An Empirical Study on the Intrinsic Privacy of Stochastic Gradient Descent

Stephanie L. Hyland and Shruti Tople
Microsoft Research
{stephanie.hyland, shruti.tople}@microsoft.com

ABSTRACT

In this work, we take the first step towards understanding whether the intrinsic randomness of stochastic gradient descent (SGD) can be leveraged for privacy, for any given dataset and model. In doing so, we hope to mitigate the trade-off between privacy and utility for models trained with differential-privacy (DP) guarantees. Our primary contribution is a large-scale empirical analysis of SGD on convex and non-convex objectives. We evaluate the inherent variability in SGD on 4 datasets and calculate the intrinsic data-dependent $\epsilon_i(\mathcal{D})$ values due to the inherent noise. For logistic regression, we observe that SGD provides intrinsic $\epsilon_i(\mathcal{D})$ values between 3.95 and 23.10 across four datasets, dropping to between 1.25 and 4.22 using the tight empirical sensitivity bound. For neural networks considered, we observe high $\epsilon_i(\mathcal{D})$ values (>40) owing to their larger parameter count. We propose a method to augment the intrinsic noise of SGD to achieve the desired target ϵ, which produces statistically significant improvements in private model performance (subject to assumptions). Our experiments provide strong evidence that the intrinsic randomness in SGD should be considered when designing private learning algorithms.

1 INTRODUCTION

Respecting the privacy of people contributing their data to train machine learning models is important for the safe use of this technique [11, 27, 30]. Private variants of learning algorithms have been proposed to address this need [5, 12, 24, 29, 31]. Unfortunately the utility of private models typically degrades, limiting their applicability. This performance loss often results from the need to add noise during or after model training, to provide the strong protections of ϵ-differential-privacy [8]. However, results to date neglect the fact that learning algorithms are often *stochastic*. Framing them as 'fixed' queries on a dataset neglects an important source of *intrinsic* noise. Meanwhile, the randomness in learning algorithms such as stochastic gradient descent (SGD) is well-known among machine learning practitioners [10, 14], and has been lauded for affording superior generalisation to its non-stochastic counterpart [16]. Moreover, the 'insensitive' nature of SGD relative to variations in its input data has been established in terms of uniform stability [13]. The data-dependent nature of this stability has also been characterised [18]. Combining these observations, we speculate that the *variability* in the model parameters produced by the stochasticity of SGD may exceed its sensitivity to perturbations in the specific input data, affording 'data-dependent intrinsic' privacy. In essence, we ask: *"Can the intrinsic, data-dependent stochasticity of SGD help with the privacy-utility trade-off?"*

Our Approach. We consider a scenario where a model is trained securely, but the final model parameters are released to the public - for example, a hospital which trains a prediction model on its own patient data and then shares it with other hospitals or a cloud provider. The adversary then has access to the fully trained model, including its architecture, and we assume details of the training procedure are public (e.g. batch size, number of training iterations, learning rate), but *not* the random seed used to initialise the model parameters and sample inputs from the dataset. We therefore focus on how SGD introduces randomness in the *final* weights of a model.

This randomness is introduced from two main sources — (1) random initialization of the model parameters and (2) random sampling of the input dataset during training. We argue that rather than viewing this variability as a pitfall of stochastic optimisation, it can instead be seen as a source of noise that can mask information about participants in the training data. This prompts us to investigate whether SGD itself can be viewed as a differentially-private mechanism, with some *intrinsic* data-dependent ϵ-value, which we refer to as $\epsilon_i(\mathcal{D})$. To calculate $\epsilon_i(\mathcal{D})$, we propose a novel method that characterises SGD as a Gaussian mechanism and estimates the intrinsic randomness for a given dataset, using a large-scale *empirical* approach. To the best of our knowledge, ours is the first work to report the empirical calculation of $\epsilon_i(\mathcal{D})$ values based on the observed distribution. Finally, we propose an augmented differentially-private SGD algorithm that takes into account the intrinsic $\epsilon_i(\mathcal{D})$ to provide better utility. We empirically compute the $\epsilon_i(\mathcal{D})$ for SGD and the utility improvement for models trained with both convex and non-convex objectives on 4 different datasets: MNIST, CIFAR10, Forest Covertype and Adult.

2 PROBLEM & BACKGROUND

We study the variability due to random sampling, and sensitivity to dataset perturbations of stochastic gradient descent (SGD).

Differential Privacy (DP).. Differential privacy hides the participation of an individual sample in the dataset [8]. (ϵ, δ)-DP ensures that for all adjacent datasets S and S', the privacy loss of any individual datapoint is bounded by ϵ with probability at least $1 - \delta$ [9]:

Definition 2.1 ((ϵ, δ)-Differential Privacy). A mechanism $\mathcal{M}$ with domain $\mathcal{I}$ and range $\mathcal{O}$ satisfies (ϵ, δ)-differential privacy if for any two neighbouring datasets $S, S' \in \mathcal{I}$ that differ only in one input and for a set $E \subseteq \mathcal{O}$, we have: $\Pr(\mathcal{M}(S) \in E) \leq e^{\epsilon} \Pr(\mathcal{M}(S') \in E) + \delta$

We consider an algorithm whose output is the weights of a trained model, thus focus on the ℓ_2-sensitivity:

Definition 2.2 (ℓ_2-Sensitivity (From Def 3.8 in [9]). Let f be a function mapping from a dataset to a vector in $\mathbb{R}^d$. Let S, S' be two datasets differing in one data point. Then the ℓ_2-sensitivity of f is defined as: $\Delta_2(f) = \max_{S,S'} \|f(S) - f(S')\|_2$

One method for making a deterministic query f differentially private is the Gaussian mechanism. This gives a way to compute the ϵ of a Gaussian-distributed query given δ, its sensitivity $\Delta_2(f)$, and its variance σ^2.

THEOREM 2.3. *Gaussian mechanism (From [9]). Let f be a function that maps a dataset to a vector in $\mathbb{R}^d$. Let $\epsilon \in (0,1)$ be arbitrary. For $c^2 > 2\ln(1.25/\delta)$, adding Gaussian noise sampled using the parameters $\sigma \geq c\Delta_2(f)/\epsilon$ guarantees (ϵ, δ)-differential privacy.*

Stochastic Gradient Descent (SGD). SGD and its derivatives are the most common optimisation methods for training machine learning models [3]. Given a loss function $\mathcal{L}(\mathbf{w}, (x, y))$ averaged over a dataset, SGD provides a stochastic approximation of the traditional gradient descent method by *estimating* the gradient of $\mathcal{L}$ at random inputs. At step t, on selecting a random sample (x_t, y_t) the gradient update function G performs $\mathbf{w}_{t+1} = G(\mathbf{w}_t) = \mathbf{w}_t - \eta\nabla_{\mathbf{w}}\mathcal{L}(\mathbf{w}_t, (x, y))$, where η is the (constant) step-size or learning rate, and w_t are the weights of the model at t. In practice, the stochastic gradient is estimated using a mini-batch of B samples.

Recently, Wu et al. [31] showed results for the sensitivity of SGD to the change in a single training example for a convex, L-Lipschitz and β-smooth loss $\mathcal{L}$. We use this sensitivity bound in our analyses to compute intrinsic $\epsilon_i(\mathcal{D})$ for convex functions. Let A denote the SGD algorithm using r as the random seed. The upper bound for sensitivity for k-passes of SGD with learning rate η is given by

$$\hat{\Delta}_S = \max_r \|A(r; S) - A(r; S')\| \leq 2kL\eta \tag{1}$$

$\hat{\Delta}_S$ gives the maximum difference in the model parameters due to the presence or absence of a single input sample. When using a batch size of B as we do, the sensitivity bound can be reduced by a factor of B i.e., $\hat{\Delta}_S \leq 2kL\eta/B$. We provide detailed explanation for this sensitivity and variability of SGD in Appendix A and B. We use this theoretical sensitivity ($\hat{\Delta}_S$) in addition to empirically-computed sensitivity ($\hat{\Delta}_S^*$) estimates in our experiments to compute the $\epsilon_i(\mathcal{D})$ for models trained using convex loss functions. For the non-convex models, no known theoretical sensitivity is established and hence we use only empirical sensitivity values. The details of computing different sensitivity values and $\epsilon_i(\mathcal{D})$ is mentioned in Appendix C.

Research Questions. We ask the following questions:

1. Does the variability in SGD exceed the sensitivity due to changes in an individual input sample?
To answer this, we present a large-scale empirical study in Section 4 across several diverse datasets. We build on results from Hardt et al. [13] and Wu et al. [31] that allow us to bound the (expected) difference in the model parameters when trained with SGD using a convex objective function.

2. Can we quantify the intrinsic privacy of SGD, if any?
To quantify the 'data-dependent intrinsic privacy' of SGD, we aim to calculate the intrinsic $\epsilon_i(\mathcal{D})$ values for any given dataset. To do this, we interpret the posterior distribution returned by SGD computed with many random seeds as a Gaussian distribution and estimate its parameters using both theoretical (for convex loss) and empirical sensitivity bounds.

3. Can the intrinsic privacy of SGD improve utility?
We propose an augmented DP-SGD algorithm based on output perturbation [2, 5, 29, 31].

3 AUGMENTED DP-SGD

In this section, we show how to account for the data-dependent intrinsic noise of SGD while adding noise using the output perturbation method that ensures differential privacy guarantees [31]. The premise of output perturbation is to train a model in secret, and then release a noisy version of the final weights. For a desired ϵ, δ, and a known sensitivity value ($\hat{\Delta}_S$, $\hat{\Delta}_S^*$) the Gaussian mechanism (Theorem 2.3) gives us the required level of noise in the output, which we call σ_{target}.

In Wu et al. [31], this σ_{target} defines the variance of the noise vector sampled and added to the model weights, to produce a (ϵ, δ)-DP model. In our case, we reduce σ_{target} to account for the noise already present in the output of SGD. Since the sum of two independent Gaussians with variance σ_a^2 and σ_b^2 is a Gaussian with variance $\sigma_a^2 + \sigma_b^2$, if the intrinsic noise of SGD is $\sigma_i(\mathcal{D})$, to achieve the desired ϵ we need to augment $\sigma_i(\mathcal{D})$ to reach σ_{target}:

$$\sigma_{\text{augment}} = \sqrt{\sigma_{\text{target}}^2 - \sigma_i(\mathcal{D})^2} \tag{2}$$

Given the likely degradation of model performance with output noise, accounting for $\sigma_i(\mathcal{D})$ is expected to help utility without compromising privacy. The resulting algorithm for augmented differentially-private SGD is shown in Algorithm 1.

Algorithm 1 Augmented differentially private SGD

Given $\sigma_i(\mathcal{D})$, ϵ_{target}, δ, $\Delta_2(f)$, model weights $\mathbf{w}_{\text{private}}$.
$c \leftarrow \sqrt{2\log(1.25)/\delta} + 1 \times 10^{-5}$
$\sigma_{\text{target}} \leftarrow c\Delta_2(f)/\epsilon_{\text{target}}$
if $\sigma_i(\mathcal{D}) < \sigma_{\text{target}}$ **then**
    $\sigma_{\text{augment}} \leftarrow \sqrt{\sigma_{\text{target}}^2 - \sigma_i(\mathcal{D})^2}$
else
    $\sigma_{\text{augment}} \leftarrow 0$
$\rho \sim \mathcal{N}(0, \sigma_{\text{augment}})$
$\mathbf{w}_{\text{public}} \leftarrow \mathbf{w}_{\text{private}} + \rho$
return $\mathbf{w}_{\text{public}}$

THEOREM 3.1. *Assuming SGD is a Gaussian mechanism with intrinsic noise $\sigma_i(\mathcal{D})$, Algorithm 1 is (ϵ, δ)-differentially private.*

The proof is a straight-forward application of the fact that the sum of Gaussians is a Gaussian, and so the construction in Algorithm 1 produces the desired value of σ to achieve a (ϵ, δ)-differentially private mechanism as per Theorem 2.3.

4 EVALUATION

To better understand both sensitivity and variability of SGD, we conduct an extensive empirical study.

Experimental Setup. We run a grid of experiments where we vary data or (non-exclusively) the random seed. We use variations of the random seed to explore the intrinsic randomness in SGD, and variations of the data to explore sensitivity to dataset perturbations.

Variation in Data. To vary the data, we consider 'neighbouring' datasets derived from a given data source $\mathcal{D}$ (e.g. two variants of `MNIST`). A pair of datasets is neighbouring if they differ in exactly one example. We construct a set of neighbouring datasets using a similar approach to Hardt et al. [13], by replacing the ith training example with the first example to create datasets indexed by i.

Dataset	Training size	Validation size	Test size	d
CIFAR2	9,000	1,000	2,000	50*
MNIST-binary	10,397	1,155	1,902	50*
Adult	29,305	3,256	16,281	100
Forest	378,783	42,086	74,272	49

Table 1: Statistics for datasets. The dimension of feature vectors is d. *Images are projected to $d = 50$ using PCA.

Model	Dataset	η	T	E	P	Hidden size
LogReg	CIFAR2	0.5	2000	20000	51	-
	MNIST-binary	0.5	1850	19600	51	-
	Adult	0.5	3400	25700	101	-
	Forest	1.0	8400	20693	50	-
NN	CIFAR2	0.5	2500	20000	521	10
	MNIST-binary	0.5	4750	9800	521	10
	Adult	0.5	1850	11250	817	8
	Forest	0.5	3500	11247	511	10

Table 2: Training and model hyperparameters. η is the learning rate. T is the number of training steps (the convergence point). E is the number of experiments performed, and P is the number of parameters in the model.

Variation in Random Seed. To vary the random seed, we simply run each experiment (on given dataset S_i) multiple times with different seeds provided to the random number generator. The random seed impacts the training procedure by impacting the *initialisation* of the weights, as well as the order of traversal of the dataset. We also consider a variant of this setting where the initialisation of the model is fixed. We follow the traditional setting of SGD where a random permutation (determined by the random seed) is applied to the training data at the start of each epoch, and batches of examples are then drawn without replacement.

Benchmark datasets. Focusing on binary classification tasks, we perform our experiments using four data sources. The sizes and dimensionality of these datasets are given in Table 1. Each dataset is normalised such that $||\mathbf{x}|| \leq 1$.

- CIFAR2[17]: We convert the (32, 32, 3)-dimensional images in CIFAR10 to $d = 50$ using principal component analysis (PCA) [23], and restrict to classes 0 and 2 (planes and birds) to form a binary classification task (hence CIFAR2).
- MNIST-binary[19]: As with CIFAR2 we use 2 classes (3 and 5) and project to $d = 50$ with PCA.
- Adult[7]: The task is to predict whether an individual's income exceeds \$50k/year based on census data from 1994. We one-hot encode categorical-valued features, dropping the first level.
- Forest[6, 7]: Forest cover type prediction from cartographic information. We convert this to a binary task by restricting to classes 1 and 2, which are the most numerous.

Model types. We consider two model classes:

(1) Logistic regression. The objective function for logistic regression is convex and Lipschitz with constant $L = \sup_{\mathbf{x}} ||\mathbf{x}||$ and smooth with $\beta = \sup_{\mathbf{x}} \|\mathbf{x}\|^2$, giving us $L = \sqrt{2}$.
(2) Neural networks. We consider fully-connected neural networks with one hidden layer, using a relu nonlinearity and a sigmoid activation on the output.

We train with a fixed learning rate and select the convergence point based on the validation performance failing to improve three times in a row, or by visual assessment of loss curves (see e.g. Figure 4). We evaluate at the convergence point to avoid studying models which overfit. We performed mild, but not extensive hyperparameter optimisation as our focus is not on finding the best-performing model; best hyperparameters are shown in Table 2. We expect these hyperparameters to also influence the variability of the weight distribution, but consider them fixed for the purpose of this investigation.

Data-dependent variability in SGD. We consider two sources of variability in the learned model: dataset perturbations, and choice of random seed. We aim to estimate:

(1) Variation due to dataset; $\Delta_S := \|A(r; S) - A(r; S')\|$
(2) Variation due to seed; $\Delta_V := \|A(r; S) - A(r'; S)\|$.
    (a) allowing for fixed initialisation (Δ_V^{fix})
    (b) and seed-dependent initialisation (Δ_V^{vary})
(3) Variation in both; $\Delta_{S+V} := \|A(r; S) - A(r'; S')\|$

where r and r' are two random seeds, S and S' are two neighbouring datasets, and A is the SGD algorithm which outputs a vector of model weights. We study how the variability *due to seed* ($\Delta_V^{\text{fix}}, \Delta_V^{\text{vary}}$) compares to the data sensitivity Δ_S. We also test the tightness of the theoretical bound proposed in [13, 31] ($\hat{\Delta}_S$) for convex objectives.

Figure 1 shows the distribution of these quantities across experiments for the four datasets and two model classes considered. We see that the seed typically has a much larger impact than the data ($\Delta_V > \Delta_S$), an effect magnified when model initialisation is further allowed to vary. We also see that the theoretical bound ($\hat{\Delta}_S$) is loose, exceeding the largest observed value of Δ_S by a factor between 3.15 and 6.46.

What is the 'intrinsic' $\sigma_i(\mathcal{D})$ and $\epsilon_i(\mathcal{D})$? We estimate intrinsic $\sigma_i(\mathcal{D})$ and $\epsilon_i(\mathcal{D})$ of SGD for each dataset as follows: We first obtain an estimate of the sensitivity of SGD for that dataset, using the theoretical bound $\hat{\Delta}_S$ where available, or the empirical sensitivity estimate $\hat{\Delta}_S^*$ which is simply taken as the max of Δ_S . We estimate the variability due to seed by treating weights as independently and identically distributed as Gaussians. Similar assumptions are commonplace on the noise distribution of SGD [20, 28], but we acknowledge it as requiring further exploration. This assumption however allows us to apply the logic of the Gaussian mechanism to estimate $\epsilon_i(\mathcal{D})$ and $\epsilon_i(\mathcal{D})^*$, combining $\delta = 1/N^2$ with the theoretical and empirical sensitivity estimates respectively.

The resulting values for each dataset and the two model classes are shown in Table 3. It is clear that $\sigma_i(\mathcal{D})$ and $\hat{\Delta}_S^*$ differ across datasets and models. Although neural networks have larger $\sigma_i(\mathcal{D})$, the sensitivity $\hat{\Delta}_S^*$ is much larger, resulting in very high $\epsilon_i(\mathcal{D})^*$. This is likely driven by the number of parameters in the model as $\hat{\Delta}_S^*$ is the ℓ_2-norm of a vector (and $\sigma_i(\mathcal{D})$ is not). Table 3 highlights that our analysis of SGD necessarily depends on the underlying dataset, motivating further study into data-dependent differentially-private mechanisms [18, 22].

How does accounting for intrinsic variability improve utility? We have seen that the intrinsic $\epsilon_i(\mathcal{D})$ of SGD can be quantified, but in many cases is insufficient alone to provide a desirable level of privacy. In this section, we demonstrate that by accounting for $\epsilon_i(\mathcal{D})$ (via $\sigma_i(\mathcal{D})$), model performance can be improved over an existing approach based solely on output perturbation. We focus here only on logistic regression. As evidenced by Table 3, the neural

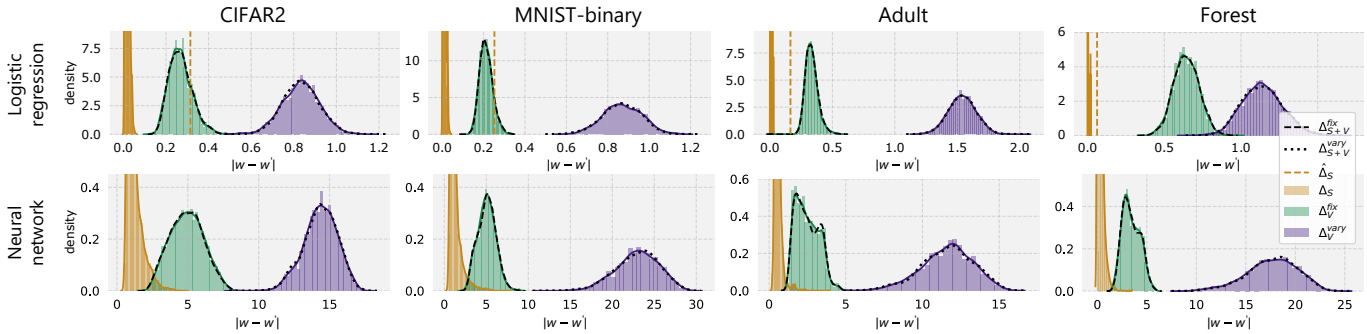

Figure 1: Distribution of $\|w - w'\|$ across pairs of experiments differing in data (Δ_S), random seed (Δ_V^{fix}, Δ_V^{vary}), or both (Δ_{S+V}). The change in w is dominated by the random seed, as evidenced by Δ_V tending to be much larger than Δ_S. Δ_V^{fix} refers to the setting where the random seed is variable, but the initialisation of the model is fixed. The vertical dashed line is the theoretical upper bound of Δ_S proposed by [31] (only available for convex objectives). The first row shows results for logistic regression, and the second is for a fully-connected neural network with one hidden layer.

δ	CIFAR2 $1.23\text{x}10^{-8}$	MNIST-binary $9.25\text{x}10^{-9}$	Adult $1.16\text{x}10^{-9}$	Forest $6.97\text{x}10^{-12}$
Logistic regression				
$\hat{\Delta}_S$	0.314	0.252	0.164	0.063
$\hat{\Delta}_S^*$	0.057	0.059	0.036	0.020
$\sigma_i(\mathcal{D})$	0.083	0.085	0.108	0.114
$\epsilon_i(\mathcal{D})$	23.10	18.17	9.77	3.95
$\epsilon_i(\mathcal{D})^*$	4.19	4.22	2.13	1.25
Neural networks				
$\hat{\Delta}_S^*$	4.813	7.688	3.891	3.307
$\sigma_i(\mathcal{D})$	0.4433	0.713	0.288	0.554
$\epsilon_i(\mathcal{D})^*$	65.922	66.009	87.136	42.939

Table 3: Theoretical sensitivity ($\hat{\Delta}_S$), empirical sensitivity ($\hat{\Delta}_S^*$), δ ($1/N^2$), intrinsic variability $\sigma_i(\mathcal{D})$ accounting for variable initialisation, intrinsic $\epsilon_i(\mathcal{D})$, and intrinsic $\epsilon_i(\mathcal{D})^*$ computed using the empirical bound. For neural networks (last 3 lines), no theoretical sensitivity bound is known.

	CIFAR2	MNIST-binary	Adult	Forest
Noiseless	0.788(4)	0.953(1)	0.8340(7)	0.771(2)
$\Delta_2(f) = \hat{\Delta}_S$				
SGD_d	0.719(2)	0.853(2)	0.53(1)	0.75(1)
SGD_r	**+.00002**	**+.0002**	**+.0020**	**+.0013**
% of gap	0.03%	0.15%	0.66%	6.46%
$\Delta_2(f) = \hat{\Delta}_S^*$				
SGD_d	0.763(4)	0.941(2)	0.810(8)	0.767(6)
SGD_r	**+.0006**	**+.0007**	**+.0067**	**+.0023**
% of gap	2.54%	5.25%	36.31%	14.13%

Table 4: We report the (binary) accuracy of private and non-private models for logistic regression using $\epsilon = 1$. Brackets indicate the standard deviation in the final digit. Bold face indicates a statistically significant improvement (paired t-test, p-val $< 10^{-6}$). The percentage improvement is over the gap between SGD_d and the noiseless performance.

networks we study do not exhibit practically useful $\epsilon_i(\mathcal{D})$ values. To study 'private' model performance, we compute the target σ of the Gaussian mechanism to produce a private model using output perturbation, and modify it for three scenarios:

(1) Noiseless ($\sigma = 0$)
(2) 'SGD as deterministic' (SGD_d); the setting in [31]. We estimate the required σ (σ_{target}) using the Gaussian mechanism and the sensitivity $\hat{\Delta}_S$ of SGD.
(3) 'SGD with unknown seed' (SGD_r); thinking of SGD as a randomised mechanism, we estimate the required σ using Eqn 2

We also include the setting where the sensitivity is computed empirically to determine σ_{target}, corresponding to the optimistic bound. As the performance of each trained model varies, we perform paired t-tests between the three settings for a fixed model, for 500 randomly-sampled models for each dataset. In Table 4, we report the utility for $\epsilon = 1$. We include $\epsilon = 0.5$ in Appendix Table 7. We see that the 'augmented DP-SGD' (SGD_r) setting produces a model with consistently and significantly superior utility to one which does not take intrinsic randomness into account. Using the empirical bound $\hat{\Delta}_S$ produces a further improvement in utility, resulting in a setting where accounting for randomness can close the gap between a private and noiseless model by 36.31%. Further work will be required to explore whether SGD can be engineered to produce increased $\sigma_i(\mathcal{D})$ without impacting sensitivity or utility, allowing for further improvements to private model performance. We present results on additional analyses in Appendix D.

5 CONCLUSION

We have taken the first steps towards examining the data-dependent inherent randomness in SGD from a privacy perspective. Using a large-scale experimental study we have quantified the variability of SGD due to random seed and related this to its data-dependent sensitivity and the notion of an 'intrinsic $\epsilon_i(\mathcal{D})$' in the sense of differential privacy. These findings demonstrate that the choice of random seed has a strictly greater impact on the resulting weights of the model than perturbations in the data for both convex and non-convex models considered. By accounting for this variability, statistically significant performance improvements can be achieved for low-dimensional models. We have further demonstrated that existing theoretical bounds on the data-dependent sensitivity of SGD on convex objectives are loose, and using optimistic empirical 'bounds', private model performance can be greatly improved.

REFERENCES

[1] Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. 2016. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 308–318.

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

[3] Léon Bottou. 2012. Stochastic gradient descent tricks. In *Neural networks: Tricks of the trade*. Springer, 421–436.

[4] Stefano Capparelli, Margherita Maria Ferrari, Emanuele Munarini, and Norma Zagaglia Salvi. 2018. A Generalization of the 'Probleme des Rencontres'. *J. Integer Seq* 21 (2018), 1828.

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

[6] Denis J. Dean and Jock A. Blackard. 1998. Comparison of neural networks and discriminant analysis in predicting forest cover types.

[7] Dheeru Dua and Casey Graff. 2017. UCI Machine Learning Repository. (2017). http://archive.ics.uci.edu/ml

[8] Cynthia Dwork. 2011. Differential privacy. *Encyclopedia of Cryptography and Security* (2011), 338–340.

[9] Cynthia Dwork and Aaron Roth. 2014. The Algorithmic Foundations of Differential Privacy. *Foundations and Trends in Theoretical Computer Science* 9 (2014), 211–407.

[10] Jonathan Frankle and Michael Carbin. 2018. The Lottery Ticket Hypothesis: Finding Sparse, Trainable Neural Networks. In *ICLR*.

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

[12] Robin C Geyer, Tassilo Klein, and Moin Nabi. 2017. Differentially private federated learning: A client level perspective. *arXiv preprint arXiv:1712.07557* (2017).

[13] Moritz Hardt, Benjamin Recht, and Yoram Singer. 2015. Train faster, generalize better: Stability of stochastic gradient descent. *arXiv preprint arXiv:1509.01240* (2015).

[14] Peter Henderson, Riashat Islam, Philip Bachman, Joelle Pineau, Doina Precup, and David Meger. 2017. Deep Reinforcement Learning that Matters. *ArXiv* abs/1709.06560 (2017).

[15] Prateek Jain, Dheeraj Nagaraj, and Praneeth Netrapalli. 2019. Sgd without replacement: Sharper rates for general smooth convex functions. *arXiv preprint arXiv:1903.01463* (2019).

[16] Nitish Shirish Keskar, Dheevatsa Mudigere, Jorge Nocedal, Mikhail Smelyanskiy, and Ping Tak Peter Tang. 2016. On Large-Batch Training for Deep Learning: Generalization Gap and Sharp Minima. *ArXiv* abs/1609.04836 (2016).

[17] Alex Krizhevsky. 2009. Learning Multiple Layers of Features from Tiny Images.

[18] Ilja Kuzborskij and Christoph H. Lampert. 2017. Data-Dependent Stability of Stochastic Gradient Descent. *ArXiv* abs/1703.01678 (2017).

[19] Yann LeCun. 1998. Gradient-based learning applied to document recognition.

[20] Stephan Mandt, Matthew D. Hoffman, and David M. Blei. 2017. Stochastic Gradient Descent as Approximate Bayesian Inference. *J. Mach. Learn. Res.* 18 (2017), 134:1–134:35.

[21] Sebastian Meiser. 2018. Approximate and Probabilistic Differential Privacy Definitions. *IACR Cryptology ePrint Archive* 2018 (2018), 277.

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

[23] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research* 12 (2011), 2825–2830.

[24] Arun Rajkumar and Shivani Agarwal. 2012. A differentially private stochastic gradient descent algorithm for multiparty classification. In *Artificial Intelligence and Statistics*. 933–941.

[25] Nornadiah Mohd Razali, Yap Bee Wah, et al. 2011. Power comparisons of shapiro-wilk, kolmogorov-smirnov, lilliefors and anderson-darling tests. *Journal of statistical modeling and analytics* 2, 1 (2011), 21–33.

[26] Ohad Shamir. 2016. Without-replacement sampling for stochastic gradient methods. In *Advances in neural information processing systems*. 46–54.

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

[28] Samuel L. Smith and Quoc V. Le. 2017. A Bayesian Perspective on Generalization and Stochastic Gradient Descent. *ICLR* abs/1710.06451 (2017).

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

[30] Florian Tramèr, Fan Zhang, Ari Juels, Michael K Reiter, and Thomas Ristenpart. 2016. Stealing Machine Learning Models via Prediction APIs.. In *USENIX Security Symposium*.

[31] Xi Wu, Fengan Li, Arun Kumar, Kamalika Chaudhuri, Somesh Jha, and Jeffrey Naughton. 2017. Bolt-on differential privacy for scalable stochastic gradient descent-based analytics. In *Proceedings of the 2017 ACM International Conference on Management of Data*. ACM, 1307–1322.

A SENSITIVITY OF SGD FOR CONVEX FUNCTIONS

Assumptions. Let $\mathcal{W} \subseteq \mathbb{R}^p$ be the hypothesis space, and $\mathcal{L} : \mathcal{W} \mapsto \mathbb{R}$ the loss function. We assume that $\forall u, v \in \mathcal{W}$:

- $\mathcal{L}$ is convex; i.e., $\mathcal{L}(u) \geq \mathcal{L}(v) + \langle \nabla(v), u - v \rangle$
- $\mathcal{L}$ is L-Lipschitz i.e., $\|\mathcal{L}(u) - \mathcal{L}(v)\| \leq L\|u - v\|$
- $\mathcal{L}$ is β-smooth; i.e., $\|\nabla\mathcal{L}(u) - \nabla\mathcal{L}(v)\| \leq \beta\|u - v\|$

We present the results for the sensitivity of SGD due to a change input datapoint as provided by Wu et al. [31]:

Theorem A.1. **From [31].** *Let A denote the SGD algorithm using r as the random seed then the upper bound for sensitivity for k-passes of SGD is given by, $\hat{\Delta}_S = \max_r \|A(r; S) - A(r; S')\| \leq 2kL\eta$*

Here, $\hat{\Delta}_S$ gives the maximum difference in the model parameters due to the presence or absence of a single input sample. Their results rely on the *boundedness* and *expansiveness* properties for the gradient update rule (G) of SGD as proposed by Hardt et al. [13]:

Property 1. *(***Boundedness of G.***) For a loss function that is L-Lipschitz and learning rate η, the gradient update of SGD is ηL bounded i.e., $\sup_{w \in \mathcal{W}} \|G(w) - w\| \leq \eta L$*

Property 2. *(***Expansiveness of G.***) For a loss function that is β-smooth, and $\eta \leq 2/\beta$, then the gradient update of SGD is 1-expansive i.e., $\sup_{w,w'} \frac{\|G(w) - G(w')\|}{\|w - w'\|} \leq 1$*

As this is not the main contribution of our paper, we refer interested readers to the original paper for a formal proof [31]. We provide here a brief intuition for achieving the bound: For a single pass of SGD over neighbouring datasets S and S' with a *fixed initialization* and *fixed sampling strategy*, the two executions G and G' will differ only at a single step — when the differing sample gets selected. In that case, from the above boundedness property, we have that $G(w) - G'(w') \leq 2L\eta$. For all the other steps, the samples selected are exactly same and hence the 1-expansiveness property applies. Therefore, after k-passes of SGD over the dataset, the difference in the model parameters will have an upper bound of $2kL\eta$. When trained using a batchsize of B, the sensitivity bound can be reduced by a factor of B i.e., $\hat{\Delta}_S \leq 2kL\eta/B$. Henceforth, the theoretical sensitivity always refers to the one with batchsize B.

B UPPER BOUND ON VARIABILITY DUE TO THE RANDOMNESS IN SGD

We use the boundedness and expansivity properties of the gradient update rule to calculate the upper bound for the variability in SGD. Here, we focus only on the difference in model parameters due to the stochastic process of selecting samples during training - including the variability in model initialisation will only increase the variability, as the difference between model weights at time $T = 0$ is non-zero.

We use a similar argument as in prior work for calculating the bound at each step of SGD. For a single pass of SGD on dataset S with *fixed initialization* but different random seeds r and r' for sampling inputs, in the 'best' case, every step encounters different samples. Thus, by boundedness, each step will add at most a $2L\eta$ deviation between the model parameters. Therefore, after k passes of SGD through a dataset of size N where each step selects differing samples, we get a variability bound of:

$$\hat{\Delta}_V = \max_{r,r'} \|A(r; S) - A(r'; S')\| \leq 2kLN\eta. \tag{3}$$

Claim 1. *The upper bound of variability due to the randomness in SGD is strictly greater than of the sensitivity of SGD due to the change in a single input sample i.e., $\hat{\Delta}_V > \hat{\Delta}_S$*

The above claim gives a weak guarantee about the inherent noise in SGD as it considers the *upper* bound of the variability. This assumed a 'worst' (or best)-case scenario where different batches are sampled at *every* step, comparing between two runs of the experiment. In reality, there is a chance for two runs of SGD to sample the same example at the same point during training. To tighten the bound on $\hat{\Delta}_V$, we can try to account for this distribution over permutations. We consider the upper bound of $\hat{\Delta}_V$ (which is at most $2kLN\eta$) to be a random variable itself, and use the Chebyshev inequality to demonstrate that it is usually larger than the sensitivity (see Section B.0.1 for proof):

Claim 2. *The bound on the variability of SGD is larger than its sensitivity with high probability.*

$$P\left[|\hat{\Delta}_V - \mathbb{E}[\hat{\Delta}_V]| \geq kL\eta(N-2)\right] \leq \frac{4}{k(N-2)^2}$$

Since N is typically large, we see that the probability $\hat{\Delta}_V$ is sufficiently far from its mean and near $\hat{\Delta}_S$ is very low.

These results cannot conclusively affirm the privacy-preserving properties of SGD as they pertain only to *upper* bounds. The lower bound is likely zero *in general* due to collapsed variability after overfitting (or converging to a unique minimum). Determining when the lower bound is nontrivial remains an open research question, however our empirical results indicate that the lower bound also tends to exceed the data-dependent sensitivity (discussed in Section 4).

B.0.1 Proof of Claim 2. For Claim 2, we need the expected value and variance of $\hat{\Delta}_V$. The bounds stated previously rely on the fact that every iteration of SGD with mis-matching samples introduce a term of $2L\eta$ to the (maximum) difference in outputs. For the upper bound, we assumed that *every* sample is mis-matching, that is we compare runs of SGD where one is a perfect derangement of the training-set traversal order of the other. In reality, between two runs with different random seeds, the same example may be encountered at the same time-point; this would constitute a permutation of the training data with a fixed point. If we assume that X_i is the number of fixed points of the training data in epoch i (relative to a fixed reference permutation), the number of *mis-matches* is therefore $N - X_i$, and the bound on the difference of weights is

$$\hat{\Delta}_V = \sum_{i}^{k} 2L\eta(N - X_i). \tag{4}$$

The probability distribution of X_i is

$$P(X_i = j) = \frac{D_{N,j}}{N!}, \tag{5}$$

where N is the number of training examples, and $D_{N,j}$ is a rencontres number giving the number of permutations of length N with j fixed points. For large N, the distribution of rencontres numbers

approaches a Poisson distribution with rate parameter $\lambda = 1$[4], and so both the expected value and variance of X_i are 1: This allows us to use standard properties of expectation and variance, and the fact that the permutation (and thus X_i) selected at each epoch is independent.

$$\mathbb{E}[\hat{\Delta}_V] = \sum_i^k 2L\eta(N - \mathbb{E}[X_i]) = 2kL\eta(N-1) \tag{6}$$

$$\mathbb{V}[\hat{\Delta}_V] = \sum_i^k \mathbb{V}[2L\eta(N - X_i)] = (2L\eta)^2 k \tag{7}$$

We then use the Chebyshev inequality to bound the probability that $\hat{\Delta}_V$ is far from its mean $\mathbb{E}[\hat{\Delta}_V]$. Doing so is interesting because we can prove that $\hat{\Delta}_V$ is unlikely to be near $\hat{\Delta}_S$. If we define $t = |\mathbb{E}[\hat{\Delta}_V] - \hat{\Delta}_S|/2 = kL\eta(N-2)$ then by Chebyshev inequality:

Claim 2. *The bound on the variability of SGD is larger than its sensitivity with high probability.*

$$P\left[|\hat{\Delta}_V - \mathbb{E}[\hat{\Delta}_V]| \geq kL\eta(N-2)\right] \leq \frac{4}{k(N-2)^2}$$

C ESTIMATING $\epsilon_I(\mathcal{D})$ FOR SGD

We think of SGD as a procedure for sampling model weights from some distribution, and aim to understand the parameters of this distribution to characterise its intrinsic privacy with respect to a dataset it is run on. While theoretically characterising SGD as a sampling mechanism is a subject of ongoing research [20], in this section, we propose an algorithm for *empirically* estimating the potential privacy properties of SGD. We outline and motivate the steps of the algorithm in what follows, and summarize the procedure in Algorithm 2.

Computing $\epsilon_i(\mathcal{D})$. We aim to compute what we call the 'data-dependent intrinsic' ϵ of SGD - $\epsilon_i(\mathcal{D})$.[1] To do this, we start by assuming that the noise of SGD is normally distributed. This is a common albeit restrictive assumption [20, 28]. We empirically test the assumption across our datasets in Section D.4 and do not find it to be strongly violated, however we consider weakening this assumption an important future step.

If A is the SGD algorithm, and S is a training dataset of size N (for example, `MNIST`), we therefore assume $A(S) = \bar{\mathbf{w}}_S + \mathbf{w}_\rho$, where $\bar{\mathbf{w}}_S$ is deterministic and *dataset-dependent* and $\mathbf{w}_\rho \sim \mathcal{N}(0, \mathbb{I}\sigma_i(\mathcal{D})^2)$ is the intrinsic random noise induced by the stochasticity of SGD. Based on Theorem 2.3 in Section 2, we then characterize SGD as a Gaussian mechanism with parameters $c^2 > 2\ln(1.25/\delta)$ and $\sigma_i(\mathcal{D}) \geq c\Delta_2(f)/\epsilon_i(\mathcal{D})$. The value of δ is arbitrary, but we set it to $\delta = 1/N^2$ following convention [9].

Assuming we know $\sigma_i(\mathcal{D})$, δ, and $\Delta_2(f)$, we calculate $\epsilon_i(\mathcal{D})$ as:

$$\epsilon_i(\mathcal{D}) = \frac{\sqrt{2\log 1.25/\delta}\Delta_2(f)}{\sigma_i(\mathcal{D})} \tag{8}$$

For $\epsilon_i(\mathcal{D}) > 1$, Theorem 2.3 does not hold. In this case we interpret $\epsilon_i(\mathcal{D})$ as a way to capture the relationship between the sensitivity $\Delta_2(f)$ and variability $\sigma_i(\mathcal{D})$ of SGD on a given dataset given δ.

[1] Although the notation does not capture it, we assume an implicit model-dependence of $\epsilon_i(\mathcal{D})$ throughout.

Algorithm 2 Estimating ϵ_i empirically

Given neighbouring datasets $\mathcal{S} = \{S_a\}_a^{|S|}$, random seeds $\mathcal{R} = \{r_i\}_i^R$, SGD algorithm A with batch size B, fixed learning rate η, number of epochs k, δ, Lipschitz constant L.
for all $S_a \in \mathcal{S}$ **do**
  for all $r \in \mathcal{R}$ **do**
    $\mathbf{w}_{r,a} \leftarrow A(S_a; r)$ ▷ Run SGD on S_a with seed r
procedure COMPUTE SENSITIVITY
  for $r \in \mathcal{R}$ **do**
    for $S_a, S_b \in \mathcal{S}$ **do**
      $\Delta_S^{rab} \leftarrow \|\mathbf{w}_{r,a} - \mathbf{w}_{r,b}\|$ ▷ Pairwise sensitivity
  $\hat{\Delta}_S \leftarrow 2kL\eta/B$ ▷ Theoretical bound
  $\hat{\Delta}_S^* \leftarrow \max_{r,a,b} \Delta_S^{rab}$ ▷ Empirical bound
procedure COMPUTE VARIANCE
  for all $S_a \in \mathcal{S}$ **do**
    $\bar{\mathbf{w}}_a \leftarrow \frac{1}{R}\sum_r \mathbf{w}_{r,a}$
    $\sigma_a \leftarrow \text{stddev}\left(\text{flatten}\left(\mathbf{w}_{r,a} - \bar{\mathbf{w}}_a\right)\right)$
  $\sigma_i \leftarrow \min_a^{|S|} \sigma_a$
procedure COMPUTE EPSILON
  $c \leftarrow \sqrt{2\log(1.25)/\delta} + 1 \times 10^{-5}$
  $\epsilon_i \leftarrow c\hat{\Delta}_S/\sigma_i$ ▷ Get values for Sensitivity & Variance
  $\epsilon_i^* \leftarrow c\hat{\Delta}_S^*/\sigma_i$ ▷ Using empirical bound
  return ϵ_i, ϵ_i^*

Computing Sensitivity. As per definition 2.2, the sensitivity of SGD is given by the largest ℓ_2-norm change in model weights obtained from neighbouring datasets. We can empirically compute (an estimate of) this value for both convex and non-convex models. Details of the set-up of this empirical study are provided in Section 4.

First, we can compute a 'pairwise' sensitivity between models trained with the same seed (r) on neighbouring datasets (S_i, S_j):

$$\Delta_S^{rij} = \|A(S_i, r) - A(S_j, r)\| \tag{9}$$

Taking the maximum of Δ_S^{rij} we obtain the 'global' (dataset-specific) sensitivity, which we estimate empirically using a subset of i, j, r, as $\hat{\Delta}_S^*$:

$$\hat{\Delta}_S^* = \max_{i,j,r} \|A(S_i, r) - A(S_j, r)\| \tag{10}$$

We assume we have access to a public dataset from which this value can be estimated.

If we consider the pairwise sensitivity, we obtain a *distribution* of $\epsilon_i(\mathcal{D})$ values which would be obtained by considering *subsets* of permissible neighbouring datasets. This variant of sensitivity computation is similar to the notion of smooth sensitivity for a given dataset instance which is emerging as a promising approach for designing better differentially-private mechanisms [22].

Lastly, for convex models, we can also take the (bound on the) theoretical sensitivity of SGD $\hat{\Delta}_S$ estimated by Hardt et al. [13], Wu et al. [31] and described in Section 2. There is no corresponding theoretical sensitivity bound for the non-convex models. We discuss the implication of these sensitivity values ($\hat{\Delta}_S^*, \Delta_S^{rij}, \hat{\Delta}_S$) in our evaluation.

Computing Variance. For computing variance, the object of central interest is $\sigma_i(\mathcal{D})$, which is the (assumed diagonal) covariance of $\mathbf{w}_\rho$ - the 'noise' added to the final weights by the stochasticity in SGD. We can obtain samples of $\mathbf{w}_\rho$ by running SGD with different

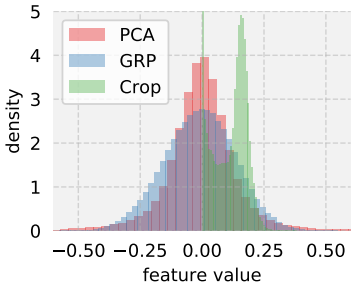

Figure 2: Distribution of feature values in three variants of MNIST-binary using either PCA preprocessing, Gaussian random projections (GRP), or cropping to the central (10×10) square and flattening (Crop).

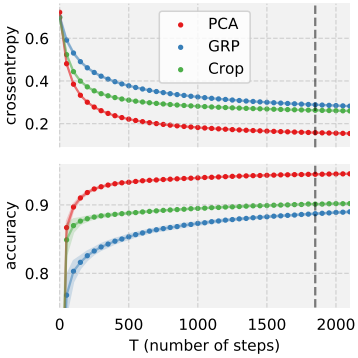

Figure 3: A comparison of training set loss (binary cross-entropy) and accuracy as a function of training steps for the three preprocessing approaches tested on MNIST-binary. The selected convergence point ($T = 1850$) is indicated by a vertical dashed line.

random seeds and subtracting the data-dependent mean value $\bar{\mathbf{w}}_S$. Estimating $\sigma_i(\mathcal{D})$ then amounts to computing the standard deviation of the (flattened) estimated $\mathbf{w}_\rho$.[2] In practice, a separate $\sigma_i(\mathcal{D})$ can be estimated for each dataset S, and we take the *minimum* observed value, although we find it to be broadly independent of S (see Section D.4). As for $\hat{\Delta}^*_S$, we assume for the purpose of this work that the user has access to a public dataset whose distribution is sufficiently similar to the private dataset, such that $\sigma_i(\mathcal{D})$ can be estimated. This is similar to the setting of fine-tuning in Wu et al. [31], where the private dataset is used predominantly to fine-tune a model already trained on a similar, but public dataset.

D FURTHER ANALYSES

In this section we augment our main findings of Section 4 with further analyses. To better understand the impact of data preprocessing we explore three variants of MNIST-binary (Section D.1), and investigate how the number of training steps T influences both sensitivity and variability in Section D.2. In Section D.3 we report results for a convolutional neural network on the full MNIST dataset. Finally, we explore the validity of our assumptions and experimental design in Section D.4.

D.1 Effect of preprocessing dataset

As we have seen, there is variation in the values of $\sigma_i(\mathcal{D})$ and $\epsilon_i(\mathcal{D})$ across datasets. To further explore the data-dependence of our findings, we performed variants of the experiment *within* a dataset (MNIST-binary), where we apply different dimensionality reduction methods before applying the logistic regression model:

(1) (PCA) Principal component analysis, as in [1], to $d = 50$
(2) (GRP) Gaussian random projections, as in [31], to $d = 50$
(3) (Crop) Cropping to the 10×10 central square of the image and flattening ($d = 100$)

In all cases, we still scale $||\mathbf{x}|| \leq 1$.

Figure 2 shows how this preprocessing changes the underlying dataset statistics. For Crop, the data remains sparse (46% of feature values are zero), while PCA and GRP produce dense symmetrical distributions with differing levels of kurtosis.

In Figure 3 we show the training curves aggregated across experiments from each of these settings using the fixed learning rate of $\eta = 0.5$. We also tested other learning rates, but they did not strongly impact the findings and so for simplicity we fix η across the experiments. As we see, PCA converges more quickly to a better-performing model, so we used this setting in all other analyses on MNIST-binary.

	PCA	GRP	Crop
# experiments	19600	14796	5000
Accuracy (noiseless)	0.953(1)	0.907(2)	0.917(1)
SGD_d ($\epsilon = 1$)	0.941(2)	0.900(4)	0.908(2)
SGD_r ($\epsilon = 1$)	**+0.0007**	**+0.0006**	**+0.0011**
$\hat{\Delta}_S$	0.2523	0.2523	0.2523
$\hat{\Delta}^*_S$	0.0586	0.0473	0.0448
$\sigma_i(\mathcal{D})$	0.0850	0.0989	0.1152
$\epsilon_i(\mathcal{D})$	18.17	15.61	13.41
$\epsilon_i(\mathcal{D})^*$	4.22	2.92	2.38

Table 5: Comparison of the accuracy, empirical sensitivity ($\hat{\Delta}^*_S$, intrinsic variability ($\sigma_i(\mathcal{D})$) and intrinsic $\epsilon_i(\mathcal{D})$ and $\epsilon_i(\mathcal{D})^*$ for the MNIST-binary variants on logistic regression. The theoretical sensitivity $\hat{\Delta}_S$ is identical. We use the empirical sensitivity bound to produce the models SGD_d and SGD_r, following Algorithm 1 in the latter case.

For simplicity, we compare all settings at $t = 1850$ steps (this is the convergence point selected for PCA used elsewhere in the paper). In Table 5 we compare the empirical sensitivity, $\sigma_i(\mathcal{D})$, and resulting 'intrinsic $\epsilon_i(\mathcal{D})$' for the three settings, as well as the noiseless performance of the three models (which can also be seen in Figure 3). In all cases, $\delta = 1/N^2$. Since the learning rate, Lipschitz constant, and number of iterations is the same for all settings, the theoretical bound $\hat{\Delta}_S$ is identical.

We see the largest utility improvement from augmented SGD for the Crop setting, owing to its low $\epsilon_i(\mathcal{D})$ value driven by a relatively higher $\sigma_i(\mathcal{D})$ and lower $\hat{\Delta}^*_S$. However, as the base performance of this model is worse, the resulting private model remains inferior to PCA. This suggests that a practitioner should focus on obtaining the highest-performing model rather than attempting to optimise for $\sigma_i(\mathcal{D})$. However, presence of the variability suggests that modifications to the data distribution can influence $\sigma_i(\mathcal{D})$, and further investigation will be required to characterise this relationship.

[2] We found that other choices for estimating an aggregate $\sigma_i(\mathcal{D})$, such as computing a per-weight $\sigma_i(\mathcal{D})^k$ and then averaging, or estimating the variance of the norm of the weights, produced largely consistent results. It is likely that a superior method for estimating $\sigma_i(\mathcal{D})$ exists, which we leave as a question for future work.

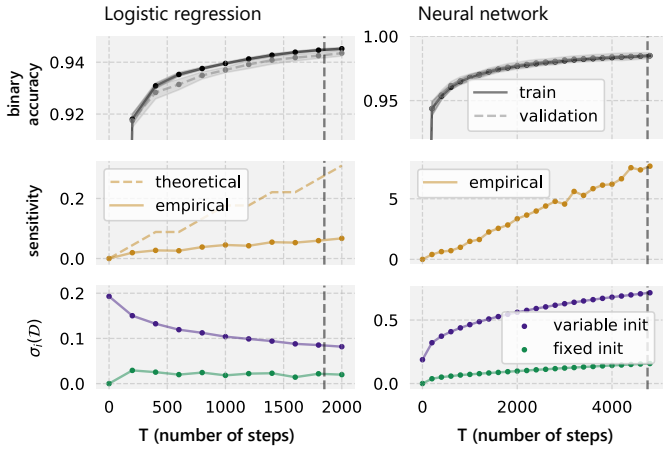

Figure 4: We show how empirical sensitivity and intrinsic variability (both with and without a fixed initial model) depend on the number of training steps. Results shown are for `MNIST-binary` for both logistic regression and neural networks. Appendix D.6 shows results on the other datasets, but they are qualitatively the same. The theoretical sensitivity exhibits a stepwise increase as it increments per epoch. For the neural network, we only report empirical sensitivity. 'fixed init' and 'variable init' indicate whether the initialisation of the model was fixed, or allowed to vary with the seed.

D.2 Dependence on training time

As highlighted by [20], training 'faster' (i.e. converging earlier) produces superior generalisation through smaller sensitivity. This is reflected by the linear dependence on the number of training steps on the theoretical bound $\hat{\Delta}_S$. However, the relationship between the *empirical* sensitivity, as well as the intrinsic variability, and the number of training steps is not known. We use our experimental set-up to explore this dependence.

In Figure 4 we plot $\hat{\Delta}_S$ (if available), estimated $\hat{\Delta}_S^*$, and $\sigma_i(\mathcal{D})$ against the number of training steps T, for `CIFAR2` and the two model classes. Results on the other datasets are included in Appendix Section D.6, but are qualitatively similar.

We can make the following observations:

- For logistic regression, empirical sensitivity $\hat{\Delta}_S^*$ grows with T, but with a slope much lower than predicted by theory, reflecting again that the theoretical bound is not tight. On neural networks, similarly $\hat{\Delta}_S^*$ grows with T. This reflects the tendency towards overfitting, and would likely be mitigated with weight decay.
- The behaviour of $\sigma_i(\mathcal{D})$ for convex models reflects convergence towards the unique minimum of the objective - given random initialisation, $\sigma_i(\mathcal{D})$ is initially large. It then decays as models 'forget' their initialisations and converge towards the minimum. Conversely, with a fixed initialisation the cross-model variability is low, and eventually converges to a steady value corresponding to oscillation around the optimum, with magnitude influenced by the learning rate.
- For the neural networks, we instead see that $\sigma_i(\mathcal{D})$ tends to increase over time regardless of initialisation, indicating that models are converging to increasingly distant locations in parameter space.

Overall we see that there is a tension between $\hat{\Delta}_S^*$ and $\sigma_i(\mathcal{D})$ for selecting T - for neural networks a large value of T would provide large $\sigma_i(\mathcal{D})$, but as $\hat{\Delta}_S^*$ grows more rapidly, the settings we examine would be better served selecting a lower T.

D.3 Multi-class classification

To check if our findings so far are specific to binary classification or 'simple' models, we additionally explore a convolutional neural network (CNN) on the full 10-class classification problem of `MNIST`. In this case, we keep the training examples in their original (28×28) shape and do not enforce $\|\mathbf{x}\| \leq 1$, simply scaling pixel values by 255. As we consider all 10 classes, we use the original dataset with 10000 test examples and 60000 training examples. From these 60000 we use 6000 as the validation set and the remaining 54000 to train the model.

For the CNN we attempt replicate the `cuda-convnet` model used in [13]. This is a CNN with three convolutional layers each followed by a (max) pooling operation, and no dropout. Each convolutional layer uses 8 filters, and the kernel sizes are (3×3), (2×2) and (2×2) respectively. The pool sizes are all (2×2). The output of the final pool is flattened and fed to a fully connected layer mapping it to a hidden size of 10 with relu nonlinearity, which is then mapped to a 10-dimensional softmax output to perform classification. The resulting model has 1448 parameters, and we run 3200 experiments testing a grid of 40 dataset instances and 40 random seeds with fixed and variable model initialisation.

Using a batch size of 32 and a learning rate of 0.1, this architecture achieves an accuracy of $93 \pm 2\%$ after 1000 training steps, which we take as the convergence point.

Figure 5 shows the distribution of Δ_S and Δ_V for this setting.

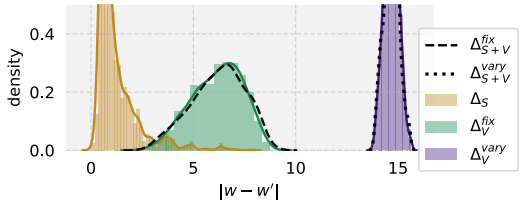

Figure 5: Distribution of differences in model weights from pairs of experiments, for a CNN on `MNIST`. For each experiment pair, we vary the dataset (Δ_S), the random seed (Δ_V), or both (Δ_{S+V}). While varying the random seed, the initialisation of the model is either fixed ($\Delta_V^{\mathbf{fix}}$) or varies with the seed ($\Delta_V^{\mathbf{vary}}$).

We see a similar story to results on other neural networks (second row of Figure 1) suggesting there is nothing unique about fully-connected feed-forward networks not shared by CNNs here. The interesting difference is the sharpness of the Δ_V^{vary} distribution. It is virtually identical to what would result from taking pairwise distances between *random* Gaussian vectors of the same dimension and scale as the learned CNN weights, suggesting a multitude of minima and no obvious 'clustering' of models.

D.4 Empirical validity of findings

Here we test the validity of assumptions and the consistency of the estimates produced by our empirical analysis.

D.4.1 Is the noise in SGD Gaussian? In designing Algorithm 2, we assumed that the noise in the weights of SGD follows a normal distribution with diagonal covariance. In this section we test the assumption that the *marginals* of the weight distribution are normally distributed, that is that $\mathbf{w}_a \sim \mathcal{N}(\mu_a, \sigma_a^2)$ for each a. This is a necessary but not sufficient condition for the joint to be normal, and thus a weaker assumption. We compare the marginals of $\mathbf{w}_a$ by conducting a the Shapiro-Wilk statistical test of normality [25]. The distribution of resulting p-values of this test are shown in Figure 6, which aggregates over weights and using multiple dataset variants.

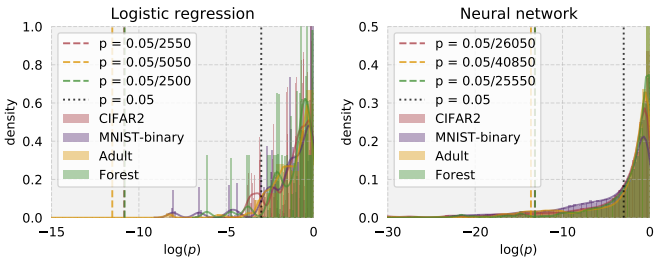

Figure 6: The marginal distribution of most model parameters is consistent with a normal distribution, shown by the distribution of p-values from Shapiro-Wilk [25] test of normality. The density is over each parameter from each model across 50 experiments. Vertical lines indicate the 'standard' $p = 0.05$ cutoff, as well as thresholds corrected for multiple hypothesis testing, using the Bonferroni correction $p = 0.05/M$, where M is the number of hypotheses, in our case this is the number of model parameters times the number of experiments, so $M = 50P$.

Small p-values indicate the hypothesis that the distribution is normal can be rejected. The thresholds for rejection are marked by two vertical lines - the line at $p = 0.05$ reflects a standard threshold for such a statistical test, however as we are performing many tests we also indicate the corrected threshold at $p = 0.05/(Pn_m)$ (Bonferroni correction, using the number of parameters (P) and the number of models whose weights we examined (n_m)). This correction is applied to avoid spurious rejections of the null hypothesis while performing multiple tests. As we can see, the majority of weights would not be rejected at $p = 0.05$, and very few would be rejected at the corrected threshold. This indicates that the distribution of most weights is *marginally* consistent with a normal distribution.

In the event that a weight is *not* normally distributed, this rules out the possibility of the joint distribution being multivariate normal. In such cases, our assumption that the posterior of SGD is normal is violated. In theory, the probability the these underlying assumptions are violated could be incorporated into δ, resulting in probabilistic differential privacy [21]. We leave this accounting to future work, and here retain the caveat that our empirical results do *not* constitute a privacy guarantee for SGD in any case, as our assumptions are overly strong in practice.

D.4.2 Did we run enough experiments? We have explored only a subset of the possible combinations of dataset perturbations and random seeds for each of our data sources, which may introduce uncertainty in our estimates of $\hat{\Delta}_S^*$ and $\sigma_i(\mathcal{D})$. To test this, in Figure 7

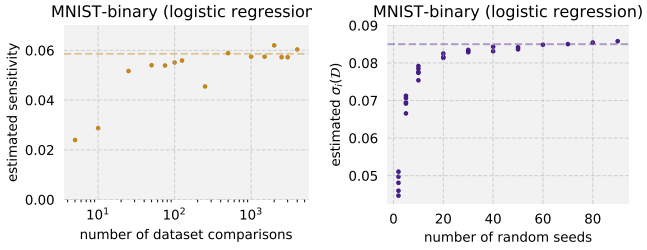

Figure 7: We show how increasing the number of experiments used impact the estimates of empirical sensitivity $\hat{\Delta}_S^*$ (left) and variability $\sigma_i(\mathcal{D})$ (right). Dashed horizontal lines show the values of $\hat{\Delta}_S^*$ and $\sigma_i(\mathcal{D})$ used across results in this paper.

	CIFAR2	MNIST-binary	Adult	Forest
Logistic regression				
# datasets	100	100	75	75
min	0.08267214	0.0849974	0.10835	0.11426
max	0.08267215	0.0849979	0.10836	0.11428
Neural networks				
# datasets	100	50	75	75
min	0.44331	0.71272	0.28796	0.5544
max	0.44339	0.71273	0.28797	0.5545

Table 6: The estimate of $\sigma_i(\mathcal{D})$ is highly stable across dataset instances $\{S_i\}$ for both logistic regression and neural networks across all four datasets. We report the minimum and maximum observed variability estimated from each dataset instance across many dataset instances, retaining digits until the first difference.

we visualise how the estimates of $\hat{\Delta}_S^*$ and $\sigma_i(\mathcal{D})$ change as we use more data (that is, include more experiments) for `MNIST-binary`. Other models and datasets are included in Appendix D.7.

As we can see, as the number of experiments used to estimate the values increases, our estimates tend towards a fixed value, suggesting that more experiments would not substantially alter the findings. We see that we are likely under-estimating the sensitivity ($\hat{\Delta}_S^*$) slightly, which is a natural consequence of it being the maximum of an unknown distribution.

In Table 6 we demonstrate that the value of $\sigma_i(\mathcal{D})$ does not depend strongly on the dataset instance used to estimate it. Combined with the observation that the $\sigma_i(\mathcal{D})$ estimate appears to converge after approximately 40 seeds (this is true across all our datasets and models), it appears that running more pairwise experiment comparisons becomes important to better estimate $\hat{\Delta}_S^*$, whose estimates are less stable.

D.5 Utility at $\epsilon = 0.5$

Table 7 replicates Table 4, using $\epsilon = 0.5$. In this more restrictive privacy setting, we see a more obvious degradation in model performance, and gains from the intrinsic noise are more slight. The largest gains tend to be made when the private model is relatively close in performance to the noiseless setting, as any reduction in the added noise has proportionally a greater effect.

	CIFAR2	MNIST-binary	Adult	Forest
$\Delta_2(f) = \hat{\Delta}_S$				
SGD_d	0.691(2)	0.789(3)	*0.277(3)*	0.71(1)
SGD_r	**+.00001**	**+.00002**	**+.0001**	**+.0005**
% of gap	0.01%	0.01%	0.03%	0.82%
$\Delta_2(f) = \hat{\Delta}^*_S$				
SGD_d	0.752(3)	0.912(2)	0.75(1)	0.759(8)
SGD_r	**+.0002**	**+.0004**	**+.0038**	**+.0015**
% of gap	0.43%	0.98%	4.70%	7.17%

Table 7: We report the (binary) accuracy of private and non-private ('gold standard') models on the four datasets for logistic regression using $\epsilon = 0.5$. SGD_d is the setting in [31] where SGD is treated as deterministic and noise is added to the weights per the Gaussian mechanism. SGD_r is the setting we propose, where the intrinsic variability ($\sigma_i(\mathcal{D})$) is used to decrease the magnitude of added noise. Reported are averages across 500 trained models, with brackets showing the standard deviation in the final digit. Bold face indicates a statistically significant improvement (paired t-test, p-val $< 10^{-6}$). The percentage improvement is over the gap between SGD_d and the noiseless performance, indicating how much 'missing' performance in the private model can be regained by accounting for the intrinsic noise. For `Adult` using the theoretical bound, the accuracy (highlighted in italics) is equivalent to the positive label prevalence, so all utility has been lost.

D.6 Dependence on number of training steps for other datasets

Figure 8 replicates Figure 4 for the other three datasets. We see a qualitatively similar story - logistic regression models (first row) approach a fixed $\sigma_i(\mathcal{D})$ owing to their convergence to the neighbourhood of the unique minimum. The empirical sensitivity of the logistic regression models either increases very slowly or appears approximately constant (`Forest`), which may reflect the underlying sensitivity of the optimum. Conversely, for the neural networks we see a steadily increasing empirical sensitivity, which may reflect the tendency for the norm of the model weights to increase during training.

D.7 Consistency of estimates for other datasets

Figure 9 replicates Figure 9 for neural networks including the multi-class CNN (from Section D.3), and the remaining datasets. We see broadly the same trend - the estimate of $\sigma_i(\mathcal{D})$ tends to 'converge' after approximately 40 seeds, while the sensitivity estimate (note the log scale on the x-axis) is less stable. This is likely because $\sigma_i(\mathcal{D})$ appears to be largely unaffected by the dataset instance (See Table 6) and so only depends on the number of seeds, while the sensitivity is influenced by both seed and the pair of dataset instances (see Equation 9).

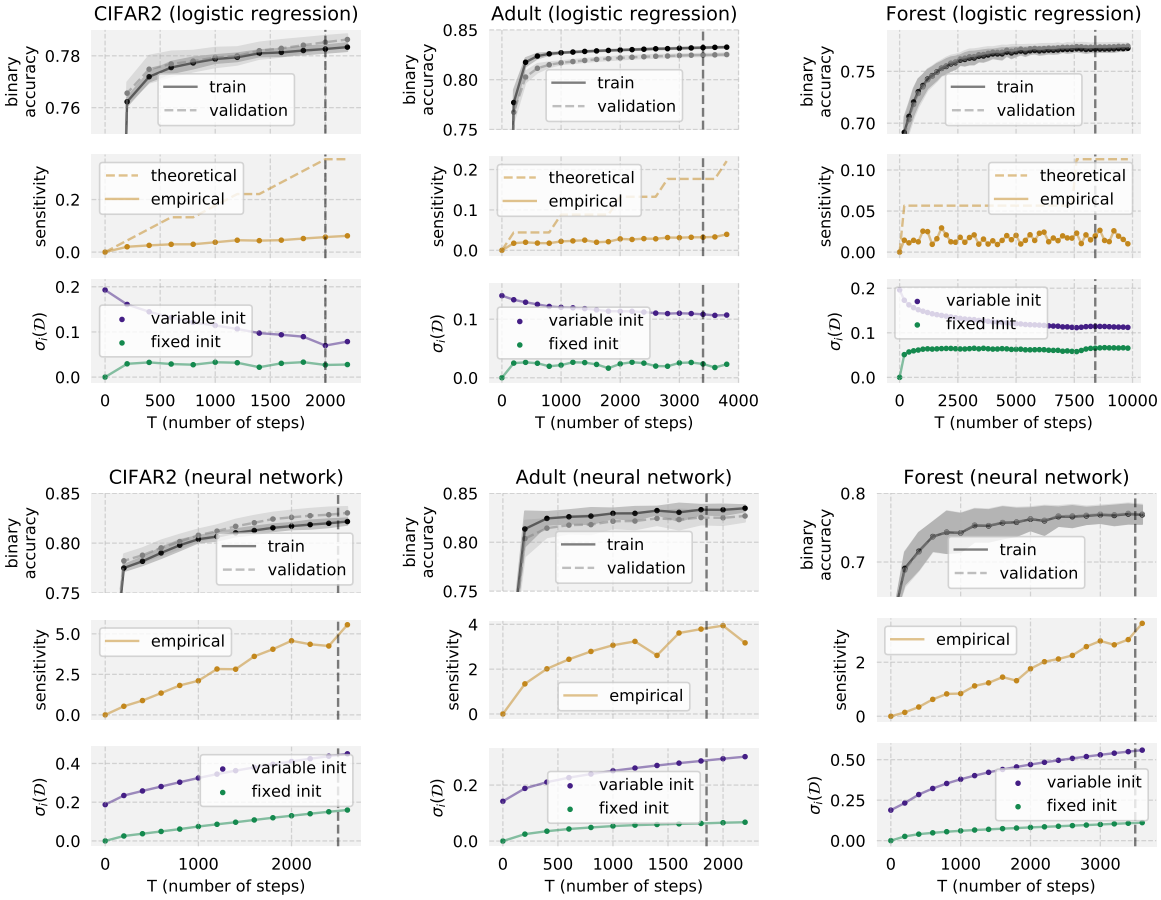

Figure 8: Results on the relationship between sensitivity, seed-dependent variability $\sigma_i(\mathcal{D})$, and steps of SGD T, for the remaining three datasets. As before, $\sigma_i(\mathcal{D})$ tends to increase with T for neural networks, while $\sigma_i(\mathcal{D})$ either decays or rises to an approximately constant value for logistic regression.

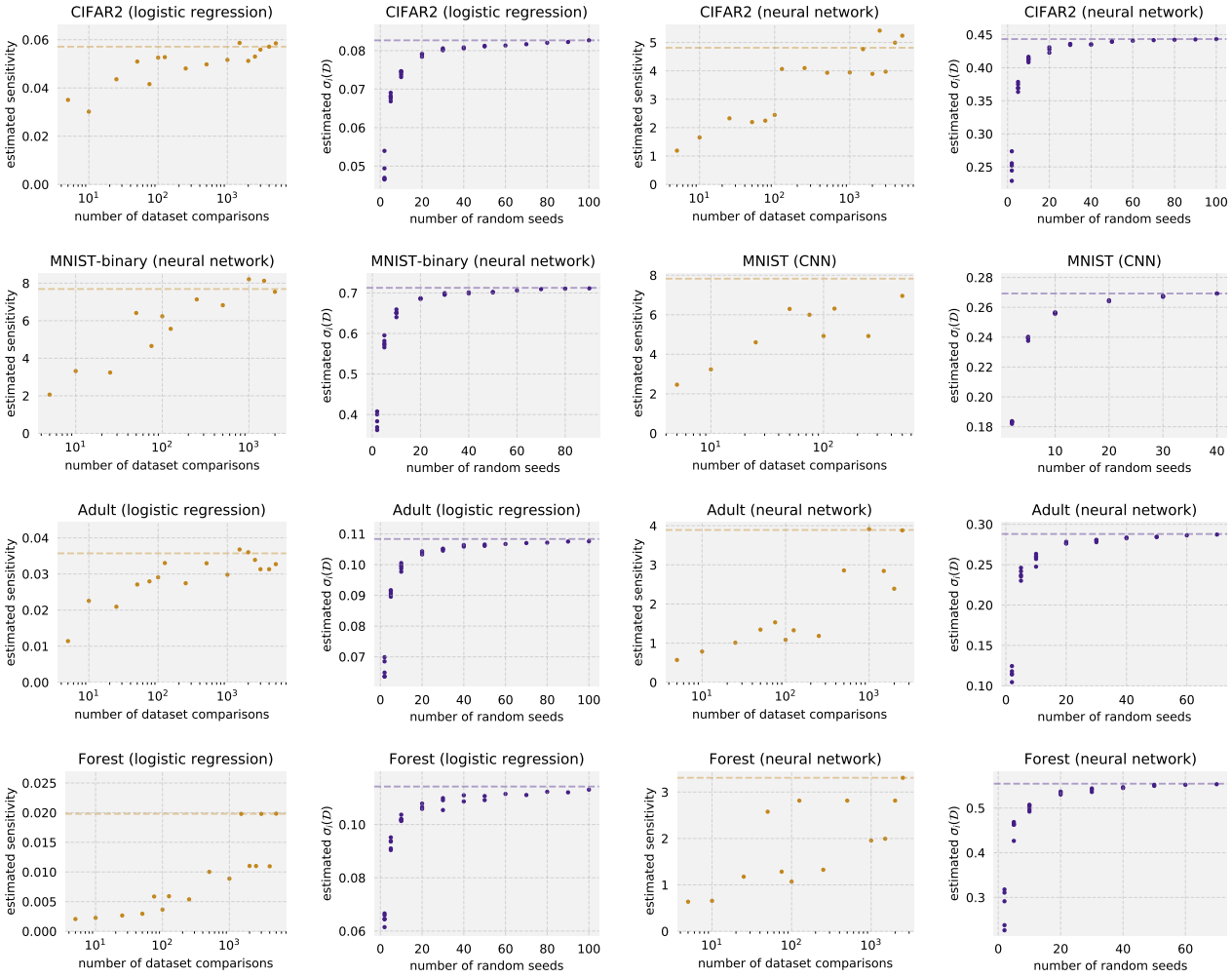

Figure 9: Demonstration of how the estimated values of $\hat{\Delta}_S^*$ and $\sigma_i(\mathcal{D})$ depend on the number of experiments used for estimation, for all datasets and models not included in Figure 7. These results indicate that although we have only run a small fraction of the possible experiments, we would not expect our estimates to change greatly with more experiments. Note that the x-axis for sensitivity estimates (in gold) is using a log scale. Horizontal dashed lines indicate the values for $\hat{\Delta}_S^*$ and $\sigma_i(\mathcal{D})$ used for analyses throughout the paper.