Federated Learning on Riemannian manifolds
A projection-based approach

Thibault Pautrel

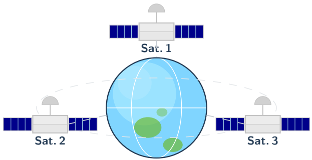
Joint work with F. Bouchard (L2S, CNRS), A. Mian (LISTIC) and G. Ginolhac (LISTIC)

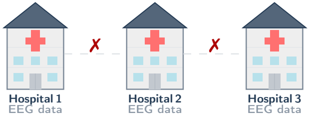
7th SLSIP Workshop – Kruje – 27/05/2026

Laboratoire
Signaux &
Systèmes

Outline

Why Federated Learning?





Naive centralized approach:

~~Pool data → Split → Train → Evaluate~~

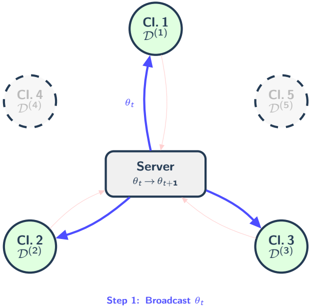
Distributed data | Heterogeneous | Partial participation | Privacy constraints | No raw sharing

What is Federated Learning ?

$$\textbf{Goal:}\quad \theta^* = \arg\min_{\theta\in\mathbb{R}^d}\left\{F(\theta) = \frac{1}{N}\sum_{i=1}^{N} F_i\big(\theta;\mathcal{D}^{(i)}\big)\right\}.$$

Workflow: random init., T communication rounds

1. **Broadcast.** Server sends θ_t to sampled subset $\mathcal{S}_t \subset [N]$.
2. **Local optimization.** Each $i \in \mathcal{S}_t$ performs SGD from θ_t: $\theta_t^{(i)} \approx \arg\min_{\theta\in\mathbb{R}^d} F_i(\theta)$
3. $\theta_t^{(i)} \to$ server.
4. **Aggregation.** $\theta_{t+1} \leftarrow \textsc{Agg}(\theta_t, \theta_t^{(i)})$

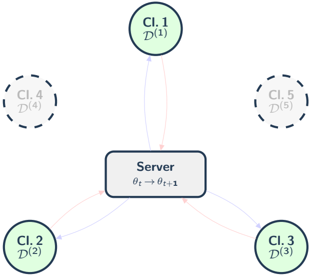


What is Federated Learning ?

$$\textbf{Goal:}\quad \theta^* = \arg\min_{\theta\in\mathbb{R}^d}\left\{F(\theta) = \frac{1}{N}\sum_{i=1}^{N} F_i\big(\theta;\mathcal{D}^{(i)}\big)\right\}.$$

Workflow: random init., T communication rounds

1. **Broadcast.** Server sends θ_t to sampled subset $\mathcal{S}_t \subset [N]$.
2. **Local optimization.** Each $i \in \mathcal{S}_t$ optimizes from θ_t: $\theta_t^{(i)} \approx \arg\min_{\theta\in\mathbb{R}^d} F_i(\theta, \mathcal{D}^{(i)})$
3. $\theta_t^{(i)} \rightarrow$ server.
4. **Aggregation.** $\theta_{t+1} \leftarrow \mathrm{AGG}(\theta_t, \theta_t^{(i)})$



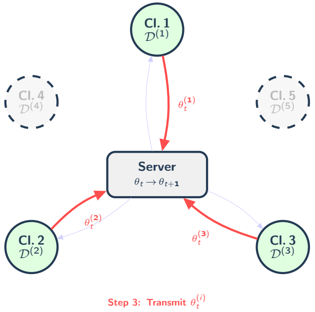
Step 2: Local SGD on each client

What is Federated Learning ?

Goal: $\theta^* = \arg\min_{\theta \in \mathbb{R}^d} \left\{ F(\theta) = \frac{1}{N} \sum_{i=1}^{N} F_i\left(\theta; \mathcal{D}^{(i)}\right) \right\}.$

Workflow: random init., T communication rounds

1. **Broadcast.** Server sends θ_t to sampled subset $\mathcal{S}_t \subset [N]$.
2. **Local optimization.** Each $i \in \mathcal{S}_t$ performs SGD from θ_t: $\theta_t^{(i)} \approx \arg\min_{\theta \in \mathbb{R}^d} F_i(\theta)$
3. $\theta_t^{(i)} \rightarrow$ server.
4. **Aggregation.** $\theta_{t+1} \leftarrow \textsc{Agg}(\theta_t, \theta_t^{(i)})$

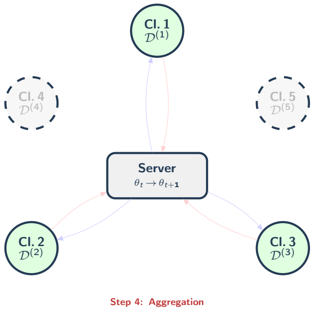


What is Federated Learning ?

$$\textbf{Goal:}\quad \theta^* = \arg\min_{\theta\in\mathbb{R}^d}\left\{F(\theta) = \frac{1}{N}\sum_{i=1}^{N} F_i\left(\theta; \mathcal{D}^{(i)}\right)\right\}.$$

Workflow: random init., T communication rounds

1. **Broadcast.** Server sends θ_t to sampled subset $\mathcal{S}_t \subset [N]$.
2. **Local optimization.** Each $i \in \mathcal{S}_t$ performs SGD from θ_t: $\theta_t^{(i)} \approx \arg\min_{\theta\in\mathbb{R}^d} F_i(\theta)$
3. $\theta_t^{(i)} \rightarrow$ server.
4. **Aggregation.** $\theta_{t+1} \leftarrow \textsc{Agg}(\theta_t, \theta_t^{(i)})$

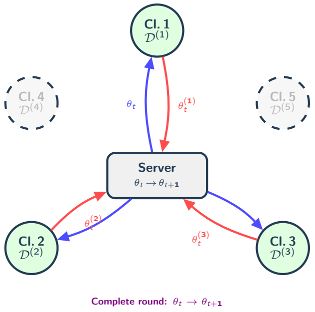


What is Federated Learning ?

$$\textbf{Goal:}\quad \theta^* = \arg\min_{\theta\in\mathbb{R}^d}\left\{F(\theta) = \frac{1}{N}\sum_{i=1}^{N} F_i\big(\theta; \mathcal{D}^{(i)}\big)\right\}.$$

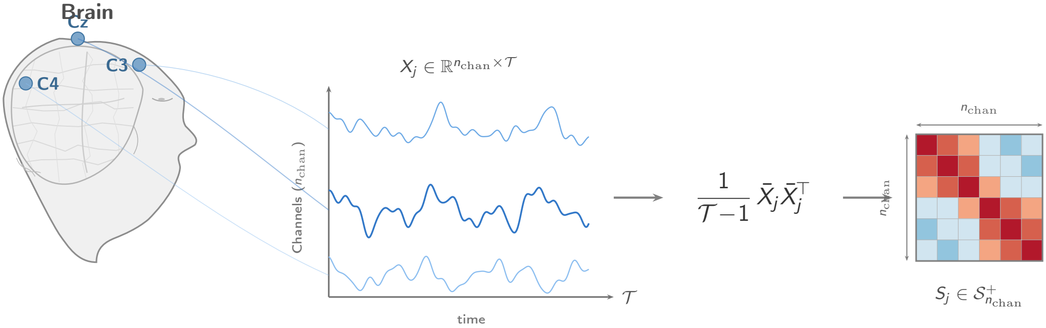
Workflow: random init., T communication rounds

1. **Broadcast.** Server sends θ_t to sampled subset $\mathcal{S}_t \subset [N]$.
2. **Local optimization.** Each $i \in \mathcal{S}_t$ optimizes from θ_t: $\theta_t^{(i)} \approx \arg\min_{\theta\in\mathbb{R}^d} F_i(\theta, \mathcal{D}^{(i)})$
3. $\theta_t^{(i)} \rightarrow$ server.
4. **Aggregation.** $\theta_{t+1} \leftarrow \textsc{Agg}(\theta_t, \theta_t^{(i)})$

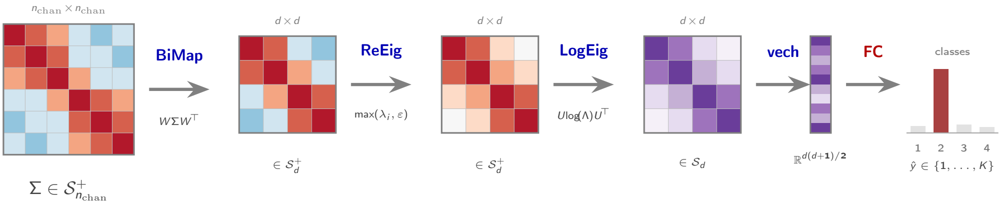


Towards Riemannian FL: covariance-based learning

EEG data



SPDnet[1]: shallow geometry-preserving network (SPD $\rightarrow$ SPD)



Learnable parameters: (W, ξ, β)

$$\mathrm{St}(d,p) = \{X \in \mathbb{R}^{d \times p} : X^\top X = I_p\}$$

[1] Zhiwu Huang and Van Gool, "A riemannian network for spd matrix learning".

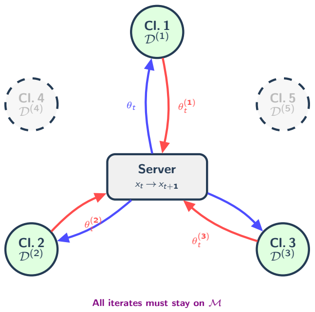
Riemannian Federated Learning: What Needs to Change?

Goal: $\theta^* = \arg\min_{\theta \in \mathcal{M}} \left\{ F(\theta) = \frac{1}{N} \sum_{i=1}^{N} F_i(\theta; \mathcal{D}^{(i)}) \right\}$

$\theta \in \mathcal{M}$, not $\mathbb{R}^d$

Initialize $\theta_0 \in \mathcal{M}$. For each round $t = 0, \ldots, T-1$:

1. **Broadcast.** Server sends θ_t to sampled subset $\mathcal{S}_t \subset [N]$.
2. **Local optimization.** Must stay on $\mathcal{M}$! $\theta_t^{(i)} \leftarrow$? **Riemannian optim.**
3. $\theta_t^{(i)} \in \mathcal{M} \rightarrow$ server.
4. **Aggregation.** Geometrically meaningful average ! $\theta_{t+1} \leftarrow$? **Riemannian aggregation**



Outline

Riemannian geometry essentials

Riemannian metric

$\langle \cdot, \cdot \rangle_\theta : T_\theta\mathcal{M} \times T_\theta\mathcal{M} \to \mathbb{R}$

$\langle \xi, \eta \rangle_\theta = \langle \xi, \eta \rangle_{\mathsf{eucl}}$

Retraction map

$\mathrm{Ret}_\theta : T_\theta\mathcal{M} \to \mathcal{M}$

Examples:

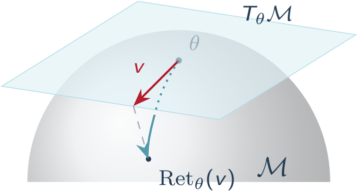
$\exp_\theta(v), \quad R_\theta(v) = \Pi(\theta + v)$

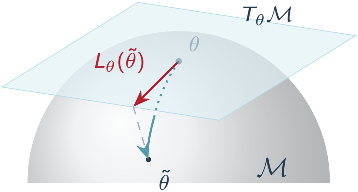
Lifting map

$L_\theta : \mathcal{M} \to T_\theta\mathcal{M}$

Examples:

$\log_\theta(\tilde{\theta}), \quad \mathrm{Proj}_{T_\theta}(\tilde{\theta} - \theta)$





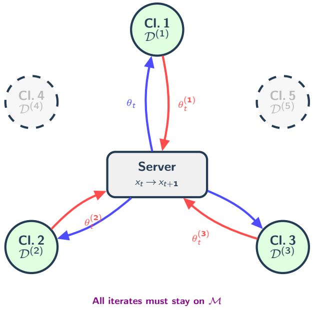
Riemannian FL: what needs to change?

Goal: $\theta^* = \arg\min_{\theta \in \mathcal{M}} \left\{ F(\theta) = \frac{1}{N} \sum_{i=1}^{N} F_i\left(\theta; \mathcal{D}^{(i)}\right) \right\}$

$\theta \in \mathcal{M}$, not $\mathbb{R}^d$

Initialize $\theta_0 \in \mathcal{M}$. For each round $t = 0, \ldots, T-1$:

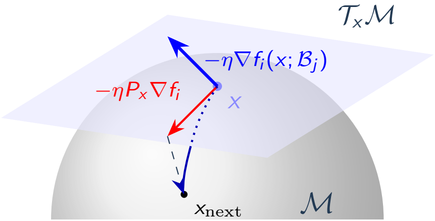
1. **Broadcast.** Server sends θ_t to sampled subset $\mathcal{S}_t \subset [N]$.
2. **Local optimization.** Must stay on $\mathcal{M}$!
   $\theta_t^{(i)} \leftarrow ?$ **Riemannian optim.**
3. Clients transmit local parameters $\theta_t^{(i)} \in \mathcal{M}$ to the server.
4. **Aggregation.** Geometrically meaningful average !
   $\theta_{t+1} \leftarrow ?$ **Riemannian aggregation**



Riemannian local optimisation

Euclidean

$$x_{\text{next}} \leftarrow x - \eta \nabla f_i(x; \mathcal{B}_j) \in \mathbb{R}^{d \times p}$$



$$P_x : \mathbb{R}^{d \times p} \to T_x\mathcal{M}$$

$$\Pi : \mathbb{R}^{d \times p} \to \mathcal{M}$$

Riemannian

$$x_{\text{next}} \leftarrow \Pi\big(x - \eta\, P_x \nabla f_i(x; \mathcal{B}_j)\big) \in \mathcal{M}$$

Riemannian gradient

Unique vector grad $\mathcal{L}(\theta) \in T_\theta\mathcal{M}$ such that

$$D\mathcal{L}(\theta)[V] = \langle \text{grad}\, \mathcal{L}(\theta), V \rangle_\theta, \quad V \in T_\theta\mathcal{M}$$

Embedded manifolds:

$$\text{grad}\, \mathcal{L}(\theta) = \text{Proj}_\theta(\nabla \mathcal{L}(\theta))$$

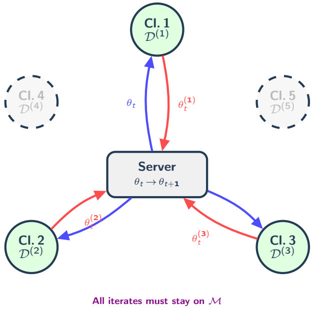
Riemannian Federated Learning: What needs to change?

Goal: $\theta^* = \arg\min_{\theta \in \mathcal{M}} \left\{ F(\theta) = \frac{1}{N} \sum_{i=1}^{N} F_i(\theta; \mathcal{D}^{(i)}) \right\}$

$\theta \in \mathcal{M}$, not $\mathbb{R}^d$

Initialize $\theta_0 \in \mathcal{M}$. For each round $t = 0, \dots, T-1$:

1. **Broadcast.** Server sends θ_t to sampled subset $\mathcal{S}_t \subset [N]$.
2. **Local optimization.** Must stay on $\mathcal{M}$!
   $\theta_t^{(i)} \leftarrow$? **Riemannian optim.**
3. Clients transmit local parameters $\theta_t^{(i)} \in \mathcal{M}$ to the server.
4. **Aggregation.** Geometrically meaningful average !
   $\theta_{t+1} \leftarrow$? **Riemannian aggregation**

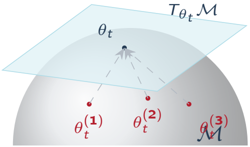


Outline

Existing approaches

Fréchet mean

$$\theta_{t+1} \leftarrow \arg\min_{\theta} \frac{1}{k} \sum_i \|\log_{\theta_t}(\theta)\|_{\theta_t}^2$$



Issue: No closed form in general $\Rightarrow$ approximated by

KFAvg

$$\theta_{t+1} \leftarrow \exp_{\theta_t}\left(\frac{1}{k} \sum_i \log_{\theta_t}\left(\theta_t^{(i)}\right)\right)$$

Issue: Exp/log maps often unknown or expensive to compute

RetAvg

$$\theta_{t+1} \leftarrow \mathrm{R}_{\theta_t}\left(\frac{1}{k} \sum_i \mathrm{R}_{\theta_t}^{-1}\left(\theta_t^{(i)}\right)\right)$$

Issue: Retraction *and* its inverse rarely available simultaneously

Proposed approaches

RetAvg

$$\theta_{t+1} \leftarrow \mathrm{R}_{\theta_t}\left(\frac{1}{k}\sum_i \mathrm{R}_{\theta_t}^{-1}\left(\theta_t^{(i)}\right)\right)$$

RLAvg

$$\theta_{t+1} \leftarrow \Pi\left(\theta_t + \frac{1}{k}\sum_i L_{\theta_t}\left(\theta_t^{(i)}\right)\right)$$

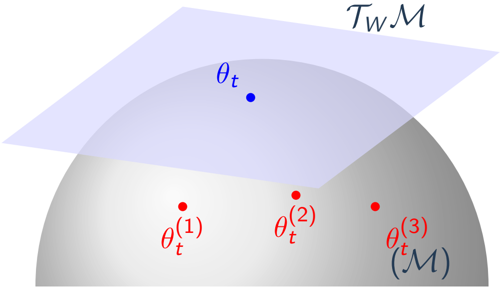
Idea: Replace R^{-1} with a lifting-based surrogate

ProjAvg

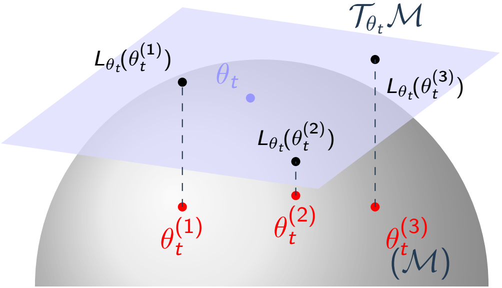
$$\theta_{t+1} \leftarrow \Pi\left(\frac{1}{k}\sum_i \theta_t^{(i)}\right)$$

Idea: Average directly in the ambient space, then project back onto $\mathcal{M}$

θ_t: current global model at round t
$\theta_t^{(i)}$: locally updated client models

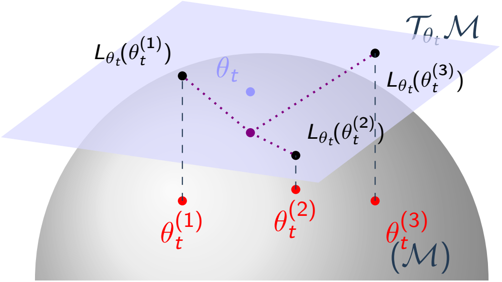


Lifting map: project each $\theta_t^{(i)}$ into the tangent space at θ_t



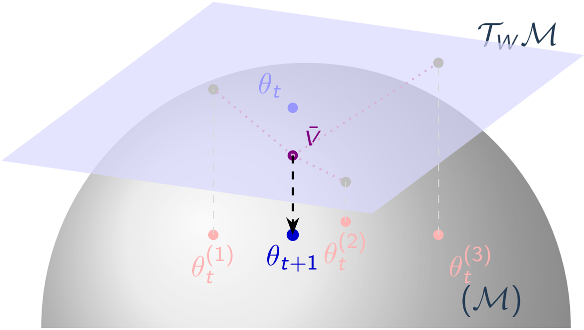
$$\mathcal{M} \ni \theta_t^{(i)} \mapsto L_{\theta_t}(\theta_t^{(i)}) = \mathrm{Proj}_{\mathcal{T}_{\theta_t}}(\theta_t^{(i)} - \theta_t) \in \mathcal{T}_{\theta_t}\mathcal{M}$$

In the tangent space, **Euclidean mean** of the lifted vectors



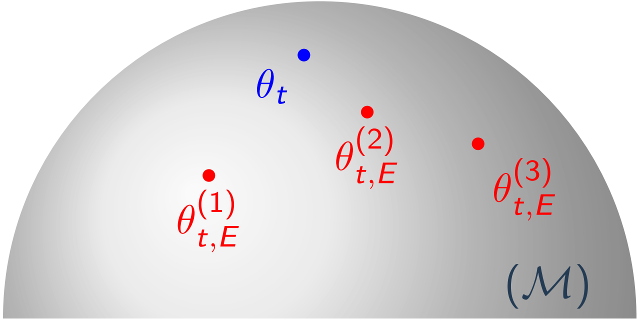
$$\bar{V} = \frac{1}{k}\sum_i L_{\theta_t}(\theta_t^{(i)}) \in \mathcal{T}_{\theta_t}\mathcal{M}$$

Retraction back to $\mathcal{M}$

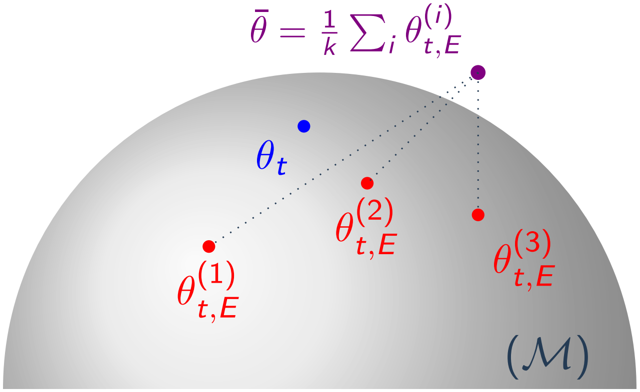


$$\theta_{t+1} = R_{\theta_t}(\bar{V}) = R_{\theta_t}\left(\frac{1}{k}\sum_i L_{\theta_t}(\theta_t^{(i)})\right) \in \mathcal{M}$$

θ_t: current global model at round t
$\theta_t^{(i)}$: locally updated client models

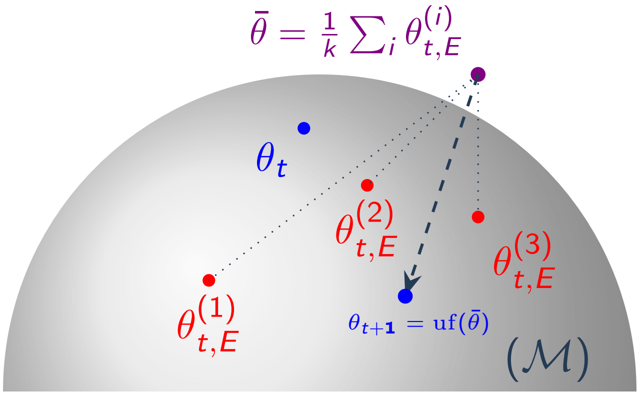


Compute the Euclidean mean in the ambient space.



$\bar{\theta}$ generally **leaves the manifold**.

Project back onto $\mathcal{M}$ via the polar retraction



$$\theta_{t+1} = \mathrm{uf}(\bar{\theta}).$$

Outline

Algorithm

Input: n clients, k clients/round, stepsize η, local steps τ, batch size B

```
Initialize x_0 ∈ 𝓜
for t = 0, ..., T − 1 do
    Sample S_t ⊂ [n], |S_t| = k
    for each client i ∈ S_t in parallel do
        x_{t,0}^{(i)} ← x_t
        x_{t,τ}^{(i)} ← LocalOpt(f_i, 𝒟_i, τ, B)          ▷ RSGD, RAdam
        Send x_{t,τ}^{(i)} to server
    ———————————— Server aggregation ————————————
    RLAvg:    x_{t+1} ← Π( x_t + (1/k) Σ_{i∈S_t} P_{x_t}( x_{t,τ}^{(i)} − x_t ) )
    ProjAvg:  x_{t+1} ← Π( (1/k) Σ_{i∈S_t} x_{t,τ}^{(i)} )
return x_T
```

Initialize $x_0 \in \mathcal{M}$

for $t = 0, \ldots, T-1$ **do**

Sample $S_t \subset [n]$, $|S_t| = k$

for each client $i \in S_t$ **in parallel do**

$x_{t,0}^{(i)} \leftarrow x_t$

$x_{t,\tau}^{(i)} \leftarrow \textsc{LocalOpt}(f_i, \mathcal{D}_i, \tau, B)$ ▷ RSGD, RAdam

Send $x_{t,\tau}^{(i)}$ to server

Server aggregation

RLAvg: $x_{t+1} \leftarrow \Pi\Big(x_t + \frac{1}{k}\sum_{i\in S_t} P_{x_t}\big(x_{t,\tau}^{(i)} - x_t\big)\Big)$

ProjAvg: $x_{t+1} \leftarrow \Pi\Big(\frac{1}{k}\sum_{i\in S_t} x_{t,\tau}^{(i)}\Big)$

return x_T

Convergence guarantees

At client level, LocalOpt=RSGD:

$$x_{t,0}^{(i)} \leftarrow x_t, \qquad x_{t,j+1}^{(i)} \leftarrow \Pi\left(x_{t,j}^{(i)} - \eta\, \widehat{grad} f_i(x_{t,j}^{(i)})\right), \qquad j = 0, \ldots, \tau - 1$$

Theorem

Under reasonable geometric and stochastic gradient assumptions, **both** ProjAvg and RLAvg achieve

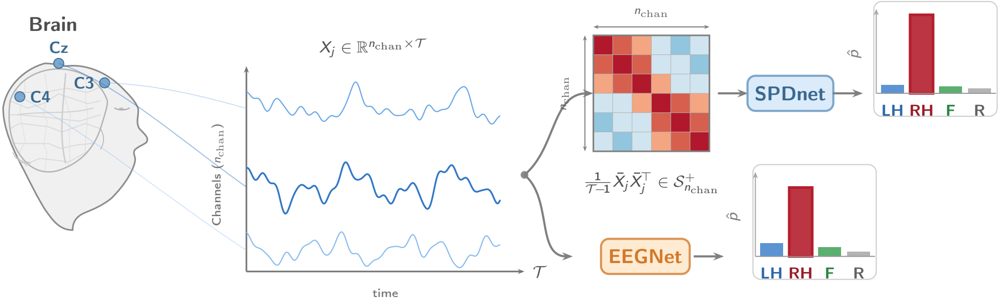
$$\frac{1}{T}\sum_{t=0}^{T-1} \mathbb{E}\|\operatorname{grad} F(x_t)\|^2 = \mathcal{O}\left(\frac{1}{\sqrt{T}}\left(1 + \frac{1}{k}\left(\frac{n-k}{n-1}\sigma^2 + \frac{\sigma_{\mathrm{sg}}^2}{\tau B}\right)\right)\right)$$

Prior work[2] exp / log maps are nonlinear $\Rightarrow$ hard to handle $\tau > 1$ with $k > 1$

[2] Li and Ma, "Federated learning on Riemannian manifolds"; Zhenwei Huang et al., "Federated learning on Riemannian manifolds with differential privacy".

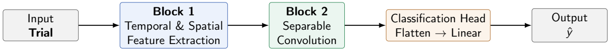
Outline

EEG data and datasets



From MOABB benchmark[a]

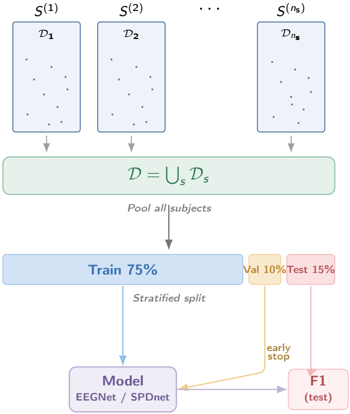
Dataset	Subjects	Channels	Classes
Weibo2014	10	60	7
PhysionetMI	103	64	4



[a] Chevallier et al., "The largest EEG-based BCI reproducibility study for open science: the MOABB benchmark".

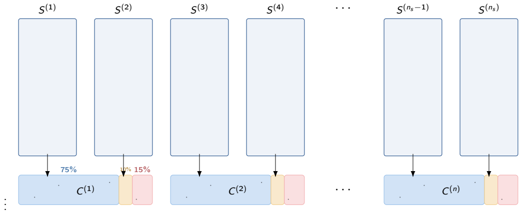
Experimental Setup

Centralized baseline



300 epochs, patience 75

Federated — subject-based clients



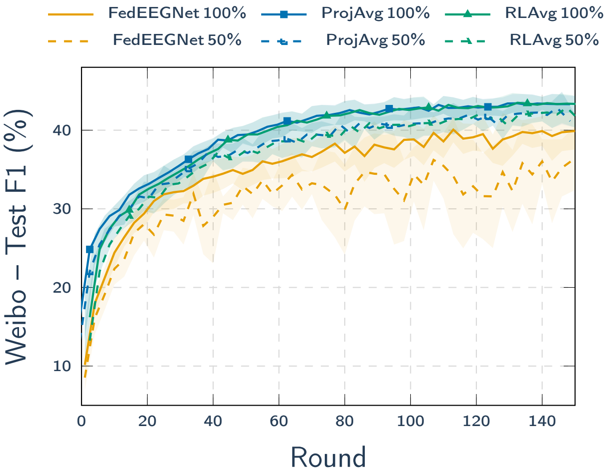
Weibo: 5 clients, PhysionetMI: 53 clients,

150 rounds, 2 local epochs, half/full participation

Results[3]

F1 score (%), mean ± std over 10 seeds

Dataset		EEGNet (FedAvg)	SPDnet (FedSPDnet)
Weibo2014	*# params*	*5 303*	*4 715*
	Centr.	50.7 ± 1.5	**51.7 ± 0.8**
	Full	39.9 ± 2.4	**43.3 ± 1.0**
	Half	36.4 ± 4.1	**41.2 ± 0.8**
PhysionetMI	*# params*	*3 284*	*2 452*
	Centr.	**50.8 ± 0.7**	43.1 ± 0.5
	Full	38.3 ± 1.8	**39.5 ± 0.6**
	Half	38.0 ± 2.7	**39.5 ± 0.9**



[3] Pautrel et al., "FedSPDnet: geometry-aware federated deep learning with SPDnet".

Outline

Privacy model: record-level differential privacy

Full dataset: $D = (\mathcal{D}_1, \ldots, \mathcal{D}_n) \in \mathcal{Z}^{n \times m}$

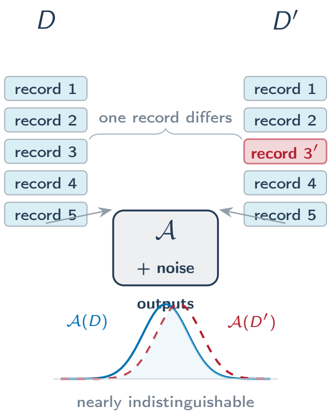
Record-level adjacency

$D \sim D'$ iff they differ in exactly **one record** of **one client** $i^\star$

(ε, δ)-Differential Privacy

$\mathcal{A}$ is (ε, δ)-DP if $\forall\, D \sim D', \forall\, S \subseteq \mathcal{Y}$:

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



DP-RFedProj

Algorithm

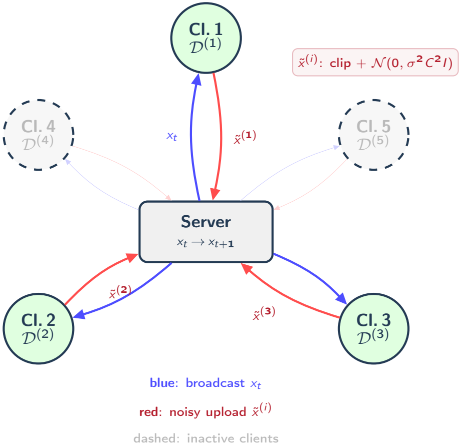
For T rounds, each selected client $i \in \mathcal{S}_t$:

1: $x \leftarrow x_t$
2: **for** $j = 0, \ldots, \tau - 1$ **do**
3: Sample $\mathcal{B}_j \subset \mathcal{D}_i$, $|\mathcal{B}_j| = B$
4: $\hat{g} \leftarrow \frac{1}{B} \sum_{b \in \mathcal{B}_j} \mathrm{clip}_C(P_x \nabla f(x; z_b))$
5: $\tilde{g} \leftarrow \hat{g} + P_x \xi, \quad \xi \sim \mathcal{N}(0, \sigma_{\mathrm{dp}}^2 I)$
6: $x \leftarrow \Pi(x - \eta P_x(\tilde{g}))$
7: **return** $\tilde{x}^{(i)} = x$

Server: $x_{t+1} = \textsc{Agg}(x_t, \{\tilde{x}^{(i)}\}_{i \in \mathcal{S}_t})$

$$\Delta_2 = \sup_{D \sim D'} \|\hat{g}_D - \hat{g}_{D'}\|_2$$

$$\Rightarrow \sigma_{\mathrm{dp}} = \frac{\Delta_2 \sqrt{2 \ln(1.25/\delta)}}{\varepsilon} \iff (\varepsilon, \delta)\text{-DP}$$



Main convergence result

Theorem

For T large enough, under reasonable geometric and stochastic gradient assumptions, **both** ProjAvg and RLAvg achieve

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\left[\|\operatorname{grad} F(x_t)\|^2\right] = O\Big(\frac{1}{\sqrt{T}}\Big[1+\frac{1}{k}\Big(\frac{n-k}{n-1}\cdot \text{noise}+\frac{\sigma_{\text{sg}}^2}{B\tau} + \frac{d_{\mathcal{M}}\,\sigma_{\text{dp}}^2}{\tau}\Big)\Big] + \text{clipping bias floor}\Big)$$

Thank you for your attention

Summary

- Riemannian FL via projection
- ProjAvg & RLAvg
- FedSPDnet

Perspectives

- Tighter privacy accounting (GDP)
- Personalization
- Convergence on general manifolds

https://thibault-pautrel.github.io/

References I

Chevallier, Sylvain et al. "The largest EEG-based BCI reproducibility study for open science: the MOABB benchmark". In: *arXiv preprint arXiv:2404.15319* (2024).

Huang, Zhenwei et al. "Federated learning on Riemannian manifolds with differential privacy". In: *arXiv preprint arXiv:2404.10029* (2024).

Huang, Zhiwu and Luc Van Gool. "A riemannian network for spd matrix learning". In: *Proceedings of the AAAI conference on artificial intelligence*. Vol. 31. 2017.

Li, Jiaxiang and Shiqian Ma. "Federated learning on Riemannian manifolds". In: *arXiv preprint arXiv:2206.05668* (2022).

Pautrel, Thibault et al. "FedSPDnet: geometry-aware federated deep learning with SPDnet". In: *Proceedings of the European Signal Processing Conference (EuSIPCO)*. Accepté. 2026.