sqrt{\ln(1/\delta)})\sqrt{\eta} + M\|\theta_0 - \theta^*\|_2 \eta,$$

for some sufficiently large constant c, and the same is true w.p. $1-\delta$ over θ'_t. Now, following the proof of Theorem 15 in Ganesh & Talwar (2020), for some constant c', we have the divergence bound $D_\alpha(P_1, P_2) \leq \varepsilon$ as long as:

$$\frac{M^4 \ln^2 \alpha}{\varepsilon^2}(p\eta^2 + \|\theta_0 - \theta^*\|_2^2 \eta^3) < c'.$$

In other words, for any fixed η, we get a divergence bound of:

$$D_\alpha(P_1, P_2) = O\left(M^2 \ln(\alpha) \cdot \max\{p\eta^2, \|\theta_0 - \theta^*\|_2^2 \eta^3\}\right),$$

as desired. □

Lemma A.10. *Let P_2 be the distribution $N(\theta_0 - \eta\nabla\mathcal{L}(\theta_0; D), 2\eta)$ and P_3 be the distribution $N(\theta^*, 2\eta)$. Then for $\eta \leq 2/M$:*

$$D_\alpha(P_2, P_3) \leq \frac{\alpha\|\theta_0 - \theta^*\|_2^2}{4\eta}.$$

Proof. By contractivity of gradient descent we have:

$$\left\|\theta_0 - \eta \nabla \mathcal{L}(\theta_0; D) - \theta^*\right\|_2 \leq \left\|\theta - \theta^*\right\|_2 .$$

Now the lemma follows from Rényi divergence bounds between Gaussians (see e.g., Example 3 of van Erven & Harremos (2014)). □

Lemma A.11. *Let P_3 be the distribution $N(\theta^*, 2\eta)$ and let P_4 be the stationary distribution of* (1). *Then for $\eta \leq 1/2M$ we have:*

$$D_\alpha(P_3, P_4) \leq \frac{\alpha}{\alpha - 1}\left(\frac{p}{2}\ln(1/\eta) - \ln(2\pi)\right) + \frac{p}{2}\ln(\alpha/4\pi\eta).$$

Proof. We have $P_3(\theta) = P_3(\theta^*)\exp(-\frac{1}{4\eta}\left\|\theta - \theta^*\right\|_2^2)$ where $P_3(\theta^*) = \left(\frac{1}{4\pi\eta}\right)^d$. By M-smoothness of the negative log density of P_4, we also have $P_4(\theta) \geq P_4(\theta^*)\exp(-\frac{M}{2}\left\|\theta - \theta^*\right\|_2^2)$. In addition, since P_4 is 1-strongly log concave, $P_4(\theta^*) \geq \left(\frac{1}{2\pi}\right)^{p/2}$ (as the 1-strongly log concave density with mode θ^* that minimizes $P_4(\theta^*)$ is the multivariate normal with mean θ^* and identity covariance). Finally, for $\alpha \geq 1$ and $\eta \leq 1/2M$, we have $\alpha/4\eta > (\alpha - 1)M/2$. Putting it all together:

$$\begin{aligned}
\exp((\alpha - 1)D_\alpha(P_3, P_4)) &= \int \frac{P_3(\theta)^\alpha}{P_4(\theta)^{\alpha-1}} d\theta \\
&= \frac{P_3(\theta^*)^\alpha}{P_4(\theta^*)^{\alpha-1}} \int \exp\left(-(\frac{\alpha}{4\eta} - (\alpha - 1)\frac{M}{2})\left\|\theta - \theta^*\right\|_2^2\right) d\theta \\
&\leq \left(\frac{1}{4\pi\eta}\right)^{\alpha p/2} (2\pi)^{\alpha(p-1)/2} \int \exp\left(-(\frac{\alpha}{4\eta} - (\alpha - 1)\frac{M}{2})\left\|\theta - \theta^*\right\|_2^2\right) d\theta \\
&= \left(\frac{1}{2\pi}\right)^{\alpha/2}\left(\frac{1}{2\eta}\right)^{\alpha p/2} \int \exp\left(-(\frac{\alpha}{4\eta} - (\alpha - 1)\frac{M}{2})\left\|\theta - \theta^*\right\|_2^2\right) d\theta \\
&\overset{(*)}{=} \left(\frac{1}{2\pi}\right)^{\alpha/2}\left(\frac{1}{2\eta}\right)^{\alpha p/2}\left(\frac{\frac{\alpha}{4\eta} - (\alpha - 1)\frac{M}{2}}{\pi}\right)^{p/2} \\
&\leq \left(\frac{1}{2\pi}\right)^{\alpha/2}\left(\frac{1}{2\eta}\right)^{\alpha p/2}\left(\frac{\alpha}{4\pi\eta}\right)^{p/2} \\
\implies D_\alpha(P_3, P_4) &\leq \frac{\alpha}{\alpha - 1}\left(\frac{p}{2}\ln(1/\eta) - \ln(2\pi)\right) + \frac{p}{2}\ln(\alpha/4\pi\eta).
\end{aligned}$$

In $(*)$, we use the fact that $\alpha/4\eta > (\alpha - 1)M/2$ to ensure the integral converges.

□

Lemma A.12. *Fix some point θ_0. Let P be the distribution θ_η that is the solution to* (1) *from θ_0 for time $\eta \leq 1/2M$. Let Q be the stationary distribution of* (1). *Then:*

$$D_\alpha(P, Q) = O\left(M^2\ln(\alpha) \cdot \max\{p\eta^2, \left\|\theta_0 - \theta^*\right\|_2^2\eta^3\} + \frac{\alpha\left\|\theta_0 - \theta^*\right\|_2^2}{\eta} + p\ln(\alpha/\eta).\right)$$

Proof. By monotonicity of Rényi divergences (see e.g., Proposition 9 of Mironov (2017)), we can assume $\alpha \geq 2$. Then by applying twice the approximate triangle inequality for Rényi divergences (see e.g. Proposition 11 of Mironov (2017)), we get:

$$D_\alpha(P_1, P_4) \leq \frac{5}{3}D_{3\alpha}(P_1, P_2) + \frac{4}{3}D_{3\alpha-1}(P_2, P_3) + D_{3\alpha-2}(P_3, P_4).$$

The lemma now follows by Lemmas A.9, A.10, A.11. □

Lemma A.4 now follows by plugging $\alpha = 2, \eta = 1/2M$ into Lemma A.12 and then using Theorem 2 of Vempala & Wibisono (2019).

Table 4: StackOverflow LSTM architecture details.

Layer	Output shape	Parameters
Input	20	0
Embedding	(20, 96)	960384
LSTM	(20, 670)	2055560
Dense	(20, 96)	64416
Dense	(20, 10004)	970388
Softmax	-	-

B MISSING DETAILS FROM SECTION 2.2

Below we provide the experimental setup and results that are missing from Section 2.2.

B.1 EXPERIMENTAL SETUP

Training hyperparameters:

CIFAR10 training: We use Jaxline (Bradbury et al., 2018) to train on CIFAR10 using DP-SGD (Berrada et al., 2022). For CIFAR10, we use a WideResNet with depth 16 and width 4. We fix clip norm to 1, batch size to 4096 and augmentation multiplicity to 16 as in (De et al., 2022). Then, we set learning rate and noise multiplier, respectively, to 2 and 10 for $\varepsilon = 1$ and to 4 and 3 for $\varepsilon = 8$. For periodic distribution shifting (PDS) CIFAR10, we set learning rate and noise multiplier, respectively, to 2 and 12 for $\varepsilon = 1$ and to 4 and 4 for $\varepsilon = 8$. We stop the training when the intended privacy budget exhausts.

CIFAR100 training: For CIFAR100 also, we use Jaxline (Bradbury et al., 2018) and use DP-SGD to *fine-tune the last, classifier layer of* a WideResNet with depth 28 and width 10 that is pre-trained on entire ImageNet data. We fix clip norm to 1, batch size to 16,384 and augmentation multiplicity to 16 as in (De et al., 2022). Then, we set learning rate and noise multiplier, respectively, to 3.5 and 21.1 for $\varepsilon = 1$ and to 4 and 9.4 for $\varepsilon = 8$. For periodic distribution shifting (PDS) CIFAR100, we set learning rate and noise multiplier, respectively, to 4 and 21.1 for $\varepsilon = 1$ and to 5 and 9.4 for $\varepsilon = 8$. We stop the training when privacy budget exhausts.

We would like to highlight that we obtain a significant improvement over the SOTA baseline of De et al. (2022): In particular, *unlike in (De et al., 2022), we fine-tune the final EMA checkpoint*, i.e., the one computed using EMA during pre-training over ImageNet. *This modification (without any additional checkpoints aggregations) gives a major accuracy boost of 5% (70.3% → 75.51%) for $\varepsilon = 1$ and of 3.2% (77.6% → 80.81%) for $\varepsilon = 1$ for the normal CIFAR100 baseline*. We obtain similarly high improvements by fine-tuning the EMA of pre-trained checkpoints (instead of just the final checkpoint) for the PDS-CIFAR100 case. We leave the further investigation of this phenomena to the future work.

StackOverflow training: We follow the SOTA in (Kairouz et al., 2021; Denisov et al., 2022) and use TFF to train a one-layer LSTM (detailed architecture in Table 4 (Reddi et al., 2020)) on StackOverflow using DP-FTRLM full version from (Denisov et al., 2022). StackOverflow is naturally user-partitioned data, and we process 100 clients in each FL round. We train for 2048 FL rounds and set clip norm, noise multiplier, server learning rate, client learning rate, and server momentum, respectively, to 1, 0.341, 0.5, 1.0, 0.95 for $\varepsilon = 18.9$ and to 1, 0.682, 0.25, 1.0, 0.95 for $\varepsilon = 8.2$. For PDS-StackOverflow, the same set of hyperparameter performs the best based on our tuning of the aforementioned hyperparameters.

Hyperparameters tuning of checkpoints aggregations: Here we provide the methodology we follow to obtain the best hyperparameters for our checkpoints aggregation methods (Section 2.1). For EMA, De et al. (2022) simply use the EMA coefficient that works the best in non-private baseline. However, for each of the settings we consider, we tune EMA coefficient in {0.85, 0.9, 0.95, 0.99, 0.999, 0.9999} and observe that the best EMA coefficients for private and non-private settings need not be the same (Table 9). For instance, for CIFAR10, for ε of 1 and 8, EMA coefficient of 0.95 and 0.99 perform the best and outperform 0.9999 by 0.6% and 0.3%, respectively. Hence, we advise future works to perform tuning of EMA coefficient. Full results are given in Table 9.

Table 5: Accuracy gains on test data due to *data-dependent* checkpoints aggregation algorithms for StackOverflow trained using DPFTRLM to achieve user-level DP.

Privacy level	Aggregation algorithms				
	None	Parameters aggregation		Outputs aggregation	
	(Baseline)	EMA	UPA	OPA	OMV
$\varepsilon = \infty$	25.24 ± 0.16	25.97 ± 0.01	**25.97 ± 0.01**	25.94 ± 0.03	25.97 ± 0.02
$\varepsilon = 18.9$	23.41 ± 0.08	23.88 ± 0.04	**23.92 ± 0.03**	23.81 ± 0.06	23.76 ± 0.07
$\varepsilon = 8.2$	22.28 ± 0.08	22.66 ± 0.05	**22.80 ± 0.05**	22.7 ± 0.01	22.69 ± 0.01

Table 6: Accuracy gains on test data due to *data-dependent* checkpoints aggregation algorithms for *periodic distribution shifting* StackOverflow trained using DPFTRLM to achieve user-level DP.

Privacy level	Aggregation algorithms				
	None	Parameters aggregation		Outputs aggregation	
	(Baseline)	EMA	UPA	OPA	OMV
$\varepsilon = \infty$	23.89 ± 0.04	**24.18 ± 0.02**	24.15 ± 0.02	24.15 ± 0.01	24.19 ± 0.01
$\varepsilon = 18.9$	21.6 ± 0.13	**22.18 ± 0.07**	22.17 ± 0.06	22.21 ± 0.08	22.19 ± 0.09
$\varepsilon = 8.2$	20.24 ± 0.29	20.84 ± 0.13	20.85 ± 0.12	**20.90 ± 0.06**	20.82 ± 0.08

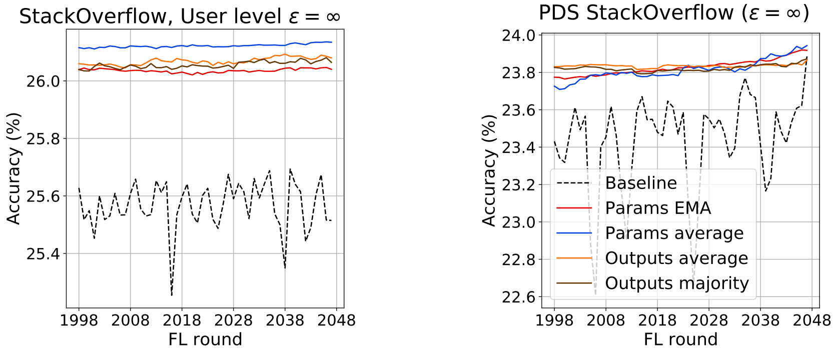

Figure 5: Accuracy gains due to checkpoints aggregations for StackOverflow (left) and periodic distribution shifting StackOverflow (right) trained using DP-FTRLM *without* any DP, i.e., $\varepsilon = \infty$.

For the past k checkpoints aggregation based methods, our tuning methodology is general and as follows: we use some validation data (which is disjoint from training data) and evaluate the efficacy of aggregating k checkpoints where we vary k over a large enough range, and select the k that performs the best on average over 5 runs. Finally, we present results on test data using the best k value. Our hyperparameters tuning method is easy to replicate hence for conciseness, here we only provide the ranges of k that we use for tuning. For CIFAR10 and CIFAR100, we tune $k \in \{3, 5, 10, 20, ..., 200\}$ for both parameters and outputs aggregation. For StackOverflow, we tune $k \in \{3, 5, 10, 20, ..., 200\}$ for parameters aggregations (i.e., UPA) and for outputs aggregation (i.e., OPA and OMV). However, as for outputs aggregations, one should store k checkpoints on device in the FL setting of StackOverflow, we reduce tuning range and tune $k \in \{3, 5, 10, 20, ..., 100\}$.

Details of data dependent checkpoints selection for StackOverflow: Here we provide precise method for selecting the best k of all checkpoints and aggregate them. We perform these experiments for StackOverflow due to availability of additional data, and omit CIFAR datasets. Specifically, we use 10,000 samples from the validation partition of the original StackOverflow data and use it as *held-out* data. Note that held-out data is disjoint from both training and validation data we use for other parts of experiments. We compute performance of all checkpoints on held-out data and order them from highest to lowest performing checkpoints. Then we tune k as detailed in the previous section. The aggregation function remains the same for all, but EMA, our aggregation methods from Section 2.1.

Recall that traditional EMA is computed over all the checkpoints computed during training. We compute EMA over best k checkpoints as $\theta^i_{ema} = (1 - \beta) \cdot \theta^{i-1}_{ema} + \beta \cdot \theta^i$, where $i \in \{1, 2, ..., k\}$. We keep EMA coefficient, β, constant through out. For EMA, we further tune β using validation data.

Table 7: Accuracy gains on test data for *CIFAR100 fine-tuned* using DP-SGD and *sample-level* DP.

Privacy level	Aggregation algorithms				
	None	Parameters aggregation		Outputs aggregation	
	(Baseline)	EMA	UPA	OPA	OMV
$\varepsilon = 8$	80.81 ± 0.11	80.88 ± 0.10	80.83 ± 0.09	**80.92** ± **0.10**	80.82 ± 0.10
$\varepsilon = 1$	75.51 ± 0.15	75.42 ± 0.13	**75.62** ± **0.12**	75.51 ± 0.16	75.57 ± 0.18

Table 8: Accuracy gains on test data for *periodic distribution shifting CIFAR100 fine-tuned* using DP-SGD and *sample-level* DP.

Privacy level	Aggregation algorithms				
	None	Parameters aggregation		Outputs aggregation	
	(Baseline)	EMA	UPA	OPA	OMV
$\varepsilon = 8$	77.16 ± 0.11	**80.53** ± **0.07**	80.53 ± 0.08	80.49 ± 0.06	80.41 ± 0.09
$\varepsilon = 1$	70.84 ± 0.16	74.83 ± 0.15	**75.81** ± **0.16**	75.02 ± 0.17	74.97 ± 0.18

Table 9: We observe that tuning the EMA coefficient can provide significant gains in accuracy over the default value of 0.9999 that De et al. (2022) use; with warm-up schedule and number of training steps that De et al. (2022) use, 0.9999 and 0.999 provide the same results. This implies that tuning EMA coefficients for each different privacy budget is required for the best performances.

Privacy level	EMA coefficient			
	0.9	0.95	0.99	0.999 (De et al. (2022))
$\varepsilon = 8$	79.41	79.35	**79.41**	79.16
$\varepsilon = 1$	56.59	**56.61**	56.06	56.05