PRIVACY-PRESERVING FEDERATED LEARNING

Aurélien Bellet (Inria)

DeepMind Paris
March 31, 2022

OUTLINE

1. Federated Learning (FL)

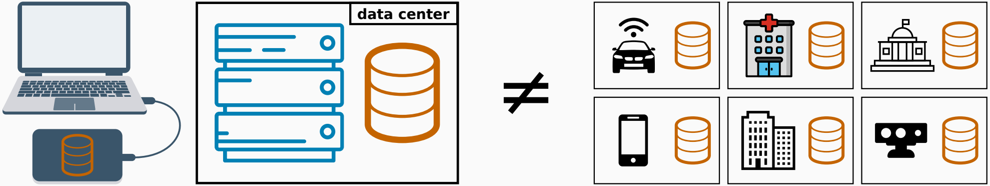
2. Privacy-Preserving FL with an untrusted server

3. Fully decentralized privacy-preserving FL

4. Wrapping up

Federated Learning (FL)

- The standard setting in Machine Learning (ML) considers a centralized dataset
- But in the real world data is often decentralized across different parties

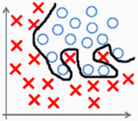


WHY DON'T WE ALWAYS CENTRALIZE DATA?

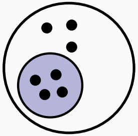
1. Sending the data may be too costly
    - Self-driving cars are expected to generate several TBs of data a day
    - Some wireless devices have limited bandwidth/power
2. **Data may be considered too sensitive**
    - Growing public awareness and regulations on data privacy
    - Keeping control of data can give a competitive advantage in business and research

HOW ABOUT EACH PARTY LEARNING ON ITS OWN?

1. The local dataset may be too small
    - Sub-par predictive performance (e.g., due to overfitting)
    - Non-statistically significant results (e.g., medical studies)



2. The local dataset may be biased
    - Not representative of the target distribution



Federated Learning (FL) aims to
collaboratively train ML models
while keeping the data decentralized

- FL is a booming topic
  - Term first coined in 2016; more than 1,000 papers in first half of 2020 alone[1]
  - First real-world deployments by companies and researchers
- FL is multidisciplinary: involves ML, optimization, statistics, privacy & security, networks, systems...

[1] https://www.forbes.com/sites/robtoews/2020/10/12/the-next-generation-of-artificial-intelligence/

Distributed learning

- Data is centrally stored (e.g., in a data center)
- The goal is to train faster → distribute data uniformly at random across workers

Federated Learning

- Data is naturally distributed → local datasets are heterogeneous (not iid, imbalance)
- Data may be sensitive → need to enforce privacy constraints
- Participants may be unreliable, unavailable (with possible time/space correlations)
- Participants may be malicious
- ...

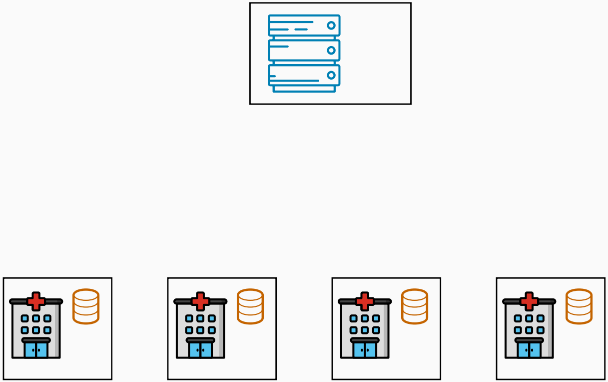
CLASSIC FL PROBLEM FORMULATION

- We consider a set of K parties (also called users or clients)
- Each party k holds a dataset $\mathcal{D}_k$
- We denote by θ the model parameters (e.g., weights of a neural network)
- We want to find the parameters that minimize the overall prediction loss:

$$\min_{\theta} \frac{1}{K} \sum_{k=1}^{K} F(\theta; \mathcal{D}_k), \quad \text{where } F \text{ is differentiable in } \theta$$

- This covers a broad class of ML problems formulated as empirical risk minimization

A BASELINE FL ALGORITHM: FEDAVG [McMahan et al., 2017]



Algorithm FedAvg (server-side)

initialize θ

for each round $t = 0, 1, \ldots$ **do**

  for each party k in parallel **do**

    $\theta_k \leftarrow \text{ClientUpdate}(k, \theta)$

  $\theta \leftarrow \frac{1}{K} \sum_{k=1}^{K} \theta_k$

Algorithm ClientUpdate(k, θ)

Parameters: # steps L, step size η

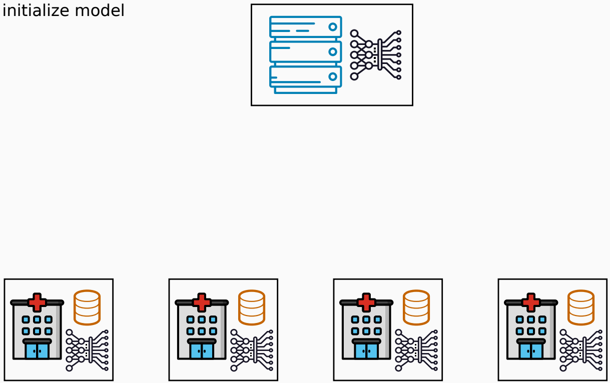
for $1, \ldots, L$ **do**

  $\theta \leftarrow \theta - \eta \nabla F(\theta; \mathcal{D}_k)$

send θ to server

- Numerous extensions / improvements: fully decentralized (no server), dealing with highly heterogeneous data, compression, fairness, and much more [Kairouz et al., 2021]

A BASELINE FL ALGORITHM: FEDAVG [McMahan et al., 2017]



Algorithm FedAvg (server-side)

initialize θ

for each round $t = 0, 1, \dots$ **do**

  for each party k in parallel **do**

    $\theta_k \leftarrow \text{ClientUpdate}(k, \theta)$

  $\theta \leftarrow \frac{1}{K} \sum_{k=1}^{K} \theta_k$

Algorithm ClientUpdate(k, θ)

Parameters: # steps L, step size η

for $1, \dots, L$ **do**

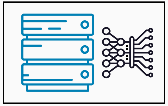
  $\theta \leftarrow \theta - \eta \nabla F(\theta; \mathcal{D}_k)$

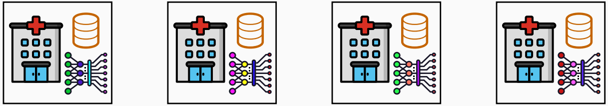
send θ to server

- Numerous extensions / improvements: fully decentralized (no server), dealing with highly heterogeneous data, compression, fairness, and much more [Kairouz et al., 2021]

A BASELINE FL ALGORITHM: FEDAVG [McMahan et al., 2017]

each party makes an update
using its local dataset





Algorithm FedAvg (server-side)

initialize θ

for each round $t = 0, 1, \ldots$ **do**

  for each party k in parallel **do**

    $\theta_k \leftarrow \mathrm{ClientUpdate}(k, \theta)$

  $\theta \leftarrow \frac{1}{K} \sum_{k=1}^{K} \theta_k$

Algorithm ClientUpdate(k, θ)

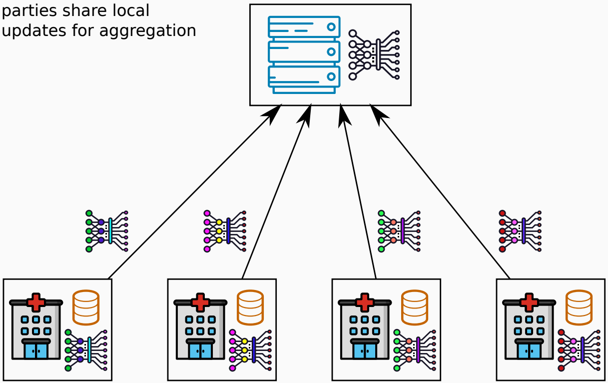
Parameters: # steps L, step size η

for $1, \ldots, L$ **do**

  $\theta \leftarrow \theta - \eta \nabla F(\theta; \mathcal{D}_k)$

send θ to server

- Numerous extensions / improvements: fully decentralized (no server), dealing with highly heterogeneous data, compression, fairness, and much more [Kairouz et al., 2021]



Algorithm FedAvg (server-side)

initialize θ

for each round $t = 0, 1, \dots$ **do**

  for each party k in parallel **do**

    $\theta_k \leftarrow \text{ClientUpdate}(k, \theta)$

  $\theta \leftarrow \frac{1}{K} \sum_{k=1}^{K} \theta_k$

Algorithm ClientUpdate(k, θ)

Parameters: # steps L, step size η

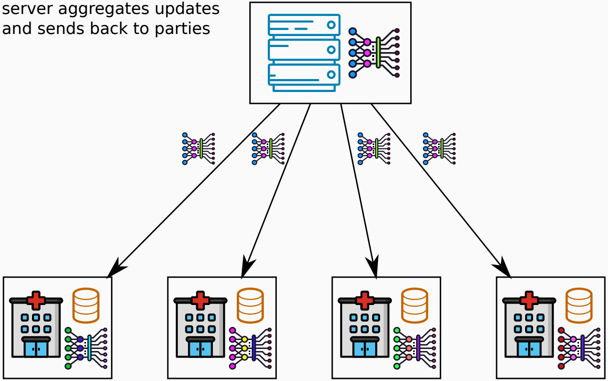
for $1, \dots, L$ **do**

  $\theta \leftarrow \theta - \eta \nabla F(\theta; \mathcal{D}_k)$

send θ to server

- Numerous extensions / improvements: fully decentralized (no server), dealing with highly heterogeneous data, compression, fairness, and much more [Kairouz et al., 2021]

A BASELINE FL ALGORITHM: FEDAVG [McMahan et al., 2017]



Algorithm FedAvg (server-side)

initialize θ

for each round $t = 0, 1, \dots$ **do**

  for each party k in parallel **do**

    $\theta_k \leftarrow \text{ClientUpdate}(k, \theta)$

  $\theta \leftarrow \frac{1}{K} \sum_{k=1}^{K} \theta_k$

Algorithm ClientUpdate(k, θ)

Parameters: # steps L, step size η

for $1, \dots, L$ **do**

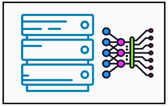
  $\theta \leftarrow \theta - \eta \nabla F(\theta; \mathcal{D}_k)$

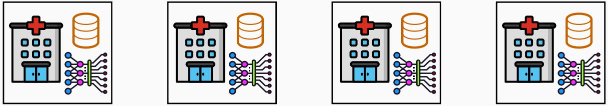
send θ to server

- Numerous extensions / improvements: fully decentralized (no server), dealing with highly heterogeneous data, compression, fairness, and much more [Kairouz et al., 2021]

A BASELINE FL ALGORITHM: FEDAVG [McMahan et al., 2017]

parties update their copy
of the model and iterate





Algorithm FedAvg (server-side)

initialize θ

for each round $t = 0, 1, \ldots$ **do**

  for each party k in parallel **do**

    $\theta_k \leftarrow \text{ClientUpdate}(k, \theta)$

  $\theta \leftarrow \frac{1}{K} \sum_{k=1}^{K} \theta_k$

Algorithm ClientUpdate(k, θ)

Parameters: # steps L, step size η

for $1, \ldots, L$ **do**

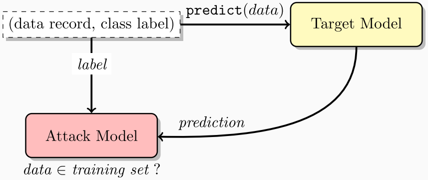
  $\theta \leftarrow \theta - \eta \nabla F(\theta; \mathcal{D}_k)$

send θ to server

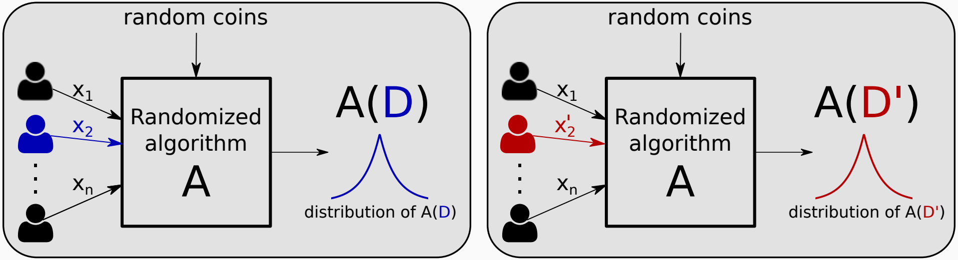
- Numerous extensions / improvements: fully decentralized (no server), dealing with highly heterogeneous data, compression, fairness, and much more [Kairouz et al., 2021]

PRIVACY ISSUES IN (FEDERATED) ML

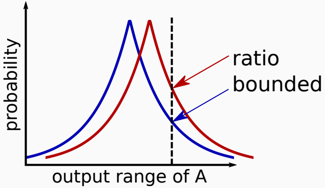
- ML models are susceptible to various attacks on data privacy
- Membership inference attacks try to infer the presence of a known individual in the training set, e.g., by exploiting the confidence in model predictions [Shokri et al., 2017]



- Reconstruction attacks try to infer some of the points used to train the model, e.g., by differencing attacks [Paige et al., 2020]
- Federated Learning offers an additional attack surface as the server and other parties observe model updates (not only the final model) [Nasr et al., 2019, Geiping et al., 2020]



- Neighboring datasets $\mathcal{D} = \{x_1, x_2, \ldots, x_n\}$ and $\mathcal{D}' = \{x_1, x'_2, x_3, \ldots, x_n\}$
- **Requirement**: $\mathcal{A}(\mathcal{D})$ and $\mathcal{A}(\mathcal{D}')$ should have "close" distribution



Definition ([Dwork et al., 2006], informal)

A randomized algorithm $\mathcal{A}$ is (ε, δ)-differentially private (DP) if for all neighboring datasets $\mathcal{D} = \{x_1, x_2, \ldots, x_n\}$ and $\mathcal{D}' = \{x_1, x_2', x_3, \ldots, x_n\}$ and all sets S:

$$\Pr[\mathcal{A}(\mathcal{D}) \in S] \leq e^{\varepsilon} \Pr[\mathcal{A}(\mathcal{D}') \in S] + \delta.$$

- For meaningful privacy guarantees, think of $\varepsilon \leq 1$ and $\delta \ll 1/n$
- Key principle: privacy is a property of the analysis, not of a particular output (in contrast to e.g., k-anonymity)
- Dwork, McSherry, Nissim & Smith won the Gödel prize for this in 2017

KEY PROPERTIES OF DIFFERENTIAL PRIVACY

- DP is immune to post-processing: it is impossible to compute a function of the output of the private algorithm and make it less differentially private
- DP is robust to arbitrary auxiliary knowledge: the guarantee is just as strong if the adversary knows all but one record
- DP is robust under composition: if multiple analyses are performed on the same data, as long as each one satisfies DP, all the information released taken together will still satisfy DP (albeit with a degradation in the parameters)

- Consider f taking as input a dataset and returning a p-dimensional real vector

Gaussian mechanism $\mathcal{A}_{\text{Gauss}}(\mathcal{D}, f, \varepsilon, \delta)$

1. Compute sensitivity $\Delta = \max_{(\mathcal{D},\mathcal{D}') \text{ are neighboring}} \|f(\mathcal{D}) - f(\mathcal{D}')\|_2$
2. For $i = 1, \dots, p$: draw $Y_i \sim \mathcal{N}(0, \sigma^2)$ independently for each i, where $\sigma = \frac{\sqrt{2\ln(1.25/\delta)}\Delta}{\varepsilon}$
3. Output $f(\mathcal{D}) + Y$, where $Y = (Y_1, \dots, Y_p) \in \mathbb{R}^p$

Theorem

Let $\varepsilon, \delta > 0$. The Gaussian mechanism $\mathcal{A}_{\text{Gauss}}(\cdot, f, \varepsilon, \delta)$ is (ε, δ)-DP.

- Noise calibrated using sensitivity of f and privacy budget (ε and δ)
- Induces a clear privacy-utility trade-off

- **Central DP**: a trusted curator collects raw data and runs a DP algorithm $\mathcal{A}$ on it $\rightarrow$ the output $\mathcal{A}(\mathcal{D})$ is only the final result
- **Local DP**: there is no trusted curator so each party must locally randomize its contributions $\rightarrow$ the output of $\mathcal{A}(\mathcal{D})$ consists of all messages sent by all parties
- Local DP is a suitable model for FL without trusted parties but, for a fixed (ϵ, δ)-DP guarantee, its utility cost is typically $\sqrt{K}$ larger

$\rightarrow$ study intermediate models allowing better utility without relying on trusted parties

Privacy-Preserving FL with an untrusted server

A KEY FUNCTIONALITY: DP AGGREGATION

- In FL algorithms with a server, interaction is needed only to aggregate local updates
- In other words: DP aggregation $+$ Composition property of DP $\implies$ DP-FL
- **Differentially private aggregation:** given a private value $\theta_k \in [0, 1]$ for each party k, we want to accurately estimate $\theta^{avg} = \frac{1}{K}\sum_k \theta_k$ under a DP constraint
- Central DP: trusted server computes θ^{avg} and adds Gaussian noise
- Local DP: each party k adds (more) Gaussian noise to θ_k before sharing it

- Assume that pairs of parties are able to exchange encrypted messages (the server may act as relay): this can be achieved e.g. through a public key infrastructure

Algorithm GOPA protocol [Sabater et al., 2020]

Each party k generates independent Gaussian noise η_k
Each party k selects a random set of m other parties
for all selected pairs of parties $k \sim l$ **do**
  Parties k and l securely exchange pairwise-canceling Gaussian noise $\Delta_{k,l} = -\Delta_{l,k}$
Each party k sends $\hat{\theta}_k = \theta_k + \sum_{k \sim l} \Delta_{k,l} + \eta_k$ to the server

- **Estimate of the average:** $\hat{\theta}^{avg} = \frac{1}{K} \sum_k \hat{\theta}_k = \theta^{avg} + \frac{1}{K} \sum_k \eta_k$

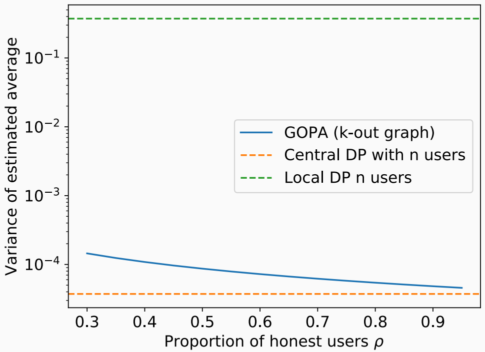
PRIVACY GUARANTEES FOR GOPA

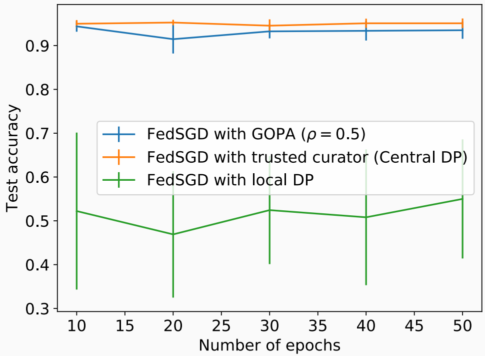
- **Adversary**: coalition of the server with a proportion $1 - \tau$ of the parties

Theorem (Privacy of GOPA [Sabater et al., 2020], informal)

- *Let each party select $m = O(\log(\tau K)/\tau)$ other parties*
- *Set the independent noise variance so as to satisfy (ϵ, δ')-DP in the central model*
- *For large enough pairwise noise variance, GOPA is (ϵ, δ)-DP with $\delta = O(\delta')$.*

- Same utility as central DP with only logarithmic number of messages per party
- Our theoretical results give practical values for the quantities above
- Our general result quantifies the effect of an arbitrary topology G on DP guarantees
- We also provide correctness guarantees against malicious parties [Sabater et al., 2020]

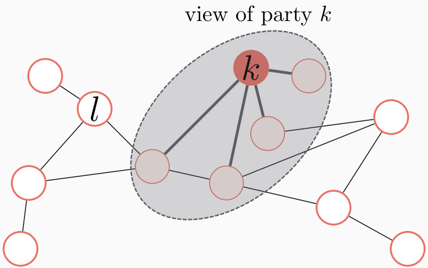




- For reasonable proportions ρ of honest parties, the variance of the estimated average produced by GOPA is similar to the trusted curator setting
- As expected, the resulting FL model also has similar accuracy

Fully decentralized privacy-preserving FL

- In fully decentralized FL, there is no global aggregation step



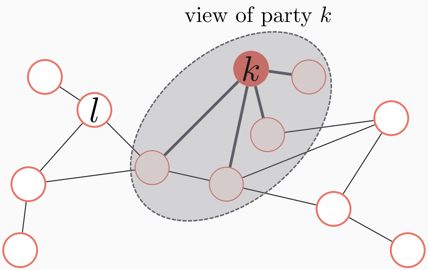
- But there is no server observing all messages, and each party k has a limited view
- Can this be used to prove stronger differential privacy guarantees?

- Let $\mathcal{O}_k$ be the set of messages sent and received by party k

Definition (Network DP [Cyffers and Bellet, 2022])

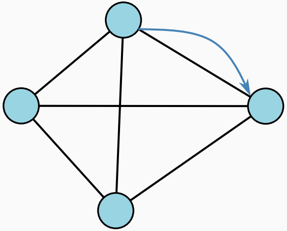
An algorithm $\mathcal{A}$ satisfies (ϵ, δ)-network DP if for all pairs of distinct parties $k, l \in \{1, \dots, n\}$ and all pairs of datasets $\mathcal{D}, \mathcal{D}'$ that differ only in the local dataset of party l, we have:

$$\Pr[\mathcal{O}_k(\mathcal{A}(\mathcal{D}))] \leq e^{\epsilon} \Pr[\mathcal{O}_k(\mathcal{A}(\mathcal{D}'))] + \delta.$$



- This is a relaxation of local DP: if $\mathcal{O}_k$ contains the full transcript of messages, then network DP boils down to local DP

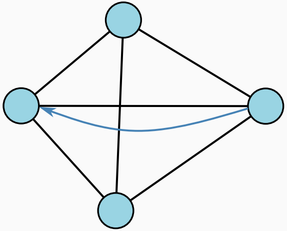
- Consider the standard objective $F(\theta; \mathcal{D}) = \frac{1}{K} \sum_{k=1}^{K} F_k(\theta; \mathcal{D}_k)$ and a complete graph
- We consider a fully decentralized algorithm where the model is updated sequentially by following a random walk



Algorithm Private decentralized SGD on a complete graph

Initialize model θ
for $t = 1$ to T **do**
  Current party updates θ by a gradient update with Gaussian noise
  Current party sends θ to a random party
return θ

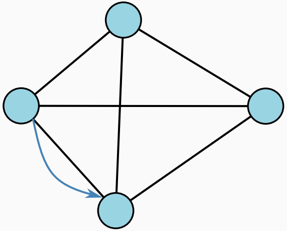
- Consider the standard objective $F(\theta; \mathcal{D}) = \frac{1}{K} \sum_{k=1}^{K} F_k(\theta; \mathcal{D}_k)$ and a complete graph
- We consider a fully decentralized algorithm where the model is updated sequentially by following a random walk



Algorithm Private decentralized SGD on a complete graph

Initialize model θ
for $t = 1$ to T **do**
  Current party updates θ by a gradient update with Gaussian noise
  Current party sends θ to a random party
return θ

- Consider the standard objective $F(\theta; \mathcal{D}) = \frac{1}{K} \sum_{k=1}^{K} F_k(\theta; \mathcal{D}_k)$ and a complete graph
- We consider a fully decentralized algorithm where the model is updated sequentially by following a random walk



Algorithm Private decentralized SGD on a complete graph

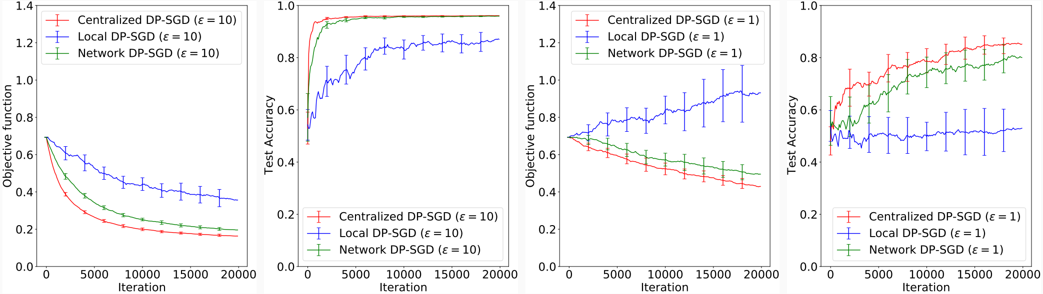
Initialize model θ
for $t = 1$ to T **do**
  Current party updates θ by a gradient update with Gaussian noise
  Current party sends θ to a random party
return θ

Theorem ([Cyffers and Bellet, 2022], informal)

To achieve a fixed (ϵ, δ)-DP guarantee with the previous algorithm, the standard deviation of the noise is $O(\sqrt{K}/\ln K)$ smaller under network DP than under local DP.

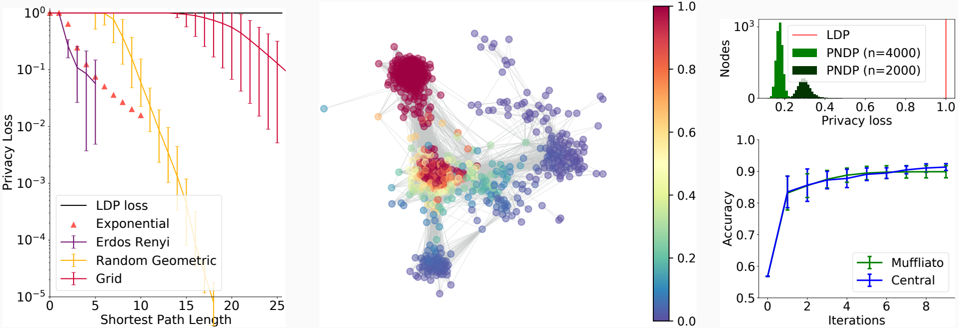
- Accounting for the limited view in fully decentralized algorithms amplifies privacy guarantees by a factor of $O(\ln K/\sqrt{K})$, nearly recovering the utility of central DP
- The proof leverages recent results on privacy amplification by iteration [Feldman et al., 2018] and exploits the randomness of the path taken by the model
- We show some robustness to collusion (albeit with smaller privacy amplification)

FULL DECENTRALIZATION: EMPIRICAL ILLUSTRATION



- Results are consistent with our theory: network DP-SGD significantly amplifies privacy guarantees compared to local DP-SGD

- In a recent work [Cyffers et al., 2022] we refine network DP to capture the privacy loss across each pair of nodes and prove amplification guarantees for gossip-based algorithms on arbitrary graphs



Wrapping up

TAKE-HOME MESSAGES

- FL allows to train machine learning models from decentralized datasets
- Not sharing data is not enough to ensure privacy: we need formal guarantees
- Differential privacy induces a privacy-utility trade-off which depends on the trust model (e.g., central versus local)
- In FL with a server, recent protocols for DP aggregation allow to achieve the same utility as the central model with reasonable computational and communication costs
- Full decentralization can amplify privacy guarantees, providing a new incentive for using such approaches beyond the usual motivation of scalability

THANK YOU FOR YOUR ATTENTION!

QUESTIONS?

[Blum, 1983] Blum, M. (1983).
Coin flipping by telephone a protocol for solving impossible problems.
ACM SIGACT News, 15(1):23–27.

[Cyffers and Bellet, 2022] Cyffers, E. and Bellet, A. (2022).
Privacy Amplification by Decentralization.
In *AISTATS*.

[Cyffers et al., 2022] Cyffers, E., Even, M., Bellet, A., and Massoulié, L. (2022).
Muffliato: Peer-to-Peer Privacy Amplification for Decentralized Optimization and Averaging.
In *NeurIPS*.

[Dwork et al., 2006] Dwork, C., McSherry, F., Nissim, K., and Smith, A. (2006).
Calibrating noise to sensitivity in private data analysis.
In *Theory of Cryptography (TCC)*.

[Feldman et al., 2018] Feldman, V., Mironov, I., Talwar, K., and Thakurta, A. (2018).
Privacy Amplification by Iteration.
In *FOCS*.

[Geiping et al., 2020] Geiping, J., Bauermeister, H., Dröge, H., and Moeller, M. (2020).
Inverting gradients - how easy is it to break privacy in federated learning?
In *NeurIPS*.

[Kairouz et al., 2021] Kairouz, P., McMahan, H. B., Avent, B., Bellet, A., Bennis, M., Bhagoji, A. N., Bonawitz, K., Charles, Z., Cormode, G., Cummings, R., D'Oliveira, R. G. L., Eichner, H., Rouayheb, S. E., Evans, D., Gardner, J., Garrett, Z., Gascón, A., Ghazi, B., Gibbons, P. B., Gruteser, M., Harchaoui, Z., He, C., He, L., Huo, Z., Hutchinson, B., Hsu, J., Jaggi, M., Javidi, T., Joshi, G., Khodak, M., Konecný, J., Korolova, A., Koushanfar, F., Koyejo, S., Lepoint, T., Liu, Y., Mittal, P., Mohri, M., Nock, R., Özgür, A., Pagh, R., Qi, H., Ramage, D., Raskar, R., Raykova, M., Song, D., Song, W., Stich, S. U., Sun, Z., Suresh, A. T., Tramèr, F., Vepakomma, P., Wang, J., Xiong, L., Xu, Z., Yang, Q., Yu, F. X., Yu, H., and Zhao, S. (2021).
Advances and Open Problems in Federated Learning.
Foundations and Trends® in Machine Learning, 14(1–2):1–210.

[McMahan et al., 2017] McMahan, H. B., Moore, E., Ramage, D., Hampson, S., and Agüera y Arcas, B. (2017).
Communication-efficient learning of deep networks from decentralized data.
In *AISTATS*.

[Nasr et al., 2019] Nasr, M., Shokri, R., and Houmansadr, A. (2019).
Comprehensive Privacy Analysis of Deep Learning: Passive and Active White-box Inference Attacks against Centralized and Federated Learning.
In *IEEE Symposium on Security and Privacy*.

[Paige et al., 2020] Paige, B., Bell, J., Bellet, A., Gascón, A., and Ezer, D. (2020).
Reconstructing Genotypes in Private Genomic Databases from Genetic Risk Scores.
In *International Conference on Research in Computational Molecular Biology RECOMB*.

[Pedersen, 1991] Pedersen, T. P. (1991).
Non-interactive and information-theoretic secure verifiable secret sharing.
In *CRYPTO*.

[Sabater et al., 2020] Sabater, C., Bellet, A., and Ramon, J. (2020).
Distributed Differentially Private Averaging with Improved Utility and Robustness to Malicious Parties.
Technical report, arXiv:2006.07218.

[Shokri et al., 2017] Shokri, R., Stronati, M., Song, C., and Shmatikov, V. (2017).
Membership Inference Attacks Against Machine Learning Models.
In *IEEE Symposium on Security and Privacy (S&P)*.

- Assume that pairs of parties are able exchange encrypted messages (the server may act as relay): this can be achieved for instance through a public key infrastructure
- Consider an arbitrary graph G over the set of parties

Algorithm GOPA protocol

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

  for all neighboring parties $\{k, l\}$ in G **do**

    k and l draw $y \sim \mathcal{N}(0, \sigma_\Delta^2)$

    set $\Delta_{k,l} \leftarrow y$, $\Delta_{l,k} \leftarrow -y$

  for each party k **do**

    k draws $\eta_k \sim \mathcal{N}(0, \sigma_\eta^2)$

    k reveals $\hat{\theta}_k \leftarrow \theta_k + \sum_{l \sim k} \Delta_{k,l} + \eta_k$

---

1. Neighbors $\{k, l\}$ in G securely exchange pairwise-canceling Gaussian noise
2. Each party k generate independent Gaussian noise
3. Party k reveals the sum of private value, pairwise and independent noise terms

- **Unbiased estimate of the average:** $\hat{\theta}^{avg} = \frac{1}{K} \sum_k \hat{\theta}_k$, with variance σ_η^2 / K

- **Adversary**: coalition of the server with a proportion $1-\rho$ of the parties

Theorem (Privacy of GOPA with random k-out graph [Sabater et al., 2020])

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

- *G be obtained by letting all parties randomly choose $m = O(\log(\rho K)/\rho)$ neighbors*
- *σ_η^2 so as to satisfy (ε, δ')-DP in the central model*
- *$\sigma_\Delta^2 = O(\sigma_\eta^2 \rho K/m)$*

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

- Same utility as central DP with only logarithmic number of messages per party
- Our theoretical results give practical values for m and σ_Δ^2
- Our general result quantifies the effect of an arbitrary topology G on DP guarantees
- We also provide correctness guarantees against malicious parties [Sabater et al., 2020]

GOPA: ENSURING CORRECTNESS

- Utility can be compromised by malicious parties tampering with the protocol (e.g., sending incorrect values to bias the outcome)
- It is impossible to force a party k to give the "right" input θ_k (this also holds in the trusted curator setting)
- We enable each party k to prove the following properties:

$$
\begin{aligned}
\theta_k &\in [0,1], && \forall k \in \{1,\dots,K\}\\
\Delta_{k,l} &= -\Delta_{l,k}, && \forall \{k,l\} \text{ neighbors in } G\\
\eta_k &\sim \mathcal{N}(0,\sigma_\eta^2), && \forall k \in \{1,\dots,K\}\\
\hat{\theta}_k &= \theta_k + \sum_{l\sim k} \Delta_{k,l} + \eta_k, && \forall k \in \{1,\dots,K\}
\end{aligned}
$$

- Parties publish an encrypted log of the computation using Pedersen commitments [Blum, 1983, Pedersen, 1991], which are additively homomorphic
- Based on these commitments, parties prove that the computation was done correctly using zero knowledge proofs

Theorem (Informal)

A party k that passes the verification proves that $\hat{\theta}_k$ was computed correctly. Additionally, by doing so, k does not reveal any additional information about θ_k.

- Costs per party remain linear in the number of neighbors
- Can prove consistency across multiple runs on same/similar data
- Can handle drop out

- Recall that we aim to minimize the objective of the form $F(\theta; \mathcal{D}) = \frac{1}{K}\sum_{k=1}^{K} F_k(\theta; \mathcal{D}_k)$
- Consider the complete graph

Algorithm Private decentralized SGD on a complete graph

Parameters: variance σ^2, # of steps T, step sizes $(\gamma(t))_{t=1}^{T}$

Initialize $\theta \in \mathbb{R}^p$
for $t = 1$ to T **do**
  Draw random party $k \sim \mathcal{U}(1, \dots, K)$
  $\eta = [\eta_1, \dots, \eta_p]$, with $\eta_i \sim \mathcal{N}(0, \sigma^2)$
  $\theta \leftarrow \theta - \gamma(t)[\nabla_\theta F_k(\theta; \mathcal{D}_k) + \eta]$
return θ

Theorem ([Cyffers and Bellet, 2022], informal)

Let $F_1(\cdot;\mathcal{D}_1), \dots, F_K(\cdot;\mathcal{D}_K)$ be convex, Lipschitz and smooth. Given $\varepsilon, \delta > 0$, let $T = \tilde{\Omega}(K^2)$ and σ^2 be such that private decentralized SGD satisfies (ε, δ)-local DP. Then the algorithm also satisfies $(\frac{\ln K}{\sqrt{K}}\varepsilon, \delta)$-network DP.

- Under network DP (i.e., full decentralization), privacy is amplified by a factor of $O(\ln K/\sqrt{K})$ compared to local DP, recovering the utility of central DP
- The proof leverages recent results on privacy amplification by iteration [Feldman et al., 2018] and exploits the randomness of the path taken by the model
- Note: for $T = o(K^2)$, the amplification effect is still strong and can be computed numerically, see [Cyffers and Bellet, 2022]