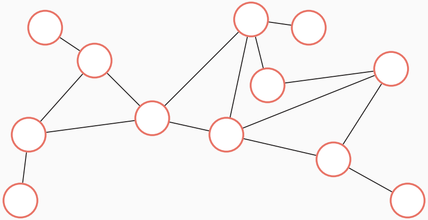
BETTER PRIVACY GUARANTEES FOR DECENTRALIZED FEDERATED LEARNING

Aurélien Bellet (Inria Lille)

Joint work with **Edwige Cyffers** (Inria Lille), Mathieu Even and Laurent Massoulié (Inria Paris)

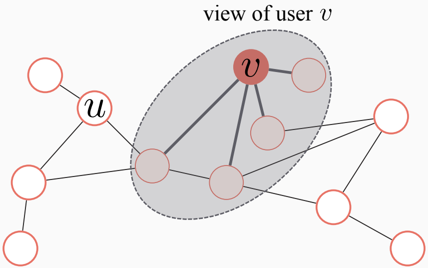
Workshop FL-Day - Decentralized Federated Learning: Approaches and Challenges
January 10, 2023

DECENTRALIZED ALGORITHMS: GOOD FOR PRIVACY?



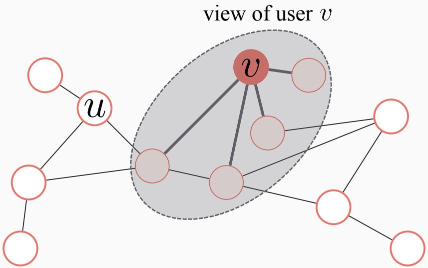
- In decentralized algorithms, such as decentralized SGD [Lian et al., 2017] [Koloskova et al., 2020], users communicate along the edges of a graph
- These algorithms are increasingly popular in machine learning due to their scalability

DECENTRALIZED ALGORITHMS: GOOD FOR PRIVACY?

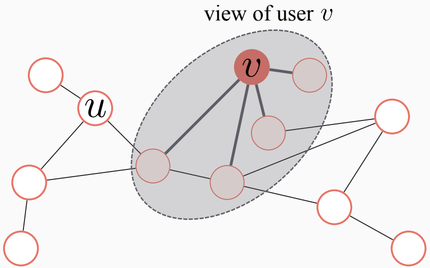


- In decentralized algorithms, such as decentralized SGD [Lian et al., 2017] [Koloskova et al., 2020], users communicate along the edges of a graph
- These algorithms are increasingly popular in machine learning due to their scalability
- **Folklore belief: "Decentralized algorithms are good for privacy because users have a limited view of the system"**

DECENTRALIZED ALGORITHMS: GOOD FOR PRIVACY?



- In decentralized algorithms, such as decentralized SGD [Lian et al., 2017] [Koloskova et al., 2020], users communicate along the edges of a graph
- These algorithms are increasingly popular in machine learning due to their scalability
- **Folklore belief: "Decentralized algorithms are good for privacy because users have a limited view of the system"**
- **Question:** is this claim really true? can we formalize and quantify these gains?



- In decentralized algorithms, such as decentralized SGD [Lian et al., 2017] [Koloskova et al., 2020], users communicate along the edges of a graph
- These algorithms are increasingly popular in machine learning due to their scalability
- **Folklore belief: "Decentralized algorithms are good for privacy because users have a limited view of the system"**
- **Question:** is this claim really true? can we formalize and quantify these gains? **Yes!**

OUTLINE

1. Background: Differential Privacy & DP-SGD

2. A relaxation of local DP for decentralized algorithms

3. Private random walk-based decentralized SGD

4. Private gossip-based decentralized SGD

5. Conclusion & Perspectives

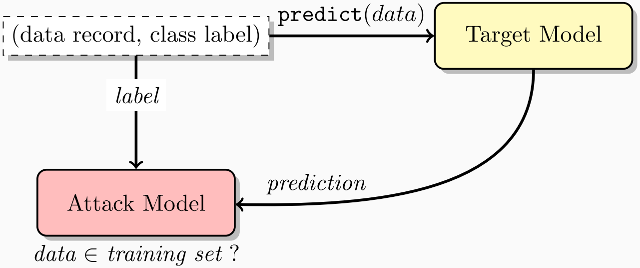
Background: Differential Privacy & DP-SGD

- ML models are susceptible to various attacks on data privacy

PRIVACY ISSUES IN MACHINE LEARNING

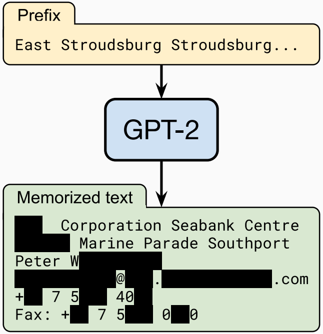
- ML models are susceptible to various attacks on data privacy
- Membership inference attack: infer whether a known individual data point was present in the training set

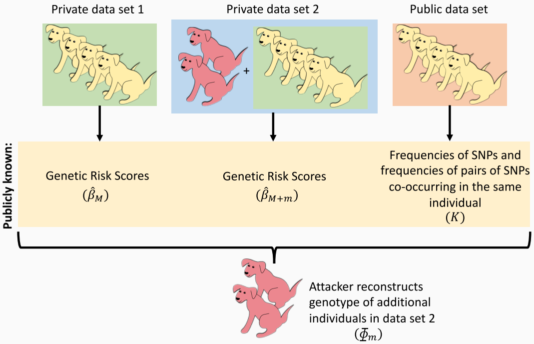
- ML models are susceptible to various attacks on data privacy
- Membership inference attack: infer whether a known individual data point was present in the training set
- For instance, one can exploit overconfidence in model predictions [Shokri et al., 2017] [Carlini et al., 2022]

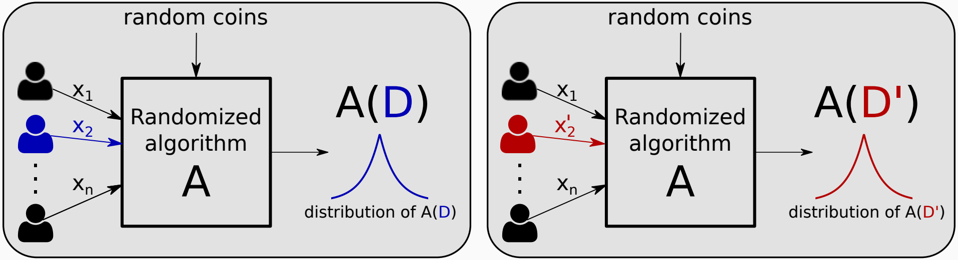


- Reconstruction attack: extract training data points from the model

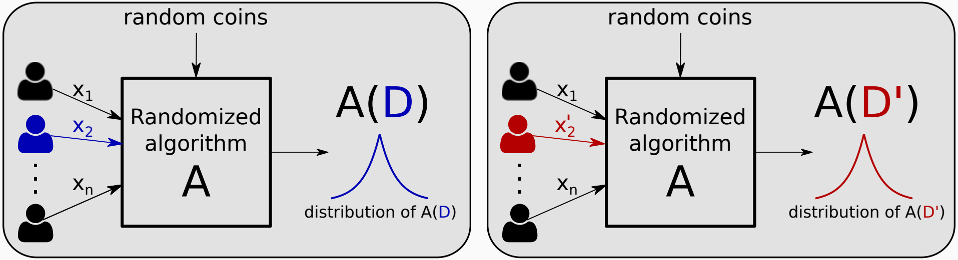
- Reconstruction attack: extract training data points from the model
- For instance, one can extract sensitive text from large language models [Carlini et al., 2021] or run differencing attacks on ML models [Paige et al., 2020]



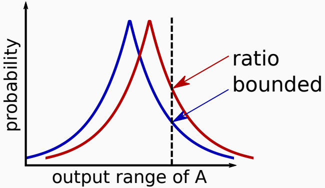




- Neighboring datasets $\mathcal{D} = \{x_1, x_2, \ldots, x_n\}$ and $\mathcal{D}' = \{x_1, x'_2, x_3, \ldots, x_n\}$



- 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 "similar" distributions



Definition (Rényi Differential Privacy [Mironov, 2017])

An algorithm $\mathcal{A}$ satisfies (α, ϵ)-Rényi Differential Privacy (RDP) for $\alpha > 1$ and $\epsilon > 0$ if for all pairs of neighboring datasets $\mathcal{D} \sim \mathcal{D}'$:

$$D_\alpha\left(\mathcal{A}(\mathcal{D}) || \mathcal{A}(\mathcal{D}')\right) \leq \epsilon\,, \tag{1}$$

where for two r.v. X, Y with densities μ_X, μ_Y, $D_\alpha(X \,||\, Y)$ is the Rényi divergence of order α:

$$D_\alpha(X \,||\, Y) = \frac{1}{\alpha - 1} \ln \int \left(\frac{\mu_X(z)}{\mu_Y(z)}\right)^\alpha \mu_Y(z) dz\,.$$

Definition (Rényi Differential Privacy [Mironov, 2017])

An algorithm $\mathcal{A}$ satisfies (α, ϵ)-Rényi Differential Privacy (RDP) for $\alpha > 1$ and $\epsilon > 0$ if for all pairs of neighboring datasets $\mathcal{D} \sim \mathcal{D}'$:

$$D_\alpha \left(\mathcal{A}(\mathcal{D}) || \mathcal{A}(\mathcal{D}') \right) \leq \epsilon \,, \tag{1}$$

where for two r.v. X, Y with densities μ_X, μ_Y, $D_\alpha(X \,||\, Y)$ is the Rényi divergence of order α:

$$D_\alpha(X \,||\, Y) = \frac{1}{\alpha - 1} \ln \int \left(\frac{\mu_X(z)}{\mu_Y(z)} \right)^\alpha \mu_Y(z) dz \,.$$

- Conversion to standard (ϵ, δ)-DP: (α, ϵ)-RDP implies $(\epsilon + \frac{\ln(1/\delta)}{\alpha - 1}, \delta)$-DP for any $\delta \in (0, 1)$

- RDP is robust to auxiliary knowledge, as seen by its Bayesian interpretation:
  - Consider an adversary who seeks to infer whether the dataset is $\mathcal{D}$ or $\mathcal{D}'$

PROPERTIES OF RDP

- RDP is robust to auxiliary knowledge, as seen by its Bayesian interpretation:
  - Consider an adversary who seeks to infer whether the dataset is $\mathcal{D}$ or $\mathcal{D}'$
  - The adversary has prior knowledge p and observes $X \sim \mathcal{A}(\mathcal{D})$

PROPERTIES OF RDP

- RDP is robust to auxiliary knowledge, as seen by its Bayesian interpretation:
  - Consider an adversary who seeks to infer whether the dataset is $\mathcal{D}$ or $\mathcal{D}'$
  - The adversary has prior knowledge p and observes $X \sim \mathcal{A}(\mathcal{D})$
  - Let the r.v. $R_{prior} = \frac{p(\mathcal{D}')}{p(\mathcal{D})}$ and $R_{post} = \frac{p(\mathcal{D}'|X)}{p(\mathcal{D}|X)} = \frac{p(X|\mathcal{D}')p(\mathcal{D}')}{p(X|\mathcal{D})p(\mathcal{D})}$ for $X \sim \mathcal{A}(\mathcal{D})$

PROPERTIES OF RDP

- RDP is robust to auxiliary knowledge, as seen by its Bayesian interpretation:
  - Consider an adversary who seeks to infer whether the dataset is $\mathcal{D}$ or $\mathcal{D}'$
  - The adversary has prior knowledge p and observes $X \sim \mathcal{A}(\mathcal{D})$
  - Let the r.v. $R_{prior} = \frac{p(\mathcal{D}')}{p(\mathcal{D})}$ and $R_{post} = \frac{p(\mathcal{D}'|X)}{p(\mathcal{D}|X)} = \frac{p(X|\mathcal{D}')p(\mathcal{D}')}{p(X|\mathcal{D})p(\mathcal{D})}$ for $X \sim \mathcal{A}(\mathcal{D})$
  - RDP bounds the α-th moment of $\frac{R_{post}}{R_{prior}}$ (for $\alpha \to \infty$, we recover "pure" ϵ-DP)

PROPERTIES OF RDP

- RDP is robust to auxiliary knowledge, as seen by its Bayesian interpretation:
  - Consider an adversary who seeks to infer whether the dataset is $\mathcal{D}$ or $\mathcal{D}'$
  - The adversary has prior knowledge p and observes $X \sim \mathcal{A}(\mathcal{D})$
  - Let the r.v. $R_{prior} = \frac{p(\mathcal{D}')}{p(\mathcal{D})}$ and $R_{post} = \frac{p(\mathcal{D}'|X)}{p(\mathcal{D}|X)} = \frac{p(X|\mathcal{D}')p(\mathcal{D}')}{p(X|\mathcal{D})p(\mathcal{D})}$ for $X \sim \mathcal{A}(\mathcal{D})$
  - RDP bounds the α-th moment of $\frac{R_{post}}{R_{prior}}$ (for $\alpha \to \infty$, we recover "pure" ϵ-DP)
  - "The adversary does not know much more after observing the output of the algorithm"

- RDP is robust to auxiliary knowledge, as seen by its Bayesian interpretation:
  - Consider an adversary who seeks to infer whether the dataset is $\mathcal{D}$ or $\mathcal{D}'$
  - The adversary has prior knowledge p and observes $X \sim \mathcal{A}(\mathcal{D})$
  - Let the r.v. $R_{prior} = \frac{p(\mathcal{D}')}{p(\mathcal{D})}$ and $R_{post} = \frac{p(\mathcal{D}'|X)}{p(\mathcal{D}|X)} = \frac{p(X|\mathcal{D}')p(\mathcal{D}')}{p(X|\mathcal{D})p(\mathcal{D})}$ for $X \sim \mathcal{A}(\mathcal{D})$
  - RDP bounds the α-th moment of $\frac{R_{post}}{R_{prior}}$ (for $\alpha \to \infty$, we recover "pure" ϵ-DP)
  - "The adversary does not know much more after observing the output of the algorithm"
- Immunity to post-processing: for any g, if $\mathcal{A}(\cdot)$ is (α, ϵ)-RDP, then so is $g(\mathcal{A}(\cdot))$

- RDP is robust to auxiliary knowledge, as seen by its Bayesian interpretation:
  - Consider an adversary who seeks to infer whether the dataset is $\mathcal{D}$ or $\mathcal{D}'$
  - The adversary has prior knowledge p and observes $X \sim \mathcal{A}(\mathcal{D})$
  - Let the r.v. $R_{prior} = \frac{p(\mathcal{D}')}{p(\mathcal{D})}$ and $R_{post} = \frac{p(\mathcal{D}'|X)}{p(\mathcal{D}|X)} = \frac{p(X|\mathcal{D}')p(\mathcal{D}')}{p(X|\mathcal{D})p(\mathcal{D})}$ for $X \sim \mathcal{A}(\mathcal{D})$
  - RDP bounds the α-th moment of $\frac{R_{post}}{R_{prior}}$ (for $\alpha \rightarrow \infty$, we recover "pure" ϵ-DP)
  - "The adversary does not know much more after observing the output of the algorithm"
- Immunity to post-processing: for any g, if $\mathcal{A}(\cdot)$ is (α, ϵ)-RDP, then so is $g(\mathcal{A}(\cdot))$
- Composition: if $\mathcal{A}_1$ is (α, ϵ_1)-RDP and $\mathcal{A}_2$ is (α, ϵ_2)-RDP, then $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ is $(\alpha, \epsilon_1 + \epsilon_2)$-RDP $\rightarrow$ simpler and tighter than composition for (ϵ, δ)-DP

ENFORCING RDP WITH THE GAUSSIAN MECHANISM

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

ENFORCING RDP WITH THE GAUSSIAN MECHANISM

- Consider f taking as input a dataset and returning a p-dimensional real vector
- Denote its sensitivity by $\Delta = \max_{\mathcal{D} \sim \mathcal{D}'} \|f(\mathcal{D}) - f(\mathcal{D}')\|_2$

- Consider f taking as input a dataset and returning a p-dimensional real vector
- Denote its sensitivity by $\Delta = \max_{\mathcal{D}\sim\mathcal{D}'} \|f(\mathcal{D}) - f(\mathcal{D}')\|_2$

Theorem (Gaussian mechanism)

Let $\sigma > 0$. The algorithm $\mathcal{A}(\cdot) = f(\cdot) + \mathcal{N}(0, \sigma^2\Delta^2)$ satisfies $(\alpha, \frac{\alpha}{2\sigma^2})$-RDP for any $\alpha > 1$.

- DP induces a privacy-utility trade-off, here in terms of the variance of the estimate

- Consider f taking as input a dataset and returning a p-dimensional real vector
- Denote its sensitivity by $\Delta = \max_{\mathcal{D}\sim\mathcal{D}'} \|f(\mathcal{D}) - f(\mathcal{D}')\|_2$

Theorem (Gaussian mechanism)

Let $\sigma > 0$. The algorithm $\mathcal{A}(\cdot) = f(\cdot) + \mathcal{N}(0, \sigma^2\Delta^2)$ satisfies $(\alpha, \frac{\alpha}{2\sigma^2})$-RDP for any $\alpha > 1$.

Theorem (Subsampled Gaussian mechanism, informal)

If $\mathcal{A}$ is executed on a random fraction q of $\mathcal{D}$, then it satisfies $(\alpha, \frac{q^2\alpha}{2\sigma^2})$-RDP.

- DP induces a privacy-utility trade-off, here in terms of the variance of the estimate
- Random subsampling amplifies privacy guarantees

- A trusted curator wants to privately release a model trained on data $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n$
- We focus here on approximately solving an Empirical Risk Minimization (ERM) problem under a DP constraint:

$$\min_{\theta \in \mathbb{R}^p} \left\{ F(\theta; \mathcal{D}) := \frac{1}{n} \sum_{i=1}^{n} \ell(\theta; x_i, y_i) \right\}, \quad \text{where loss } \ell \text{ is differentiable in } \theta$$

- Note: in some cases, DP implies generalization [Bassily et al., 2016, Jung et al., 2021]

Algorithm Differentially Private SGD (DP-SGD) [Bassily et al., 2014, Abadi et al., 2016]

Initialize $\theta^{(0)} \in \mathbb{R}^p$ (must be independent of $\mathcal{D}$)
for $t = 0, \ldots, T-1$ **do**
Pick $i_t \in \{1, \ldots, n\}$ uniformly at random
$\eta^{(t)} \leftarrow (\eta_1^{(t)}, \ldots, \eta_p^{(t)}) \in \mathbb{R}^p$ where each $\eta_j^{(t)} \sim \mathcal{N}(0, \sigma^2\Delta^2)$
$\theta^{(t+1)} \leftarrow \theta^{(t)} - \gamma^{(t)}\big(\nabla\ell(\theta^{(t)}; x_{i_t}, y_{i_t}) + \eta^{(t)}\big)$
Return $\theta^{(T)}$

- The sensitivity $\Delta = \sup_\theta \sup_{x,y,x',y'} \|\nabla\ell(\theta^{(t)}; x, y) - \nabla\ell(\theta^{(t)}; x', y')\|$ can be controlled by assuming $\ell(\cdot; x, y)$ Lipschitz for all x, y, or using gradient clipping [Abadi et al., 2016]

- Utility analysis: same as non-private SGD (with additional noise due to privacy)

PRIVACY-UTILITY TRADE-OFF OF DP-SGD

- Utility analysis: same as non-private SGD (with additional noise due to privacy)
- Privacy analysis: DP-SGD is $(\alpha, \frac{\alpha T}{2n^2\sigma^2})$ by subsampled Gaussian mechanism + composition over T iterations

- Utility analysis: same as non-private SGD (with additional noise due to privacy)
- Privacy analysis: DP-SGD is $(\alpha, \frac{\alpha T}{2n^2\sigma^2})$ by subsampled Gaussian mechanism + composition over T iterations
- Setting σ^2 to satisfy (ϵ, δ)-DP and choosing T to balance optimization and privacy errors, we get the following suboptimality gap:

Convex, Lipschitz, smooth loss	$\tilde{O}\left(\frac{\sqrt{p}\ln(1/\delta)}{n\epsilon}\right)$
Convex, Lipschitz, smooth loss, strongly convex F	$\tilde{O}\left(\frac{p\ln(1/\delta)}{n^2\epsilon^2}\right)$

PRIVACY-UTILITY TRADE-OFF OF DP-SGD

- Utility analysis: same as non-private SGD (with additional noise due to privacy)
- Privacy analysis: DP-SGD is $(\alpha, \frac{\alpha T}{2n^2\sigma^2})$ by subsampled Gaussian mechanism + composition over T iterations
- Setting σ^2 to satisfy (ϵ, δ)-DP and choosing T to balance optimization and privacy errors, we get the following suboptimality gap:

Convex, Lipschitz, smooth loss	$\tilde{O}\left(\frac{\sqrt{p}\ln(1/\delta)}{n\epsilon}\right)$
Convex, Lipschitz, smooth loss, strongly convex F	$\tilde{O}\left(\frac{p\ln(1/\delta)}{n^2\epsilon^2}\right)$

- This is optimal [Bassily et al., 2014]: cannot do better without additional assumptions

REMOVING THE TRUSTED CURATOR: LOCAL DP

- So far we considered the central DP model, which relies on a trusted curator to collect and process raw data → the output $\mathcal{A}(\mathcal{D})$ is only the final result

REMOVING THE TRUSTED CURATOR: LOCAL DP

- So far we considered the central DP model, which relies on a trusted curator to collect and process raw data $\rightarrow$ the output $\mathcal{A}(\mathcal{D})$ is only the final result
- Central DP is good for utility but is an unrealistic trust model in applications where many parties contribute sensitive data, as in federated learning

REMOVING THE TRUSTED CURATOR: LOCAL DP

- So far we considered the central DP model, which relies on a trusted curator to collect and process raw data $\rightarrow$ the output $\mathcal{A}(\mathcal{D})$ is only the final result
- Central DP is good for utility but is an unrealistic trust model in applications where many parties contribute sensitive data, as in federated learning
- Instead we can consider for local DP, where each party must locally randomize its contributions $\rightarrow$ the output of $\mathcal{A}(\mathcal{D})$ consists of all messages sent by all parties

REMOVING THE TRUSTED CURATOR: LOCAL DP

- So far we considered the central DP model, which relies on a trusted curator to collect and process raw data $\rightarrow$ the output $\mathcal{A}(\mathcal{D})$ is only the final result
- Central DP is good for utility but is an unrealistic trust model in applications where many parties contribute sensitive data, as in federated learning
- Instead we can consider for local DP, where each party must locally randomize its contributions $\rightarrow$ the output of $\mathcal{A}(\mathcal{D})$ consists of all messages sent by all parties
- Unfortunately local DP induces a large cost in utility: for averaging n private p-dimensional values in ball of radius Δ under (α, ϵ)-RDP, we have

$$\mathbb{E}[\|x^{\mathrm{out}} - \bar{x}\|^2] = \Theta\Big(\frac{\alpha p \Delta^2}{n\epsilon}\Big) \text{ for local DP}, \quad \text{and} \quad \mathbb{E}[\|x^{\mathrm{out}} - \bar{x}\|^2] = \Theta\Big(\frac{\alpha p \Delta^2}{n^2\epsilon}\Big) \text{ for central DP}$$

REMOVING THE TRUSTED CURATOR: LOCAL DP

- So far we considered the central DP model, which relies on a trusted curator to collect and process raw data $\rightarrow$ the output $\mathcal{A}(\mathcal{D})$ is only the final result
- Central DP is good for utility but is an unrealistic trust model in applications where many parties contribute sensitive data, as in federated learning
- Instead we can consider for local DP, where each party must locally randomize its contributions $\rightarrow$ the output of $\mathcal{A}(\mathcal{D})$ consists of all messages sent by all parties
- Unfortunately local DP induces a large cost in utility: for averaging n private p-dimensional values in ball of radius Δ under (α, ϵ)-RDP, we have

$$\mathbb{E}[\|x^{\text{out}} - \bar{x}\|^2] = \Theta\Big(\frac{\alpha p \Delta^2}{n\epsilon}\Big) \text{ for local DP}, \quad \text{and} \quad \mathbb{E}[\|x^{\text{out}} - \bar{x}\|^2] = \Theta\Big(\frac{\alpha p \Delta^2}{n^2\epsilon}\Big) \text{ for central DP}$$

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

A relaxation of local DP for decentralized algorithms

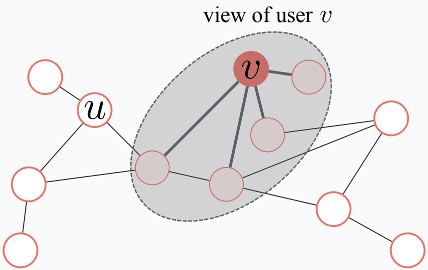
- A connected graph $G = (\mathcal{V}, \mathcal{E})$ on a set of $|\mathcal{V}| = n$ users (nodes)

- A connected graph $G = (\mathcal{V}, \mathcal{E})$ on a set of $|\mathcal{V}| = n$ users (nodes)
- Each user $v \in \mathcal{V}$ holds a local dataset $\mathcal{D}_v$

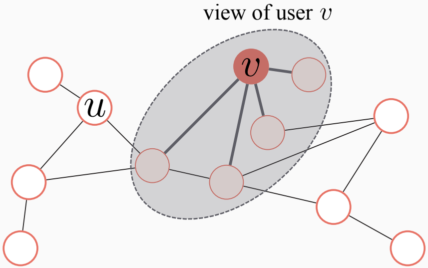
DECENTRALIZED ALGORITHMS

- A connected graph $G = (\mathcal{V}, \mathcal{E})$ on a set of $|\mathcal{V}| = n$ users (nodes)
- Each user $v \in \mathcal{V}$ holds a local dataset $\mathcal{D}_v$
- A decentralized algorithm relies only on communication along the edges $\mathcal{E}$ of G

- A connected graph $G = (\mathcal{V}, \mathcal{E})$ on a set of $|\mathcal{V}| = n$ users (nodes)
- Each user $v \in \mathcal{V}$ holds a local dataset $\mathcal{D}_v$
- A decentralized algorithm relies only on communication along the edges $\mathcal{E}$ of G
- Each user v thus has a limited view: it only observes the messages that it receives



- A connected graph $G = (\mathcal{V}, \mathcal{E})$ on a set of $|\mathcal{V}| = n$ users (nodes)
- Each user $v \in \mathcal{V}$ holds a local dataset $\mathcal{D}_v$
- A decentralized algorithm relies only on communication along the edges $\mathcal{E}$ of G
- Each user v thus has a limited view: it only observes the messages that it receives



- We want to use this to prove stronger privacy guarantees than under local DP

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

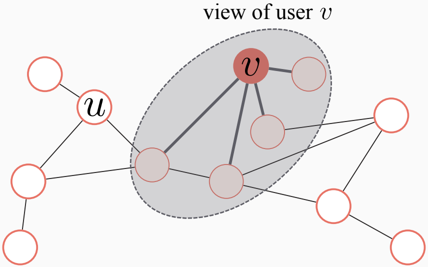
- Let $\mathcal{O}_v$ be the set of messages sent and received by party v
- Denote by $\mathcal{D} \sim_u \mathcal{D}'$ two datasets $\mathcal{D} = (\mathcal{D}_1, \ldots, \mathcal{D}_u, \ldots, \mathcal{D}_n)$ and $\mathcal{D}' = (\mathcal{D}_1, \ldots, \mathcal{D}'_u, \ldots, \mathcal{D}_n)$ that differ only in the local dataset of user u

- Let $\mathcal{O}_v$ be the set of messages sent and received by party v
- Denote by $\mathcal{D} \sim_u \mathcal{D}'$ two datasets $\mathcal{D} = (\mathcal{D}_1, \ldots, \mathcal{D}_u, \ldots, \mathcal{D}_n)$ and $\mathcal{D}' = (\mathcal{D}_1, \ldots, \mathcal{D}'_u, \ldots, \mathcal{D}_n)$ that differ only in the local dataset of user u

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

An algorithm $\mathcal{A}$ satisfies (α, ϵ)-Network DP (NDP) if for all pairs of distinct users $u, v \in \mathcal{V}$ and neighboring datasets $\mathcal{D} \sim_u \mathcal{D}'$:

$$D_\alpha(\mathcal{O}_v(\mathcal{A}(\mathcal{D})) \,||\, \mathcal{O}_v(\mathcal{A}(\mathcal{D}'))) \leq \epsilon\,.$$



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

- We will also consider privacy guarantees that are specific to each pair of nodes, rather than uniform over all pairs

Definition (Pairwise Network DP [Cyffers et al., 2022])

For $f : \mathcal{V} \times \mathcal{V} \to \mathbb{R}^+$, an algorithm $\mathcal{A}$ satisfies (α, f)-Pairwise Network DP (PNDP) if for all pairs of distinct users $u, v \in \mathcal{V}$ and neighboring datasets $\mathcal{D} \sim_u \mathcal{D}'$:

$$D_\alpha(\mathcal{O}_v(\mathcal{A}(\mathcal{D})) \,||\, \mathcal{O}_v(\mathcal{A}(\mathcal{D}'))) \leq f(u, v) \,. \tag{2}$$

- We will also consider privacy guarantees that are specific to each pair of nodes, rather than uniform over all pairs

Definition (Pairwise Network DP [Cyffers et al., 2022])

For $f : \mathcal{V} \times \mathcal{V} \to \mathbb{R}^+$, an algorithm $\mathcal{A}$ satisfies (α, f)-Pairwise Network DP (PNDP) if for all pairs of distinct users $u, v \in \mathcal{V}$ and neighboring datasets $\mathcal{D} \sim_u \mathcal{D}'$:

$$D_\alpha\left(\mathcal{O}_v(\mathcal{A}(\mathcal{D})) \,||\, \mathcal{O}_v(\mathcal{A}(\mathcal{D}'))\right) \leq f(u, v)\,. \tag{2}$$

- For comparison with central and local DP baselines, we will report the mean privacy loss $\bar{\varepsilon}_v = \frac{1}{n} \sum_{u \in \mathcal{V} \setminus \{v\}} f(u, v)$ under the constraint $\bar{\varepsilon} = \max_{v \in \mathcal{V}} \bar{\varepsilon}_v \leq \epsilon$

- We will also consider privacy guarantees that are specific to each pair of nodes, rather than uniform over all pairs

Definition (Pairwise Network DP [Cyffers et al., 2022])

For $f : \mathcal{V} \times \mathcal{V} \to \mathbb{R}^+$, an algorithm $\mathcal{A}$ satisfies (α, f)-Pairwise Network DP (PNDP) if for all pairs of distinct users $u, v \in \mathcal{V}$ and neighboring datasets $\mathcal{D} \sim_u \mathcal{D}'$:

$$D_\alpha(\mathcal{O}_v(\mathcal{A}(\mathcal{D})) \,||\, \mathcal{O}_v(\mathcal{A}(\mathcal{D}'))) \leq f(u, v) \,. \tag{2}$$

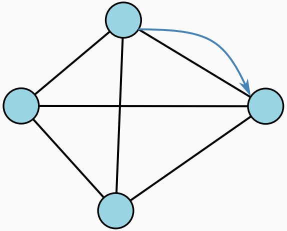
- For comparison with central and local DP baselines, we will report the mean privacy loss $\bar{\varepsilon}_v = \frac{1}{n} \sum_{u \in \mathcal{V} \setminus \{v\}} f(u, v)$ under the constraint $\bar{\varepsilon} = \max_{v \in \mathcal{V}} \bar{\varepsilon}_v \leq \epsilon$
- Note: $\bar{\varepsilon}_v$ is not a proper privacy guarantee (we simply use it to summarize our gains)

Private random walk-based decentralized SGD

- Consider the standard objective $F(\theta; \mathcal{D}) = \frac{1}{n} \sum_{v=1}^{n} F_v(\theta; \mathcal{D}_v)$

- Consider the standard objective $F(\theta; \mathcal{D}) = \frac{1}{n} \sum_{v=1}^{n} F_v(\theta; \mathcal{D}_v)$
- We consider a decentralized SGD algorithm where the model is updated sequentially by following a random walk, aka incremental gradient [Johansson et al., 2009]

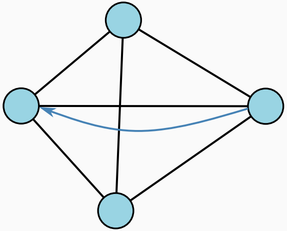
- Consider the standard objective $F(\theta; \mathcal{D}) = \frac{1}{n} \sum_{v=1}^{n} F_v(\theta; \mathcal{D}_v)$
- We consider a decentralized SGD algorithm where the model is updated sequentially by following a random walk, aka incremental gradient [Johansson et al., 2009]
- We focus here on the complete graph



Algorithm Private random walk-based SGD [Cyffers and Bellet, 2022]

Initialize $\theta \in \mathbb{R}^p$
for $t = 1$ to T **do**
  Draw random user $v \sim \mathcal{U}(1, \ldots, n)$
  $\eta = [\eta_1, \ldots, \eta_p]$, where each $\eta_j \sim \mathcal{N}(0, \sigma^2\Delta^2)$
  $\theta \leftarrow \theta - \gamma[\nabla_\theta F_v(\theta; \mathcal{D}_v) + \eta]$
return θ

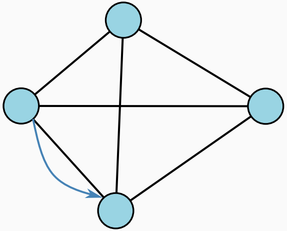
- Consider the standard objective $F(\theta; \mathcal{D}) = \frac{1}{n} \sum_{v=1}^{n} F_v(\theta; \mathcal{D}_v)$
- We consider a decentralized SGD algorithm where the model is updated sequentially by following a random walk, aka incremental gradient [Johansson et al., 2009]
- We focus here on the complete graph



Algorithm Private random walk-based SGD [Cyffers and Bellet, 2022]

Initialize $\theta \in \mathbb{R}^p$
for $t = 1$ to T **do**
  Draw random user $v \sim \mathcal{U}(1, \ldots, n)$
  $\eta = [\eta_1, \ldots, \eta_p]$, where each $\eta_j \sim \mathcal{N}(0, \sigma^2 \Delta^2)$
  $\theta \leftarrow \theta - \gamma[\nabla_\theta F_v(\theta; \mathcal{D}_v) + \eta]$
return θ

- Consider the standard objective $F(\theta; \mathcal{D}) = \frac{1}{n}\sum_{v=1}^{n} F_v(\theta; \mathcal{D}_v)$
- We consider a decentralized SGD algorithm where the model is updated sequentially by following a random walk, aka incremental gradient [Johansson et al., 2009]
- We focus here on the complete graph



Algorithm Private random walk-based SGD [Cyffers and Bellet, 2022]

Initialize $\theta \in \mathbb{R}^p$
for $t = 1$ to T **do**
  Draw random user $v \sim \mathcal{U}(1, \ldots, n)$
  $\eta = [\eta_1, \ldots, \eta_p]$, where each $\eta_j \sim \mathcal{N}(0, \sigma^2\Delta^2)$
  $\theta \leftarrow \theta - \gamma[\nabla_\theta F_v(\theta; \mathcal{D}_v) + \eta]$
return θ

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

Let $F_1(\cdot;\mathcal{D}_1), \ldots, F_n(\cdot;\mathcal{D}_n)$ be convex and smooth. Given $\alpha > 1$, $\epsilon > 0$, let $T = \tilde{\Omega}(n^2)$ and σ^2 be such that private random walk-based decentralized SGD on the complete graph satisfies (α, ϵ)-local RDP. Then the algorithm also satisfies $(\alpha, \frac{\ln^2 n}{n}\epsilon)$-network DP.

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

Let $F_1(\cdot; \mathcal{D}_1), \ldots, F_n(\cdot; \mathcal{D}_n)$ be convex and smooth. Given $\alpha > 1$, $\epsilon > 0$, let $T = \tilde{\Omega}(n^2)$ and σ^2 be such that private random walk-based decentralized SGD on the complete graph satisfies (α, ϵ)-local RDP. Then the algorithm also satisfies $(\alpha, \frac{\ln^2 n}{n}\epsilon)$-network DP.

- In other words, accounting for the limited view in decentralized algorithms allows to recover the privacy-utility trade-off of DP-SGD under central DP! (up to a log factor)

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

Let $F_1(\cdot; \mathcal{D}_1), \ldots, F_n(\cdot; \mathcal{D}_n)$ be convex and smooth. Given $\alpha > 1$, $\epsilon > 0$, let $T = \tilde{\Omega}(n^2)$ and σ^2 be such that private random walk-based decentralized SGD on the complete graph satisfies (α, ϵ)-local RDP. Then the algorithm also satisfies $(\alpha, \frac{\ln^2 n}{n}\epsilon)$-network DP.

- In other words, accounting for the limited view in decentralized algorithms allows to recover the privacy-utility trade-off of DP-SGD under central DP! (up to a log factor)
- Note: for $T = o(n^2)$, the amplification effect is still strong and can be computed numerically, see [Cyffers and Bellet, 2022]

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

Let $F_1(\cdot; \mathcal{D}_1), \dots, F_n(\cdot; \mathcal{D}_n)$ be convex and smooth. Given $\alpha > 1$, $\epsilon > 0$, let $T = \tilde{\Omega}(n^2)$ and σ^2 be such that private random walk-based decentralized SGD on the complete graph satisfies (α, ϵ)-local RDP. Then the algorithm also satisfies $(\alpha, \frac{\ln^2 n}{n}\epsilon)$-network DP.

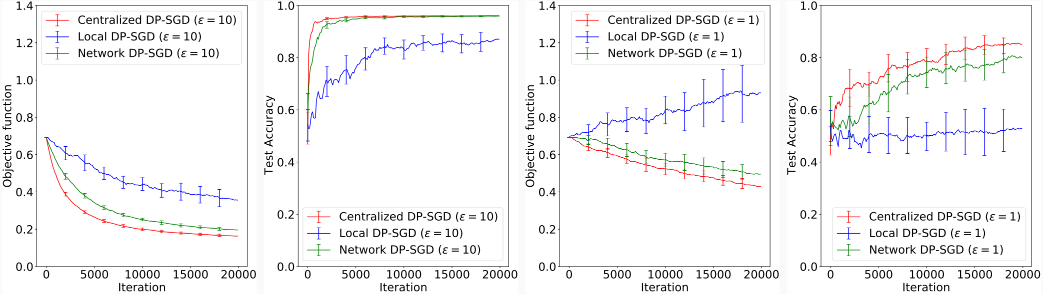
- In other words, accounting for the limited view in decentralized algorithms allows to recover the privacy-utility trade-off of DP-SGD under central DP! (up to a log factor)
- Note: for $T = o(n^2)$, the amplification effect is still strong and can be computed numerically, see [Cyffers and Bellet, 2022]
- Utility analysis: same as DP-SGD!

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

Let $F_1(\cdot; \mathcal{D}_1), \ldots, F_n(\cdot; \mathcal{D}_n)$ be convex and smooth. Given $\alpha > 1$, $\epsilon > 0$, let $T = \tilde{\Omega}(n^2)$ and σ^2 be such that private random walk-based decentralized SGD on the complete graph satisfies (α, ϵ)-local RDP. Then the algorithm also satisfies $(\alpha, \frac{\ln^2 n}{n}\epsilon)$-network DP.

- In other words, accounting for the limited view in decentralized algorithms allows to recover the privacy-utility trade-off of DP-SGD under central DP! (up to a log factor)
- Note: for $T = o(n^2)$, the amplification effect is still strong and can be computed numerically, see [Cyffers and Bellet, 2022]
- Utility analysis: same as DP-SGD!
- Privacy analysis: leverages privacy amplification by iteration [Feldman et al., 2018] and exploits the randomness of the walk through “weak convexity” of Rényi divergence

EMPIRICAL ILLUSTRATION



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

Private gossip-based decentralized SGD

GOSSIP-BASED DECENTRALIZED SGD

- Random walk-based SGD is sequential (no parallel computation)

GOSSIP-BASED DECENTRALIZED SGD

- Random walk-based SGD is sequential (no parallel computation)
- A popular parallel alternative is gossip-based decentralized SGD [Lian et al., 2017] [Koloskova et al., 2020], which builds upon gossip averaging [Boyd et al., 2006]

GOSSIP-BASED DECENTRALIZED SGD

- Random walk-based SGD is sequential (no parallel computation)
- A popular parallel alternative is gossip-based decentralized SGD [Lian et al., 2017] [Koloskova et al., 2020], which builds upon gossip averaging [Boyd et al., 2006]
- A gossip matrix over the graph $G = (\mathcal{V}, \mathcal{E})$ is a matrix $W \in \mathbb{R}^{n \times n}$ which:
  - is symmetric with nonnegative entries
  - is stochastic, i.e., $W\mathbb{1} = \mathbb{1}$
  - for any $v, w \in \mathcal{V}$, $W_{v,w} > 0$ implies $\{v, w\} \in \mathcal{E}$ or $v = w$

- Random walk-based SGD is sequential (no parallel computation)
- A popular parallel alternative is gossip-based decentralized SGD [Lian et al., 2017] [Koloskova et al., 2020], which builds upon gossip averaging [Boyd et al., 2006]
- A gossip matrix over the graph $G = (\mathcal{V}, \mathcal{E})$ is a matrix $W \in \mathbb{R}^{n \times n}$ which:
  - is symmetric with nonnegative entries
  - is stochastic, i.e., $W\mathbb{1} = \mathbb{1}$
  - for any $v, w \in \mathcal{V}$, $W_{v,w} > 0$ implies $\{v, w\} \in \mathcal{E}$ or $v = w$

Algorithm GOSSIP_AVERAGING$(\{x_v\}_{v\in\mathcal{V}}, W, K)$ [Boyd et al., 2006]

for all nodes v in parallel **do**
  $x_v^0 \leftarrow x_v$
for $k = 0$ to $K - 1$ **do**
  for all nodes v in parallel **do**
    $x_v^{k+1} \leftarrow \sum_{w\in\mathcal{N}_v} W_{v,w} x_w^k$, where $\mathcal{N}_v = \{w : W_{v,w} > 0\}$
return $x_1^K, \ldots, x_n^K$

- Consider again $F(\theta; \mathcal{D}) = \frac{1}{n} \sum_{v=1}^{n} F_v(\theta; \mathcal{D}_v)$ with $F_v(\theta; \mathcal{D}_v) = \frac{1}{|\mathcal{D}_v|} \sum_{(x_v, y_v) \in \mathcal{D}_v} \ell(\theta; x_v, y_v)$

- Consider again $F(\theta;\mathcal{D}) = \frac{1}{n}\sum_{v=1}^{n} F_v(\theta;\mathcal{D}_v)$ with $F_v(\theta;\mathcal{D}_v) = \frac{1}{|\mathcal{D}_v|}\sum_{(x_v,y_v)\in\mathcal{D}_v} \ell(\theta;x_v,y_v)$

Algorithm Gossip-based decentralized SGD [Lian et al., 2017, Koloskova et al., 2020]

Initialize $\theta_1^{(0)},\dots,\theta_n^{(0)} \in \mathbb{R}^p$
for $t = 0$ to $T-1$ **do**
  for all nodes v in parallel **do**
    $\hat{\theta}_v^t \leftarrow \theta_v^t - \gamma\nabla_\theta\ell(\theta_v^t;x_v^t,y_v^t)$ where $(x_v^t,y_v^t)\sim\mathcal{D}_v$
  $\theta_v^{t+1} \leftarrow \textsc{Gossip\_Averaging}\big(\{\hat{\theta}_v^t\}_{v\in\mathcal{V}}, W, K\big)$
return $\theta_1^T,\dots,\theta_n^T$

- Consider again $F(\theta; \mathcal{D}) = \frac{1}{n}\sum_{v=1}^{n} F_v(\theta; \mathcal{D}_v)$ with $F_v(\theta; \mathcal{D}_v) = \frac{1}{|\mathcal{D}_v|}\sum_{(x_v,y_v)\in\mathcal{D}_v} \ell(\theta; x_v, y_v)$

Algorithm Gossip-based decentralized SGD [Lian et al., 2017, Koloskova et al., 2020]

Initialize $\theta_1^{(0)}, \dots, \theta_n^{(0)} \in \mathbb{R}^p$
for $t = 0$ to $T - 1$ **do**
  for all nodes v in parallel **do**
    $\hat{\theta}_v^t \leftarrow \theta_v^t - \gamma \nabla_\theta \ell(\theta_v^t; x_v^t, y_v^t)$ where $(x_v^t, y_v^t) \sim \mathcal{D}_v$
  $\theta_v^{t+1} \leftarrow \textsc{Gossip\_Averaging}\big(\{\hat{\theta}_v^t\}_{v\in\mathcal{V}}, W, K\big)$
return $\theta_1^T, \dots, \theta_n^T$

- Note: to improve the dependence on the topology in the convergence rate we actually use accelerated gossip [Berthier et al., 2020]

- To make the algorithm private, we simply add Gaussian noise before gossiping

Algorithm PRIVATE_GOSSIP_AVERAGING$\left(\{x_v\}_{v\in\mathcal{V}}, W, K, \sigma^2\right)$

for all nodes v in parallel **do**
  $\tilde{x}_v^0 \leftarrow x_v + \eta_v$ where $\eta_v \sim \mathcal{N}(0, \sigma^2)$
$x_1^K, \ldots, x_n^K \leftarrow$ GOSSIP_AVERAGING$\left(\{\tilde{x}_v^0\}_{v\in\mathcal{V}}, W, K\right)$
return $x_1^K, \ldots, x_n^K$

- To make the algorithm private, we simply add Gaussian noise before gossiping

Algorithm PRIVATE_GOSSIP_AVERAGING$\left(\{x_v\}_{v\in\mathcal{V}}, W, K, \sigma^2\right)$

for all nodes v in parallel **do**
  $\tilde{x}_v^0 \leftarrow x_v + \eta_v$ where $\eta_v \sim \mathcal{N}(0, \sigma^2)$
$x_1^K, \dots, x_n^K \leftarrow$ GOSSIP_AVERAGING$\left(\{\tilde{x}_v^0\}_{v\in\mathcal{V}}, W, K\right)$
return $x_1^K, \dots, x_n^K$

Algorithm Private gossip-based decentralized SGD [Cyffers et al., 2022]

Initialize $\theta_1^{(0)}, \dots, \theta_n^{(0)} \in \mathbb{R}^p$
for $t = 0$ to $T - 1$ **do**
  for all nodes v in parallel **do**
    $\hat{\theta}_v^t \leftarrow \theta_v^t - \gamma \nabla_\theta \ell(\theta_v^t; x_v^t, y_v^t)$ where $(x_v^t, y_v^t) \sim \mathcal{D}_v$
  $\theta_v^{t+1} \leftarrow$ PRIVATE_GOSSIP_AVERAGING$\left(\{\hat{\theta}_v^t\}_{v\in\mathcal{V}}, W, K, \gamma^2\sigma^2\Delta^2\right)$
return $\theta_1^T, \dots, \theta_n^T$

Theorem ([Cyffers et al., 2022])

After K iterations, Private Gossip Averaging is (α, f)-PNDP with

$$f(u,v) = \frac{\alpha\Delta^2}{2\sigma^2}\sum_{k=0}^{K-1}\sum_{w:\{v,w\}\in\mathcal{E}}\frac{(W^k)^2_{u,w}}{\|(W^k)_{w,:}\|^2}$$
$$\leq \frac{\alpha\Delta^2 n}{2\sigma^2}\max_{\{v,w\}\in\mathcal{E}} W^{-2}_{v,w}\sum_{k=1}^{K}\mathbb{P}(X^k = v | X^0 = u)^2,$$

where $(X^k)_k$ is the random walk on graph G, with transitions W.

- As desired, this exhibits the fact that, for two nodes u and v, privacy guarantees improve with their “distance” in the graph

PRIVACY-UTILITY TRADE-OFF OF PRIVATE GOSSIP AVERAGING

- Recall central DP achieves $O(\frac{\alpha p \Delta^2}{n^2 \epsilon})$ and local DP achieves $O(\frac{\alpha p \Delta^2}{n \epsilon})$

PRIVACY-UTILITY TRADE-OFF OF PRIVATE GOSSIP AVERAGING

- Recall central DP achieves $O(\frac{\alpha p \Delta^2}{n^2 \epsilon})$ and local DP achieves $O(\frac{\alpha p \Delta^2}{n \epsilon})$
- Setting the mean privacy loss $\bar{\varepsilon}_v = \frac{1}{n} \sum_{u \in \mathcal{V} \setminus \{v\}} f(u, v)$ to satisfy $\bar{\varepsilon} = \max_{v \in \mathcal{V}} \bar{\varepsilon}_v \leq \epsilon$, for private gossip averaging we get (ignoring log terms):

Graph	Arbitrary
Utility (MSE)	$\frac{\alpha p \Delta^2 d}{n^2 \epsilon \sqrt{\lambda_W}}$

- We match the utility of central DP up to an additional $d/\sqrt{\lambda_W}$ factor, where d is the max degree and λ_W of the spectral gap of W

PRIVACY-UTILITY TRADE-OFF OF PRIVATE GOSSIP AVERAGING

- Recall central DP achieves $O(\frac{\alpha p \Delta^2}{n^2 \epsilon})$ and local DP achieves $O(\frac{\alpha p \Delta^2}{n \epsilon})$
- Setting the mean privacy loss $\overline{\varepsilon}_v = \frac{1}{n} \sum_{u \in \mathcal{V} \setminus \{v\}} f(u, v)$ to satisfy $\overline{\varepsilon} = \max_{v \in \mathcal{V}} \overline{\varepsilon}_v \leq \epsilon$, for private gossip averaging we get (ignoring log terms):

Graph	Arbitrary	Complete
Utility (MSE)	$\frac{\alpha p \Delta^2 d}{n^2 \epsilon \sqrt{\lambda_W}}$	$\frac{\alpha p \Delta^2}{n \epsilon}$

- We match the utility of central DP up to an additional $d/\sqrt{\lambda_W}$ factor, where d is the max degree and λ_W of the spectral gap of W

PRIVACY-UTILITY TRADE-OFF OF PRIVATE GOSSIP AVERAGING

- Recall central DP achieves $O(\frac{\alpha p \Delta^2}{n^2 \epsilon})$ and local DP achieves $O(\frac{\alpha p \Delta^2}{n \epsilon})$
- Setting the mean privacy loss $\bar{\varepsilon}_v = \frac{1}{n} \sum_{u \in \mathcal{V} \setminus \{v\}} f(u, v)$ to satisfy $\bar{\varepsilon} = \max_{v \in \mathcal{V}} \bar{\varepsilon}_v \leq \epsilon$, for private gossip averaging we get (ignoring log terms):

Graph	Arbitrary	Complete	Ring
Utility (MSE)	$\frac{\alpha p \Delta^2 d}{n^2 \epsilon \sqrt{\lambda_W}}$	$\frac{\alpha p \Delta^2}{n \epsilon}$	$\frac{\alpha p \Delta^2}{n \epsilon}$

- We match the utility of central DP up to an additional $d / \sqrt{\lambda_W}$ factor, where d is the max degree and λ_W of the spectral gap of W

PRIVACY-UTILITY TRADE-OFF OF PRIVATE GOSSIP AVERAGING

- Recall central DP achieves $O(\frac{\alpha p \Delta^2}{n^2 \epsilon})$ and local DP achieves $O(\frac{\alpha p \Delta^2}{n \epsilon})$
- Setting the mean privacy loss $\bar{\varepsilon}_v = \frac{1}{n} \sum_{u \in \mathcal{V} \setminus \{v\}} f(u, v)$ to satisfy $\bar{\varepsilon} = \max_{v \in \mathcal{V}} \bar{\varepsilon}_v \leq \epsilon$, for private gossip averaging we get (ignoring log terms):

Graph	Arbitrary	Complete	Ring	Expander
Utility (MSE)	$\frac{\alpha p \Delta^2 d}{n^2 \epsilon \sqrt{\lambda_W}}$	$\frac{\alpha p \Delta^2}{n \epsilon}$	$\frac{\alpha p \Delta^2}{n \epsilon}$	$\frac{\alpha p \Delta^2}{n^2 \epsilon}$

- We match the utility of central DP up to an additional $d/\sqrt{\lambda_W}$ factor, where d is the max degree and λ_W of the spectral gap of W
- Some graphs (e.g., expanders) make this constant: we get privacy **and** efficiency!

PRIVACY-UTILITY TRADE-OFF OF PRIVATE GOSSIP AVERAGING

- Recall central DP achieves $O(\frac{\alpha p \Delta^2}{n^2 \epsilon})$ and local DP achieves $O(\frac{\alpha p \Delta^2}{n \epsilon})$
- Setting the mean privacy loss $\bar{\varepsilon}_v = \frac{1}{n} \sum_{u \in \mathcal{V} \setminus \{v\}} f(u, v)$ to satisfy $\bar{\varepsilon} = \max_{v \in \mathcal{V}} \bar{\varepsilon}_v \leq \epsilon$, for private gossip averaging we get (ignoring log terms):

Graph	Arbitrary	Complete	Ring	Expander
Utility (MSE)	$\frac{\alpha p \Delta^2 d}{n^2 \epsilon \sqrt{\lambda_W}}$	$\frac{\alpha p \Delta^2}{n \epsilon}$	$\frac{\alpha p \Delta^2}{n \epsilon}$	$\frac{\alpha p \Delta^2}{n^2 \epsilon}$

- We match the utility of central DP up to an additional $d/\sqrt{\lambda_W}$ factor, where d is the max degree and λ_W of the spectral gap of W
- Some graphs (e.g., expanders) make this constant: we get privacy **and** efficiency!
- Note: we also have extensions to time-varying graphs and randomized gossip

Theorem ([Cyffers et al., 2022])

Let F be μ-strongly convex, F_v be L-smooth and $\mathbb{E}[\|\nabla \ell(\theta^\star; x_v, y_v) - \nabla F(\theta^\star)\|^2] \leq \rho_v^2$. Let $\bar{\rho}^2 = \frac{1}{n}\sum_{v \in \mathcal{V}} \rho_v^2$. For any $\epsilon > 0$, and appropriate choices of T and K, there exists f such that the algorithm is (α, f)-PNDP, with:

$$\forall v \in \mathcal{V}\,, \quad \bar{\varepsilon}_v = \frac{1}{n} \sum_{u \in \mathcal{V} \setminus \{v\}} f(u, v) \leq \epsilon \quad \text{and} \quad \mathbb{E}[F(\bar{\theta}^{1:T}) - F(\theta^\star)] \leq \tilde{\mathcal{O}}\left(\frac{\alpha p \Delta^2 d}{n^2 \mu \epsilon \sqrt{\lambda_W}} + \frac{\bar{\rho}^2}{nL}\right),$$

where d_v is the degree of node v and λ_W is the spectral gap associated with W.

- The term $\frac{\bar{\rho}^2}{nL}$ is privacy-independent and dominated by the first term

Theorem ([Cyffers et al., 2022])

Let F be μ-strongly convex, F_v be L-smooth and $\mathbb{E}[\|\nabla\ell(\theta^\star; x_v, y_v) - \nabla F(\theta^\star)\|^2] \leq \rho_v^2$. Let $\bar{\rho}^2 = \frac{1}{n}\sum_{v\in\mathcal{V}} \rho_v^2$. For any $\epsilon > 0$, and appropriate choices of T and K, there exists f such that the algorithm is (α, f)-PNDP, with:

$$\forall v \in \mathcal{V}, \quad \bar{\varepsilon}_v = \frac{1}{n}\sum_{u\in\mathcal{V}\setminus\{v\}} f(u,v) \leq \epsilon \quad \text{and} \quad \mathbb{E}[F(\bar{\theta}^{1:T}) - F(\theta^\star)] \leq \tilde{\mathcal{O}}\left(\frac{\alpha p \Delta^2 d}{n^2 \mu \epsilon \sqrt{\lambda_W}} + \frac{\bar{\rho}^2}{nL}\right),$$

where d_v is the degree of node v and λ_W is the spectral gap associated with W.

- The term $\frac{\bar{\rho}^2}{nL}$ is privacy-independent and dominated by the first term
- The first term has the same form as before, so same conclusions apply!

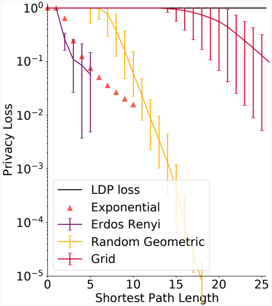
Theorem ([Cyffers et al., 2022])

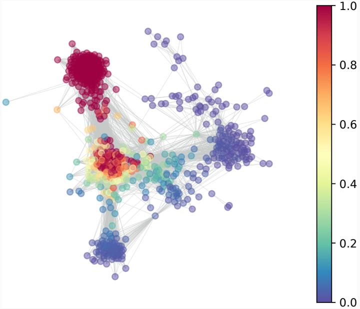
Let F be μ-strongly convex, F_v be L-smooth and $\mathbb{E}[\|\nabla\ell(\theta^\star; x_v, y_v) - \nabla F(\theta^\star)\|^2] \leq \rho_v^2$. Let $\bar{\rho}^2 = \frac{1}{n}\sum_{v\in\mathcal{V}} \rho_v^2$. For any $\epsilon > 0$, and appropriate choices of T and K, there exists f such that the algorithm is (α, f)-PNDP, with:

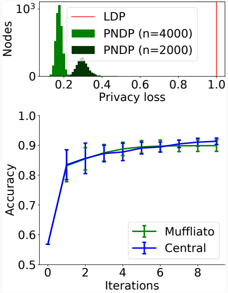
$$\forall v \in \mathcal{V}, \quad \bar{\varepsilon}_v = \frac{1}{n}\sum_{u\in\mathcal{V}\setminus\{v\}} f(u,v) \leq \epsilon \quad \text{and} \quad \mathbb{E}[F(\bar{\theta}^{1:T}) - F(\theta^\star)] \leq \tilde{\mathcal{O}}\left(\frac{\alpha p \Delta^2 d}{n^2 \mu \epsilon \sqrt{\lambda_W}} + \frac{\bar{\rho}^2}{nL}\right),$$

where d_v is the degree of node v and λ_W is the spectral gap associated with W.

- The term $\frac{\bar{\rho}^2}{nL}$ is privacy-independent and dominated by the first term
- The first term has the same form as before, so same conclusions apply!
- In particular, with an expander graph, we match the privacy-utility trade-off of centralized SGD with a trusted curator (up to log terms)

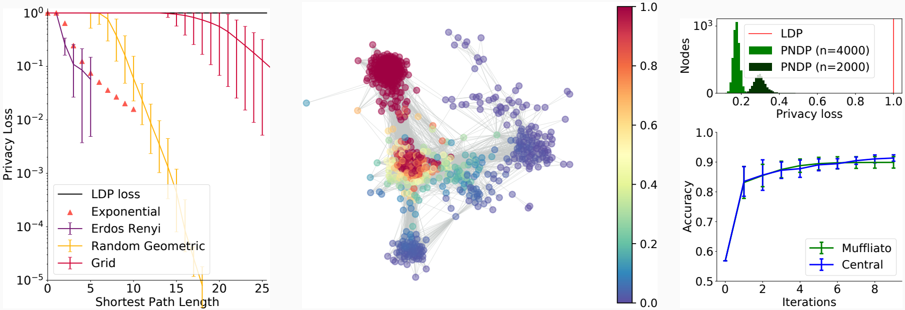




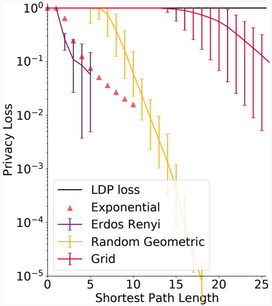


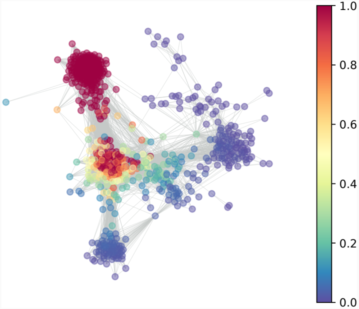
- Users get local DP guarantees w.r.t. their direct neighbors but stronger privacy w.r.t. to other users depending on their distance and the mixing properties of the graph

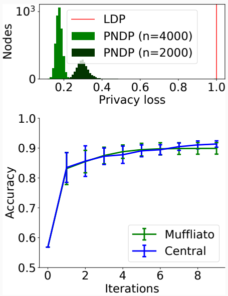
EMPIRICAL ILLUSTRATION



- Users get local DP guarantees w.r.t. their direct neighbors but stronger privacy w.r.t. to other users depending on their distance and the mixing properties of the graph
- This fits the privacy expectations of users in many use-cases (e.g., social networks)







- Users get local DP guarantees w.r.t. their direct neighbors but stronger privacy w.r.t. to other users depending on their distance and the mixing properties of the graph
- This fits the privacy expectations of users in many use-cases (e.g., social networks)
- For learning, we can randomize the graph after each local computation step to make the privacy loss concentrate!

Conclusion & Perspectives

Take-home message

- Decentralized learning can amplify differential privacy guarantees, providing a new incentive for using such approaches beyond the usual motivation of scalability

Take-home message

- Decentralized learning can amplify differential privacy guarantees, providing a new incentive for using such approaches beyond the usual motivation of scalability

Perspectives

- Privacy and utility guarantees for random walk-based decentralized SGD on arbitrary graphs [Johansson et al., 2009], possibly with multiple parallel walks [Hendrikx, 2022]

Take-home message

- Decentralized learning can amplify differential privacy guarantees, providing a new incentive for using such approaches beyond the usual motivation of scalability

Perspectives

- Privacy and utility guarantees for random walk-based decentralized SGD on arbitrary graphs [Johansson et al., 2009], possibly with multiple parallel walks [Hendrikx, 2022]
- Capturing the redundancy in gossip-based communication (i.e., correlated noise) to further improve privacy guarantees (recall that even constants matter in DP!)

THANK YOU FOR YOUR ATTENTION!

QUESTIONS?

REFERENCES I

[Abadi et al., 2016] Abadi, M., Chu, A., Goodfellow, I. J., McMahan, H. B., Mironov, I., Talwar, K., and Zhang, L. (2016).
Deep learning with differential privacy.
In *CCS*.

[Bassily et al., 2016] Bassily, R., Nissim, K., Smith, A., Steinke, T., Stemmer, U., and Ullman, J. (2016).
Algorithmic stability for adaptive data analysis.
In *STOC*.

[Bassily et al., 2014] Bassily, R., Smith, A. D., and Thakurta, A. (2014).
Private Empirical Risk Minimization: Efficient Algorithms and Tight Error Bounds.
In *FOCS*.

[Berthier et al., 2020] Berthier, R., Bach, F., and Gaillard, P. (2020).
Accelerated gossip in networks of given dimension using jacobi polynomial iterations.
SIAM Journal on Mathematics of Data Science, 2(1):24–47.

[Boyd et al., 2006] Boyd, S., Ghosh, A., Prabhakar, B., and Shah, D. (2006).
Randomized gossip algorithms.
IEEE Transactions on Information Theory, 52(6):2508–2530.

[Carlini et al., 2022] Carlini, N., Chien, S., Nasr, M., Song, S., Terzis, A., and Tramer, F. (2022).
Membership inference attacks from first principles.
In *S&P*.

[Carlini et al., 2021] Carlini, N., Tramèr, F., Wallace, E., Jagielski, M., Herbert-Voss, A., Lee, K., Roberts, A., Brown, T., Song, D., Erlingsson, Ú., Oprea, A., and Raffel, C. (2021).
Extracting training data from large language models.
In *USENIX Security*.

[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*.

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

[Hendrikx, 2022] Hendrikx, H. (2022).
A principled framework for the design and analysis of token algorithms.
Technical report, arXiv:2205.15015.

[Johansson et al., 2009] Johansson, B., Rabi, M., and Johansson, M. (2009).
A randomized incremental subgradient method for distributed optimization in networked systems.
SIAM Journal on Optimization, 20(3):1157–1170.

[Jung et al., 2021] Jung, C., Ligett, K., Neel, S., Roth, A., Sharifi-Malvajerdi, S., and Shenfeld, M. (2021).
A New Analysis of Differential Privacy's Generalization Guarantees (Invited Paper).

[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*.

[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*.

[Mironov, 2017] Mironov, I. (2017).
Rényi Differential Privacy.
In *CSF*.

[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*.

[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).*