PRIVACY-PRESERVING DECENTRALIZED MACHINE LEARNING

Aurélien Bellet (Inria)

HeKA seminar, Inserm/Inria
March 29, 2021

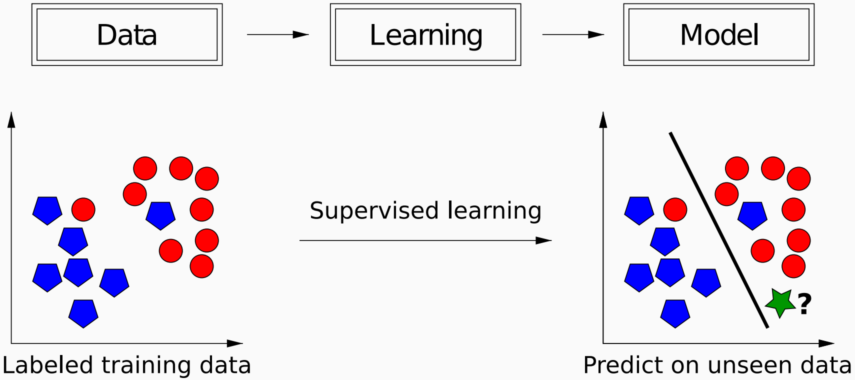
1. Decentralized Machine Learning

2. Privacy in Decentralized Machine Learning

3. Applications to the medical domain

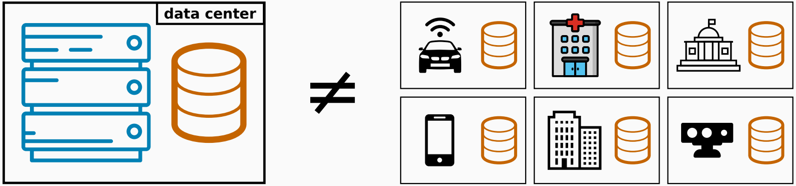
4. Wrapping up

Decentralized Machine Learning


Data
Learning
Model
Supervised learning
?
Labeled training data
Predict on unseen data

A SHIFT OF PARADIGM: FROM CENTRALIZED TO DECENTRALIZED DATA

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

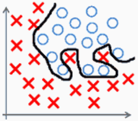


Why can't we just centralize the data?

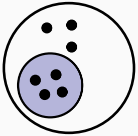
1. Sending the data may be too costly
   - Self-driving cars are expected to generate several TBs of data a day 
   - Some wireless devices have limited bandwidth/power 
2. **Data may be considered too sensitive**
   - **We see a growing public awareness and regulations on data privacy** 
     (we could try to anonymize the data, but it is generally difficult to prevent all possible re-identification attacks without destroying utility)
   - 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)

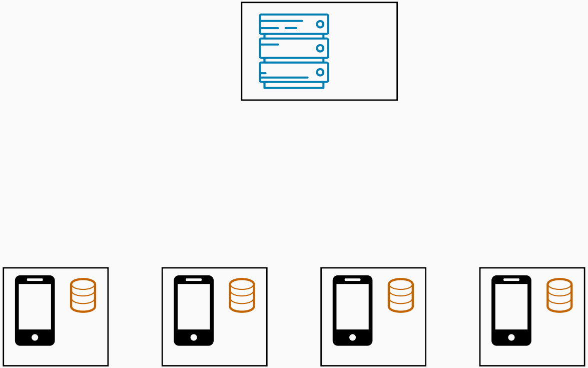


2. The local dataset may be biased
    - Not representative of the target distribution

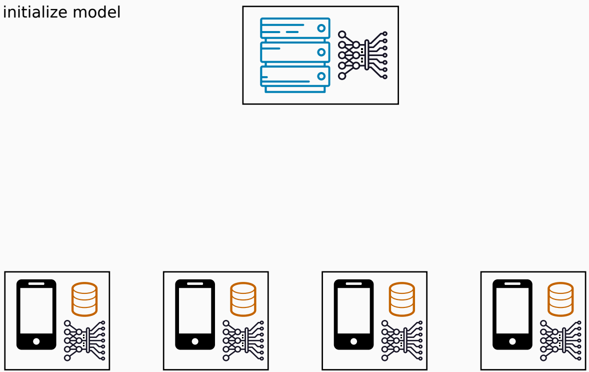


A BROAD DEFINITION OF DECENTRALIZED MACHINE LEARNING

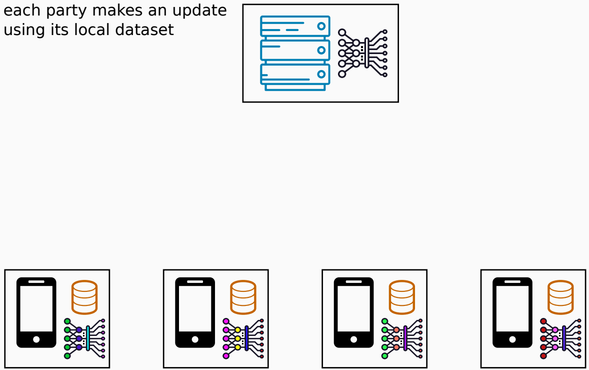
- **Decentralized Machine Learning (DML)**, also called Federated Learning, aims to collaboratively train a ML model while keeping the data decentralized



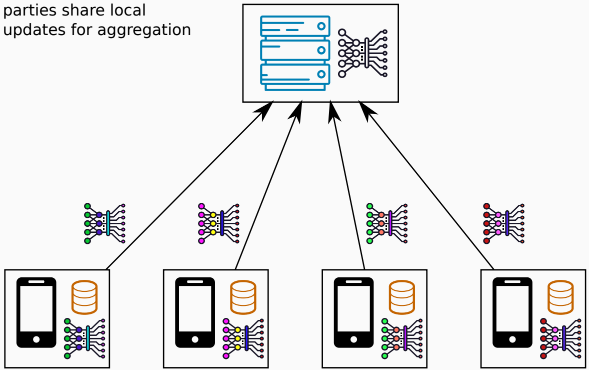
- **Decentralized Machine Learning (DML)**, also called Federated Learning, aims to collaboratively train a ML model while keeping the data decentralized



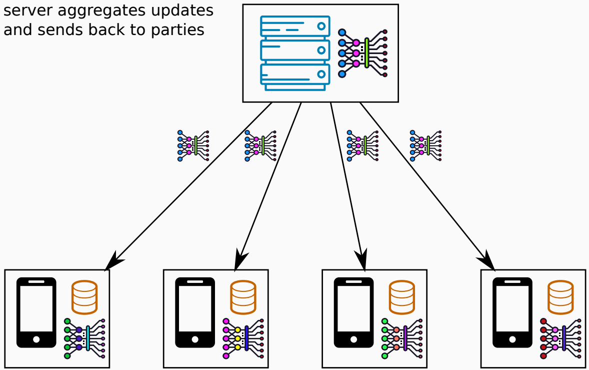
- **Decentralized Machine Learning (DML)**, also called Federated Learning, aims to collaboratively train a ML model while keeping the data decentralized



- **Decentralized Machine Learning (DML)**, also called Federated Learning, aims to collaboratively train a ML model while keeping the data decentralized



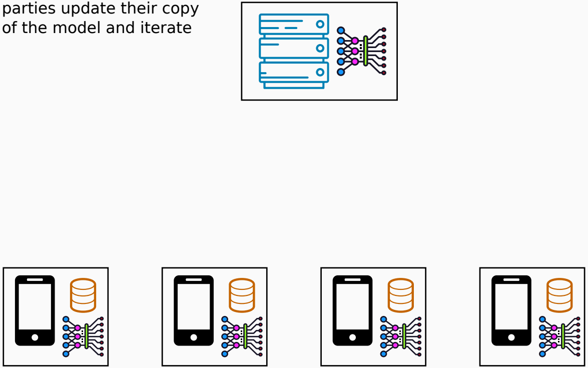
- **Decentralized Machine Learning (DML)**, also called Federated Learning, aims to collaboratively train a ML model while keeping the data decentralized



A BROAD DEFINITION OF DECENTRALIZED MACHINE LEARNING

- **Decentralized Machine Learning (DML)**, also called Federated Learning, aims to collaboratively train a ML model while keeping the data decentralized

parties update their copy of the model and iterate



- We would like the final model to be as good as the centralized solution (ideally), or at least better than what each party can learn on its own

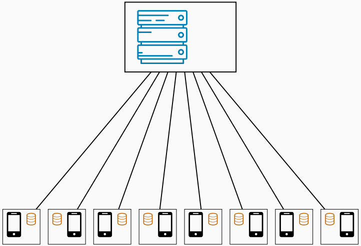
Data distribution

- In distributed learning, data is centrally stored (e.g., in a data center)
    - The main goal is just to train faster
    - We control how data is distributed across workers: usually, it is distributed uniformly at random across workers
- In DML, data is naturally distributed and generated locally
    - Data is **not** independent and identically distributed (non-i.i.d.), and it is imbalanced

Additional challenges that arise in DML

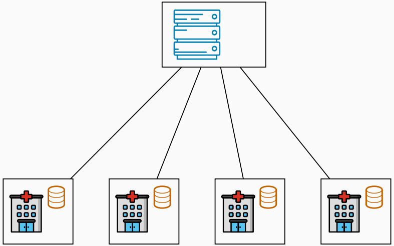
- Enforcing privacy constraints
- Dealing with the possibly limited reliability/availability of participants
- Achieving robustness against malicious parties
- ...

Cross-device DML



- Massive number of parties (up to 10^{10})
- Small dataset per party (could be size 1)
- Limited availability and reliability
- Some parties may be malicious

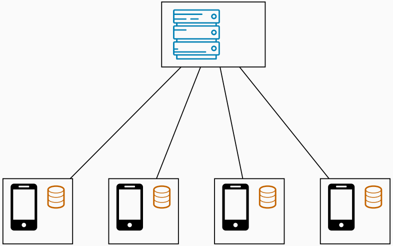
Cross-silo DML



- 2-100 parties
- Medium to large dataset per party
- Reliable parties, almost always available
- Parties are typically honest

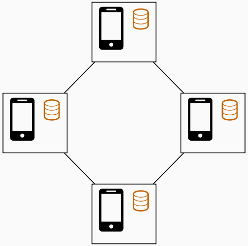
SERVER ORCHESTRATED VS. FULLY DECENTRALIZED DML

Server-orchestrated DML



- Server-client communication
- Global coordination, global aggregation
- Server is a single point of failure and may become a bottleneck

Fully decentralized DML



- Direct communication between parties
- No global coordination, local aggregation
- Naturally scales to a large number of participants

- We consider a set of K parties (clients)
- Each party k holds a dataset $\mathcal{D}_k$ of n_k points, so there is $n = \sum_k n_k$ points in total
- 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 error:

$$\min_{\theta} \sum_{k=1}^{K} \frac{n_k}{n} \text{Loss}(\theta; \mathcal{D}_k)$$

A STANDARD DML ALGORITHM: FEDAVG [McMahan et al., 2017]

- **Main idea:** clients update model with gradient descent to make it better on local data, server performs a weighted average of client updates

Algorithm FedAvg (server-side)

initialize θ
for each round $t = 0, 1, \dots$ **do**
  for each client k in parallel **do**
    $\theta_k \leftarrow \text{ClientUpdate}(k, \theta)$
  $\theta \leftarrow \sum_{k=1}^{K} \frac{n_k}{n} \theta_k$

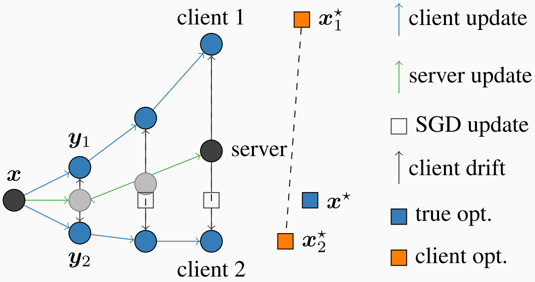
Algorithm ClientUpdate(k, θ)

Parameters: number of local steps L
learning rate η

for each local step $1, \dots, L$ **do**
  $\theta \leftarrow \theta - \eta \nabla \text{Loss}(\theta; \mathcal{D}_k)$
send θ to server

- $L > 1$ allows to reduce the number of communication rounds
- Can be extended to the fully decentralized case [Lian et al., 2017, Koloskova et al., 2020]

A KEY CHALLENGE: DEALING WITH HETEROGENEOUS DATA



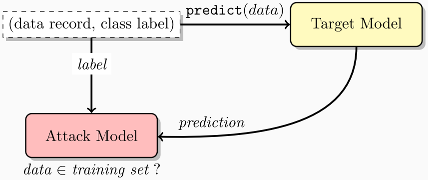
(Figure taken from [Karimireddy et al., 2020])

- When local datasets are non-i.i.d., FedAvg suffers from client drift
- Recent work on correcting updates [Karimireddy et al., 2020, Li et al., 2020]
- Can also learn personalized models [Smith et al., 2017, Zantedeschi et al., 2020]

Privacy in Decentralized Machine Learning

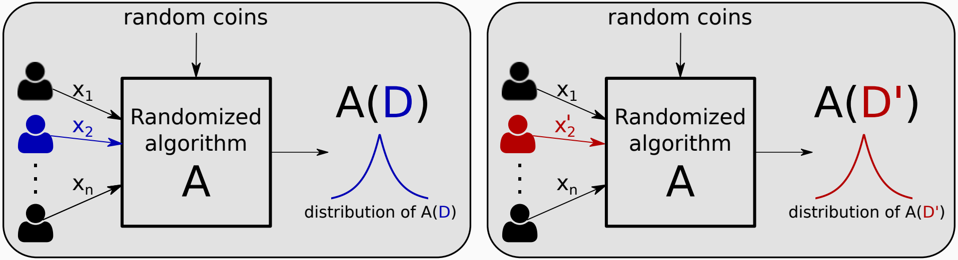
PRIVACY ISSUES IN (DECENTRALIZED) ML

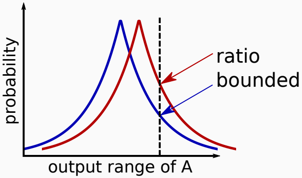
- ML models are susceptible to various attacks on data privacy
- Membership inference attacks try to infer the presence of a known individual in the training set, e.g., by exploiting the confidence in model predictions [Shokri et al., 2017]



- Reconstruction attacks try to infer some of the points used to train the model, e.g., by differencing attacks [Paige et al., 2020]
- Decentralized ML offers an additional attack surface because the server and/or other clients see intermediate model updates (not only the final model) [Nasr et al., 2019]

DIFFERENTIAL PRIVACY IN A NUTSHELL





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

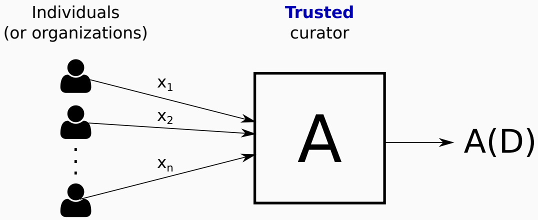
$\mathcal{A}$ is (ε, δ)-differentially private (DP) if for all neighboring datasets $\mathcal{D} = \{x_1, x_2, \ldots, x_n\}$ and $\mathcal{D}' = \{x_1, x_2', x_3, \ldots, x_n\}$ and all possible sets of outputs S:

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

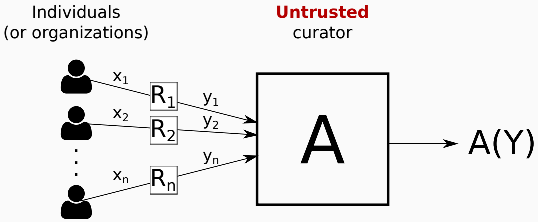
KEY PROPERTIES OF DIFFERENTIAL PRIVACY

- 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 (worst-case model): the guarantee is just as strong if the adversary knows all but one record and regardless of the adversary strategy and computational power
- 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)

Centralized setting (also called global setting or trusted curator setting): $\mathcal{A}$ is differentially private wrt dataset $\mathcal{D}$



Decentralized/federated setting (also called local setting or untrusted curator setting): each $\mathcal{R}_k$ is DP wrt record x_k (or local dataset $\mathcal{D}_k$)



- Most server-orchestrated DML algorithms follow the same high-level strategy:

  for $t = 1$ to T **do**
  At each party k: compute $\theta_k \leftarrow \textsc{LocalUpdate}(\theta, \theta_k)$, send θ_k to server
  At server: compute $\theta \leftarrow \frac{1}{K} \sum_k \theta_k$, send θ back to the participants

- Therefore:

$$\text{DP aggregation} + \text{Composition property of DP} \implies \text{DP-DML}$$

- **Differentially private aggregation:** given a private value $x_k \in \mathbb{R}$ for each party k, we want to accurately estimate $x^{avg} = \frac{1}{K} \sum_k x_k$ under an (ε, δ)-DP constraint

EXISTING APPROACHES TO DP AGGREGATION

- Centralized setting: trusted curator adds (Gaussian) noise to the average x^{avg}
- Decentralized setting: each party k adds noise to x_k before sharing it
- For a fixed DP guarantee, the error is $O(\sqrt{K})$ larger in the decentralized case!
- Cryptographic primitives such as secure aggregation [Bonawitz et al., 2017] and secure shuffling [Balle et al., 2019] can be used to close this gap but pose practical implementation challenges

A SIMPLE PROTOCOL FOR DP AGGREGATION: GOPA [Sabater et al., 2020]

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

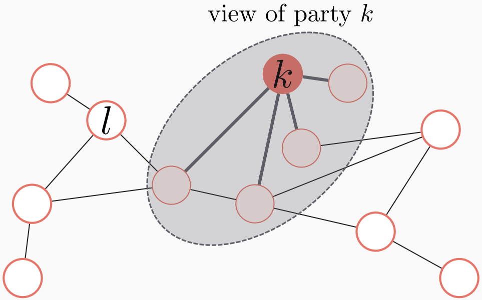
for each party k **do**
- k draws $\eta_k \sim \mathcal{N}(0, \sigma_\eta^2)$
- k reveals $\hat{x}_k \leftarrow x_k + \sum_{l \sim k} \Delta_{k,l} + \eta_k$

---

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

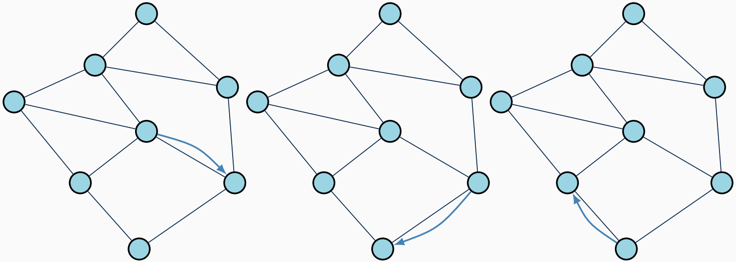
- Accurate: the result $\hat{x}^{avg} = \frac{1}{K} \sum_k \hat{x}_k$ can match the accuracy of the centralized setting
- Scalable: it is sufficient for each party to communicate with $O(\log K)$ others
- Robust: it can handle some collusions, dropouts and malicious behavior

PRIVACY BENEFITS OF FULL DECENTRALIZATION [Cyffers and Bellet, 2020]



- In the fully decentralized case, each party has a limited view of the system
- Can this be used to prove stronger differential privacy guarantees?

- Consider algorithms that sequentially update the estimate (e.g., ML model) by following a walk over the network graph [Ram et al., 2009, Mao et al., 2020]

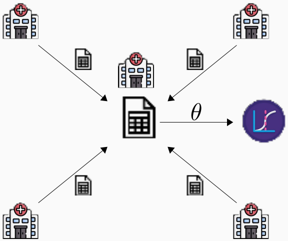


- We have shown that for some topologies (directed ring, complete graph), such algorithms can match the privacy-utility trade-off of the centralized setting
- Analysis relies on recent privacy amplification results [Balle et al., 2018] [Erlingsson et al., 2019, Feldman et al., 2018]

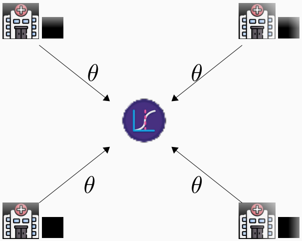
Applications to the medical domain

MULTI-CENTRIC MEDICAL STUDIES

Classic multi-centric study



Decentralized multi-centric study



- Multi-centric studies augment the statistical power of studies
- Decentralized studies could be easier to set up, could minimize privacy risks, and their results could be updated more regularly

ONGOING COLLABORATION WITH LILLE UNIVERSITY HOSPITAL

- Development of a decentralized machine learning library
- Proof of concept across hospitals of the G4 alliance as short term objective
- Identification of end-users needs and appropriate workflow with clinicians
- Understanding the regulatory requirements, in relation with CNIL

Wrapping up

CONCLUDING REMARKS

- Strong interest in ML community for decentralized/federated approaches, see recent survey [Kairouz et al., 2019]
- Can have differential privacy guarantees for these algorithms with the same utility as in the centralized setting:
  - via private aggregation, with a reasonable computational and communication overhead
  - via certain fully decentralized algorithms
- Compared to sharing "anonymized" data, DML restricts the usage to a specific ML analysis but can offer more robust privacy guarantees and/or better utility
- Clear applications to the medical domain

THANK YOU FOR YOUR ATTENTION!

REFERENCES I

[Balle et al., 2018] Balle, B., Barthe, G., and Gaboardi, M. (2018).
Privacy Amplification by Subsampling: Tight Analyses via Couplings and Divergences.
In *NeurIPS*.

[Balle et al., 2019] Balle, B., Bell, J., Gascón, A., and Nissim, K. (2019).
The Privacy Blanket of the Shuffle Model.
In *CRYPTO*.

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

[Bonawitz et al., 2017] 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*.

[Cyffers and Bellet, 2020] Cyffers, E. and Bellet, A. (2020).
Privacy Amplification by Decentralization.
Technical report, arXiv:2012.05326.

[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)*.

[Erlingsson et al., 2019] Erlingsson, U., Feldman, V., Mironov, I., Raghunathan, A., and Talwar, K. (2019).
Amplification by Shuffling: From Local to Central Differential Privacy via Anonymity.
In *SODA*.

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

[Kairouz et al., 2019] 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., 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., Konečný, J., Korolova, A., Koushanfar, F., Koyejo, S., Lepoint, T., Liu, Y., Mittal, P., Mohri, M., Nock, R., Özgür, A., Pagh, R., Raykova, M., Qi, H., Ramage, D., Raskar, R., 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. (2019).
Advances and Open Problems in Federated Learning.
Technical report, arXiv:1912.04977.

[Karimireddy et al., 2020] Karimireddy, S. P., Kale, S., Mohri, M., Reddi, S. J., Stich, S. U., and Suresh, A. T. (2020).
SCAFFOLD: Stochastic Controlled Averaging for On-Device Federated Learning.
In *ICML*.

[Koloskova et al., 2020] Koloskova, A., Loizou, N., Boreiri, S., Jaggi, M., and Stich, S. U. (2020).
A Unified Theory of Decentralized SGD with Changing Topology and Local Updates.
In *ICML*.

[Li et al., 2020] Li, T., Sahu, A. K., Zaheer, M., Sanjabi, M., Talwalkar, A., and Smith, V. (2020).
Federated Optimization in Heterogeneous Networks.
In *MLSys*.

[Lian et al., 2017] Lian, X., Zhang, C., Zhang, H., Hsieh, C.-J., Zhang, W., and Liu, J. (2017).
Can Decentralized Algorithms Outperform Centralized Algorithms? A Case Study for Decentralized Parallel Stochastic Gradient Descent.
In *NIPS*.

[Mao et al., 2020] Mao, X., Yuan, K., Hu, Y., Gu, Y., Sayed, A. H., and Yin, W. (2020).
Walkman: A Communication-Efficient Random-Walk Algorithm for Decentralized Optimization.
IEEE Transactions on Signal Processing, 68:2513–2528.

[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*.

[Ram et al., 2009] Ram, S., Nedić, A., and Veeravalli, V. (2009).
Incremental stochastic subgradient algorithms for convex optimization.
SIAM Journal on Optimization, 20(2):691–717.

[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).*

[Smith et al., 2017] Smith, V., Chiang, C.-K., Sanjabi, M., and Talwalkar, A. S. (2017).
Federated Multi-Task Learning.
In *NIPS*.

[Zantedeschi et al., 2020] Zantedeschi, V., Bellet, A., and Tommasi, M. (2020).
Fully Decentralized Joint Learning of Personalized Models and Collaboration Graphs.
In *AISTATS*.

- **Adversary**: proportion $1-\rho$ of colluding malicious parties who observe all communications they take part in
- Denote by H the set of honest-but-curious parties, and by G^H the honest subgraph
- GOPA can achieve (ε, δ)-DP for any $\varepsilon, \delta > 0$ for connected G^H and large enough $\sigma_\eta^2, \sigma_\Delta^2$
- We show that σ_η^2 can be as small as in the centralized setting (matching its utility)
- We show that the required σ_Δ^2 depends on the topology of G^H

Theorem (Case of random m-out graph)

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

- *G be obtained by letting all parties randomly choose $m = O(\log(\rho n)/\rho)$ neighbors*
- *σ_η^2 so as to satisfy (ε, δ)-DP in the centralized (trusted curator) setting*
- *$\sigma_\Delta^2 = O(\sigma_\eta^2 |H|/m)$*

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

- Trusted curator utility with logarithmic number of messages per party
- Our theoretical results give practical values for m and σ_Δ^2

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 to give the "right" input (this also holds in the trusted curator setting)
- We enable each party u to prove the following properties:

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

GOPA: ENSURING CORRECTNESS

- 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 $\hat{x}_k$ was computed correctly. Additionally, by doing so, k does not reveal any additional information about x_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

- Each party k holds a local dataset $\mathcal{D}_k$, joint dataset $\mathcal{D} = \mathcal{D}_1 \cup \cdots \cup \mathcal{D}_K$
- $\mathcal{D} \sim_k \mathcal{D}'$ means that datasets $\mathcal{D}$ and $\mathcal{D}'$ differ only on party k's data
- $\mathcal{O}_k(\mathcal{A}(\mathcal{D})$: view of party k (local memory and messages received)

Definition (Network differential Privacy)

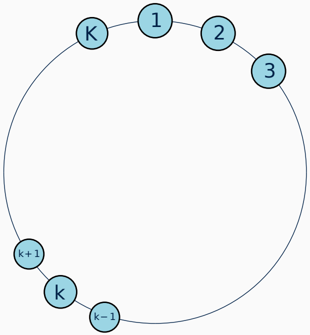
An algorithm $\mathcal{A}$ is (ε, δ)-network differentially private if for all pairs of parties (k, l) and for all datasets $\mathcal{D} \sim_k \mathcal{D}'$:

$$\Pr(\mathcal{O}_l(\mathcal{A}(\mathcal{D}))) \leq e^{\varepsilon} \Pr(\mathcal{O}_l(\mathcal{A}(\mathcal{D}'))) + \delta.$$

- Each party k has M values $x_k^1, \ldots, x_k^M$ and we want to estimate $\bar{x} = \sum_{k=1}^{K} \sum_{m=1}^{M} x_k^m$
- Let Perturb$(;\sigma)$ satisfy (ε, δ)-local DP

Algorithm Private real summation on a ring

$\tau \leftarrow 0$; $a \leftarrow 0$
for $m = 1$ to M **do**
  for $k = 1$ to K **do**
    if $a = 0$ **then**
      $\tau \leftarrow \tau + \text{Perturb}(x_k^m; \sigma)$
      $a = K - 2$
    else
      $\tau \leftarrow \tau + x_k^m$
      $a \leftarrow a - 1$
return τ



Simple example: real summation on a ring

Theorem (Privacy-utility guarantee)

Let $\varepsilon, \delta > 0$. The previously introduced algorithm

- *outputs an unbiased estimate of $\bar{x}$ with standard deviation $\sqrt{\lfloor MK/(K-1) \rfloor}\sigma$,*
- *satisfies $(\sqrt{2M \ln(1/\delta')}\varepsilon + M\varepsilon(e^{\varepsilon} - 1), M\delta + \delta')$-network DP for any $\delta' > 0$.*

- Same privacy-utility trade-off as a trusted aggregator
- Gain of $O(1/\sqrt{K})$ compared to local DP