DeepMind

# Unlocking High-Accuracy Differentially Private Image Classification through Scale

Paper: arxiv.org/abs/2204.13650

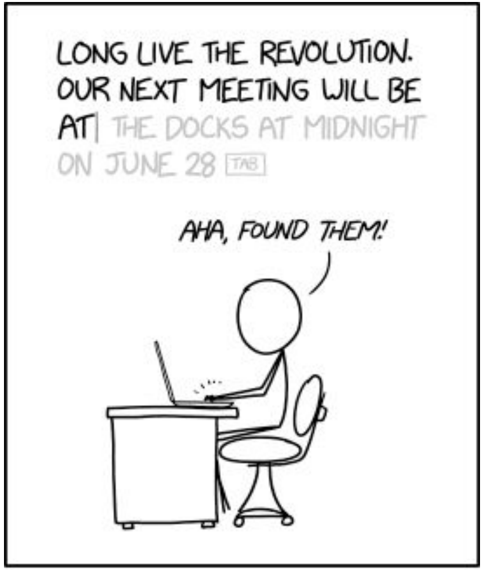
Code: github.com/deepmind/jax_privacy

Soham De

With Leonard Berrada*, Jamie Hayes, Samuel L Smith, Borja Balle

12/05/2022

# Models trained with current ML pipelines can leak training data!




WHEN YOU TRAIN PREDICTIVE MODELS ON INPUT FROM YOUR USERS, IT CAN LEAK INFORMATION IN UNEXPECTED WAYS.

# Models trained with current ML pipelines can leak training data!

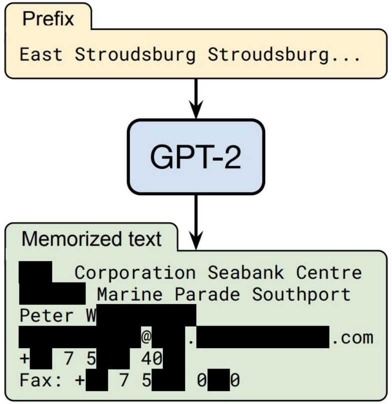
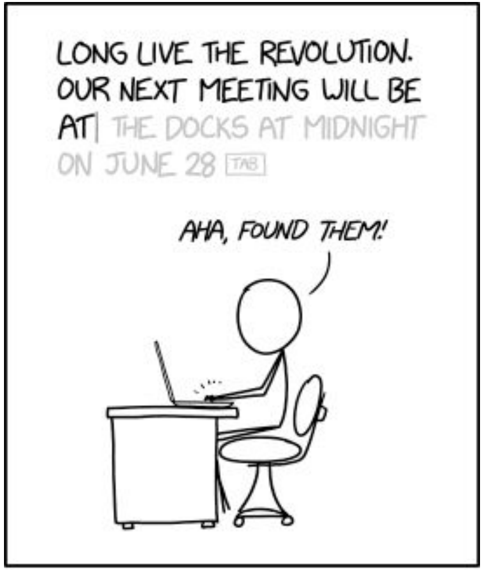



Figure 1: **Our extraction attack.** Given query access to a neural network language model, we extract an individual person's name, email address, phone number, fax number, and physical address. The example in this figure shows information that is all accurate so we redact it to protect privacy.

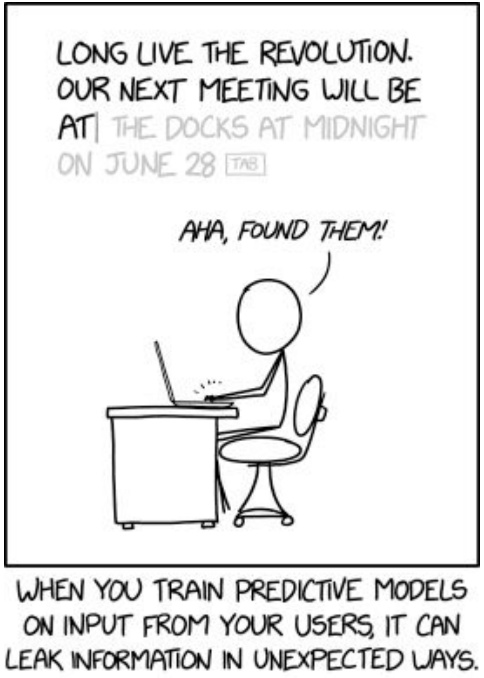



WHEN YOU TRAIN PREDICTIVE MODELS ON INPUT FROM YOUR USERS, IT CAN LEAK INFORMATION IN UNEXPECTED WAYS.

Carlini, Nicholas, et al. "**Extracting training data from large language models**." 30th USENIX Security Symposium (USENIX Security 21). 2021.

# Models trained with current ML pipelines can leak training data!


Fig. 1: Examples of training data points reconstructed from a 55K parameter CNN classifier trained on CIFAR-10.




Borja Balle, Giovanni Cherubin, and Jamie Hayes. "**Reconstructing Training Data with Informed Adversaries**." arXiv:2201.04845 (2022).

# Summary of Results

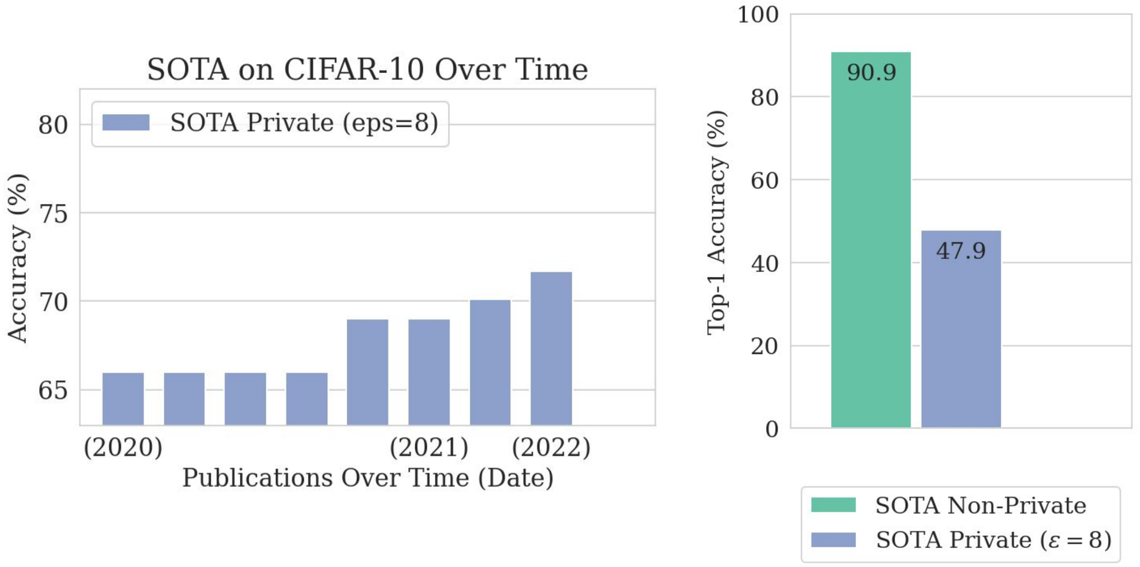
## *Goal*

***Train models with Differential Privacy (DP) to high accuracy → unlock ML on sensitive data***

# Summary of Results

## *Goal*

***Train models with Differential Privacy (DP) to high accuracy → unlock ML on sensitive data***



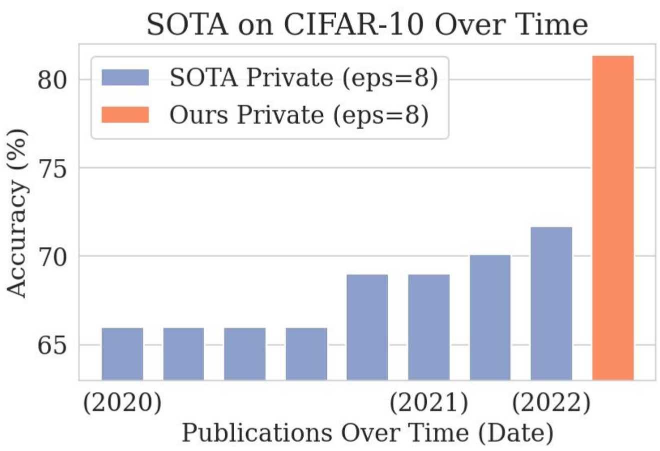
# Summary of Results

## *Goal*

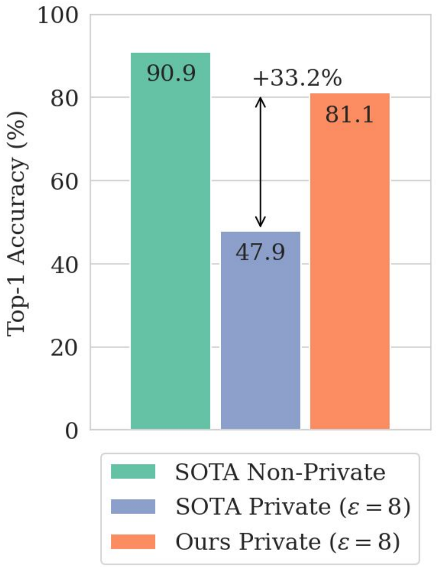
***Train models with Differential Privacy (DP) to high accuracy → unlock ML on sensitive data***

## *Results*

***SOTA on CIFAR10 and ImageNet by large margins***




Largest improvement to date on CIFAR-10




Practically useful levels of performance on ImageNet

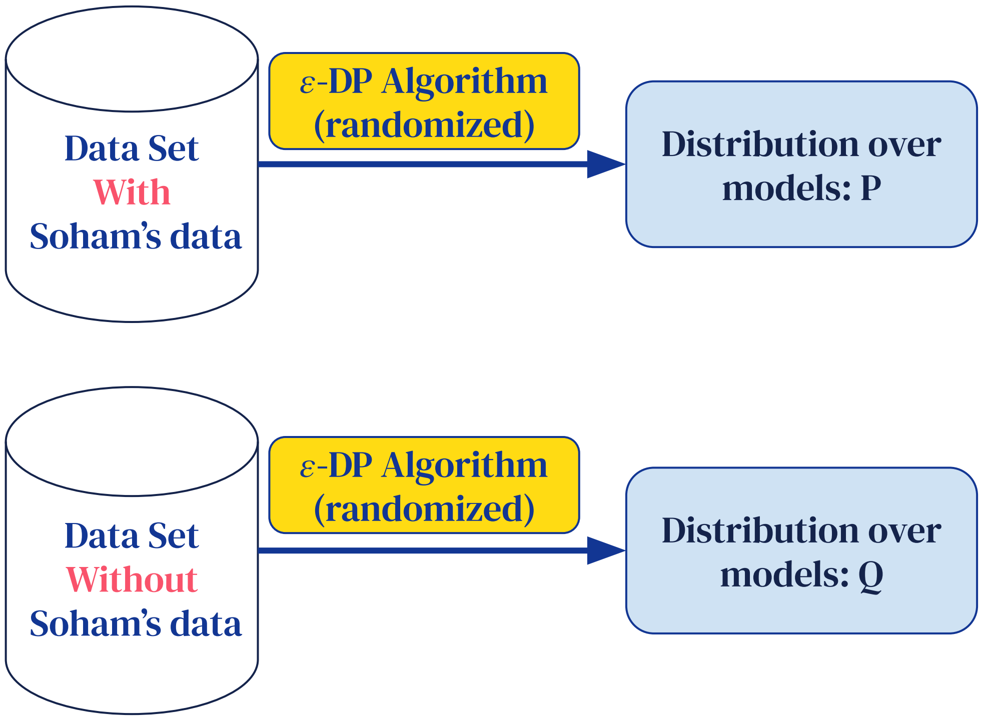
# Talk outline

- What is Differential Privacy (DP)?
- Differentially Private Stochastic Gradient Descent (DP-SGD)
- Improving convergence and trainability of deep networks
- Leveraging pre-training
- Interplay between noise, batch size and compute budget

# Talk outline

- What is Differential Privacy (DP)?
- Differentially Private Stochastic Gradient Descent (DP-SGD)
- Improving convergence and trainability of deep networks
- Leveraging pre-training
- Interplay between noise, batch size and compute budget

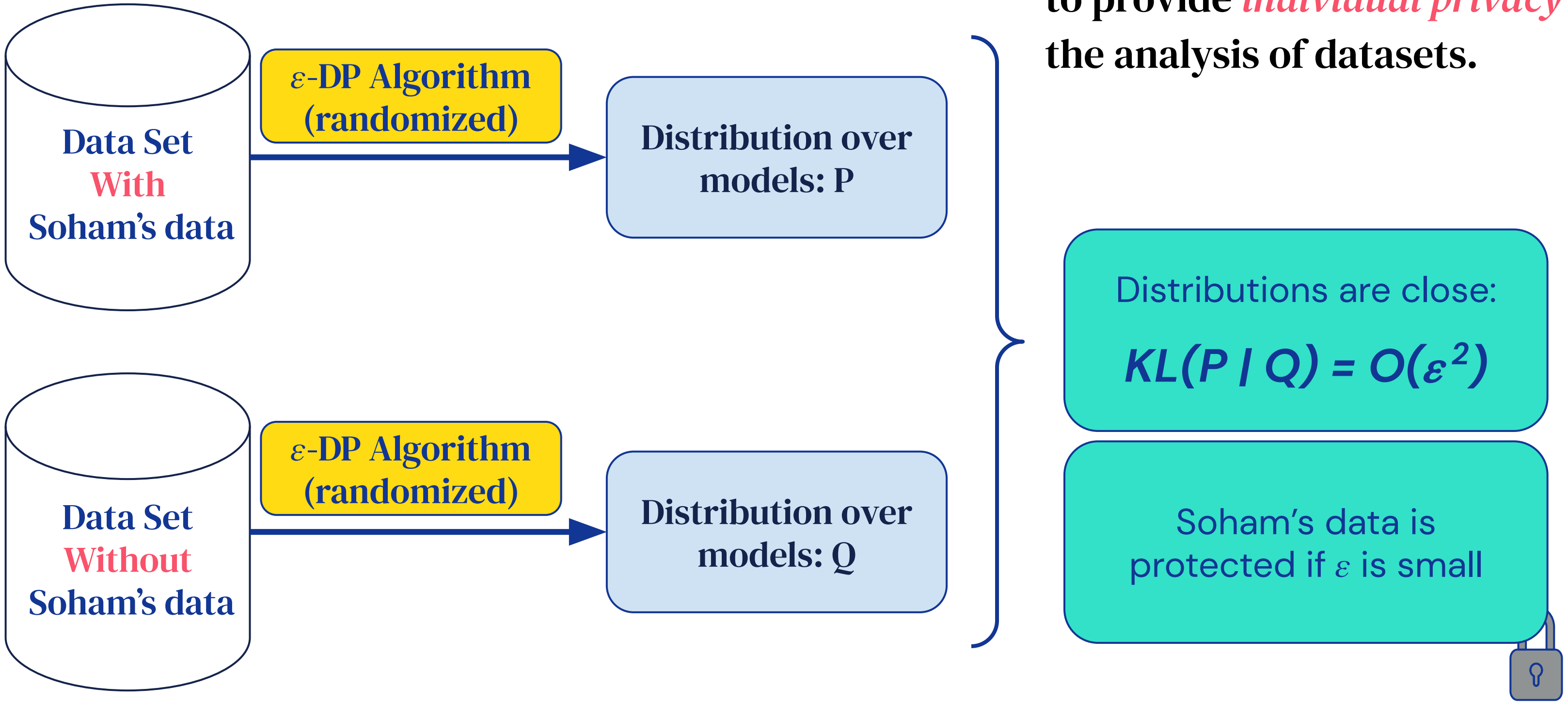
# What is differential privacy?




**Differential privacy is a methodology to provide *individual privacy* during the analysis of datasets.**

# What is differential privacy?

**Differential privacy is a methodology to provide *individual privacy* during the analysis of datasets.**




*DP provides a <u>formal</u> privacy guarantee (defined by $\varepsilon$) against data leakage*

# Formal definition:

**Definition** (Differential Privacy). Let $A : \mathcal{X} \to \mathcal{Y}$ be a randomized algorithm, and let $\varepsilon > 0$, $\delta \in [0, 1]$. We say that $A$ is $(\varepsilon, \delta)$-DP if for any two datasets $D, D' \in \mathcal{X}$ differing by a single element, we have that

$$\forall Y \subset \mathcal{Y}, \ \mathbb{P}[A(D) \in Y] \leq \exp(\varepsilon)\mathbb{P}[A(D') \in Y] + \delta.$$

Dwork, Cynthia, and Aaron Roth. "**The algorithmic foundations of differential privacy.**"
Found. Trends Theor. Comput. Sci. 9.3-4 (2014): 211-407.

# Formal definition

**Definition** (Differential Privacy). Let $A : \mathcal{X} \to \mathcal{Y}$ be a randomized algorithm, and let $\varepsilon > 0$, $\delta \in [0,1]$. We say that $A$ is $\boxed{(\varepsilon, \delta)\text{-DP}}$ if for any $\boxed{\text{two datasets } D, D' \in \mathcal{X} \text{ differing by a single element}}$, we have that

$$\boxed{\forall Y \subset \mathcal{Y},\ \mathbb{P}[A(D) \in Y] \le \exp(\varepsilon)\mathbb{P}[A(D') \in Y] + \delta.}$$

**this implies :** $KL(A(D)|A(D')) = O(\epsilon^2)$

Dwork, Cynthia, and Aaron Roth. "**The algorithmic foundations of differential privacy.**"
Found. Trends Theor. Comput. Sci. 9.3-4 (2014): 211-407.

# An example: Private Averaging


***d*-dimensional vectors of maximum norm C**

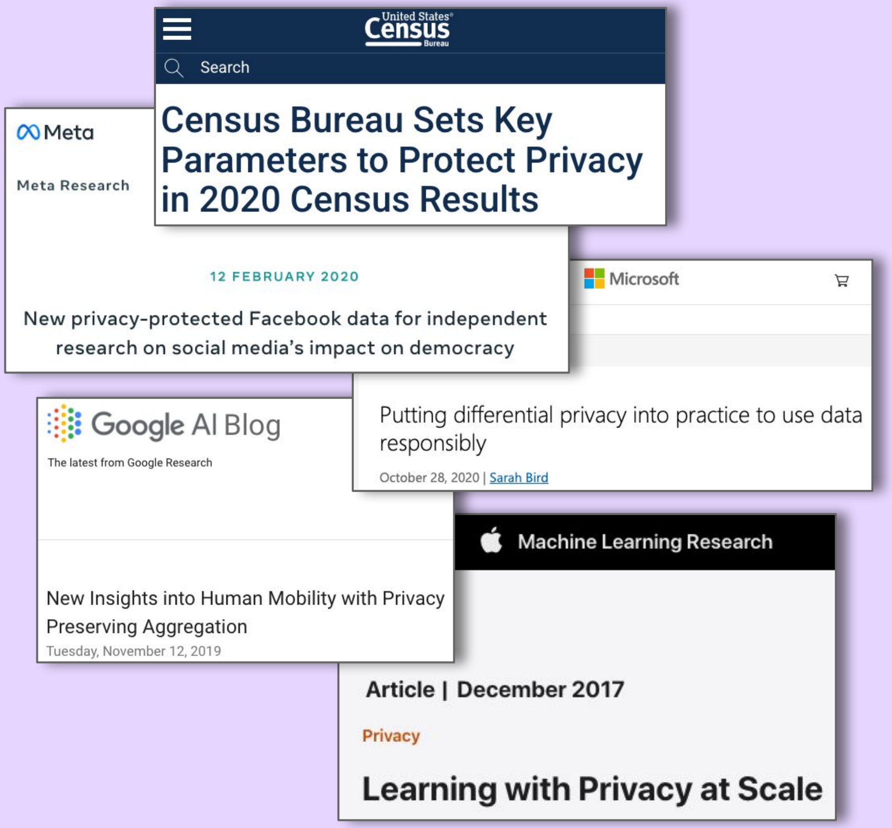
$\tilde{\mathbf{x}} = \hat{\mathbf{x}} + \mathbf{Z}$

$$Z \sim \mathcal{N}\left(0, \frac{C^2}{n^2}\sigma^2 I\right)$$ $\rightarrow$ $\sigma$ depends on privacy guarantee $(\varepsilon, \delta)$

**Privacy-utility trade-off:** $$\mathrm{E}[||\tilde{x} - \hat{x}||] \approx \frac{\sqrt{d}}{\epsilon n}$$

# Why DP is desirable for deploying models on sensitive data

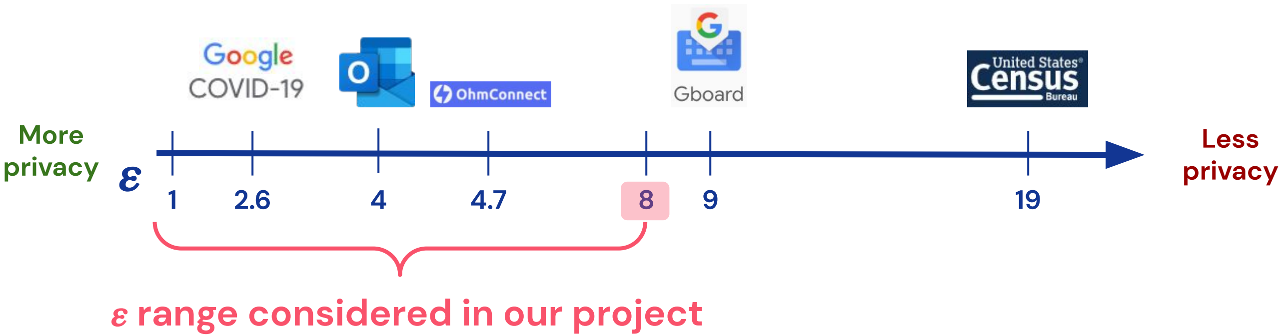
- **Robust to powerful adversaries:**
  - Adversaries with unbounded computation & arbitrary side-knowledge
- **Does not rely on obscurity**
  - Algorithms can be public
- **Quantifiable**
  - with privacy budget $\varepsilon$



# Calibrating the Privacy Budget

The choice of $\varepsilon$ is a policy question that should be informed by:

- Normative privacy requirements of each application
- Utility/accuracy requirements



# Talk outline

- What is Differential Privacy (DP)?
- Differentially Private Stochastic Gradient Descent (DP-SGD)
- Improving convergence and trainability of deep networks
- Leveraging pre-training
- Interplay between noise, batch size and compute budget

# Private ML: Differentially Private SGD (DP-SGD)

$$w^{(t+1)} = w^{(t)} - \eta_t \underbrace{\boxed{\left( \frac{1}{|B|} \sum_{i \in B} \mathtt{clip}_C\left(\nabla l_i(w^{(t)})\right) + \frac{\sigma C}{|B|} \xi \right)}}_{\text{Privatized Average Gradient}}$$

- **Setting**: a trusted party trains the ML model on a private dataset
- Updates only use privatized gradients → **model can be released at any point**

Abadi, Martin, et al. "**Deep learning with differential privacy.**" Proceedings of the 2016 ACM SIGSAC conference on computer and communications security. 2016.

# Private ML: Differentially Private SGD (DP-SGD)

$$w^{(t+1)} = w^{(t)} - \eta_t \left( \frac{1}{|B|} \sum_{i \in B} \mathtt{clip}_C\left(\nabla l_i(w^{(t)})\right) + \frac{\sigma C}{|B|} \xi \right)$$

Clip gradient per sample to norm C

Add Gaussian noise

# Private ML: Differentially Private SGD (DP-SGD)

$$w^{(t+1)} = w^{(t)} - \eta_t \left( \frac{1}{|B|} \sum_{i \in B} \mathtt{clip}_C\left(\nabla l_i(w^{(t)})\right) + \frac{\sigma C}{|B|} \xi \right)$$

Clip gradient per sample to norm C

Add Gaussian noise

**The total privacy loss $\varepsilon$ of the training procedure:**

- Increases with number of iterations
- Decreases with added noise
- Increases with batch size

# Challenges of DP-SGD

- **Bounded privacy budget ε**
  - → tradeoff between 1) # iterations & 2) amount of noise
  - → different hyper-parameter & regularization settings
-
-

# Challenges of DP-SGD

- **Bounded privacy budget ε**
  - → tradeoff between **1) # iterations** & **2) amount of noise**
  - → different hyper-parameter & regularization settings
- Clipping per sample + Noise
  - → **Privatized gradient is biased and has high variance**
-

# Challenges of DP-SGD

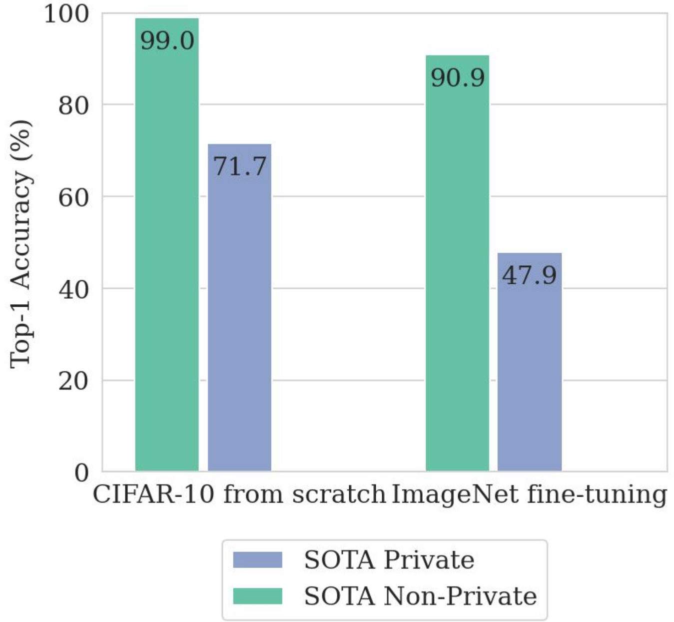
- **Bounded privacy budget ε**
  - → tradeoff between 1) # iterations & 2) amount of noise
  - → different hyper-parameter & regularization settings
- Clipping per sample + Noise
  - → **Privatized gradient is biased and has high variance**
- Making standard models work
  - → **L2 norm of noise scales with model dimension**
  - → **Cannot use batch normalization**

# Prior Work: DP Training in Computer Vision

## Large accuracy drop for DP training

Community focus on:

- Specialized architectures
- Reduction of model dimensionality
- Modifications to DP-SGD





Kurakin, Alexey, et al.. "**Toward Training at ImageNet Scale with Differential Privacy.**" arXiv, 2022.
Tramer, Florian, and Dan Boneh. "**Differentially Private Learning Needs Better Features (or Much More Data).**" ICLR, 2020.
Yu, Da, et al. "**Large Scale Private Learning via Low-rank Reparametrization.**" ICML 2021.
Papernot, Nicolas, et al. "**Tempered Sigmoid Activations for Deep Learning with Differential Privacy.**" AAAI, 2021.

# Our approach

- **Standard deep learning architectures**
  *(unlike community)*

- **Push the limits of vanilla DP-SGD**
  *(using enough compute & careful hyperparam tuning)*

- **Improve trainability & convergence of DP-SGD**
  *(using tricks from non-private training)*

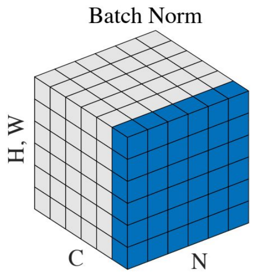
**Getting all the details right was crucial for good performance**

# Talk outline

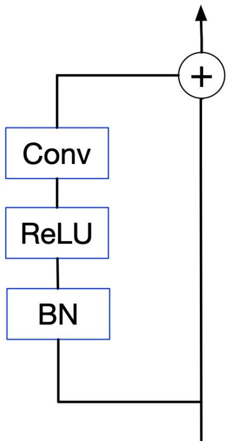
- What is Differential Privacy (DP)?
- Differentially Private Stochastic Gradient Descent (DP-SGD)
- Improving convergence and trainability of deep networks
- Leveraging pre-training
- Interplay between noise, batch size and compute budget

# Improving convergence & trainability

| CIFAR-10 classification under $(8, 10^{-5})$-DP | Accuracy (%) | | | |
|---|---|---|---|---|
| | Validation | | Training | |
| Baseline (WRN-40-4 w/o batch normalization) | 50.8 | (0.7) | 51.2 | (0.7) |

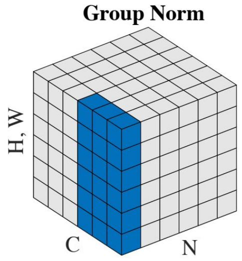



**Standard deep networks for vision rely on batch normalization for good performance**

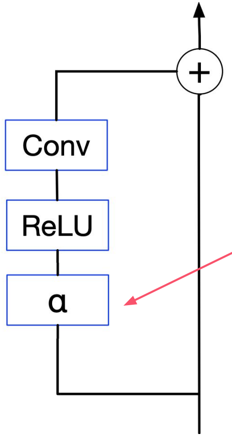


# Improving convergence & trainability

| CIFAR-10 classification under $(8, 10^{-5})$-DP | Accuracy (%) | | | |
|---|---|---|---|---|
| | Validation | | Training | |
| Baseline (WRN-40-4 w/o batch normalization) | 50.8 | (0.7) | 51.2 | (0.7) |
| + Group normalization (16 groups) | 66.3 | (0.6) | 67.9 | (0.3) |




**Replacing with alternate normalizers or normalizer-free methods can recover the benefits of batch normalization**




crucial to place it on the residual branch

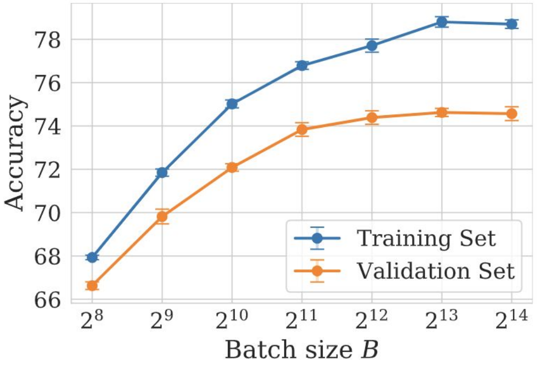
De & Smith. "**Batch normalization biases residual blocks towards the identity function in deep networks.**" NeurIPS (2020).
Brock et al.. "**Characterizing signal propagation to close the performance gap in unnormalized ResNets.**" ICLR (2021).

# Improving convergence & trainability

| CIFAR-10 classification under (8, $10^{-5}$)-DP | Accuracy (%) Validation | | Accuracy (%) Training | |
|---|---|---|---|---|
| Baseline (WRN-40-4 w/o batch normalization) | 50.8 | (0.7) | 51.2 | (0.7) |
| + Group normalization (16 groups) | 66.3 | (0.6) | 67.9 | (0.3) |
| + Larger batch size (batch size of 4096) | 70.0 | (0.6) | 73.4 | (0.9) |

**Larger batch sizes help by reducing the scale of the added noise and improving signal-to-noise ratio of privatized gradient**

$$w^{(t+1)} = w^{(t)} - \eta_t \frac{1}{|B|} \sum_{i \in B} \mathtt{clip}_C\left(\nabla l_i(w^{(t)})\right) - \eta_t \boxed{\frac{\sigma C}{|B|}} \xi$$





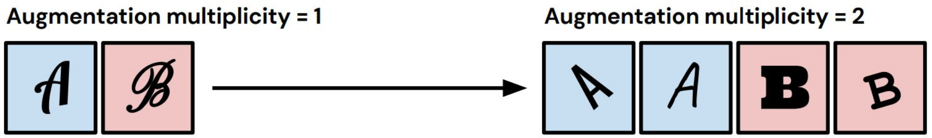
Anil, Rohan, et al. "**Large-scale differentially private BERT**." arXiv:2108.01624 (2021).
Li, Xuechen, et al. "**Large language models can be strong differentially private learners**." ICLR (2022).

# Improving convergence & trainability

| CIFAR-10 classification under $(8, 10^{-5})$-DP | Accuracy (%) | | | |
|---|---|---|---|---|
| | Validation | | Training | |
| Baseline (WRN-40-4 w/o batch normalization) | 50.8 | (0.7) | 51.2 | (0.7) |
| + Group normalization (16 groups) | 66.3 | (0.6) | 67.9 | (0.3) |
| + Larger batch size (batch size of 4096) | 70.0 | (0.6) | 73.4 | (0.9) |
| + Weight standardization | 71.2 | (1.0) | 74.7 | (1.3) |

# Improving convergence & trainability

| CIFAR-10 classification under (8, $10^{-5}$)-DP | Accuracy (%) Validation | | Accuracy (%) Training | |
|---|---|---|---|---|
| Baseline (WRN-40-4 w/o batch normalization) | 50.8 | (0.7) | 51.2 | (0.7) |
| + Group normalization (16 groups) | 66.3 | (0.6) | 67.9 | (0.3) |
| + Larger batch size (batch size of 4096) | 70.0 | (0.6) | 73.4 | (0.9) |
| + Weight standardization | 71.2 | (1.0) | 74.7 | (1.3) |
| + Augmentation multiplicity (16 augmentations) | 78.4 | (0.9) | 79.4 | (0.9) |

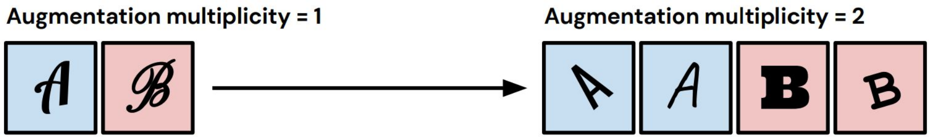




Fort et al. "**Drawing Multiple Augmentation Samples Per Image During Training Efficiently Decreases Test Error**." arXiv:2105.13343
Hoffer, et al. "**Augment your batch: better training with larger batches**." CVPR (2020).

# Improving convergence & trainability

| CIFAR-10 classification under $(8, 10^{-5})$-DP | Accuracy (%) Validation | | Accuracy (%) Training | |
|---|---|---|---|---|
| Baseline (WRN-40-4 w/o batch normalization) | 50.8 | (0.7) | 51.2 | (0.7) |
| + Group normalization (16 groups) | 66.3 | (0.6) | 67.9 | (0.3) |
| + Larger batch size (batch size of 4096) | 70.0 | (0.6) | 73.4 | (0.9) |
| + Weight standardization | 71.2 | (1.0) | 74.7 | (1.3) |
| + Augmentation multiplicity (16 augmentations) | 78.4 | (0.9) | 79.4 | (0.9) |



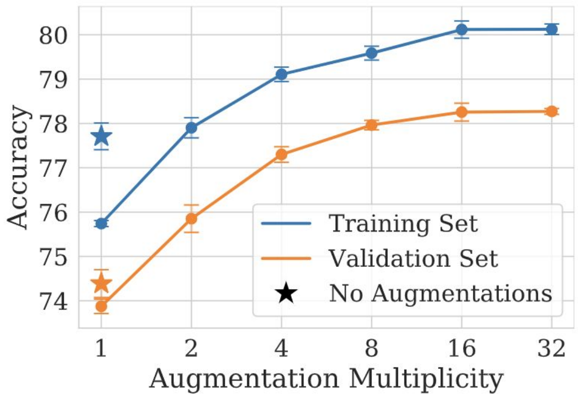

$$w^{(t+1)} = w^{(t)} - \eta_t \frac{1}{|B|} \sum_{i \in B} \texttt{clip}_C \left( \boxed{\frac{1}{|K_i|} \sum_{j \in K_i} \nabla l_j(w^{(t)})} \right) - \eta_t \frac{\sigma C}{|B|} \xi$$

Average over augmentations

# Improving convergence & trainability

| CIFAR-10 classification under $(8, 10^{-5})$-DP | Accuracy (%) | | | |
|---|---|---|---|---|
| | Validation | | Training | |
| Baseline (WRN-40-4 w/o batch normalization) | 50.8 | (0.7) | 51.2 | (0.7) |
| + Group normalization (16 groups) | 66.3 | (0.6) | 67.9 | (0.3) |
| + Larger batch size (batch size of 4096) | 70.0 | (0.6) | 73.4 | (0.9) |
| + Weight standardization | 71.2 | (1.0) | 74.7 | (1.3) |
| + Augmentation multiplicity (16 augmentations) | 78.4 | (0.9) | 79.4 | (0.9) |

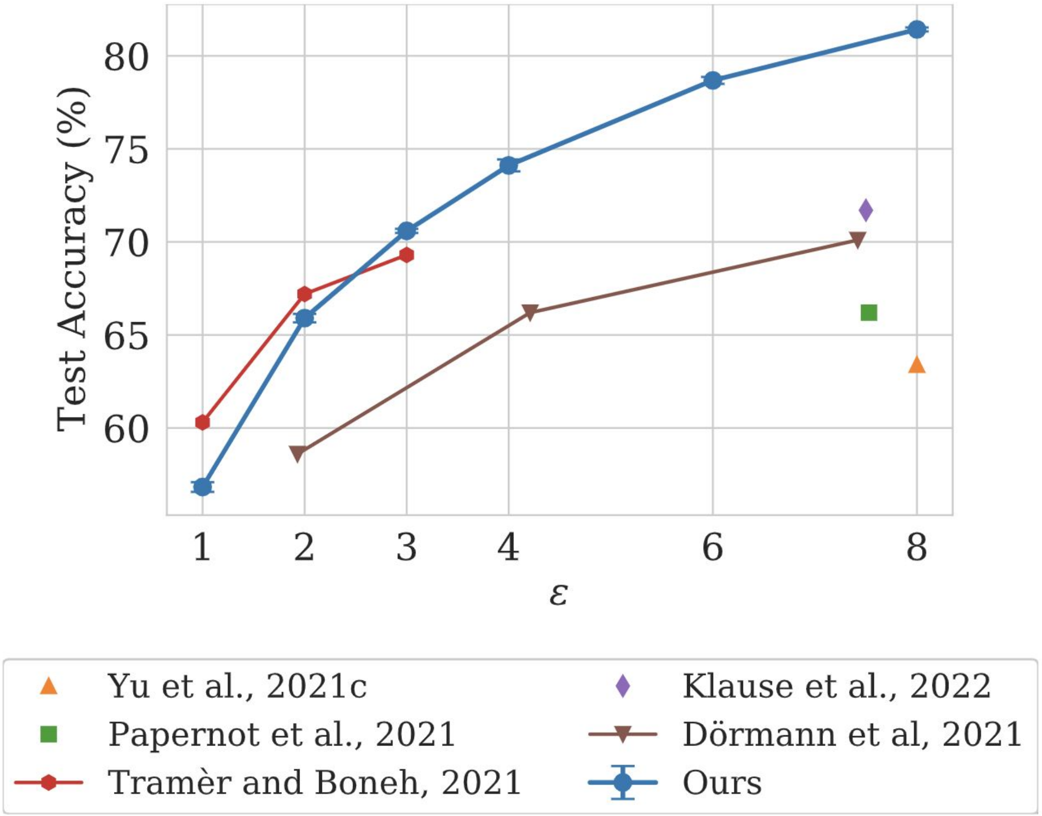
Reduces the variance introduced by data augmentation without incurring any privacy cost



# Improving convergence & trainability

| CIFAR-10 classification under (8, $10^{-5}$)-DP | Accuracy (%) | | | |
|---|---|---|---|---|
| | Validation | | Training | |
| Baseline (WRN-40-4 w/o batch normalization) | 50.8 | (0.7) | 51.2 | (0.7) |
| + Group normalization (16 groups) | 66.3 | (0.6) | 67.9 | (0.3) |
| + Larger batch size (batch size of 4096) | 70.0 | (0.6) | 73.4 | (0.9) |
| + Weight standardization | 71.2 | (1.0) | 74.7 | (1.3) |
| + Augmentation multiplicity (16 augmentations) | 78.4 | (0.9) | 79.4 | (0.9) |
| + Parameter averaging (exponential moving average) | 79.7 | (0.2) | 81.5 | (0.2) |

# Putting it all together: CIFAR-10 w/o extra data




***81.4% test accuracy at $\varepsilon$ = 8***

***Our best results are with a WRN 40-4 & scaling up batch size, augmentation multiplicity & compute***

**we train significantly larger networks with DP than previous work*

# Putting it all together: ImageNet w/o extra data

Top-1 and top-5 accuracy when training on ImageNet using DP-SGD without additional data.

| Method | Model | $(\varepsilon, \delta)$ | Accuracy (%) Top-1 | Accuracy (%) Top-5 |
|---|---|---|---|---|
| Kurakin et al. (2022) | ResNet-18 | $(13.2, 10^{-6})$ | 6.9 | – |
| Ours | NF-ResNet-50 | $(8.0, 8 \cdot 10^{-7})$ | **32.4** | **55.8** |

*Significant benefits on ImageNet as well with a 50-layer Normalizer Free (NF) ResNet + tricks*

**But accuracy is low → compute is a limiting factor on large datasets**

# Talk outline

- What is Differential Privacy (DP)?
- Differentially Private Stochastic Gradient Descent (DP-SGD)
- Improving convergence and trainability of deep networks
- Leveraging pre-training
- Interplay between noise, batch size and compute budget

# Pre-training can have remarkable benefits!

**Fine-tuning on CIFAR-10:**

- We use checkpoints of Wide-ResNets pre-trained non-privately on ImageNet-32
- Fine-tune on CIFAR using DP-SGD

# Pre-training can have remarkable benefits!

| Fine-tuning Method | $\varepsilon$ | Test Accuracy (%) | |
|---|---|---|---|
| | | Median | Std. Dev. |
| Yu et al. (2021b) | 1 | 94.3 | – |
| | 2 | 94.8 | – |
| Tramèr and Boneh (2021) | 2 | 92.7 | – |
| Classifier layer | 1 | 93.1 | (0.03) |
| | 2 | 93.6 | (0.05) |
| | 4 | 94.0 | (0.08) |
| | 8 | 94.2 | (0.07) |
| All layers | 1 | **94.8** | (0.08) |
| | 2 | **95.4** | (0.15) |
| | 4 | **96.1** | (0.06) |
| | 8 | **96.6** | (0.08) |

**Fine-tuning on CIFAR-10:**

- We use checkpoints of Wide-ResNets pre-trained non-privately on ImageNet-32
- Fine-tune on CIFAR using DP-SGD
- **Fine-tuning all layers is better**

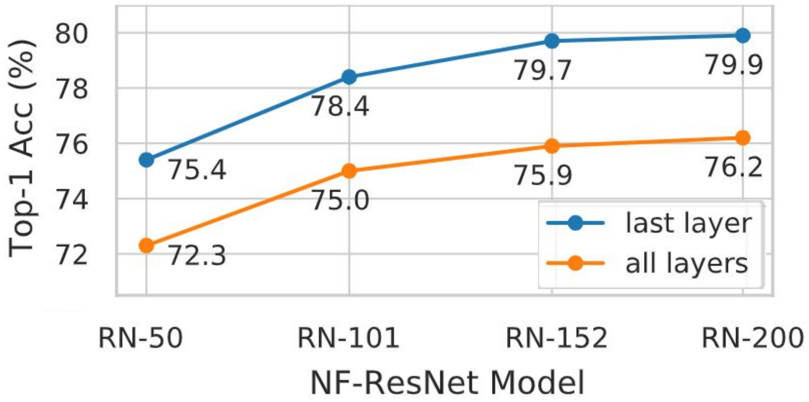
Bu et al. "**Scalable and Efficient Training of Large Convolutional Neural Networks with Differential Privacy**". arXiv: 2205.10683

# Differentially Private Fine-tuning on ImageNet

- NF-ResNets pre-trained non-privately on JFT-300M
- Fine-tune on ImageNet using DP-SGD under small compute budget

Brock et al. “**High-performance large-scale image recognition without normalization**.” ICML (2021).

# Differentially Private Fine-tuning on ImageNet




ImageNet classification using extra data

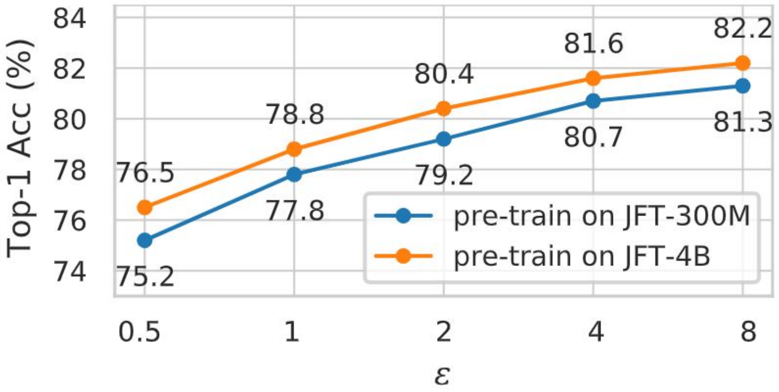
- NF-ResNets pre-trained non-privately on JFT-300M
- Fine-tune on ImageNet using DP-SGD under small compute budget

Results:

- **Better (larger) pre-trained model leads to better downstream results**
- **Fine-tuning only last layer better (*small distribution shift?*)**

Brock et al. "**High-performance large-scale image recognition without normalization**." ICML (2021).

# Better pre-training dataset leads to better downstream results




**ImageNet classification using extra data**

Fine-tuning last layer only.

**Scaling up:**

- model size: 200-layer NF-ResNet
- batch size: $2^{18}$
- training epochs: ~800 epochs for $\varepsilon = 8$

**Results:**

- Better pre-training dataset → better fine-tuning results

Brock et al. "**High-performance large-scale image recognition without normalization**." ICML (2021).
Mehta et al. "Large scale transfer learning for differentially private image classification". arXiv: 2205.02973

# Scaling up to NFNet-F3 pre-trained on JFT-4B

Larger model with more capacity than NF-ResNet-200

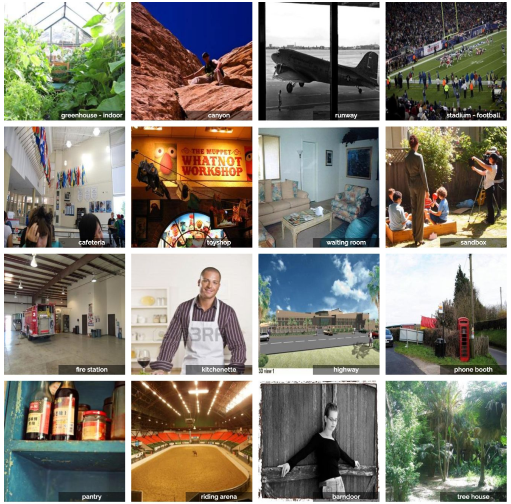
| Accuracy (%) | $\varepsilon$ | | | | | | Non-private |
|---|---|---|---|---|---|---|---|
| | 0.1 | 0.5 | 1.0 | 2.0 | 4.0 | 8.0 | |
| Top-1 | 77.6 | 83.8 | 84.4 | 85.6 | 86.0 | 86.7 | 88.5 |
| Top-5 | 93.0 | 96.7 | 96.6 | 97.5 | 97.4 | 98.0 | 98.7 |

Strong performance, even at low $\varepsilon$

~2% gap between private ($\varepsilon$ = 8) and non-private performance

Brock et al. "**High-performance large-scale image recognition without normalization**." ICML (2021).

# Fine-tuning from JFT-300M to Places365




| $\varepsilon$ | Fine-tuning Method | Accuracy (%) | |
|---|---|---|---|
| | | Top-1 | Top-5 |
| 8 | Classifier layer | 54.4 | 84.4 |
| | All layers | **55.1** | **84.6** |
| – | Classifier layer | 54.3 | 85.2 |
| | All layers | **57.0** | **87.1** |

NF-ResNet-50

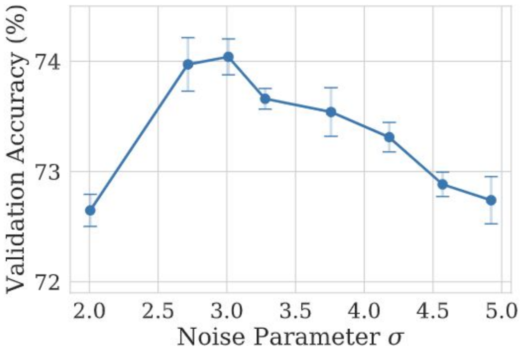
Fine-tuning all layers performs better for this dataset

(*larger distribution shift w.r.t JFT-300M?*)

# Talk outline

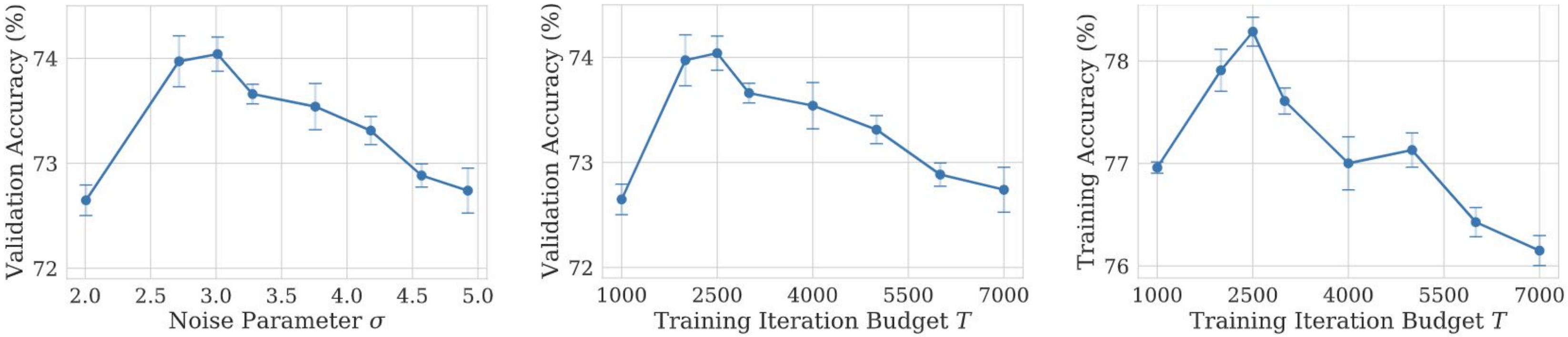
- Background: DP & DP-SGD
- Improving convergence and trainability of deep networks
- Leveraging pre-training
- Interplay between noise, batch size and compute budget

# DP-SGD requires careful hyper-parameter tuning




At fixed batch size: there is an optimal noise scale

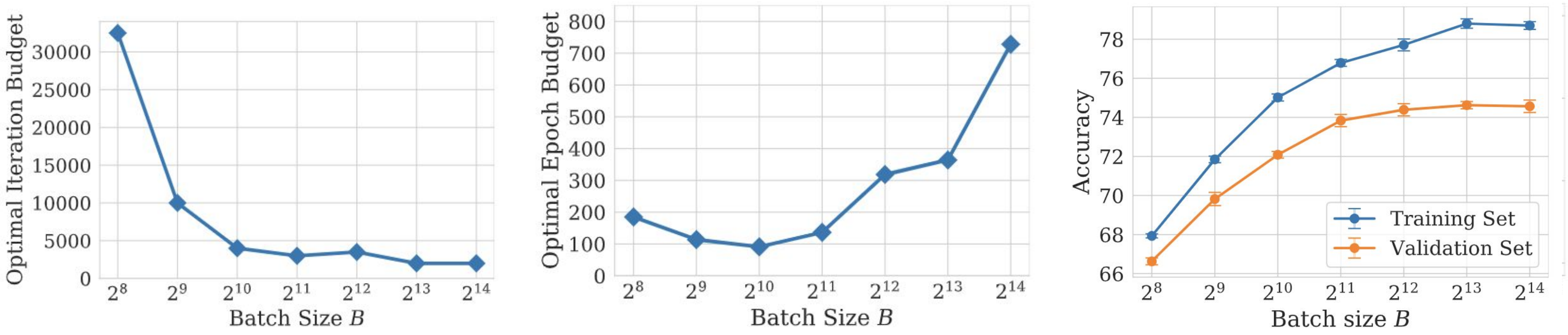
# DP-SGD requires careful hyper-parameter tuning




**At fixed batch size: there is an optimal noise scale → optimal compute budget**

(contrary to non-private training on training set!)

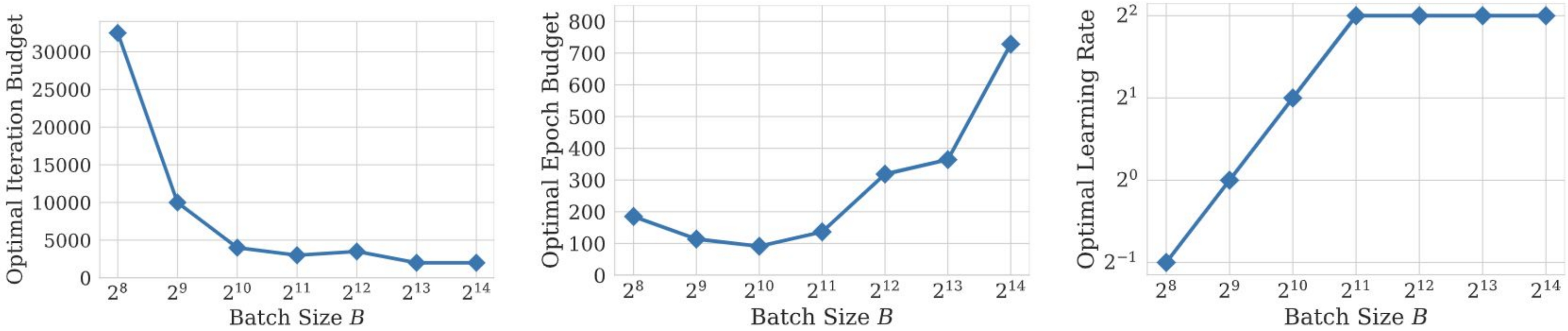
# This optimal compute budget increases with batch size




**Leveraging larger batch sizes requires using more epochs after a threshold**

→ DP-training requires more compute than non-private training for optimal performance

# This optimal compute budget increases with batch size




**Leveraging larger batch sizes requires using more epochs after a threshold**

→ DP-training requires more compute than non-private training for optimal performance

Batch size threshold determined by when the optimal learning rate becomes constant

# Summary

→ Standard vision models can work surprisingly well with DP-SGD when combined with:

- tricks to improve convergence & trainability
- careful hyper-parameter tuning
- enough compute

→ Pre-training + standard models → practical levels of performance with DP-SGD

# Are these results enough for practical use of DP-SGD?

Several additional important considerations may be involved:

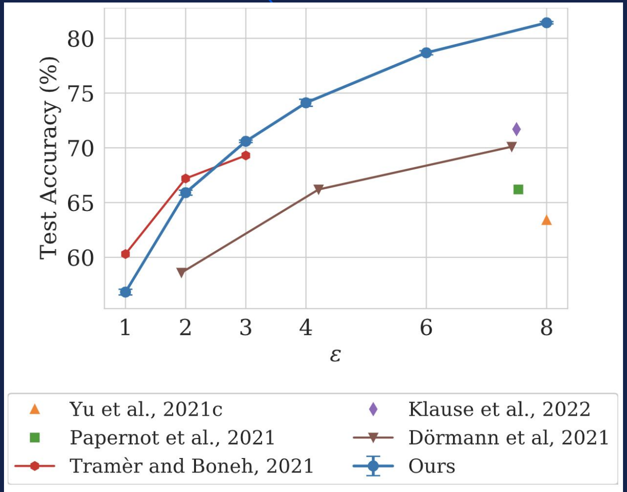
- Record-level vs user-level privacy
- Choice of the privacy budget
- Careful evaluation to avoid disparate impact on under-represented groups
- Sensitivity of the pre-training dataset

DeepMind

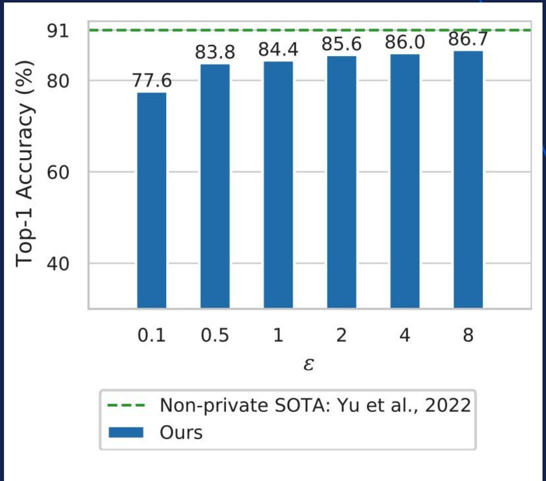
# Thank you! Questions?

Paper: arxiv.org/abs/2204.13650

Code: github.com/deepmind/jax_privacy




(a) CIFAR-10 without extra data




(b) ImageNet with extra data