Multi-Epoch Matrix Factorization Mechanisms for Private Machine Learning

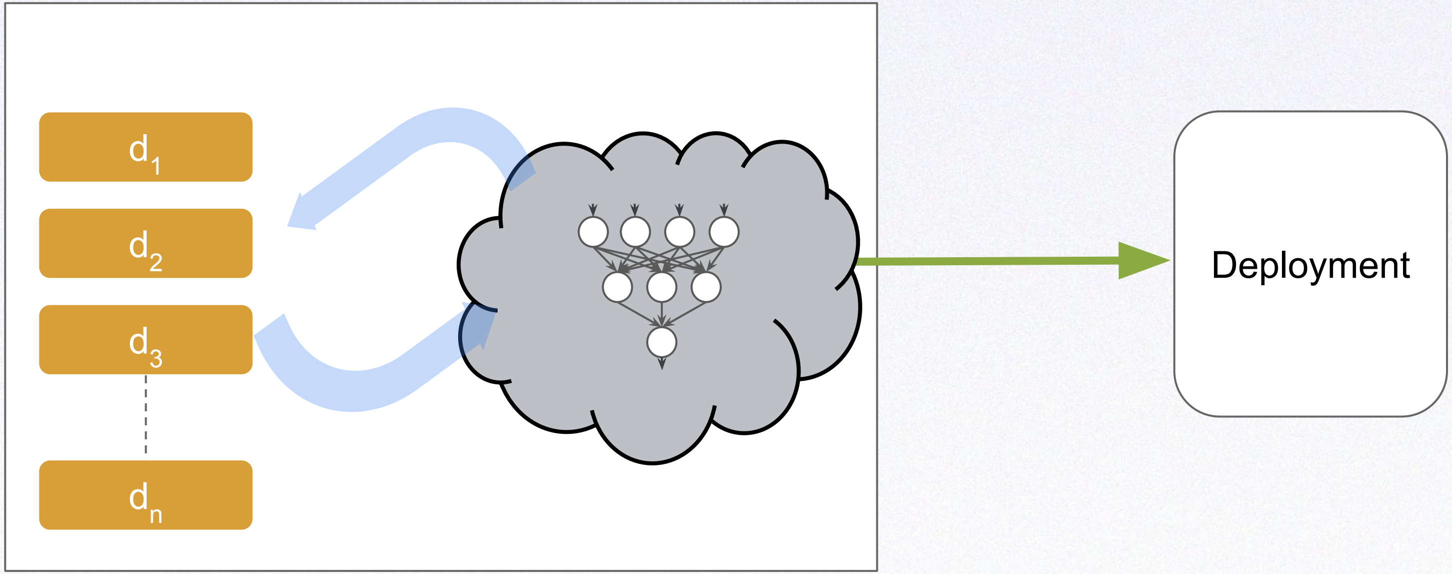
Christopher A. Choquette-Choo

H. Brendan McMahan, Keith Rush, Abhradeep Thakurta

Google DeepMind   Google Research

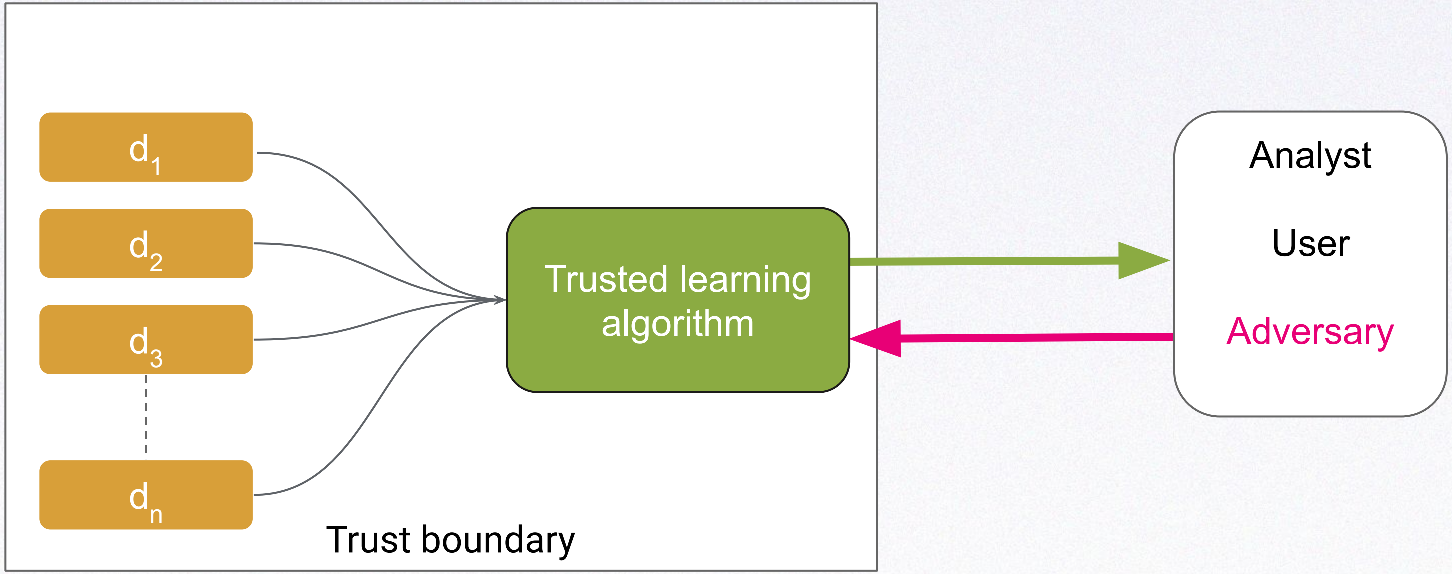
www.christopherchoquette.com
@chris_choquette

Standard Central Machine Learning Setup



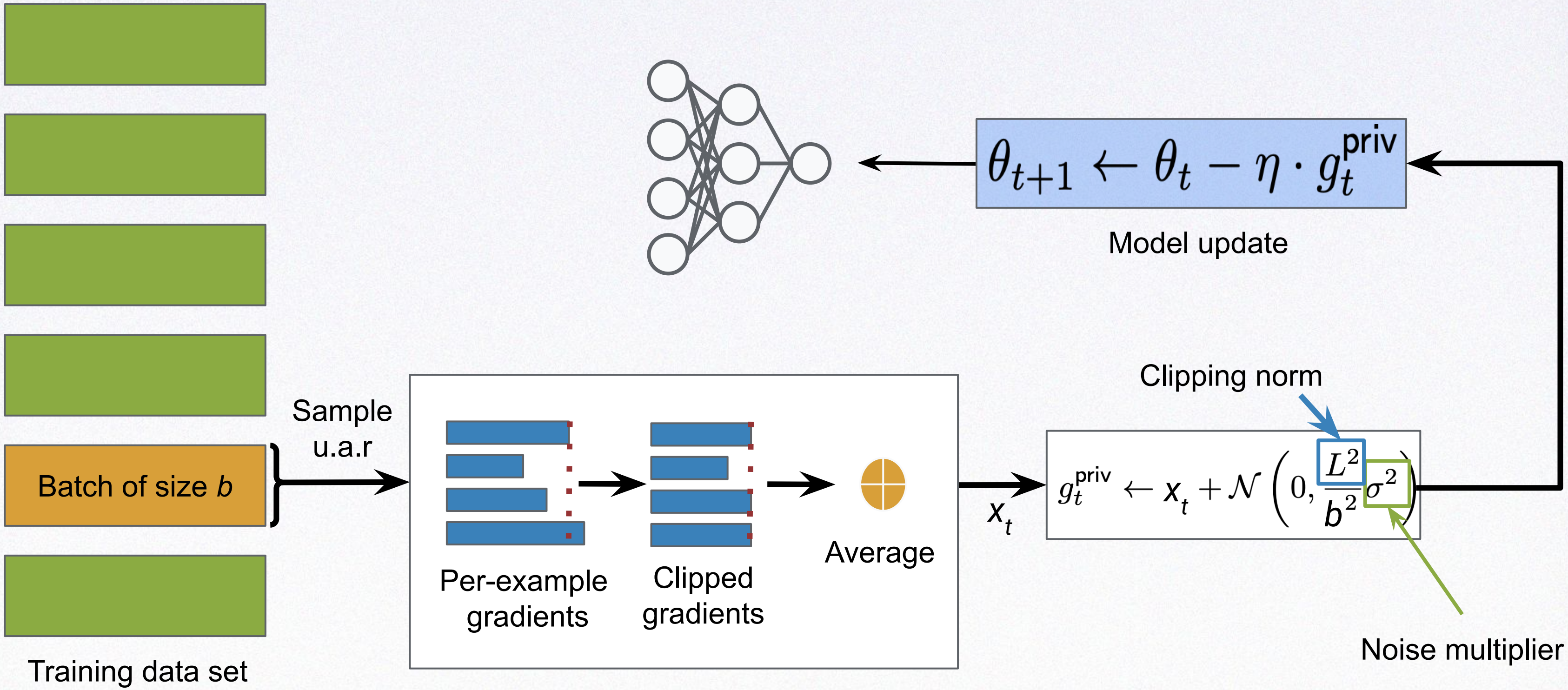
Learn a high utility model via stochastic gradient descent (SGD).

Our Goal: To do this with Differential Privacy (DP)

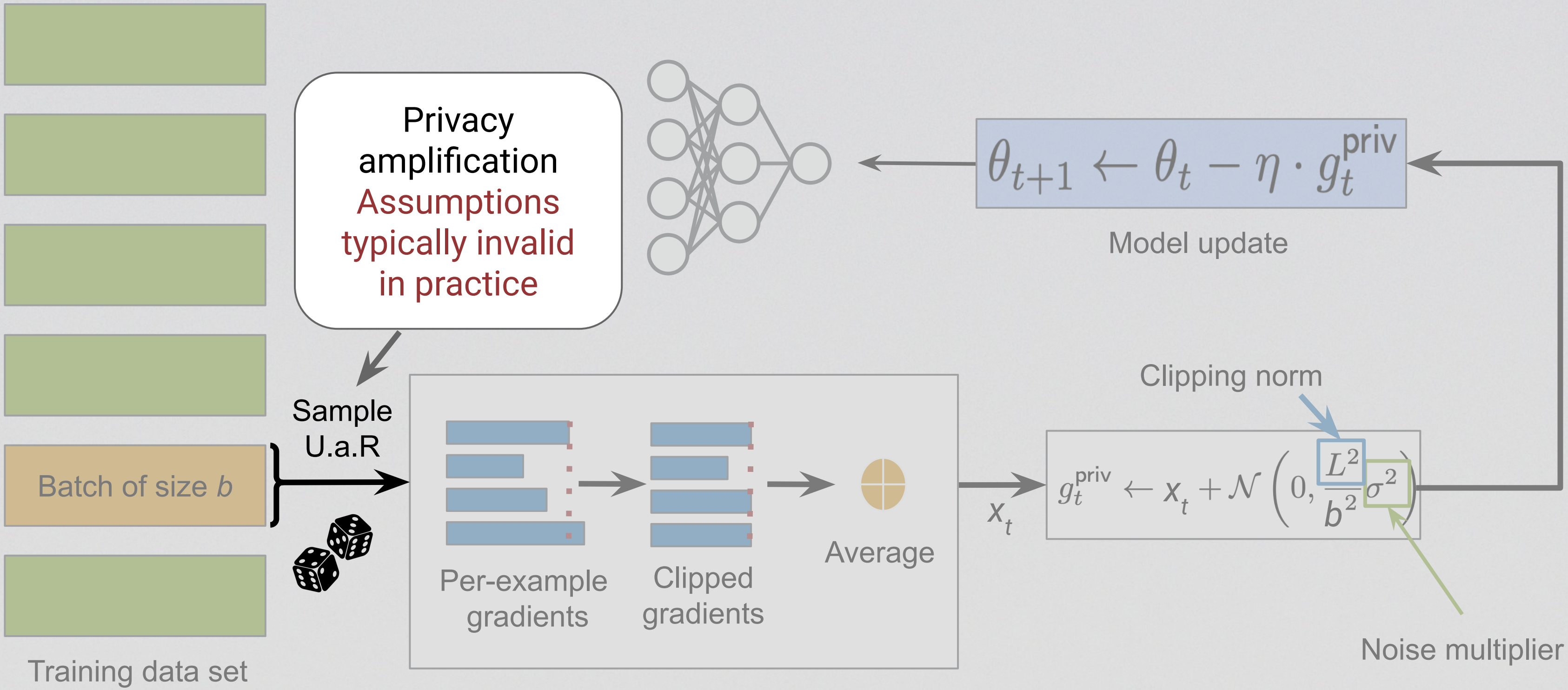


How can we maximize utility and privacy while minimizing computation

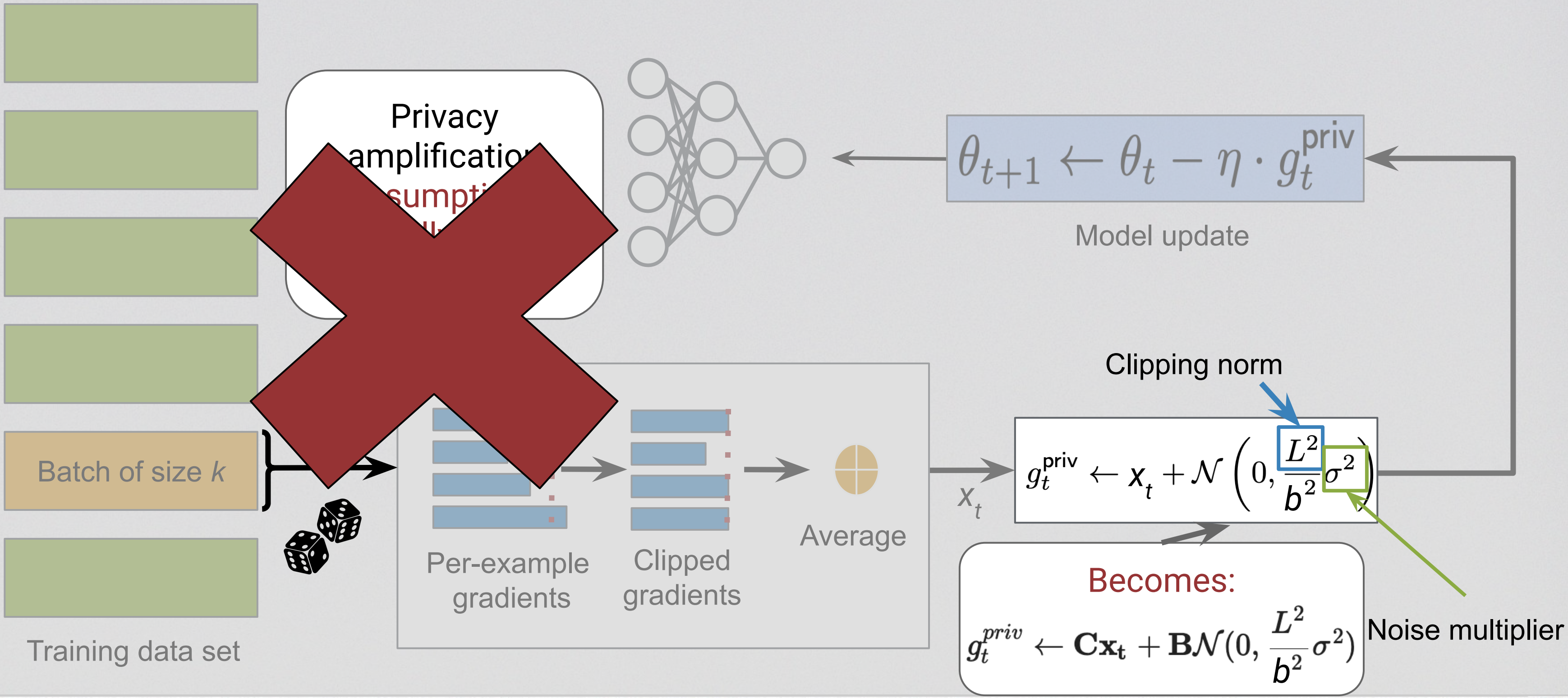
DP-SGD: Most Common Trusted Learning Algorithm



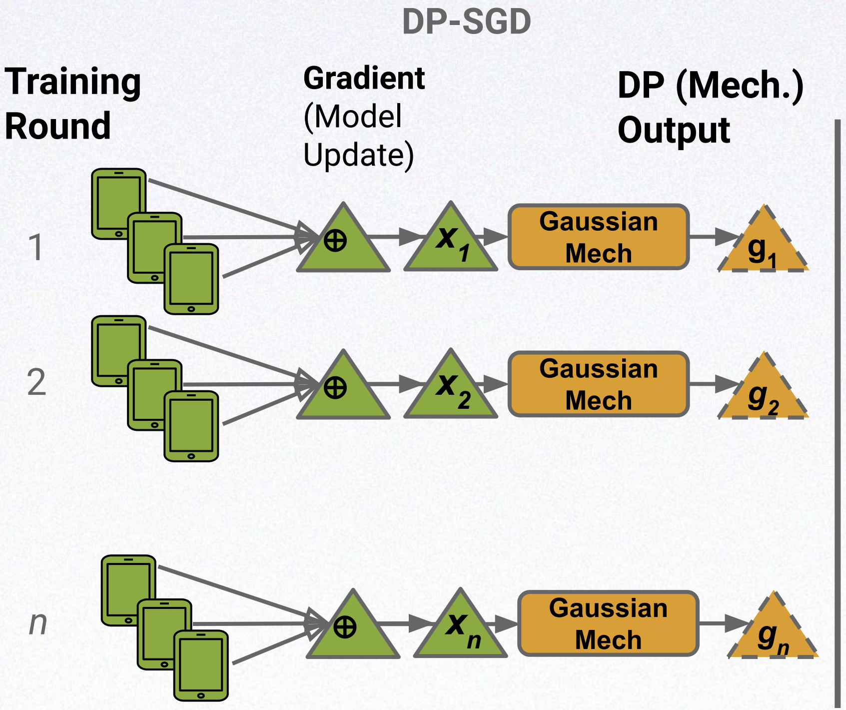
DP-SGD: Most Common Trusted Learning Algorithm



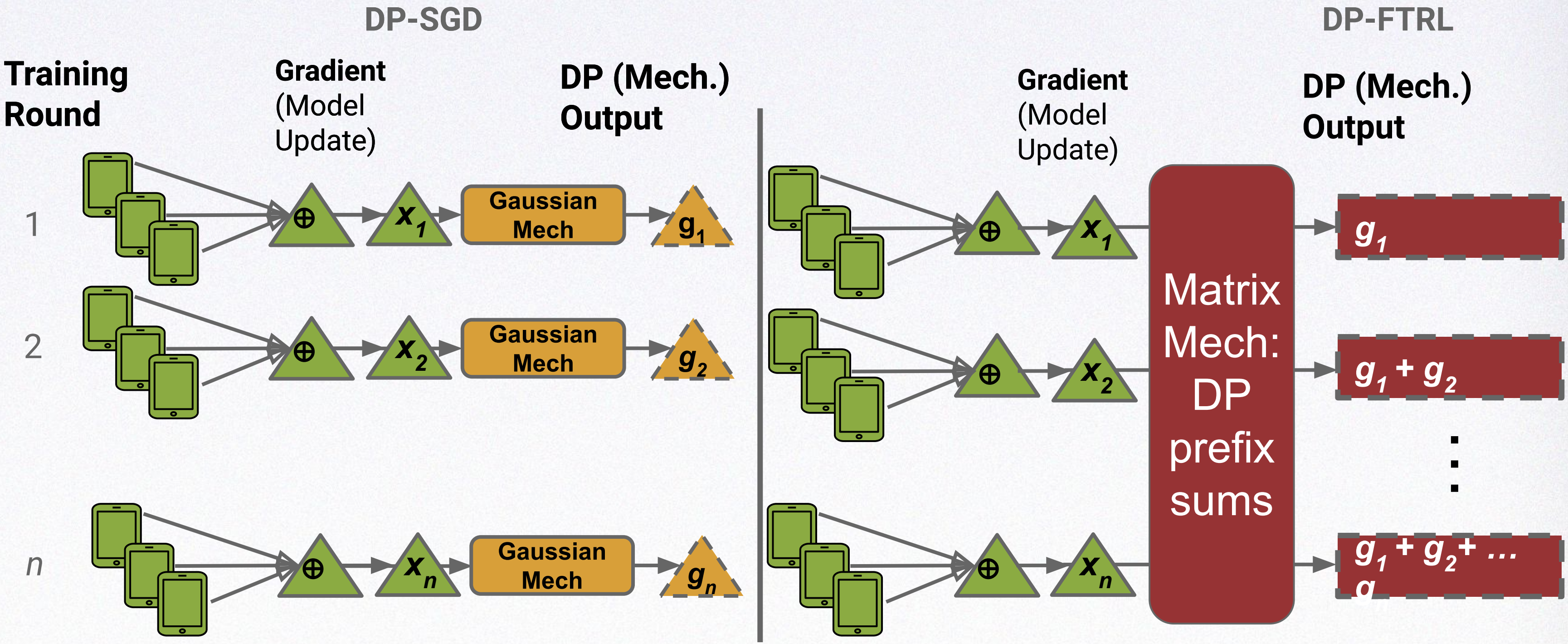
DP-SGD: Most Common Trusted Learning Algorithm



DP-SGD to DP-Follow the Regularized Leader (FTRL)



DP-SGD to DP-Follow the Regularized Leader (FTRL)

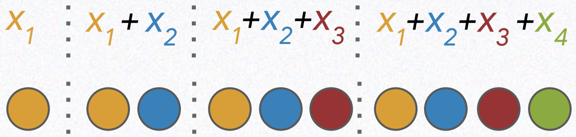


DP-FTRL Formulation and Matrix Factorization

Stream of adaptively chosen data vectors (a.k.a. gradients)

$x_1 \quad x_2 \quad x_3 \quad \cdots \quad x_n$

Prefix sum as mat.-vec. product :



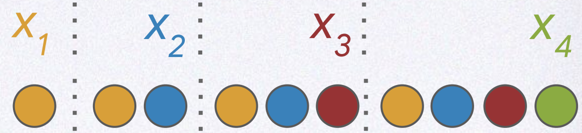
Query matrix: A Data vector: x

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}$$

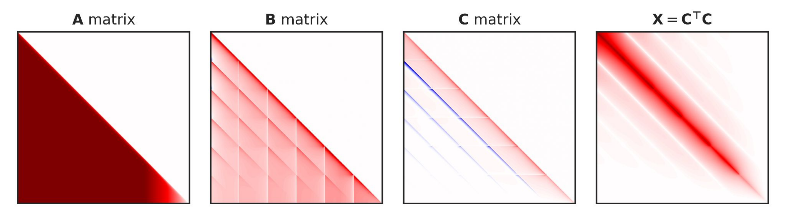
DP-FTRL Formulation and Matrix Factorization

Workload: $\boldsymbol{Ax}$

Gradient prefix sums



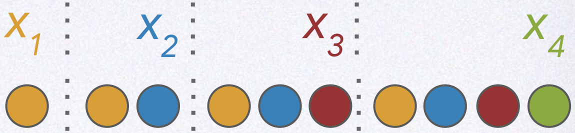
Factorize: $\boldsymbol{Ax=BCx}$

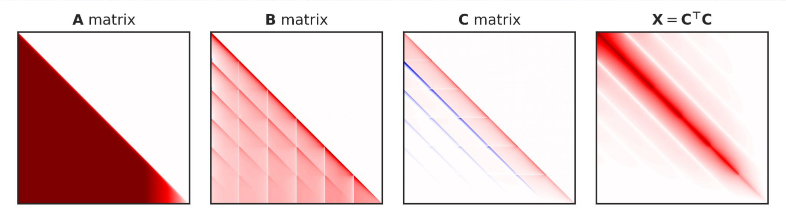


DP-FTRL Formulation and Matrix Factorization

Workload: $\boxed{Ax}$

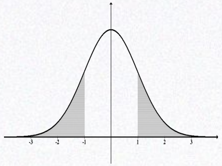
Gradient prefix sums





Factorize: $\boxed{Ax=BCx}$

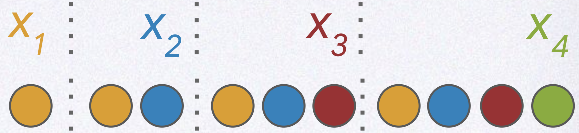
With DP: $\boxed{Ax=B(Cx+\underline{z})}$

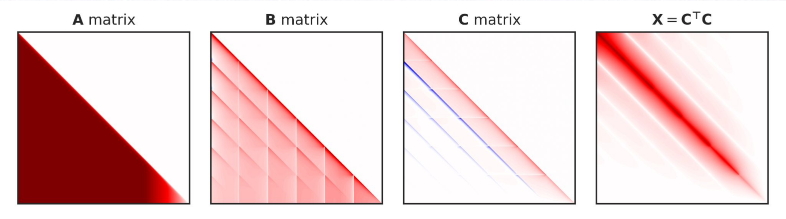


DP-FTRL Formulation and Matrix Factorization

Workload: Ax

Gradient prefix sums





Factorize: $Ax=BCx$

With DP: $Ax=\underline{B}(\underline{C}x+z)$

Decoder

Encoder

What are $\boldsymbol{B}$ and $\boldsymbol{C}$?

Objective: factorize $\boldsymbol{A=BC}$ such that…

$||\boldsymbol{Bz}||_2$ is minimized

$\boldsymbol{Cx+z}$ satisfies DP

Decoder

Encoder

What are ***B*** and ***C***?

Objective: factorize ***A=BC*** such that…

$$\min_{\mathbf{C}} \|\mathbf{A}\mathbf{C}^{\dagger}\|_F^2 \text{ where our loss is } \text{sens}^2_{\mathfrak{D}}(\mathbf{C})\|\mathbf{B}\|_F^2$$
$$\text{sens}(\mathbf{C}) = 1; \qquad \text{as } \forall \alpha,\ \mathcal{L}(\mathbf{B}, \mathbf{C}) = \mathcal{L}(\alpha \mathbf{B}, \mathbf{C}/\alpha)$$

$\|\boldsymbol{Bz}\|_2$ is minimized

Decoder

Cx+z satisfies DP

Encoder

What are *B* and *C*?

Objective: factorize ***A*=*BC*** such that...

$$\min_{\mathbf{C}} \|\mathbf{A}\mathbf{C}^{\dagger}\|_F^2 \text{ where our loss is } \mathrm{sens}_{\mathfrak{D}}^2(\mathbf{C})\|\mathbf{B}\|_F^2$$

$$\mathrm{sens}(\mathbf{C}) = 1; \quad \text{as } \forall \alpha,\ \mathcal{L}(\mathbf{B}, \mathbf{C}) = \mathcal{L}(\alpha\mathbf{B}, \mathbf{C}/\alpha)$$

$\|\boldsymbol{Bz}\|_2$ is minimized

Decoder

Cx+z satisfies DP

Encoder

Bound sens($\boldsymbol{Cx}$)

Desiderata

1. **Tight:** Leave nothing on the table.
2. **Efficient:** Or optimization of the factorization will be slow.
3. **Practical:** Can be implemented easily to not break assumptions/guarantees.

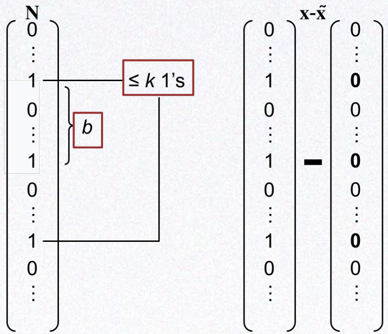
Challenges

1. Sensitivity for scalar contributions does not always extend to vector contributions.
2. In general, sensitivity is NP-hard to compute.

Bound sens($\boldsymbol{Cx}$)

define $\mathfrak{D}^d_\Pi = \{\mathbf{x} - \tilde{\mathbf{x}} | (\mathbf{x}, \tilde{\mathbf{x}}) \in \mathcal{N}\}$, set of all participation patterns.

$$\mathrm{sens}_{\mathfrak{D}}(\mathbf{C}) = \sup_{(\mathbf{x},\tilde{\mathbf{x}})\in \mathbf{N}} \|\mathbf{Cx} - \mathbf{C}\tilde{\mathbf{x}}\|_F = \sup_{\mathbf{u}\in\mathfrak{D}} \|\mathbf{Cu}\|_F$$

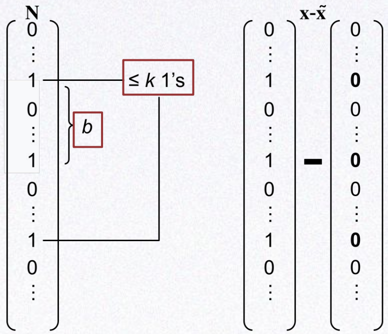


- Formalize participation patterns
  - When users are allowed to participate.
  - Enforced via data pipelines.
- Require a notion of “neighbouring streams”
  - Zero-out contributions of an example/user
  - The delta must still be a valid participation pattern

Bound sens($\boldsymbol{Cx}$)

define $\mathfrak{D}^d_\Pi = \{\mathbf{x} - \tilde{\mathbf{x}} | (\mathbf{x}, \tilde{\mathbf{x}}) \in \mathcal{N}\}$, set of all participation patterns.

$$\text{sens}_{\mathfrak{D}}(\mathbf{C}) = \sup_{(\mathbf{x},\tilde{\mathbf{x}})\in \mathbf{N}} \|\mathbf{Cx} - \mathbf{C}\tilde{\mathbf{x}}\|_F = \sup_{\mathbf{u}\in\mathfrak{D}} \|\mathbf{Cu}\|_F$$



- Formalize participation patterns
  - When users are allowed to participate.
  - Enforced via data pipelines.

- Require a notion of "neighbouring streams"
  - Zero-out contributions of an example/user
  - The delta must still be a valid participation pattern

- Identify and propose (*k*, *b*)-participation
  - Participate at most *k* times, exactly *b* steps apart.

    Practical
    - Shuffle just once prior to training.

    Efficient
    - Requires only $\mathcal{O}(k^2 b)$ time.

    Tight
    - Bounds exactly the *k* contributions per data point.

What are *B* and *C*?

Objective: factorize ***A=BC*** such that…

$$\min_{\mathbf{C}} \|\mathbf{A}\mathbf{C}^{\dagger}\|_F^2 \text{ where our loss is } \mathrm{sens}_{\mathfrak{D}}^2(\mathbf{C})\|\mathbf{B}\|_F^2$$

$$\mathrm{sens}(\mathbf{C}) = 1; \quad \text{as } \forall \alpha,\ \mathcal{L}(\mathbf{B}, \mathbf{C}) = \mathcal{L}(\alpha\mathbf{B}, \mathbf{C}/\alpha)$$

$\|\boldsymbol{Bz}\|_2$ is minimized

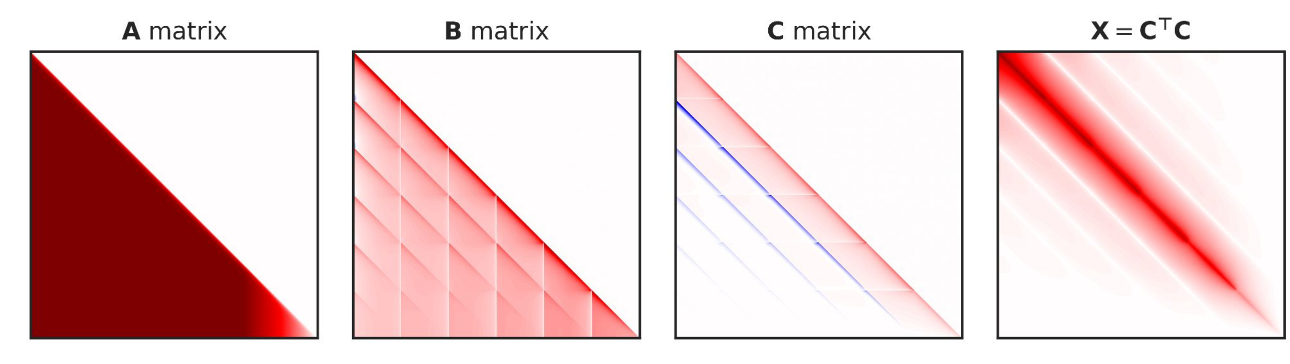
Cx+z satisfies DP

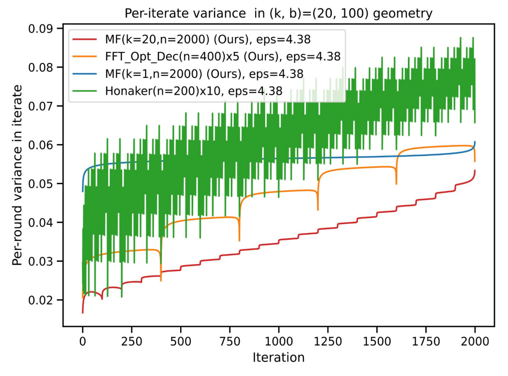
Decoder

Encoder

$||\boldsymbol{Bz}||_2$ is minimized

- We define a mathematical program for minimizing $||\boldsymbol{Bz}||_2$.
- We solve for the lagrangian and dual functions in closed-form.
- We solve for the gradient of the dual.
    - Allows us to directly leverage prior fixed-point iteration optimization algorithms to optimize ***B,C***.
- Noise added is lower than all prior algorithms.

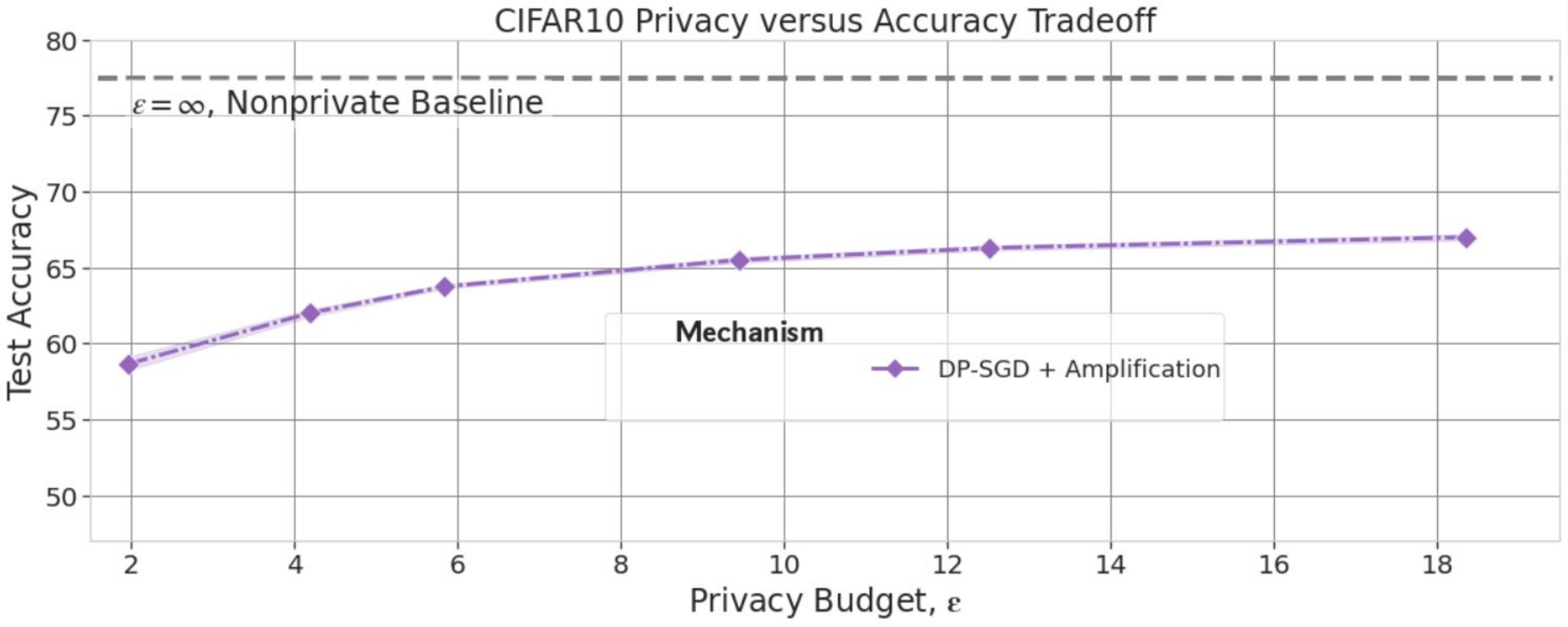




Empirical Details

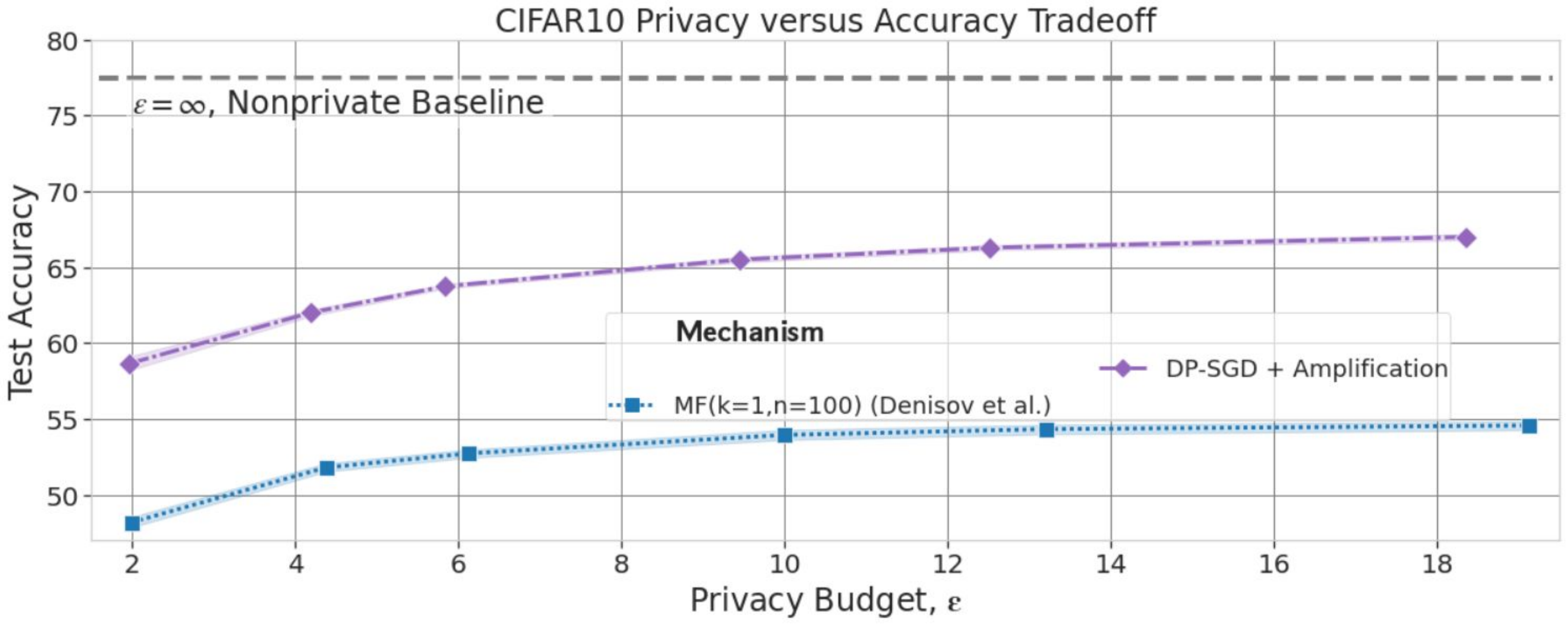
- Focus on CIFAR-10 (see paper for Stack Overflow next work prediction)
- Train for $k=20$ epochs, this gives $b=100$
- Factorize $\boldsymbol{B},\boldsymbol{C}$ matrices for (20,100)-participation
  - We also propose two other methods: FFT and stamping. See the paper for more details!
- Scale as $\boldsymbol{\alpha B}, 1/\boldsymbol{\alpha}\ \boldsymbol{C}$ so that sens($\boldsymbol{C}$) = 1
- Choose noise standard deviation σ so that a single Gaussian event achieves (ε,δ)-DP.
- Generate per-step noise as $(\boldsymbol{AC}^{-1})_{[:,:i]}\mathcal{N}(0,\sigma)$

CIFAR-10



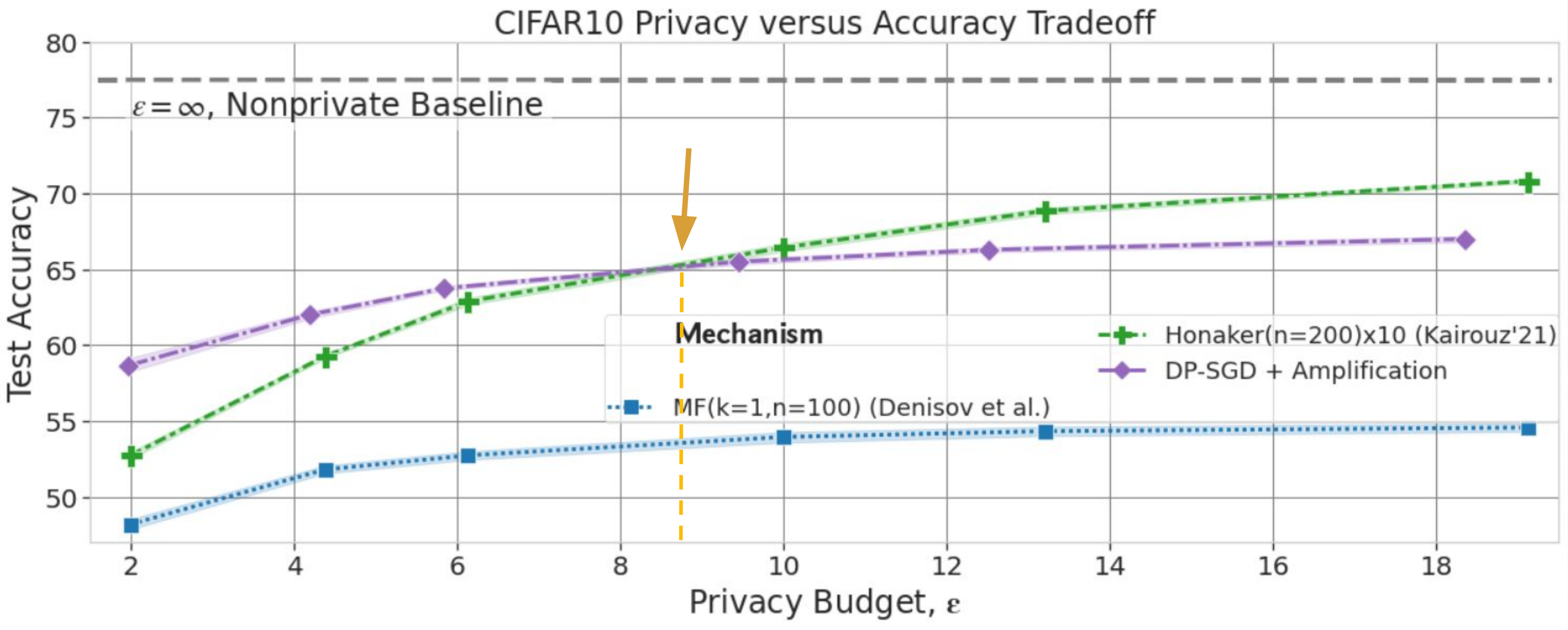
CIFAR10 Privacy versus Accuracy Tradeoff
80
75
70
65
60
55
50
Test Accuracy
$\varepsilon = \infty$, Nonprivate Baseline
Mechanism
DP-SGD + Amplification
2
4
6
8
10
12
14
16
18
Privacy Budget, ε

CIFAR-10



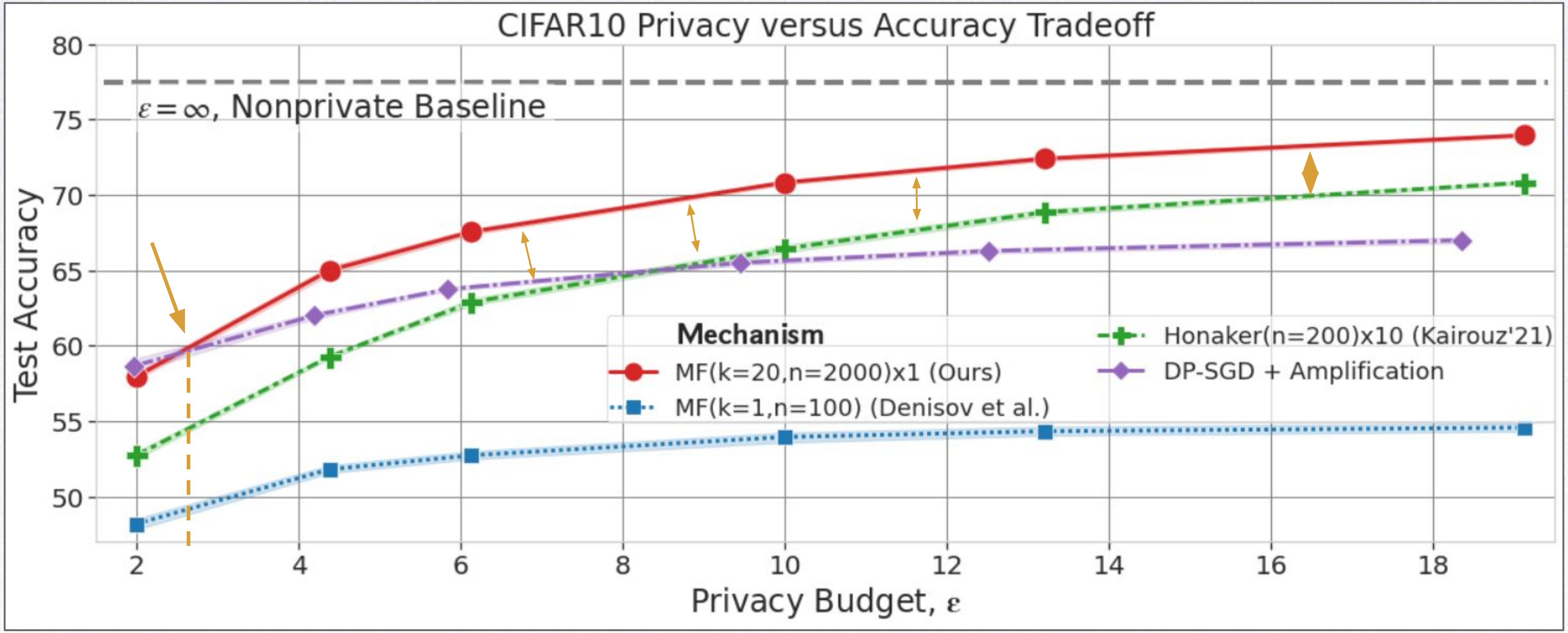
CIFAR10 Privacy versus Accuracy Tradeoff
$\varepsilon = \infty$, Nonprivate Baseline
Test Accuracy
80
75
70
65
60
55
50
2
4
6
8
10
12
14
16
18
Privacy Budget, ε
Mechanism
DP-SGD + Amplification
MF(k=1,n=100) (Denisov et al.)

CIFAR-10



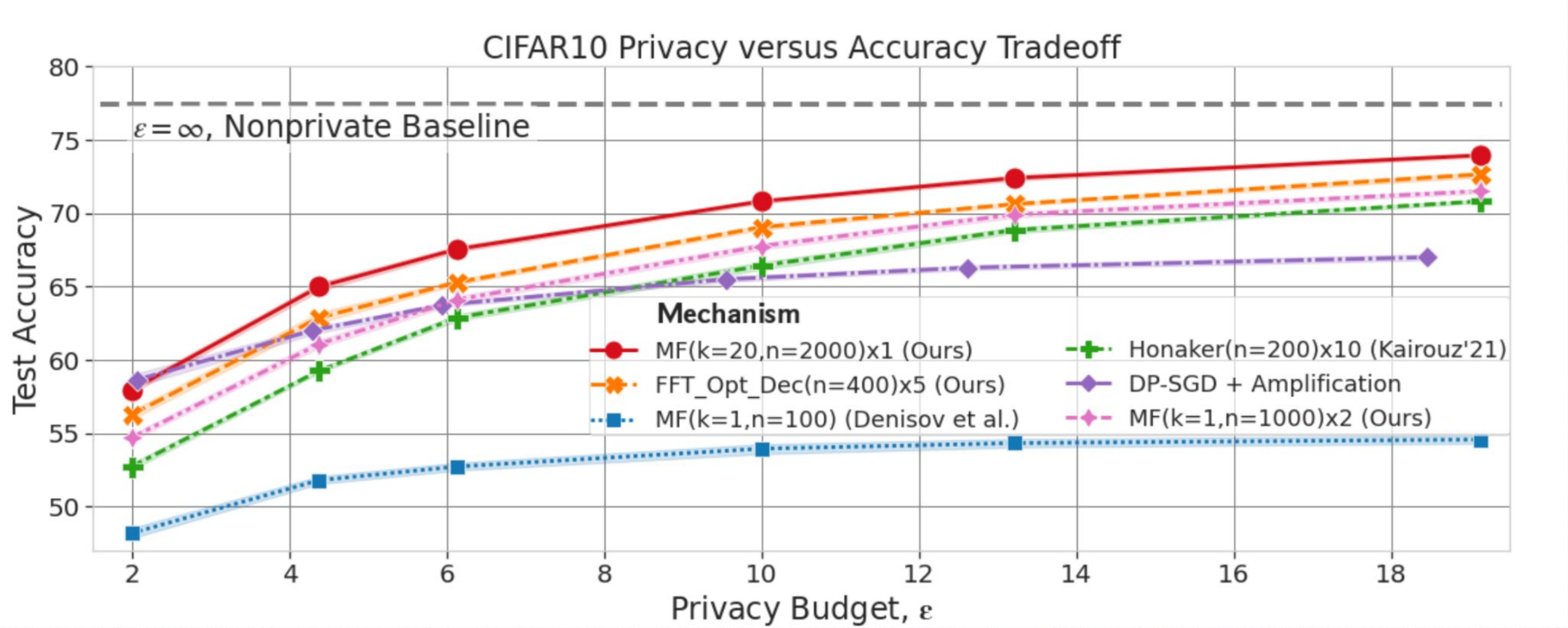
CIFAR10 Privacy versus Accuracy Tradeoff
ε = ∞, Nonprivate Baseline
Test Accuracy
80
75
70
65
60
55
50
2
4
6
8
10
12
14
16
18
Privacy Budget, ε
Mechanism
Honaker(n=200)x10 (Kairouz'21)
DP-SGD + Amplification
MF(k=1,n=100) (Denisov et al.)

CIFAR-10



CIFAR10 Privacy versus Accuracy Tradeoff
Test Accuracy
80
75
70
65
60
55
50
$\varepsilon = \infty$, Nonprivate Baseline
2
4
6
8
10
12
14
16
18
Privacy Budget, ε
Mechanism
MF(k=20,n=2000)x1 (Ours)
MF(k=1,n=100) (Denisov et al.)
Honaker(n=200)x10 (Kairouz'21)
DP-SGD + Amplification

CIFAR-10



CIFAR10 Privacy versus Accuracy Tradeoff
$\varepsilon = \infty$, Nonprivate Baseline
Test Accuracy
80
75
70
65
60
55
50
2
4
6
8
10
12
14
16
18
Privacy Budget, ε
Mechanism
MF(k=20,n=2000)x1 (Ours)
FFT_Opt_Dec(n=400)x5 (Ours)
MF(k=1,n=100) (Denisov et al.)
Honaker(n=200)x10 (Kairouz'21)
DP-SGD + Amplification
MF(k=1,n=1000)x2 (Ours)

Conclusion

- New differentially private mechanisms.
- Achieve state-of-the-art privacy-utility tradeoffs (~5% points above DP-SGD).
- Realistic assumptions that can be implemented in practice.
- Applicable with tricks used for DP-SGD (no changes other than minimizing noise)
- Computationally practical

www.christopherchoquette.com
@chris_choquette