Private Averaging with Untrusted Parties

César Sabater

June 24th, 2022

ANR PMR Workshop

Joint work with Aurélien Bellet and Jan Ramon

Centre de Recherche en Informatique, Signal et Automatique de Lille

Introduction

Privacy

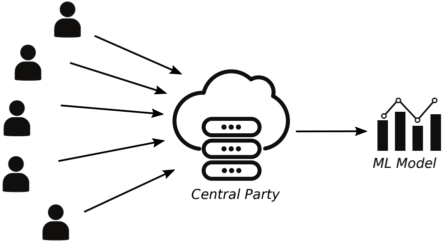
GOssip noise for Private Averaging

Conclusion

Centralized Machine Learning

- Machine Learning (ML) offers solutions in domains such as machine vision, natural language processing, medical research
- It requires large amounts of data
- Data often belongs to individuals or organizations

Centralized Setting



Data contains private information of individuals and is sensitive
Untrusted central parties $\rightarrow$ privacy concerns

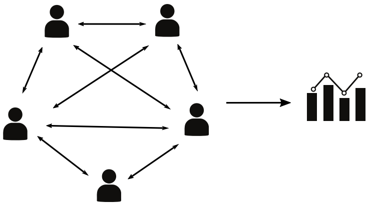
Measures for Privacy

Legislation: GDPR, PIPEDA, ...

- ask for consent to gather data
- define privacy-preserving practices
- withdraw data under request

Legislation is important, but not sufficient by itself
(e.g: it is impossible to prove that data has been forgotten)

Technical Measures: algorithms to prevent data exposure



(Semi-)Decentralized Setting: keep data locally, interact to compute models

Goals

Setting

- untrusted parties
- large number of participants

Important Challenges:

1. **First Challenge:** improve accuracy and scalability of privacy preserving algorithms
2. **Second Challenge:** reduce vulnerability to malicious participants and dropouts

Introduction

Privacy

GOssip noise for Private Averaging

Conclusion

Differential Privacy (DP)

- $X = (X_1, \ldots, X_n)$: dataset of n individuals (X_i belongs to i)
- $\mathcal{A}$: stochastic algorithm
- two datasets X and X' are neighboring if they only differ in the contribution of one individual

Differential Privacy (DP)

- $X = (X_1, \ldots, X_n)$: dataset of n individuals (X_i belongs to i)
- $\mathcal{A}$: stochastic algorithm
- two datasets X and X' are neighboring if they only differ in the contribution of one individual

Definition (Differential Privacy [Dwork, 2006])

For $\varepsilon > 0$ and $\delta \in (0, 1)$, $\mathcal{A}$ satisfies (ε,δ)-Differential Privacy if for all neighboring datasets X and X' and all subsets of outcomes $\mathcal{O}$ we have

$$Pr(\mathcal{A}(X) \in \mathcal{O}) \leq e^{\varepsilon} Pr(\mathcal{A}(X') \in \mathcal{O}) + \delta$$

Differential Privacy (DP)

- $X = (X_1, \ldots, X_n)$: dataset of n individuals (X_i belongs to i)
- $\mathcal{A}$: stochastic algorithm
- two datasets X and X' are neighboring if they only differ in the contribution of one individual

Definition (Differential Privacy [Dwork, 2006])

For $\varepsilon > 0$ and $\delta \in (0, 1)$, $\mathcal{A}$ satisfies (ε,δ)-Differential Privacy if for all neighboring datasets X and X' and all subsets of outcomes $\mathcal{O}$ we have

$$Pr(\mathcal{A}(X) \in \mathcal{O}) \leq e^{\varepsilon} Pr(\mathcal{A}(X') \in \mathcal{O}) + \delta$$

- smaller ε implies more privacy
- δ is a (small enough) value for unlikely events
- precisely quantifies the information leakage

Privacy Mechanisms

- let $\mathcal{A}$ be an algorithm with input X
- (ε, δ)-DP can be achieved adding noise to the outcome of $\mathcal{A}$

Privacy Mechanisms

- let $\mathcal{A}$ be an algorithm with input X
- (ε, δ)-DP can be achieved adding noise to the outcome of $\mathcal{A}$

Mechanisms

- generate the required noise η according to some distribution (Gaussian, Laplacian, Exponential, ..)
- reveal $\mathcal{A}(X) + \eta$
- calibrate variance of η depending on ε, δ, and how sensitive is $\mathcal{A}$ to individual contributions

Privacy Mechanisms

- let $\mathcal{A}$ be an algorithm with input X
- (ε, δ)-DP can be achieved adding noise to the outcome of $\mathcal{A}$

Mechanisms

- generate the required noise η according to some distribution (Gaussian, Laplacian, Exponential, ..)
- reveal $\mathcal{A}(X) + \eta$
- calibrate variance of η depending on ε, δ, and how sensitive is $\mathcal{A}$ to individual contributions

Two popular settings

- **Central DP**
- **Local DP**

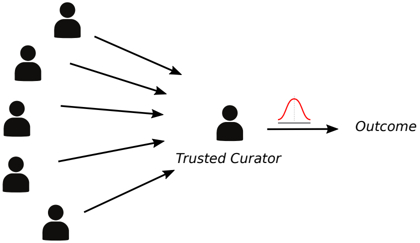
Central DP (CDP)

Classical Centralized Setting: assumes a Trusted Curator

Central DP (CDP)

Classical Centralized Setting: assumes a Trusted Curator

Trusted Curator Setting



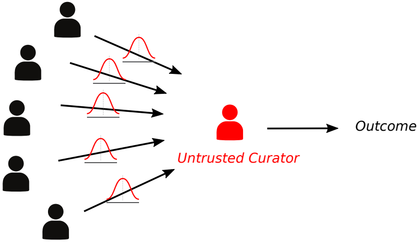
Local DP (LDP)

Decentralized Setting: no party is trusted

Local DP (LDP)

Decentralized Setting: no party is trusted

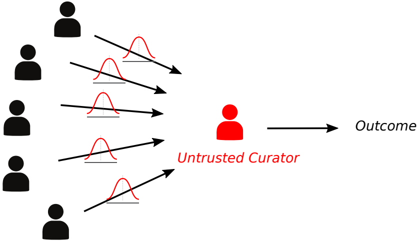
Local DP[Duchi et al., 2013]: inputs are considered public



Local DP (LDP)

Decentralized Setting: no party is trusted

Local DP[Duchi et al., 2013]: inputs are considered public

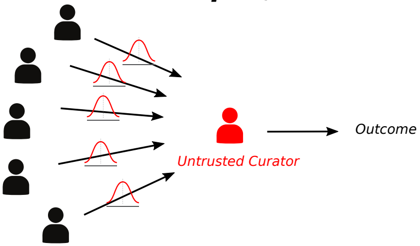


- requires substantially more noise than CDP for the same privacy
- For $\mathcal{A}(X) = \frac{1}{n}\sum_{i=1}^{n} X_i$, the noise variance in LDP *n* times bigger than in CDP

Local DP (LDP)

Decentralized Setting: no party is trusted

Local DP[Duchi et al., 2013]: inputs are considered public



if a trusted curator is available accuracy is substantially better

Introduction

Privacy

GOssip noise for Private Averaging

Conclusion

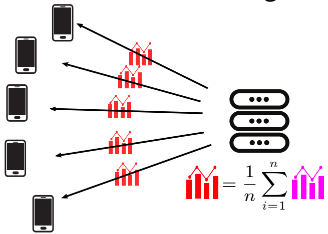
Private Averaging

- Set U of n users
- Each user $u \in U$ has a private value $X_u \in [0, 1]$
- **Goal:** compute the average $\frac{1}{n} \sum_{u \in U} X_u$ while satisfying differential privacy (DP)

Private Averaging

- Set U of n users
- Each user $u \in U$ has a private value $X_u \in [0, 1]$
- **Goal:** compute the average $\frac{1}{n}\sum_{u \in U} X_u$ while satisfying differential privacy (DP)

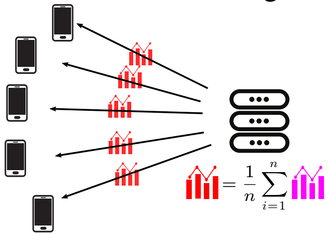
Federated Learning



Private Averaging

- Set U of n users
- Each user $u \in U$ has a private value $X_u \in [0, 1]$
- **Goal:** compute the average $\frac{1}{n}\sum_{u \in U} X_u$ while satisfying differential privacy (DP)

Federated Learning



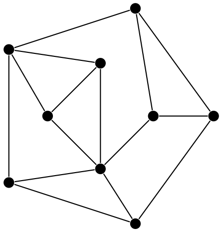
- can be used to compute other models and statistics: decision trees, linear regression, Hosmer-Lemeshow tests ..

Key Features

1. **Accuracy in the order Trusted Curator DP**
   - unlike local Differential Privacy
2. **Logarithmic communication and computation cost** per party
   - unlike secure Aggregation [Bonawitz et al., 2017], except for recent (concurrent) work [Bell et al., 2020]
3. **Guaranteed Correctness** in the presence of malicious users that might want to bias the computation.

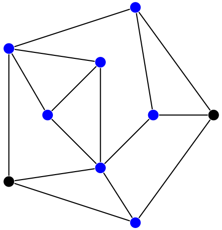
Setting

- Users communicate using secure channels through graph G



Setting

- Users communicate using secure channels through graph G

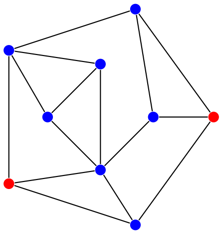


A proportion ρ of honest (but curious) users:

- follow the protocol
- might try to infer information
- do not collude with other users

Setting

- Users communicate using secure channels through graph G

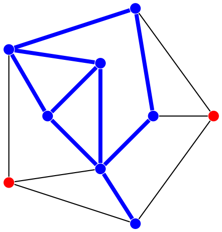


Adversary: a proportion of $(1 - \rho)$ malicious users:

- deviate from the protocol
- try to (1) infer information and (2) bias the computation
- collude in organized attacks

Setting

- Users communicate using secure channels through graph G



The sub-graph of honest users is G^H

- channels whose information the is not seen by the adversary
- not known by honest parties

Protocol

Algorithm 1 GOPA protocol

Input: graph G, variances $\sigma_\Delta^2, \sigma_\eta^2 \in \mathbb{R}^+$

for all neighbor pairs $\{u, v\} \in E(G)$ **do**

  1a. u and v draw random pairwise noise $x \sim \mathcal{N}(0, \sigma_\Delta^2)$

  1b. set $\Delta_{u,v} \leftarrow x$, $\Delta_{v,u} \leftarrow -x$

end for

for each user $u \in U$ **do**

  2. u draws a random independent noise $\eta_u \sim \mathcal{N}(0, \sigma_\eta^2)$

  3. u reveals $\hat{X}_u \leftarrow X_u + \sum_{u \sim v} \Delta_{u,v} + \eta_u$

end for

Unbiased estimate of the average: $\hat{X}^{avg} = \frac{1}{n} \sum_u \hat{X}_u$
with variance σ_η^2 / n.

Privacy Guarantees - General Result

The adversary sees:

1. who communicates with who (structure of G)
2. pairwise noise involving a malicious peer
   ($\Delta_{u,v}$: u or v is malicious)
3. independent noise of malicious peers (η_u: u malicious)

Privacy Guarantees - General Result

The adversary sees:

1. who communicates with who (structure of G)
2. pairwise noise involving a malicious peer ($\Delta_{u,v}$: u or v is malicious)
3. independent noise of malicious peers (η_u: u malicious)

General Result

GOPA can achieve (ε, δ)-DP with trusted curator accuracy when

- the subgraph G^H of honest users is connected
- pairwise variance σ_Δ^2 is large enough

The required σ_Δ^2 depends on the connectivity of G^H

Privacy Guarantees - General Results

- We proved utility of the central setting as long as G^H is connected
- How to ensure that G^H is good enough?

Privacy Guarantees - Random Graphs

- *k-out random graph*: each user chooses k neighbors at random
- if $k = O_\rho(\log(n))$ then G^H is sufficiently connected with high probability

Theorem (k-out Random Graphs)

Let $\varepsilon, \delta \in (0,1)$ and

- *each user chooses $k = O(\log(\rho n)/\rho)$ neighbors*
- *$\sigma_\eta^2 = O(\log(1/\delta)/\rho n \varepsilon^2) \rightarrow$ in the order of trusted curator noise*
- *$\sigma_\Delta^2 = O(\sigma_\eta^2 \rho n/k)$*

Then GOPA *is (ε, δ')-differentially private with $\delta' = O(\delta)$.*

Privacy Guarantees - Random Graphs

- *k*-out random graph: each user chooses k neighbors at random
- if $k = O_\rho(\log(n))$ then G^H is sufficiently connected with high probability

Theorem (k-out Random Graphs)

Let $\varepsilon, \delta \in (0, 1)$ and

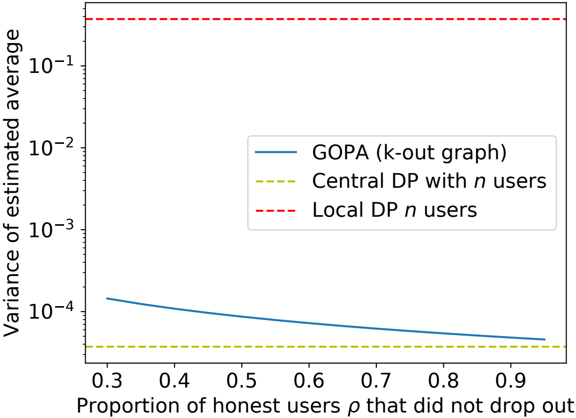
- *each user chooses $k = O(\log(\rho n)/\rho)$ neighbors*
- $\sigma_\eta^2 = O(\log(1/\delta)/\rho n\varepsilon^2)$ $\rightarrow$ *in the order of trusted curator noise*
- $\sigma_\Delta^2 = O(\sigma_\eta^2 \rho n/k)$

Then GOPA *is (ε, δ')-differentially private with $\delta' = O(\delta)$.*

- Trusted curator accuracy with logarithmic number of messages per user
- we show that k and σ_Δ can be even smaller in practice (using simulations)

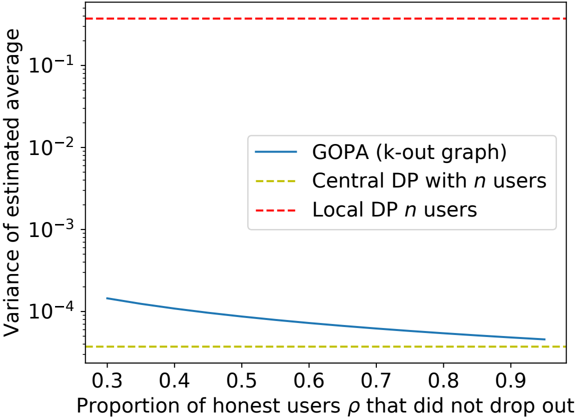
An Illustration

$n = 10000$, (ε, δ)-DP for $\varepsilon = 0.1$, $\delta = 10/(\rho n)^2$



An Illustration

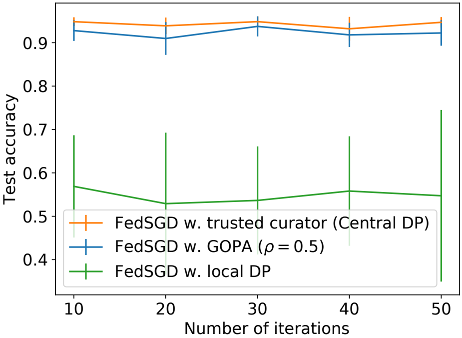
$n = 10000$, (ε, δ)-DP for $\varepsilon = 0.1$, $\delta = 10/(\rho n)^2$



- utility close to CDP even if ρ is small
- substantially more efficient than LDP

An Experiment

- (ε, δ)-DP Federated Learning for Logistic Regression
- Each user has 1 or 2 data points (each step samples one point)



$n = 10000$, $\rho = 0.5$ (prop. honest users), $\varepsilon = 1$, $\delta = 10/(\rho n)^2$

- CDP and GOPA have similar performance
- LDP does not arrive to learn anything

Dropouts

- pairwise noise can be rolled back
- have the same privacy impact than a malicious user (degrades G^H)
- ρ: proportion of honest users that did not dropout

Dropouts

- pairwise noise can be rolled back
- have the same privacy impact than a malicious user (degrades G^H)
- ρ: proportion of honest users that did not dropout

If dropouts are more than the expected by ρ:

- Gaussian uncancelled noise has a bounded impact
- GOPA can tolerate a few extra dropouts

We have shown

1. how to obtain trusted curator utility
2. how to have tractable communication
3. how to deal with dropouts

Now we show:

- robustness against malicious participants

Ensuring Correctness

Goal: prevent that a malicious user u poisons $\hat{X}_u$ (as much as possible)

Ensuring Correctness

Goal: prevent that a malicious user u poisons $\hat{X}_u$
(as much as possible)

Ensuring Correctness

Goal: prevent that a malicious user u poisons $\hat{X}_u$ (as much as possible)

Ensure that:

$$\begin{aligned} X_u &\in [0,1], && \forall u \in U \\ \Delta_{u,v} &= -\Delta_{v,u}, && \forall \{u,v\} \quad \text{neighbors in } G \\ \eta_u &\sim \mathcal{N}(0, \sigma_\eta^2), && \forall u \in U \\ \hat{X}_u = X_u &+ \sum_{u \sim v} \Delta_{u,v} + \eta_u. && \forall u \in U \end{aligned}$$

- u can lie about X_u, but this is also true in the central setting

Ensuring Correctness of Computations

Parties share a bulletin board (e.g. block chain)

- users can post public messages
- other parties can query messages

Ensuring Correctness of Computations

Parties share a bulletin board (e.g. block chain)

- users can post public messages
- other parties can query messages

Each user $u \in U$:

- publishes an encrypted log of the computation
- prove correctness of the computations
  using Commitments and Zero Knowledge Proofs

Ensuring Correctness of Computations

Parties share a bulletin board (e.g. block chain)

- users can post public messages
- other parties can query messages

Each user $u \in U$:

- publishes an encrypted log of the computation
- prove correctness of the computations
  using Commitments and Zero Knowledge Proofs

Assume deterrence: malicious users avoid getting detected by cheating

Cryptographic Tools

Commitments

Allow to commit to a value while keeping it hidden

Zero Knowledge Proofs (ZKP)

Allow prove properties about committed values without revealing anything else

Cryptographic Tools

Commitments

Allow to commit to a value while keeping it hidden

It is a function $C : M \to \mathbb{C}$:

- $C(x)$ does not reveal anything about x (hiding)
- infeasible to find x and x' such that $C(x) = C(x')$ (binding)

Zero Knowledge Proofs (ZKP)

Allow prove properties about committed values without revealing anything else

Cryptographic Tools

Commitments

Allow to commit to a value while keeping it hidden

It is a function $C : M \to \mathbb{C}$:

- $C(x)$ does not reveal anything about x (hiding)
- infeasible to find x and x' such that $C(x) = C(x')$ (binding)

Zero Knowledge Proofs (ZKP)

Allow prove properties about committed values without revealing anything else

- parties can prove arithmetic relations ($+$ and $\times$) over commitments in $\mathbb{Z}$ or $\mathbb{Z}_p$
- parties can prove boolean formulas ($\wedge$ and $\vee$) over provable statements
- there is negligible probability of success in proving false relations

GOPA: Verification Protocol

Each user $u \in U$

- commits to X_u, η_u, $\Delta_{u,v}$'s and $\hat{X}_u$ (as soon as generated)

GOPA: Verification Protocol

Each user $u \in U$

- commits to X_u, η_u, $\Delta_{u,v}$'s and $\hat{X}_u$ (as soon as generated)
- and uses ZKPs to prove

$$\begin{aligned} X_u &\in [0,1], & \\ \Delta_{u,v} &= -\Delta_{v,u}, & \forall \{v \in N(u)\} \\ \eta_u &\sim \mathcal{N}(0, \sigma_\eta^2), & \text{(customizable precision)} \\ \hat{X}_u = X_u + \sum_{v \in N(u)} &\Delta_{u,v} + \eta_u. & \end{aligned}$$

GOPA: Verification Protocol

Each user $u \in U$

- commits to X_u, η_u, $\Delta_{u,v}$'s and $\hat{X}_u$ (as soon as generated)
- and uses ZKPs to prove

$$
\begin{aligned}
X_u &\in [0,1], & & \\
\Delta_{u,v} &= -\Delta_{v,u}, & & \forall\{v \in N(u)\} \\
\eta_u &\sim \mathcal{N}(0, \sigma_\eta^2), & & \text{(customizable precision)} \\
\hat{X}_u = X_u + \sum_{v \in N(u)} \Delta_{u,v} &+ \eta_u. & &
\end{aligned}
$$

- ensures correctness of GOPA
- can prove consistency of multiple GOPA runs over related data

GOPA: Verification Protocol

Each user $u \in U$

- commits to X_u, η_u, $\Delta_{u,v}$'s and $\hat{X}_u$ (as soon as generated)
- and uses ZKPs to prove

$$\begin{aligned} X_u &\in [0,1], & \\ \Delta_{u,v} &= -\Delta_{v,u}, & \forall\{v \in N(u)\} \\ \eta_u &\sim \mathcal{N}(0, \sigma_\eta^2), & \text{(customizable precision)} \\ \hat{X}_u = X_u + \sum_{v \in N(u)} & \Delta_{u,v} + \eta_u. & \end{aligned}$$

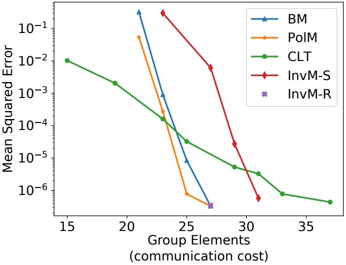
- ensures correctness of GOPA
- can prove consistency of multiple GOPA runs over related data
- verifying distributions: some elaboration

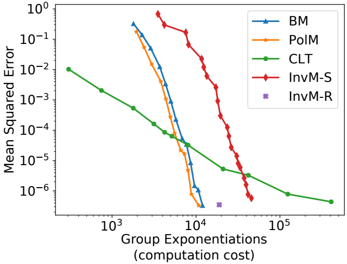
Proving $\eta_u \sim \mathcal{N}(0, \sigma_\eta^2)$

For each technique, we measure

- Quality: MSE to an ideal Gaussian over 10^7 samples
- Cost per sample: communication and computation

for different precision parameters.



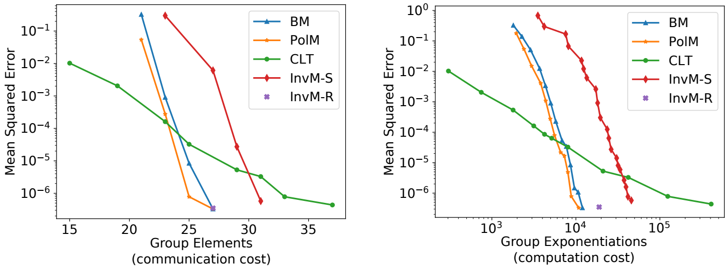


Proving $\eta_u \sim \mathcal{N}(0, \sigma_\eta^2)$

For each technique, we measure

- Quality: MSE to an ideal Gaussian over 10^7 samples
- Cost per sample: communication and computation

for different precision parameters.



- **quality impacts on privacy**
- if quality is more important: PolM and BM
  ($<$ 0.5 seconds, $<$ 1 KByte)
- otherwise: CLT can generate fast samples (10 milliseconds)

Introduction

Privacy

GOssip noise for Private Averaging

Conclusion

Summary

In this work we

- provide a protocol to privately compute statistics and models through averaging
- prove that it achieves similar accuracy than the central setting
- prove that it can achieve good balance between communication and amount of DP noise
- provide robustness against malicious users
  - similar to the central setting
  - with tractable computational cost

Future Work and Perspectives

1. Further Experimental Study
   - determine parameters that impact in the runtime
   - exploit vectorization
   - simulate realistic malicious behavior and dropouts

Future Work and Perspectives

1. Further Experimental Study
   - determine parameters that impact in the runtime
   - exploit vectorization
   - simulate realistic malicious behavior and dropouts
2. Computation of Statistics
   - what can be computed using private averaging as a building block?
   - how can our algorithms be combined with other building blocks (MPC, Shuffling, ...)

Future Work and Perspectives

1. Further Experimental Study
   - determine parameters that impact in the runtime
   - exploit vectorization
   - simulate realistic malicious behavior and dropouts
2. Computation of Statistics
   - what can be computed using private averaging as a building block?
   - how can our algorithms be combined with other building blocks (MPC, Shuffling, ...)
3. Derivation of more efficient privacy bounds
   - better composition bounds for specific mechanisms (e.g. tighter than current advanced composition)
   - exploit noise that is already present in the data or computation

Future Work and Perspectives

1. Further Experimental Study
   - determine parameters that impact in the runtime
   - exploit vectorization
   - simulate realistic malicious behavior and dropouts
2. Computation of Statistics
   - what can be computed using private averaging as a building block?
   - how can our algorithms be combined with other building blocks (MPC, Shuffling, ...)
3. Derivation of more efficient privacy bounds
   - better composition bounds for specific mechanisms (e.g. tighter than current advanced composition)
   - exploit noise that is already present in the data or computation
4. Verifying correct training of models
   - proving correct computation of training is challenging
   - verification cost must be tractabe for Federated Learning
   - could we use the model to prove it is good enough?

Thanks for listening !

References I

Bell, J. H., Bonawitz, K. A., Gascón, A., Lepoint, T., and Raykova, M. (2020).
Secure Single-Server Aggregation with (Poly)Logarithmic Overhead.
In *CCS*.

Bonawitz, K., Ivanov, V., Kreuter, B., Marcedone, A., McMahan, H. B., Patel, S., Ramage, D., Segal, A., and Seth, K. (2017).
Practical Secure Aggregation for Privacy-Preserving Machine Learning.
In *CCS*.

References II

Duchi, J. C., Jordan, M. I., and Wainwright, M. J. (2013).
Local Privacy and Statistical Minimax Rates.
In *2013 IEEE 54th Annual Symposium on Foundations of Computer Science*, pages 429–438.
ISSN: 0272-5428.

Dwork, C. (2006).
Differential Privacy.
In *ICALP*.

Dwork, C., Kenthapadi, K., McSherry, F., Mironov, I., and Naor, M. (2006).
Our Data, Ourselves: Privacy Via Distributed Noise Generation.
In Vaudenay, S., editor, *Advances in Cryptology - EUROCRYPT 2006*, pages 486–503, Berlin, Heidelberg. Springer Berlin Heidelberg.

References III

Goldwasser, S., Rothblum, G. N., Shafer, J., and Yehudayoff, A. (2021).
Interactive Proofs for Verifying Machine Learning.
In Lee, J. R., editor, *12th Innovations in Theoretical Computer Science Conference (ITCS 2021)*, volume 185 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 41:1–41:19, Dagstuhl, Germany. Schloss Dagstuhl–Leibniz-Zentrum für Informatik.

Walther, J. S. (1971).
A unified algorithm for elementary functions.
In *Proceedings of the May 18-20, 1971, spring joint computer conference*, AFIPS '71 (Spring), pages 379–385, New York, NY, USA. Association for Computing Machinery.