Projection-based Riemannian federated learning with partial participation

Thibault Pautrel[a,*], Florent Bouchard[a], Guillaume Ginolhac[b], Ammar Mian[b]

[a] *Université Paris Saclay, CNRS, CentraleSupélec, L2S, Gif-sur-Yvette, France*
[b] *Université Savoie Mont Blanc, LISTIC, Annecy, France*

Abstract

Federated learning on Riemannian manifolds enables collaborative training without centralized data pooling when model parameters are intrinsically constrained. Existing methods either rely on geometric operations lacking closed-form expressions on key manifolds (such as the Stiefel manifold), employ optimizer-specific gradient streams that incur information loss through successive transports, or – even when computationally cheap – require drift correction terms and full client participation. We propose two aggregation strategies, RFedProj and RFedRL, that are optimizer-agnostic, lightweight (requiring only standard projections and retractions), and support partial participation without auxiliary correction. Both achieve identical convergence rates under these relaxed conditions and data heterogeneity, with bounds explicitly characterizing how participation ratio and heterogeneity interact – mirroring classical Euclidean federated guarantees. Experiments on EEG motor imagery classification with SPDNet on compact Stiefel manifolds validate competitive performance against centralized and Euclidean baselines.

Keywords: federated learning, Riemannian optimization, SPD matrices

*Corresponding author
Email addresses: thibault.pautrel@centralesupelec.fr (Thibault Pautrel), florent.bouchard@cnrs.fr (Florent Bouchard), guillaume.ginolhac@univ-smb.fr (Guillaume Ginolhac), ammar.mian@univ-smb.fr (Ammar Mian)

1. Introduction

1.1. Motivation

Federated learning (FL) enables collaborative model training across distributed clients without exchanging raw data, making it well-suited to applications where data is scarce or subject-specific and where centralized pooling is precluded by confidentiality or regulatory constraints [1, 2, 3]. In the standard protocol, clients perform local training on their own data and periodically send model updates to a central server, which aggregates them into a global model. The canonical FedAvg algorithm [1] performs this aggregation via arithmetic averaging – computationally simple and theoretically well-understood – with extensions such as FedProx [3] and SCAFFOLD [4] handling partial client participation and data heterogeneity. Deploying modern deep learning in this framework, however, is costly: architectures with millions of parameters incur prohibitive communication overhead when model updates are exchanged at each round [5, 6]. For domains such as medical imaging, brain-computer interfaces, and radar classification, an alternative approach leverages the discriminative power of spatial covariance matrices [7, 8], motivating compact models that preserve their symmetric positive definite (SPD) geometry [9]. Riemannian neural networks operating on SPD matrices, such as SPDNet [10] and variants [11, 12, 13], achieve competitive accuracy with significantly fewer parameters – a parameter efficiency particularly attractive for federated settings where communication costs scale with model size. Beyond SPD matrices, extending FL to Riemannian manifolds addresses applications where parameters are intrinsically constrained like orthogonal matrices and low-rank subspaces on the Grassmann manifold [14]. These models require Riemannian optimization [15, 16], and their federated training faces a challenge absent in Euclidean FL: naive averaging of client parameters may leave the manifold or fail to produce geometrically meaningful aggregates.

Existing Riemannian FL methods address this aggregation challenge by approximating the Fréchet mean [17] via one step of Karcher flow [18, 19]: client models are lifted to the tangent space via the Riemannian logarithm, averaged, and mapped back via the exponential map. Yet, on the Stiefel manifold – central to SPDNet architectures – the logarithm lacks a closed-form expression and requires iterative solvers [20], while the exponential map involves costly matrix exponentials and is numerically unstable [21]. Retractions offer a practical alternative as first-order approximations of the exponential map which project tangent vectors back onto the manifold [15, 16].

On compact submanifolds such as Stiefel, projection-based retractions – which map ambient Euclidean updates onto the constraint set – admit efficient closed-form expressions [22]. Absil and Malick [22] established that such retractions satisfy convergence requirements for first-order optimization methods; since federated aggregation schemes like FedAvg are inherently first-order, these guarantees extend naturally to the federated setting. Beyond computational efficiency, compact smooth submanifolds are proximally smooth, ensuring well-behaved nearest-point projections [23, 24] that enable direct transfer of Euclidean convergence analyses. This regularity has been exploited for stochastic projected gradient methods [25] and decentralized optimization [26]. We adapt these tools to the federated setting, addressing the joint challenge of partial client participation and data heterogeneity.

1.2. Related work

Li and Ma [18] introduced RFedSVRG, the first general Riemannian FL framework based on variance-reduced stochastic gradients, with subsequent extensions incorporating differential privacy [19] and second-order Hessian information [27, 28]. Yet, all these methods rely on Riemannian logarithms, exponential maps and parallel transport, and their convergence guarantees hold only for single local steps or single-agent scenarios – the difficulty of bounding accumulated errors through repeated log/exp operations precludes analysis with multiple local epochs. Moreover, the SVRG backbone requires full gradient computation at each outer iteration, making it inherently incompatible with partial participation. Zhang et al. [29] addressed multiple local epochs on compact submanifolds via projection-based updates with drift correction, later extended to zeroth-order oracles [30]. Both achieve sublinear convergence under data heterogeneity but assume full client participation. The combination of projection-based aggregation, partial participation, and multiple local epochs remains unexplored. An alternative paradigm is gradient-stream aggregation [31], where clients accumulate stochastic gradients via vector transport along their local optimization path; the server averages these streams and retracts to obtain the next global model. However, this approach is inherently tied to SGD and precludes alternative client optimizers such as Adam [32] or momentum-based methods [33].

1.3. Contributions

We propose two projection-based Riemannian federated aggregation methods – RFedProj and RFedRL – that eliminate exponential maps, logarithms,

Table 1: Comparison of Riemannian federated learning methods.

Method	Manifold	Aggregation	Client sampling	Local epochs	Drift corr.	Geometric ops
[18]	General	Parameters	Full / single†	$\tau \geq 1^\dagger$	Yes	Exp/Log, parallel transport
[19]	General	Parameters	Full	$\tau = 1$	Yes/No	Exp/Log, parallel transport
[27]	General	Parameters	Full	$\tau = 1$	Yes	Exp/Log, parallel transport
[28]	General	Parameters	Full	$\tau = 1$	Yes	Exp/Log, parallel transport
[29]	Compact	Parameters	Full	$\tau \geq 1$	Yes	Projection only
[30]	Compact	Parameters	Full	$\tau \geq 1$	Yes	Projection only
[31]	General	Gradients	Arbitrary	$\tau \geq 1$	No	Retraction, vector transport
Ours	Compact	Parameters	Arbitrary	$\tau \geq 1$	No	Projection only

†Convergence for $\tau \geq 1$ requires single-client participation ($k = 1$); full participation ($k = n$) only analyzed for $\tau = 1$.

parallel transport and gradient-stream transport while remaining optimizer-agnostic and supporting arbitrary client sampling with multiple local epochs. Our framework applies to compact smooth submanifolds, combining the algorithmic structure of RFedAvg [18] with geometric tools from [29] and drawing inspiration from retracted-lifted barycenters [21]. Table 1 summarizes the key characteristics of existing approaches and our contributions. The main contributions are:

- **Projection-based aggregation.** We introduce two server-side aggregation mechanisms: **RFedProj**, which projects the Euclidean mean of client models onto the manifold; and **RFedRL**, which lifts client displacements onto the tangent space at the current global iterate, averages, and retracts via nearest-point projection; requiring storage of the global iterate, unlike RFedProj.

- **Unified convergence analysis.** Under Lipschitz-smooth client objectives and proximal smoothness of the manifold, we establish that both RFedProj and RFedRL achieve $\mathcal{O}(1/\sqrt{T})$ convergence under partial participation:

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

  where σ^2 quantifies client heterogeneity, σ_{sg}^2 captures stochastic gradient noise, k/n is the participation ratio, and B is the minibatch size. The variance term $\frac{n-k}{k(n-1)}\sigma^2$ explicitly captures how heterogeneity interacts

with participation – mirroring Euclidean FL bounds [34] and confirming that these trade-offs extend to manifolds using only cheap projections. The two methods differ only in higher-order curvature terms; the analysis relies on regularity of nearest-point projections [23, 25, 26] and a tube-stability argument ensuring iterates remain within the well-behaved tubular neighborhood.

- **Numerical experiments.** We validate our approach on EEG-based motor imagery classification using SPDNet [10] with Stiefel-constrained parameters across three MOABB benchmark datasets [35]: Weibo2014, Schirrmeister2017, and PhysionetMI. We call this federated architecture FedSPDNet, which can employ either RFedProj or RFedRL for server-side aggregation. As Euclidean baseline, we consider FedEEGNet – FedAvg applied to EEGNet [36]. Comparing the two, we find that: (i) RFedProj and RFedRL yield statistically indistinguishable performance, validating theoretical equivalence and favoring the simpler RFedProj in practice; (ii) FedSPDNet exhibits lower performance degradation relative to its centralized baseline than FedEEGNet, demonstrating implicit regularization against client drift; and (iii) FedSPDNet scales more gracefully with federation size, with significantly smaller degradation when doubling client count compared to FedEEGNet. These results suggest that projection-based Riemannian FL is well-suited for multi-site BCI deployments.

2. Riemannian geometry background

Manifold setup and metric. Let $\mathcal{M}$ be a compact $\mathcal{C}^\infty$ submanifold embedded in $\mathbb{R}^{d\times p}$. The Frobenius inner product $\langle \xi, \eta \rangle = \operatorname{tr}(\xi^\top \eta)$ on the ambient space restricts to each tangent space $T_x\mathcal{M}$ – the space of velocity vectors of smooth curves through x – turning $\mathcal{M}$ into a Riemannian manifold; we write $\|\cdot\|$ for the associated norm. The orthogonal projector $P_x : \mathbb{R}^{d\times p} \to T_x\mathcal{M}$ maps any ambient vector w to its closest element in $T_x\mathcal{M}$:

$$P_x(w) = \arg\min_{u \in T_x\mathcal{M}} \|w - u\|^2.$$

Riemannian gradient. The Riemannian gradient of a smooth function $f : \mathcal{M} \to \mathbb{R}$ at $x \in \mathcal{M}$ is the unique vector $\operatorname{grad} f(x) \in T_x\mathcal{M}$ satisfying $\mathrm{D}f(x)[\xi] = \langle \operatorname{grad} f(x), \xi \rangle$ for all $\xi \in T_x\mathcal{M}$. In our embedded setting, if f admits a smooth extension $\tilde{f} : \mathbb{R}^{d\times p} \to \mathbb{R}$, then $\operatorname{grad} f(x) = P_x(\nabla \tilde{f}(x))$, where $\nabla \tilde{f}(x)$ is the Euclidean gradient [15].

Geodesics and exponential map. A geodesic is a curve that locally minimizes arc length – the Riemannian analogue of straight lines in curved spaces. The Riemannian distance $d(x, y)$ is the length of a shortest geodesic connecting x and y. The exponential map $\mathrm{Exp}_x : T_x\mathcal{M} \to \mathcal{M}$ sends $\xi \in T_x\mathcal{M}$ to $\gamma(1)$, where γ is the geodesic with $\gamma(0) = x$ and $\dot{\gamma}(0) = \xi$; its local inverse is the logarithmic map Log_x. On key manifolds such as the Stiefel manifold, the exponential map is computationally expensive and the logarithmic map lacks a closed-form expression [20].

Retractions and projections. In Riemannian optimization, retractions serve as computationally efficient substitutes for the exponential map [15, 16]. A smooth mapping $\mathrm{R}_x : T_x\mathcal{M} \to \mathcal{M}$, defined for each $x \in \mathcal{M}$, is a retraction if $\mathrm{R}_x(0) = x$ and $\mathrm{DR}_x(0)[u] = u$ for all $u \in T_x\mathcal{M}$ – that is, it coincides with the exponential map to first order. On compact submanifolds, a natural choice is the projection-based retraction [22]. Define the nearest-point projection onto $\mathcal{M}$ as

$$\mathcal{P}_{\mathcal{M}}(y) := \arg\min_{x \in \mathcal{M}} \tfrac{1}{2}\|y - x\|^2,$$

which is unique when y lies sufficiently close to $\mathcal{M}$. The projection-based retraction is then

$$\mathrm{R}_x(u) := \mathcal{P}_{\mathcal{M}}(x + u), \qquad x \in \mathcal{M},\ u \in T_x\mathcal{M}. \tag{1}$$

Proximal smoothness and tubular neighborhoods. The projection-based retraction (1) is well-defined only when the projection is unique. Proximal smoothness formalizes this requirement: for $\gamma > 0$, define the open tube $U_{\mathcal{M}}(\gamma) := \{y \in \mathbb{R}^{d\times p} : \mathrm{dist}(y, \mathcal{M}) < \gamma\}$, where $\mathrm{dist}(y, \mathcal{M}) := \inf_{x\in\mathcal{M}} \|y - x\|$. The manifold $\mathcal{M}$ is *γ-proximally smooth* if $\mathcal{P}_{\mathcal{M}}(y)$ is unique for all $y \in U_{\mathcal{M}}(\gamma)$. Any compact smooth embedded submanifold is proximally smooth for some $\gamma > 0$ [23, 25]. Within the tube, the projection enjoys strong regularity properties that enable our analysis (see Supplementary material, Section S1).

Example: the Stiefel manifold. The Stiefel manifold $\mathrm{St}(d, p) = \{X \in \mathbb{R}^{d\times p} : X^\top X = I_p\}$, consisting of $d \times p$ matrices with orthonormal columns, is a compact, proximally-smooth $\mathcal{C}^\infty$ submanifold of $\mathbb{R}^{d\times p}$. It arises naturally in SPDNet [10], where spatial filtering layers are parameterized by orthogonal matrices. The tangent space at X is

$$T_X\mathrm{St}(d, p) = \{\xi \in \mathbb{R}^{d\times p} : X^\top\xi + \xi^\top X = 0\},$$

with orthogonal projector

$$P_X(\xi) = \xi - X\,\mathrm{sym}(X^\top \xi), \quad \mathrm{sym}(A) := \tfrac{1}{2}(A + A^\top). \tag{2}$$

The nearest-point projection admits the closed form

$$\mathcal{P}_{\mathrm{St}(d,p)}(A) = A(A^\top A)^{-1/2} =: \mathrm{uf}(A), \tag{3}$$

where $\mathrm{uf}(\cdot)$ denotes the orthogonal polar factor, computable via a single thin SVD [22].

3. Riemmanian Federated Learning framework

3.1. Problem formulation

We consider a cross-device federated learning (FL) protocol involving a central server and n distributed clients, each holding a private local dataset that cannot be shared due to confidentiality, communication, or regulatory constraints. The global learning task is the manifold-constrained empirical risk minimization

$$x_\star \in \arg\min_{x \in \mathcal{M}} F(x), \qquad F(x) := \frac{1}{n}\sum_{i=1}^{n} f_i(x), \tag{4}$$

where for $i = 1, \ldots, n$ the functions $f_i : \mathcal{M} \to \mathbb{R}$ are $\mathcal{C}^1$ client local objectives. Each client i has access to a local data distribution $\mathcal{D}_i$, and its objective is defined as the expected risk $f_i(x) = \mathbb{E}_{z\sim\mathcal{D}_i}[f_i(x; z)]$, where z denotes a data sample and $f_i(x; z)$ is the per-sample loss. In practice, the local distributions $\{\mathcal{D}_i\}_{i=1}^n$ are often heterogeneous (non-i.i.d.), posing significant challenges for convergence analysis [3]. Table 2 summarizes the key notation used throughout the paper.

3.2. Federated training protocol

The training workflow (see Figure 1) proceeds through a sequence of *communication rounds* orchestrated by the central server. At round t, the server holds a global model $x_t \in \mathcal{M}$ and broadcasts it to a selected subset of clients. Due to system constraints such as limited bandwidth, device availability, or energy considerations, only a fraction of clients participate in each round: a subset $S_t \subset [n]$ of size $k \in \{1, \ldots, n\}$ is sampled uniformly at random without replacement. This *partial participation* regime is standard in cross-device FL, where many clients may be enrolled but only a small fraction are active at any given time [5, 6].

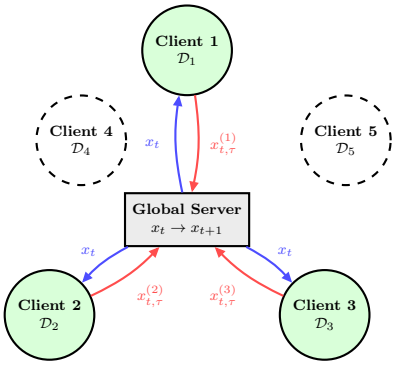

Figure 1: At each round t, the central server coordinates training by sharing global model parameters x_t with selected clients S_t, which update based on their local datasets $\mathcal{D}_i$ and return model updates $\{x_{t,\tau}^{(i)}\}_{i \in S_t}$. Raw data is never transmitted.

Local optimization. Upon receiving the global model x_t, each selected client $i \in S_t$ initializes its local model to x_t and performs $\tau \geq 1$ local optimization steps. Riemannian FL methods [18] typically rely on the exponential map for gradient descent [37]:

$$x_{t,\ell+1}^{(i)} \leftarrow \operatorname{Exp}_{x_{t,\ell}^{(i)}}\big(-\eta \operatorname{grad} f_i(x_{t,\ell}^{(i)})\big). \tag{5}$$

In our framework, we replace the exponential map with the projection-based retraction (1) [22] and use stochastic gradient estimators $\widehat{\operatorname{grad}} f_i(x_{t,\ell}^{(i)})$:

$$x_{t,\ell+1}^{(i)} \leftarrow R_{x_{t,\ell}^{(i)}}(-\eta \widehat{\operatorname{grad}} f_i(x_{t,\ell}^{(i)}) = \mathcal{P}_{\mathcal{M}}\big(x_{t,\ell}^{(i)} - \eta \widehat{\operatorname{grad}} f_i(x_{t,\ell}^{(i)})\big). \tag{6}$$

After completing τ local steps, client i sends its updated model $x_{t,\tau}^{(i)} \in \mathcal{M}$ to the server. Only model parameters are communicated – not raw data – thereby preserving data locality.

Server aggregation challenge. The server aggregates the received client models $\{x_{t,\tau}^{(i)}\}_{i \in S_t}$ into an updated global model $x_{t+1} \in \mathcal{M}$. In Euclidean FL, this is typically a simple arithmetic mean (FedAvg [1]):

$$x_{t+1} \leftarrow \frac{1}{k} \sum_{i \in S_t} x_{t,\tau}^{(i)}. \tag{7}$$

However, when the parameter space is a Riemannian manifold, naive Euclidean averaging is problematic: on the Stiefel manifold, the mean lies outside $\mathcal{M}$;

on SPD matrices, it remains in $\mathcal{M}$ but does not coincide with geometrically meaningful averages. A natural alternative is the Fréchet mean [17]:

$$x_{t+1} \leftarrow \arg\min_{x \in \mathcal{M}} \frac{1}{k} \sum_{i \in S_t} d^2(x, x_{t,\tau}^{(i)}), \tag{8}$$

which generally lacks a closed-form solution. Existing methods approximate it via one step of Karcher flow [18, 19]:

$$x_{t+1} \leftarrow \mathrm{Exp}_{x_t}\left(\frac{1}{k} \sum_{i \in S_t} \mathrm{Log}_{x_t}(x_{t,\tau}^{(i)}) \right), \tag{9}$$

or using gradient-stream aggregation [31], where each client accumulates stochastic gradients transported back to the tangent space at x_t:

$$x_{t+1} \leftarrow R_{x_t}\left(-\alpha_t \frac{1}{k} \sum_{i \in S_t} \zeta_\tau^{(i)} \right), \quad \text{with } \zeta_\tau^{(i)} = \sum_{s=0}^{\tau-1} \mathcal{T}_{x_{t,s}^{(i)} \to x_t}\left(g_{t,s}^{(i)} \right), \tag{10}$$

where $g_{t,s}^{(i)}$ is the stochastic gradient at client i's local iterate $x_{t,s}^{(i)}$ and $\mathcal{T}_{x_{t,s}^{(i)} \to x_t}$ denotes vector transport to $T_{x_t}\mathcal{M}$. This approach represents model displacements through accumulated stochastic gradients transported to a common tangent space – an expression that is exact in the Euclidean setting and constitutes a first-order approximation on manifolds, avoiding exponential and logarithmic maps entirely. However, the formulation is inherently tied to SGD: since $\zeta_\tau^{(i)}$ is constructed by summing gradient directions along the local path, it assumes updates follow the negative gradient, precluding alternative optimizers such as Adam or momentum-based methods.

These approaches face distinct limitations: Karcher flow (9) requires logarithmic and exponential maps lacking closed-form expressions on key manifolds; gradient-stream aggregation (10) avoids these primitives but restricts clients to SGD. Recent projection-based methods [29] are computationally efficient but rely on drift correction and full participation. We next propose two lightweight alternatives that aggregate final client iterates rather than gradient streams, requiring neither costly geometric operations nor drift correction, while supporting partial participation and arbitrary client optimizers.

3.3. Proposed server aggregation strategies

The federated aggregation (9) can be interpreted as computing a Riemannian barycenter of local model updates with respect to the current global

Table 2: Summary of notations.

Symbol	Description	Symbol	Description
$\mathcal{M}$	Compact submanifold of $\mathbb{R}^{d\times p}$	P_x	Projector onto $T_x\mathcal{M}$
$T_x\mathcal{M}$	Tangent space at x	$\mathcal{P}_{\mathcal{M}}$	Nearest-point projection onto $\mathcal{M}$
$\|\cdot\|$	Frobenius norm	$\operatorname{grad} f(x)$	Riemannian gradient at x
n	Total clients	τ	Local steps per round
k	Clients per round	η	Local learning rate
S_t	Selected clients at round t	T	Communication rounds
$F(x)$	Global objective $\frac{1}{n}\sum_{i=1}^n f_i(x)$	x_t	Global model at round t
$f_i(x)$	Local objective of client i	$x_{t,\ell}^{(i)}$	Local model of client i
$\mathcal{D}_i$	Data distribution of client i	$\widetilde{\operatorname{grad}} f_i$	Stochastic gradient estimator

iterate. Motivated by the analysis of retracted-lifted (RL) barycenters on compact submanifolds in [21], we propose two computationally lightweight alternatives that operate in the ambient Euclidean space and exploit the well-behaved projection operators guaranteed by proximal smoothness.

Let $x_t \in \mathcal{M}$ denote the current global iterate at round t, and let $\{x_{t,\tau}^{(i)}\}_{i\in S_t} \subset \mathcal{M}$ be the updated model parameters returned by the selected client subset S_t of size k.

RFedProj: Projection Averaging. The simplest approach computes the Euclidean mean of client iterates in the ambient space and projects the result onto $\mathcal{M}$:

$$x_{t+1}^{\mathrm{RFedProj}} \;\leftarrow\; \mathcal{P}_{\mathcal{M}}\left(\frac{1}{k}\sum_{i\in S_t} x_{t,\tau}^{(i)}\right). \tag{11}$$

This scheme requires no tangent-space computation and does not require the server to store the previous iterate x_t.

RFedRL: Retracted-Lifted Averaging. This scheme directly implements the federated analogue of the RL-barycenter construction: each client displacement $x_{t,\tau}^{(i)} - x_t$ is lifted to the tangent space $T_{x_t}\mathcal{M}$ via the orthogonal projector P_{x_t}, the lifts are averaged, and the result is mapped back to $\mathcal{M}$ via the projection-based retraction (1):

$$x_{t+1}^{\mathrm{RFedRL}} \;\leftarrow\; \mathcal{P}_{\mathcal{M}}\left(x_t + \frac{1}{k}\sum_{i\in S_t} P_{x_t}\big(x_{t,\tau}^{(i)} - x_t\big)\right). \tag{12}$$

Compared to gradient-stream aggregation (10), which reconstructs model displacements by summing vector-transported gradients along the local opti-

Algorithm 1 Riemannian FL Framework: RFedProj / RFedRL

Require: n: total clients; k: clients per round; τ: local steps; $\textsc{Agg} \in \{\textsc{RFedProj}, \textsc{RFedRL}\}$; $\textsc{LocalOpt}$: client optimizer

Initialize $x_0 \in \mathcal{M}$
for $t = 0$ to $T - 1$ **do**
  Sample $S_t \subset [n]$ uniformly without replacement, $|S_t| = k$
  for all $i \in S_t$ **in parallel do**
    $x_{t,\tau}^{(i)} \leftarrow \textsc{LocalOpt}(x_t, f_i, \mathcal{D}_i, \tau)$ ▷ Any Riemannian optimizer
    Client i sends $x_{t,\tau}^{(i)}$ to server
  $x_{t+1} \leftarrow \textsc{Agg}\big(\{x_{t,\tau}^{(i)}\}_{i \in S_t}, x_t\big)$ ▷ Eq. (11) or (12)
return x_T

mization path, RFedRL directly lifts the final displacement $x_{t,\tau}^{(i)} - x_t$ to the tangent space – capturing the cumulative effect of all local updates in a single projection, without vector transport. Both RFedProj and RFedRL preserve manifold feasibility while relying exclusively on standard linear algebra and nearest-point projections, avoiding Riemannian logarithms and exponential maps. Since the server aggregates only final client iterates and performs a single projection step, they are inherently optimizer-agnostic: each client may employ a distinct local solver – SGD, momentum methods, or adaptive algorithms such as Adam – and the aggregation remains well-defined regardless of how individual clients reached their final iterates. This decoupling is particularly valuable in heterogeneous deployments where clients have different computational capabilities or optimizer preferences [3, 5].

3.4. Algorithm

The proposed framework decouples server-side aggregation from client-side optimization. Algorithm 1 presents the generic protocol: clients perform τ local updates using any manifold-compatible optimizer, and the server aggregates via RFedProj or RFedRL.

For theoretical analysis (Section 4), we instantiate $\textsc{LocalOpt}$ with projected stochastic gradient descent:

$$x_{t,j+1}^{(i)} \leftarrow \mathcal{P}_{\mathcal{M}}\Big(x_{t,j}^{(i)} - \eta\, \widehat{\operatorname{grad}} f_i(x_{t,j}^{(i)})\Big). \tag{13}$$

While the aggregation is optimizer-agnostic, our convergence analysis assumes local SGD. Experiments use Adam, which falls outside the theory but reflects

common practice where adaptive methods outperform SGD despite lacking formal guarantees in the federated setting.

4. Convergence analysis

4.1. Assumptions

We state the geometric and statistical assumptions underlying our convergence analysis.

Assumption 4.1. The manifold $\mathcal{M}$ is compact and is 2γ-proximally smooth.

These conditions, standard in projection-based Riemannian optimization [30, 29], ensure the nearest-point projection $\mathcal{P}_{\mathcal{M}}$ is single-valued and smooth within the tube $U_{\mathcal{M}}(2\gamma)$. In our analysis, the step size η and number of local updates τ are chosen to keep all iterates within $\overline{U_{\mathcal{M}}(\gamma)} \subset U_{\mathcal{M}}(2\gamma)$ – the geometric analogue of a stability region in Euclidean optimization.

Assumption 4.2 (Stochastic gradients with minibatches)**.** For each client i, there is a data distribution $\mathcal{D}_i$ such that

$$f_i(x) \;=\; \mathbb{E}_{z\sim\mathcal{D}_i}\big[f_i(x;z)\big],$$

and the algorithm observes stochastic gradients via minibatches of size B:

$$\widehat{\operatorname{grad}} f_i(x) \;=\; \frac{1}{B}\sum_{b=1}^{B} \operatorname{grad} f_i(x;z_b), \qquad z_b \sim_{\text{i.i.d.}} \mathcal{D}_i.$$

We assume for all i, all $x \in \mathcal{M}$, and all minibatches:

(a) **Unbiasedness:** $\mathbb{E}[\widehat{\operatorname{grad}} f_i(x) \mid x] = \operatorname{grad} f_i(x)$.

(b) **Bounded variance (per sample):** there exists $\sigma_{\text{sg}}^2 < \infty$ s.t.

$$\mathbb{E}\big[\|\widehat{\operatorname{grad}} f_i(x) - \operatorname{grad} f_i(x)\|^2 \mid x\big] \;\le\; \frac{\sigma_{\text{sg}}^2}{B}.$$

(c) **Uniform boundedness:** the per-sample gradients are uniformly bounded,

$$\max_i \sup_{x\in\mathcal{M},z} \|\operatorname{grad} f_i(x;z)\| \;\le\; G.$$

In particular, $\|\widehat{\operatorname{grad}} f_i(x)\| \le G$ almost surely.

These conditions are standard in stochastic optimization on manifolds [37, 38] and federated learning [4]. We also assume that, conditional on the past, client subsampling S_t and all minibatch draws $\{\mathcal{B}_{t,j}^{(i)}\}$ are independent across t, across i, and across j.

We state the following tube smoothness assumption, similar to Assumption 4.1 in [26].

Assumption 4.3 (Tube-local smoothness)**.** Each client objective f_i admits a $\mathcal{C}^1$ extension to $U_{\mathcal{M}}(2\gamma)$ with Euclidean gradient ℓ_i-Lipschitz:

$$\|\nabla f_i(x) - \nabla f_i(y)\| \leq \ell_i \|x - y\| \qquad \forall x, y \in U_{\mathcal{M}}(2\gamma).$$

Under Assumption 4.3, the global objective F has its gradient $\bar{\ell}$-Lipschitz, where $\bar{\ell} := \frac{1}{n}\sum_{i=1}^n \ell_i$. Requiring a C^1 extension with Euclidean ℓ_i-Lipschitz gradients lets us apply standard smooth analysis to both client-side drifts and server-side averaging.

Assumption 4.4 (Global objective lower bound)**.** The global objective F is such that

$$F_* := \inf_{x \in \mathcal{M}} F(x) > -\infty.$$

To quantify data heterogeneity across clients, we adopt the bounded gradient dissimilarity assumption, standard in federated optimization [4, 34].

Assumption 4.5 (Client-gradient heterogeneity)**.** There exists $\sigma^2 \geq 0$ such that

$$\sup_{x \in \mathcal{M}} \frac{1}{n} \sum_{i=1}^n \left\| \operatorname{grad} f_i(x) - \operatorname{grad} F(x) \right\|^2 \leq \sigma^2.$$

The parameter σ^2 quantifies how far individual client gradients deviate from the global gradient at any point on $\mathcal{M}$.

4.2. Main result

Theorem 4.6 shows that both RFedProj (11) and RFedRL (12) achieve the same convergence rate, with guarantees that parallel classical Euclidean results [34, 39, 4] while improving upon existing Riemannian FL methods [29, 18]. Indeed, compared to prior work: (i) we avoid the exponential maps, logarithms, and parallel transport required by [18, 19], using only projections; (ii) we support multiple local epochs ($\tau > 1$), whereas [18, 19] guarantee

convergence only for $\tau = 1$; (iii) unlike [29, 30], we handle partial participation without drift correction; and (iv) unlike gradient-stream methods [31], our bounds hold for any client optimizer.

Theorem 4.6. *Under the previous assumptions, for both RFedProj and RFedRL, pick*

$$\alpha_T = \min\{\gamma/(4G),\, 1/(4L_{\mathcal{M}}),\, c/\sqrt{T}\}, \quad c > 0.$$

Then

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

The bound decomposes into three terms: (i) an optimization error decaying at rate $O(1/\sqrt{T})$, (ii) a heterogeneity term σ^2 scaled by the participation ratio $(n-k)/((n-1)k)$, and (iii) stochastic gradient variance σ_{sg}^2/B reduced by minibatch averaging. The factor $(n-k)/((n-1)k)$ captures the communication-convergence trade-off: it vanishes under full participation ($k = n$), equals 1 for single-client sampling ($k = 1$), and scales as $\approx 1/k$ for $k \ll n$. The $\mathcal{O}(1/\sqrt{T})$ rate requires $T \geq T_0 := 16c^2 \max\{G^2/\gamma^2, L_{\mathcal{M}}^2\}$ with $L_{\mathcal{M}} = \bar{\ell} + G/(2\gamma)$; for smaller T, the stepsize is constrained by geometry and convergence proceeds at $\mathcal{O}(1/T)$. All implicit constants depend polynomially on problem parameters (G, $\bar{\ell}$, γ, $L_{\mathrm{d}\mathcal{P}}$) but are independent of n, k, and T.

4.3. Proof outline

Throughout the analysis, the key quantity governing both tube stability and convergence rate is the aggregate stepsize $\alpha := \eta\tau$, representing the total step taken per communication round. Our convergence analysis proceeds in four main steps. The detailed proofs can be found in Supplementary material (Sections S2–S5).

4.3.1. Tube-stability and linearization

By Assumption 4.1, the projection $\mathcal{P}_{\mathcal{M}}$ is single-valued and smooth on $U_{\mathcal{M}}(2\gamma)$. The stepsize constraint $\alpha \leq \gamma/(4G)$ ensures that all client iterates and their averages remain within the smaller tube $\overline{U_{\mathcal{M}}(\gamma)} \subset U_{\mathcal{M}}(2\gamma)$, providing a safety margin that guarantees well-posedness of all projection operations throughout the algorithm.

Lemma 4.7 (Tube-stability and linearization)**.** *If $\alpha \le \gamma/(4G)$, then for all rounds t, clients $i \in S_t$, and local steps $j \in \{0, \dots, \tau\}$:*

(a) ***Path length:*** *$\|x_{t,j}^{(i)} - x_t\| \le 2G\eta j \le \gamma/2$.*

(b) ***Feasibility:*** *All retractions $x_{t,j+1}^{(i)} = \mathcal{P}_{\mathcal{M}}\Big(x_{t,j}^{(i)} - \eta\, \widehat{\operatorname{grad}} f_i(x_{t,j}^{(i)})\Big)$ are uniquely defined.*

(c) ***Linearization:*** *Let $\zeta_{t,j}^{(i)} := \widehat{\operatorname{grad}} f_i(x_{t,j}^{(i)}) - \operatorname{grad} f_i(x_{t,j}^{(i)})$. We have*

$$x_{t,\tau}^{(i)} - x_t = -\eta \sum_{j=0}^{\tau-1} \operatorname{grad} f_i(x_t) - \eta \sum_{j=0}^{\tau-1} \zeta_{t,j}^{(i)} + O(\alpha^2(1+\ell_i)). \tag{14}$$

(d) ***Aggregation feasibility:*** *Both $s_t := \frac{1}{k}\sum_{i\in S_t} x_{t,\tau}^{(i)}$ and $u_t := \frac{1}{k}\sum_{i\in S_t} P_{x_t}(x_{t,\tau}^{(i)} - x_t)$ are such that $\|s_t - x_t\| \le 2G\alpha, \|u_t\| \le 2G\alpha$ and therefore lie within distance $\gamma/2$ of $\mathcal{M}$, so $\mathcal{P}_{\mathcal{M}}(s_t)$ and $\mathcal{P}_{\mathcal{M}}(x_t + u_t)$ are uniquely defined.*

The result (14) overcomes the loss of linearity at *client level*, created by the use of true exponential and logarithmic maps in [18, 19]. Indeed, in their approach, the main obstacle to tackling $\tau > 1$ with multiple selected clients is the combination of non-linear key quantities:

$$\operatorname{Log}_{x_{t,j}^{(i)}}(x_{t,j+1}^{(i)}) \leftarrow -\eta \operatorname{grad} f_i(x_j^{(i)}).$$

In our approach, the linear approximation given by the retraction with the quadratic remainder characterization (Supplementary Material, Lemma S1.2).

$$\mathcal{P}_{\mathcal{M}}(x+u) = x + P_x u + O(\|u\|^2) \tag{15}$$

enables to bypass these difficulties and obtain (14) via telescoping.

4.3.2. Bias quantification

Recall that

$$x_{t+1}^{\text{RFedProj}} = \mathcal{P}_{\mathcal{M}}\left(\frac{1}{k}\sum_{i\in S_t} x_{t,\tau}^{(i)}\right), \quad x_{t+1}^{\text{RFedRL}} = \mathcal{P}_{\mathcal{M}}\left(x_t + \frac{1}{k}\sum_{i\in S_t} P_{x_t}(x_{t,\tau}^{(i)} - x_t)\right).$$

We express the server update displacement as $-\alpha \cdot (\text{sample gradient}) + (\text{bias})$, where the bias arises from curvature and gradient drift during local steps. The two aggregation schemes (RFedRL and RFedProj) differ only in higher-order curvature terms.

Lemma 4.8 (Aggregation bias)**.** *Under $\alpha \leq \gamma/(4G)$, both RFedRL and RFedProj satisfy*

$$x_{t+1} - x_t = -\alpha \frac{1}{k} \sum_{i \in S_t} \operatorname{grad} f_i(x_t) + \bar{b}_t + \xi_t,$$

where $\mathbb{E}_t[\|\bar{b}_t\|] = O(\alpha^2), \mathbb{E}_t[\|\bar{b}_t\|^2] = O(\alpha^4)$, *and* $\mathbb{E}_t[\xi_t] = 0, \mathbb{E}_t[\|\xi_t\|^2] = O\left(\frac{\alpha^2}{kB}\sigma_{sg}^2\right)$.

This results bypasses the loss of linearity at *server level* occuring when using true exponential and logarithmic maps (see [18, 19]):

$$\operatorname{Log}_{x_t}(x_{t+1}) \leftarrow \frac{1}{k} \sum_{i \in S_t} \operatorname{Log}_{x_t}(x_{t,\tau}^{(i)})$$

Keeping the same notation as in Lemma 4.7, let us denote by s_t the extrincic average $\frac{1}{k}\sum_{i \in S_t} x_{t,\tau}^{(i)}$ and u_t the tangent average $\frac{1}{k}\sum_{i \in S_t} P_{x_t}(x_{t,\tau}^{(i)} - x_t)$. Establishing Lemma 4.8 relies on (14) from Lemma 4.7 (c) and the central following identity:

$$P_{x_t}(s_t - x_t) = P_{x_t}\left(\frac{1}{k}\sum_{i \in S_t} x_{t,\tau}^{(i)} - x_t\right) = \frac{1}{k}\sum_{i \in S_t} P_{x_t}(x_{t,\tau}^{(i)} - x_t) = u_t, \tag{16}$$

so that the quadratic remainder decomposition (15) yields the same bias decomposition for both aggregation strategies RFedProj and RFedRL. Indeed, since $s_t = x_t + (s_t - x_t)$,

$$x_{t+1}^{\text{RFedProj}} - x_t = \mathcal{P}_{\mathcal{M}}(s_t) - x_t = P_{x_t}(s_t - x_t) + O(\|s_t - x_t\|^2) \tag{17}$$

and

$$x_{t+1}^{\text{RFedRL}} - x_t = \mathcal{P}_{\mathcal{M}}(x_t + u_t) - x_t = u_t + O(\|u_t\|^2), \tag{18}$$

with $O(\|u_t\|^2) = O(\|s_t - x_t\|^2) = O(\alpha^2)$ using inequalities from Lemma 4.7 (d).

4.3.3. One-round expected descent

In order to obtain a descent inequality, we rely on the fact (Lemma 4.2 in [26]) that, under our assumptions, there exists an explicit geometric constant $L_{\mathcal{M}} > 0$ such that

$$F(x_{t+1}) \leq F(x_t) + \langle \operatorname{grad} F(x_t), x_{t+1} - x_t \rangle + \frac{L_{\mathcal{M}}}{2}\|x_{t+1} - x_t\|^2, \tag{19}$$

which is also central to the approaches carried out in [29, 30]. Inequality (19) generalizes the classical inequality

$$F(x_{t+1}) \leq F(x_t) + \langle \operatorname{grad} F(x_t), \log_{x_t}(x_{t+1}) \rangle + \frac{L}{2} \| \log_{x_t}(x_{t+1}) \|^2, \tag{20}$$

used in [18, 19] without resorting to exponential and logarithmic mappings. Combining manifold smoothness with controlled bias and sampling variance, we establish a per-round descent inequality in expectation that explicitly separates the contributions of global gradient, client heterogeneity, and aggregation bias.

Lemma 4.9 (One-round expected descent)**.** *Assume* $\alpha \leq \min\left\{ \frac{\gamma}{4G}, \frac{1}{4L_{\mathcal{M}}} \right\}$. *Then, for either RFedRL or RFedProj aggregation, conditioning on* $\mathcal{F}_t$ *we have*

$$\mathbb{E}_t[F(x_{t+1})] \;\leq\; F(x_t) \;-\; \frac{\alpha}{4} \| \operatorname{grad} F(x_t) \|^2 \;+\; O\!\left(\alpha^2 \left(\frac{n-k}{k(n-1)} \, \sigma^2 + \frac{\sigma_{sg}^2}{kB} \right) \right) \\ +\; O(\alpha^3) + \; O(\alpha^4), \tag{21}$$

where $L_{\mathcal{M}} = \bar{\ell} + \frac{G}{2\gamma}$.

4.3.4. Telescoping and stationarity

Summing the one-round descent over T rounds and applying Young's inequality to absorb the linear gradient term yields the $\mathcal{O}(1/\sqrt{T})$ stationarity rate with explicit dependence on partial participation ratio $(n-k)/(k(n-1))$.

Lemma 4.10 (Stationarity from one-round descent)**.** *Let* $\alpha = \eta\tau$ *satisfy the stepsize cap of Lemma 4.9,* $\alpha \leq \min\left\{ \frac{\gamma}{4G}, \frac{1}{4L_{\mathcal{M}}} \right\}$, *and assume* F *is bounded below on* $\mathcal{M}$ *with* $F_\star := \inf_{x \in \mathcal{M}} F(x) > -\infty$. *Then for any* $T \geq 1$,

$$\frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}\big[\| \operatorname{grad} F(x_t) \|^2 \big] \;\leq\; \frac{4\,(F(x_0) - F_\star)}{\alpha\, T} + O\!\left(\alpha \left(\frac{n-k}{k(n-1)} \sigma^2 + \frac{\sigma_{sg}^2}{kB} \right) \right) \\ +\; O(\alpha^2) + \; O(\alpha^3). \tag{22}$$

The proofs of the previous four key Lemmas can be found in the Supplementary material (Sections S2–S5). With Lemma 4.10 in hand, Theorem 4.6 follows by setting $\alpha_T = c/\sqrt{T}$, which satisfies the stepsize cap for sufficiently large T.

5. Numerical experiments

We study the proposed framework on EEG-based motor imagery classification, a task where data confidentiality concerns naturally motivate federated learning and where manifold geometry plays a central role. Indeed, spatial covariance matrices have proven to be highly effective descriptors for EEG signals [7], motivating the use of geometric methods that respect their SPD structure. Using three datasets from the MOABB benchmark [35], we compare two architectures: EEGNet [36], a convolutional network operating on raw signals in Euclidean space, and SPDNet [10], a geometric deep network operating on SPD covariance matrices whose layer parameters are constrained to the Stiefel manifold. This setup allows us to assess our projection-based aggregation schemes in both Euclidean and Riemannian settings.

Traditional BCI pipelines [40, 41] typically employ within-subject cross-validation, using calibration data from each user to build personalized decoders. In contrast, we target population-level models that generalize across subjects without per-user calibration, reflecting realistic federated scenarios where new clients may join with minimal or no local data.

To disentangle the challenges of cross-subject generalization from those introduced by federation, we consider two complementary settings. First, a *centralized baseline* (Section 5.3) pools all available data, establishing an upper bound on achievable performance when data sharing is unrestricted. Second, a *federated setting* (Section 5.4) distributes data across clients that never share raw recordings, allowing us to evaluate whether our projection-based aggregation schemes can approach centralized performance while preserving data locality.

5.1. Datasets

We evaluate on three single-session motor imagery EEG datasets from the MOABB benchmark [35], summarized in Table 3: Weibo2014 [42] provides a challenging 7-class task with 60 channels and 10 subjects; Schirrmeister2017 [43] features high-density 128-channel recordings from 14 subjects with a standard 4-class paradigm; and PhysionetMI [44] offers the largest subject pool (106 subjects, 64 channels, 4 classes), enabling scalability evaluation. Three PhysionetMI subjects were excluded due to incomplete recordings.

Each trial (X_j, y_j) consists of a multichannel EEG recording $X_j \in \mathbb{R}^{n_{\text{chan}} \times \mathcal{T}}$ where $\mathcal{T}$ denotes the number of time samples, and class label $y_j \in \{1, \ldots, n_{\text{cl}}\}$.

Table 3: Overview of EEG motor imagery datasets. Trials/class denotes the approximate number per subject. S.R. denotes the Sampling Rate.

Dataset	n_{subj}	n_{chan}	n_{cl}	S.R. (Hz)	Trials/class	Epoch (s)
Weibo2014	10	60	7	200	80	$[3, 7]$
Schirrmeister2017	14	128	4	500	120	$[0, 4]$
PhysionetMI	106	64	4	160	23	$[0, 3]$

All datasets undergo identical preprocessing: signals are band-pass filtered in $[8, 32]$ Hz to isolate motor imagery-related frequency bands.

5.2. Model architectures

We employ two complementary architectures operating on different input representations: EEGNet processes raw temporal signals while SPDNet operates on SPD covariance matrices. Both architectures are trained using identical optimization protocols (Table 4) in centralized (Section 5.3) and federated (Section 5.4) settings.

Table 4: Training hyperparameters for all experiments.

Parameter	**Value**
Batch size	64
Optimizer	Adam
Initial learning rate	10^{-3}
Loss function	Cross-entropy
LR scheduler	ReduceLROnPlateau (patience 20, factor 0.5)
Random seeds	10 independent runs

EEGNet [36] is a compact convolutional neural network designed for EEG signals, operating directly on raw trials $X_j \in \mathbb{R}^{n_{\text{chan}} \times \mathcal{T}}$ after per-trial, per-channel z-score normalization. As illustrated in Figure 2, the architecture comprises two convolutional blocks (detailed in Supplementary Material, Figure S1) followed by a classification head. Block 1 applies temporal convolution to extract frequency-specific features, followed by depthwise spatial convolution to capture electrode correlations. Block 2 refines features via separable convolutions. To accommodate varying sampling rates across datasets, we set the temporal kernel size to $K_t = \max(\lfloor 0.5 \cdot f_s \rceil, 32)$, ensuring a consistent receptive field of approximately 500 ms.

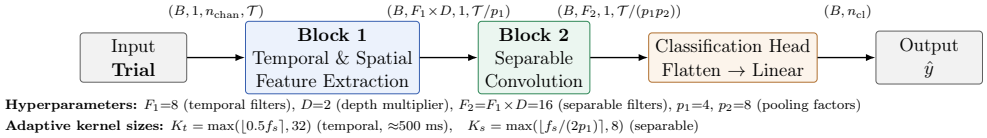

Figure 2: EEGNet overall pipeline

SPDNet [10] operates on symmetric positive definite (SPD) sample covariance matrices (Figure 3). Given a centered trial $\bar{X}_j$, the input covariance is computed as $\Sigma_j = \frac{1}{\mathcal{T}-1} \bar{X}_j \bar{X}_j^\top \in \mathcal{S}^{++}_{n_{\text{chan}}}$. The architecture considered consists of a single BiMap layer $\Sigma \mapsto W^\top \Sigma W$ with $W \in \text{St}(n_{\text{chan}}, d)$, which reduces dimensionality while preserving SPD structure; a ReEig layer that rectifies eigenvalues below threshold $\varepsilon > 0$ to maintain positive definiteness; and a LogEig layer that maps to the tangent space via $\Sigma = U \Lambda U^\top \mapsto U \log(\Lambda) U^\top$. The final classifier computes $\hat{y} = \text{softmax}(\xi \, \text{vec}(\Sigma) + \beta)$, where $\text{vec}(\cdot)$ denotes full vectorization. The learnable parameters (W, ξ, β) are optimized end-to-end with Adam [32]: Stiefel weight W via tangent-space local trivialization [45], softmax parameters ξ, β in the standard Euclidean way. Backpropagation through eigendecomposition follows [46]; Table 5 details the per-dataset configuration.

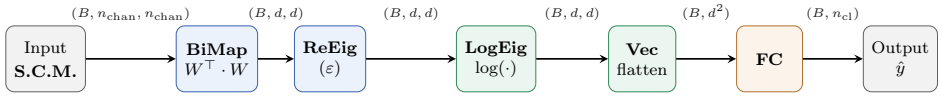

Figure 3: SPDNet pipeline: BiMap-ReEig block on the SPD manifold, followed by LogEig, vectorization, and a fully-connected classifier. Tensor dimensions are shown as $(B, \cdot, \cdot)$ where B denotes batch size, n_{chan} the number of channels, and d the reduced dimension after BiMap.

Following preliminary studies on SPDNet hyperparameters on each dataset (see Supplementary Material, Section S7), we consider the configurations detailed in Table 5.

Table 5: SPDNet architecture configuration per dataset.

Dataset	**Hidden dim** d	ϵ **(ReEig)**
Weibo2014	22	10^{-1}
Schirrmeister2017	24	10^{-2}
PhysionetMI	18	10^{-2}

Formally, let $\mathcal{D} = \{(Z_j, y_j) \in \mathcal{X} \times \{1, \ldots, n_{\text{cl}}\}\}_{j=1}^{N}$ denotes a dataset of N labeled samples, where for EEGNet, $Z_j^{(i)} = X_j^{(i)}, \mathcal{X} = \mathbb{R}^{n_{\text{chan}} \times \mathcal{T}}$, for SPDNet, $Z_j^{(i)} = \Sigma_j^{(i)}, \mathcal{X} = \mathcal{S}_{n_{\text{chan}}}^{++}$ and $y_j^{(i)}$ is the corresponing class label. Given a model forward pass f_θ with learnable parameters θ, the empirical risk on $\mathcal{D}$ is the cross-entropy loss $F(\theta; \mathcal{D}) = -\frac{1}{|\mathcal{D}|} \sum_{j=1}^{|\mathcal{D}|} \sum_{k=1}^{n_{\text{cl}}} \mathbf{1}_{y_j=k} \log\big(f_\theta(Z_j)_k\big)$, where $f_\theta(Z_j)_k$ denotes the k-th softmax output.

5.3. Centralized baselines

To establish a centralized baseline representing unrestricted data sharing, we pool all subjects and perform a stratified split into training (75%), validation (10%), and test (15%) sets, with each subject contributing to all three. Training proceeds for at most 300 epochs with early stopping (patience 75 epochs), retaining the checkpoint with lowest validation loss [47]. Table 6 summarizes the results.

Table 6: Centralized test F1 (%) on EEG motor imagery datasets. Mean ± std over 10 seeds. Best results per dataset in **bold**.

Dataset	**Model**	**#Parameters**	**Centralized**
Weibo2014	EEGNet	5,303	50.7 ± 1.5
	SPDNet	4,715	$\mathbf{51.7 \pm 0.8}$
Schirrmeister2017	EEGNet	9,348	$\mathbf{81.5 \pm 0.9}$
	SPDNet	5,380	74.5 ± 0.6
PhysionetMI	EEGNet	3,284	$\mathbf{50.8 \pm 0.7}$
	SPDNet	2,452	43.1 ± 0.5

5.4. Federated learning setup

We simulate a multi-institutional setting by partitioning data across n clients. To disentangle the effects of federation from data heterogeneity, we consider two partitioning strategies:

- **IID partitioning.** A single global stratified split is performed on the pooled data, matching the centralized baseline exactly, before uniform distribution to clients. Each client receives a representative mixture of all subjects and classes, eliminating client-level distribution shift.

- **Subject-based partitioning (non-IID).** Each client receives data from a disjoint subset of subjects assigned sequentially, and independently splits its local data. This captures realistic inter-institutional heterogeneity arising from subject-specific EEG characteristics.

Comparing these two modes quantifies the performance gap attributable to non-IID data distributions. The number of clients is chosen to balance data across clients in the subject-based (non-IID) setting, and kept identical for the IID baseline to ensure fair comparison.

Each client $i \in \{1, \dots, n\}$ has only access to its own dataset

$$\mathcal{D}_i := \left\{ (Z_j^{(i)}, y_j^{(i)}) \in \mathcal{X} \times \{1, \dots, n_{\text{cl}}\} \right\}_{j=1}^{|\mathcal{D}_i|}, \quad \text{with} \quad \mathcal{D} = \bigsqcup_{i=1}^{n} \mathcal{D}_i.$$

It is split into training (75%), validation (10%), and test (15%) sets using stratified sampling. Given learnable parameters θ, let us denote by $F_i(\theta) := F(\theta; \mathcal{D}_i)$ the local empirical risk for client i. On the server, the objective is to minimize the global empirical risk over all clients $\arg\min_\theta \ \frac{1}{N} \sum_{i=1}^{N} F_i(\theta)$.

Training proceeds over T communication rounds. At each round t, a subset $\mathcal{S}_t \subseteq \{1, \dots, N\}$ of k clients participates: (1) the server broadcasts global parameters θ_t; (2) each client $i \in \mathcal{S}_t$ performs τ local epochs, yielding $\theta_{t,\tau}^{(i)}$; (3) the server aggregates updates to obtain θ_{t+1}.

To examine partial participation effects, we consider participation rates $\rho = k/n \in \{1.0, 0.8, 0.5, 0.2\}$, where clients are sampled uniformly without replacement each round. We set $T = 150$ rounds with $\tau = 2$ local epochs, matching the centralized budget of 300 epochs. Setting $\tau > 1$ reduces communication frequency but amplifies client drift under heterogeneous data, as the bias term in Lemma 4.10 scales with $\alpha = \eta\tau$. We do not employ early stopping to ensure consistent comparison across strategies.

We report macro-averaged F1 on a pooled test set aggregating all local test sets, evaluated at every round to monitor convergence.

Server aggregation strategies. For FedEEGNet, we use FedAvg [1] with batch normalization statistics excluded from aggregation, as local statistics better capture client-specific distributions [48].

For FedSPDNet, aggregation is hybrid: BiMap weights $W \in \text{St}(n_{\text{chan}}, d)$ use Riemannian strategies from Section 3.3, while classifier parameters (ξ, β) use standard averaging. Table 7 summarizes the update rules, where $\text{uf}(\cdot)$ denotes the orthogonal polar factor (3) and P_{W_t} the tangent projector (2).

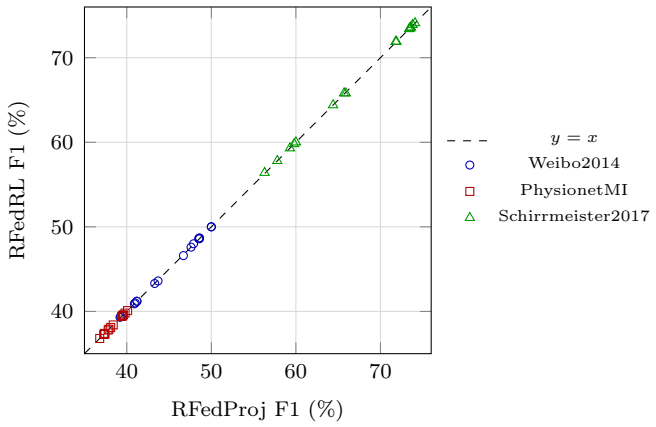

Figure 4: RFedProj vs RFedRL F1 scores across all configurations. Points lie on the diagonal, confirming equivalence (max $|\Delta\text{F1}| < 0.2\%$).

Table 7: Server aggregation strategies for FedSPDNet, with $k = |\mathcal{S}_t|$ participating clients.

Parameter	**Strategy**	**Update rule**
$W \in \mathrm{St}(n_{\text{chan}}, d)$	RFedProj	$W_{t+1} = \mathrm{uf}\big(\frac{1}{k}\sum_{i\in\mathcal{S}_t} W_{t,\tau}^{(i)}\big)$
	RFedRL	$W_{t+1} = \mathrm{uf}\big(W_t + \frac{1}{k}\sum_{i\in\mathcal{S}_t} P_{W_t}(W_{t,\tau}^{(i)} - W_t)\big)$
ξ, β	FedAvg	$\xi_{t+1} = \frac{1}{k}\sum_{i\in S_t} \xi_{t,\tau}^{(i)},\ \beta_{t+1} = \frac{1}{k}\sum_{i\in S_t} \beta_{t,\tau}^{(i)}$

Both strategies share asymptotic complexity $O(kn_{\text{chan}}d + n_{\text{chan}}d^2)$ for aggregating k matrices and computing the polar factor via thin SVD. However, RFedProj incurs lower constant factors, requiring only averaging followed by polar retraction, whereas RFedRL additionally computes tangent projections and requires storing W_t.

5.5. Results and analysis

Aggregation scheme equivalence. RFedProj and RFedRL yield equivalent results: F1 scores differ by less than 0.2% (Figure 4) with near-identical convergence curves (Supplementary Material, Figures S3–S5), validating Theorem 4.6. Given RFedProj's lower computational overhead, we adopt it as the default for FedSPDNet.

Robustness to federation overhead. FedSPDNet exhibits systematically lower degradation relative to its centralized baseline (Supplementary Material,

Table 8: Test F1 scores (%) for federated learning on EEG motor imagery datasets. Results are mean ± std over 10 seeds; best per row in **bold**. Centralized baselines: Weibo2014 (EEGNet 50.7 ± 1.5, SPDNet $\mathbf{51.7 \pm 0.8}$), PhysionetMI (EEGNet $\mathbf{50.8 \pm 0.7}$, SPDNet 43.1 ± 0.5), Schirrmeister2017 (EEGNet $\mathbf{81.5 \pm 0.9}$, SPDNet 74.5 ± 0.6).

			IID		**Non-IID**	
Dataset	n	**Part.**	FedEEGNet	FedSPDNet	FedEEGNet	FedSPDNet
Weibo2014	5	100%	44.1 ± 2.1	$\mathbf{50.0 \pm 1.6}$	39.9 ± 2.4	$\mathbf{43.3 \pm 1.0}$
		80%	42.9 ± 1.6	$\mathbf{50.0 \pm 1.8}$	39.0 ± 2.2	$\mathbf{43.7 \pm 1.3}$
		50%	40.9 ± 2.1	$\mathbf{48.6 \pm 1.7}$	36.4 ± 4.1	$\mathbf{41.2 \pm 0.8}$
		20%	37.5 ± 3.1	$\mathbf{47.6 \pm 2.0}$	32.9 ± 4.1	$\mathbf{39.2 \pm 2.3}$
	10	100%	39.9 ± 1.3	$\mathbf{48.6 \pm 1.7}$	37.1 ± 2.7	$\mathbf{41.0 \pm 1.0}$
		80%	39.4 ± 1.0	$\mathbf{48.5 \pm 1.8}$	36.2 ± 2.4	$\mathbf{41.2 \pm 0.9}$
		50%	38.8 ± 1.8	$\mathbf{47.9 \pm 1.5}$	34.3 ± 3.7	$\mathbf{40.9 \pm 0.7}$
		20%	34.4 ± 2.8	$\mathbf{46.7 \pm 1.6}$	32.8 ± 3.3	$\mathbf{39.3 \pm 1.9}$
PhysionetMI	53	100%	40.1 ± 1.8	$\mathbf{40.1 \pm 0.9}$	38.3 ± 1.8	$\mathbf{39.5 \pm 0.6}$
		80%	$\mathbf{40.2 \pm 1.7}$	39.8 ± 0.9	38.0 ± 2.6	$\mathbf{39.4 \pm 0.7}$
		50%	$\mathbf{40.3 \pm 2.0}$	39.6 ± 0.8	38.0 ± 2.7	$\mathbf{39.5 \pm 0.9}$
		20%	38.9 ± 1.3	$\mathbf{39.5 \pm 0.8}$	37.9 ± 2.3	$\mathbf{38.4 \pm 1.1}$
	106	100%	33.8 ± 2.4	$\mathbf{38.1 \pm 1.1}$	30.0 ± 2.3	$\mathbf{37.3 \pm 0.6}$
		80%	33.9 ± 2.6	$\mathbf{37.8 \pm 1.0}$	30.2 ± 2.8	$\mathbf{37.3 \pm 0.6}$
		50%	33.6 ± 2.2	$\mathbf{37.9 \pm 0.9}$	31.5 ± 2.8	$\mathbf{37.4 \pm 0.7}$
		20%	33.7 ± 2.6	$\mathbf{37.4 \pm 1.3}$	31.4 ± 2.0	$\mathbf{36.8 \pm 0.9}$
Schirrmeister2017	7	100%	$\mathbf{76.5 \pm 1.1}$	74.1 ± 0.7	$\mathbf{70.2 \pm 1.2}$	65.9 ± 0.7
		80%	$\mathbf{75.7 \pm 1.1}$	73.8 ± 0.7	$\mathbf{67.9 \pm 2.9}$	65.7 ± 0.9
		50%	$\mathbf{75.3 \pm 1.1}$	73.6 ± 0.6	$\mathbf{65.8 \pm 3.0}$	64.4 ± 0.8
		20%	69.9 ± 4.1	$\mathbf{71.8 \pm 0.8}$	$\mathbf{58.1 \pm 5.6}$	57.8 ± 3.4
	14	100%	$\mathbf{74.3 \pm 1.6}$	73.5 ± 0.4	$\mathbf{62.1 \pm 2.7}$	59.8 ± 1.0
		80%	73.4 ± 2.3	$\mathbf{73.5 \pm 0.6}$	$\mathbf{62.0 \pm 2.2}$	60.0 ± 1.2
		50%	$\mathbf{73.6 \pm 1.0}$	73.4 ± 0.7	$\mathbf{62.1 \pm 2.5}$	59.3 ± 1.6
		20%	69.2 ± 1.3	$\mathbf{71.9 \pm 0.5}$	$\mathbf{58.5 \pm 2.6}$	56.3 ± 2.6

Figure S6): 3–24% versus 13–35% for FedEEGNet on Weibo2014, 7–15% versus 21–41% on PhysionetMI, and 1–4% versus 6–15% on Schirrmeister2017 (IID). This robustness persists even when FedSPDNet achieves lower absolute F1, suggesting Riemannian aggregation provides geometric regularization absent in Euclidean averaging.

Scalability with client count. FedSPDNet scales more gracefully with federation size. Doubling clients on PhysionetMI ($n = 53 \rightarrow 106$) increases FedEEGNet's degradation by 12–16 pp versus 4–5 pp for FedSPDNet, with similar patterns on other datasets.

Robustness to partial participation. Higher participation generally improves performance in small federations. At large scale ($n = 106$), however, FedEEG-

Net exhibits non-monotonic behavior under non-IID conditions – 50% participation outperforms full participation, likely due to gradient divergence [4] – while FedSPDNet remains stable (37.3–37.4% F1). Across datasets, FedSPDNet consistently exhibits 1.5–3× lower variance across seeds under non-IID, indicating more stable convergence valuable for clinical deployments.

Communication efficiency. SPDNet requires 11–42% fewer parameters (Table 6), reducing per-round communication. FedSPDNet achieves superior early-round performance (34% F1 within 50 rounds vs 21–29% for FedEEGNet on PhysionetMI), though FedEEGNet exhibits steeper late-phase improvement. FedSPDNet is thus preferable in communication-constrained settings; architecture selection should also consider the channel-to-subject ratio.

6. Conclusion

We introduced RFedProj and RFedRL, two projection-based Riemannian federated learning methods for compact submanifolds that eliminate exponential maps, logarithms, and parallel transport while supporting partial participation and multiple local epochs without drift correction. Both achieve $\mathcal{O}(1/\sqrt{T})$ convergence with explicit heterogeneity and participation dependence, extending classical Euclidean federated bounds to the Riemannian setting. Experiments on EEG motor imagery classification validate theoretical predictions: the two aggregation schemes perform equivalently, favoring the simpler RFedProj. Compared to Euclidean FedAvg, FedSPDNet exhibits lower federation overhead, better scalability, greater robustness to partial participation, and reduced variance – properties valuable for multi-site clinical deployments. Open directions include extending the theory to adaptive optimizers, non-compact geometries such as the SPD manifold, and differential privacy guarantees.

Acknowledgments

This research was supported by DATAIA Convergence Institute as part of the "Programme d'Investissement d'Avenir", (ANR-17-CONV-0003) operated by L2S.

References

[1] B. McMahan, E. Moore, D. Ramage, S. Hampson, B. A. y Arcas, Communication-efficient learning of deep networks from decentralized data, in: Artificial intelligence and statistics, PMLR, 2017, pp. 1273–1282.

[2] J. Konečný, H. B. McMahan, D. Ramage, P. Richtárik, Federated optimization: Distributed machine learning for on-device intelligence, arXiv preprint arXiv:1610.02527 (2016).

[3] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, V. Smith, Federated optimization in heterogeneous networks, Proceedings of Machine learning and systems 2 (2020) 429–450.

[4] S. P. Karimireddy, S. Kale, M. Mohri, S. Reddi, S. Stich, A. T. Suresh, Scaffold: Stochastic controlled averaging for federated learning, in: International conference on machine learning, PMLR, 2020, pp. 5132–5143.

[5] P. Kairouz, H. B. McMahan, B. Avent, A. Bellet, M. Bennis, A. N. Bhagoji, K. Bonawitz, Z. Charles, G. Cormode, R. Cummings, et al., Advances and open problems in federated learning, Foundations and Trends® in Machine Learning 14 (1–2) (2021) 1–210.

[6] K. Bonawitz, H. Eichner, W. Grieskamp, D. Huba, A. Ingerman, V. Ivanov, C. Kiddon, J. Konečnỳ, S. Mazzocchi, B. McMahan, et al., Towards federated learning at scale: A system design, in: Proceedings of Machine Learning and Systems, Vol. 1, 2019, pp. 374–388.

[7] A. Barachant, S. Bonnet, M. Congedo, C. Jutten, Multiclass brain–computer interface classification by riemannian geometry, IEEE transactions on biomedical engineering 59 (4) (2012) 920–928.

[8] P. Li, J. Xie, Q. Wang, W. Zuo, Is second-order information helpful for large-scale visual recognition?, in: Proceedings of the IEEE international conference on computer vision, 2017, pp. 2070–2078.

[9] R. Bhatia, Positive definite matrices, in: Positive Definite Matrices, Princeton university press, 2009.

[10] Z. Huang, L. Van Gool, A riemannian network for spd matrix learning, in: Proceedings of the AAAI conference on artificial intelligence, Vol. 31, 2017.

[11] D. Brooks, O. Schwander, F. Barbaresco, J.-Y. Schneider, M. Cord, Riemannian batch normalization for spd neural networks, Advances in Neural Information Processing Systems 32 (2019).

[12] R. Kobler, J.-i. Hirayama, Q. Zhao, M. Kawanabe, Spd domain-specific batch normalization to crack interpretable unsupervised domain adaptation in eeg, Advances in Neural Information Processing Systems 35 (2022) 6219–6235.

[13] D. Jafuno, A. Mian, G. Ginolhac, N. Stelzenmuller, Classification of buried objects from ground penetrating radar images by using second order deep learning models, IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing (2025).

[14] A. Edelman, T. A. Arias, S. T. Smith, The geometry of algorithms with orthogonality constraints, SIAM journal on Matrix Analysis and Applications 20 (2) (1998) 303–353.

[15] P.-A. Absil, R. Mahony, R. Sepulchre, Optimization algorithms on matrix manifolds, Princeton University Press, 2008.

[16] N. Boumal, An introduction to optimization on smooth manifolds, Cambridge University Press, 2023.

[17] H. Karcher, Riemannian center of mass and so called karcher mean, arXiv preprint arXiv:1407.2087 (2014).

[18] J. Li, S. Ma, Federated learning on riemannian manifolds, arXiv preprint arXiv:2206.05668 (2022).

[19] Z. Huang, W. Huang, P. Jawanpuria, B. Mishra, Federated learning on riemannian manifolds with differential privacy, arXiv preprint arXiv:2404.10029 (2024).

[20] R. Zimmermann, K. Huper, Computing the riemannian logarithm on the stiefel manifold: Metrics, methods, and performance, SIAM Journal on Matrix Analysis and Applications 43 (2) (2022) 953–980.

[21] F. Bouchard, N. Laurent, S. Said, N. L. Bihan, Beyond r-barycenters: an effective averaging method on stiefel and grassmann manifolds, arXiv preprint arXiv:2501.11555 (2025).

[22] P.-A. Absil, J. Malick, Projection-like retractions on matrix manifolds, SIAM Journal on Optimization 22 (1) (2012) 135–158.

[23] F. H. Clarke, R. J. Stern, P. R. Wolenski, Proximal smoothness and the lower-c2 property, J. Convex Anal 2 (1-2) (1995) 117–144.

[24] R. Poliquin, R. Rockafellar, Prox-regular functions in variational analysis, Transactions of the American Mathematical Society 348 (5) (1996) 1805–1838.

[25] D. Davis, D. Drusvyatskiy, Z. Shi, Stochastic optimization over proximally smooth sets, SIAM Journal on Optimization 35 (1) (2025) 157–179.

[26] K. Deng, J. Hu, Decentralized projected riemannian gradient method for smooth optimization on compact submanifolds embedded in the euclidean space, Numerische Mathematik (2025) 1–38.

[27] H. Xiao, T. Yan, K. Wang, Riemannian svrg using barzilai–borwein method as second-order approximation for federated learning, Symmetry 16 (9) (2024) 1101.

[28] H. Xiao, T. Yan, S. Zhao, Riemannian svrg with barzilai-borwein scheme for federated learning, Journal of Industrial and Management Optimization 21 (2) (2025) 1546–1567.

[29] J. Zhang, J. Hu, A. M.-C. So, M. Johansson, Nonconvex federated learning on compact smooth submanifolds with heterogeneous data, Advances in Neural Information Processing Systems 37 (2024) 109817–109844.

[30] H. Wang, Z. Pan, C. He, J. Li, B. Jiang, Federated learning on riemannian manifolds: A gradient-free projection-based approach, arXiv preprint arXiv:2507.22855 (2025).

[31] Z. Huang, W. Huang, P. Jawanpuria, B. Mishra, Riemannian federated learning via averaging gradient stream, arXiv preprint arXiv:2409.07223 (2024).

[32] K. D. B. J. Adam, et al., A method for stochastic optimization, arXiv preprint arXiv:1412.6980 1412 (6) (2014).

[33] F. Alimisis, A. Orvieto, G. Becigneul, A. Lucchi, Momentum improves optimization on riemannian manifolds, in: International conference on artificial intelligence and statistics, PMLR, 2021, pp. 1351–1359.

[34] J. Wang, Z. Charles, Z. Xu, G. Joshi, H. B. McMahan, M. Al-Shedivat, G. Andrew, S. Avestimehr, K. Daly, D. Data, et al., A field guide to federated optimization, arXiv preprint arXiv:2107.06917 (2021).

[35] S. Chevallier, I. Carrara, B. Aristimunha, P. Guetschel, S. Sedlar, B. Lopes, S. Velut, S. Khazem, T. Moreau, The largest eeg-based bci reproducibility study for open science: the moabb benchmark, arXiv preprint arXiv:2404.15319 (2024).

[36] V. J. Lawhern, A. J. Solon, N. R. Waytowich, S. M. Gordon, C. P. Hung, B. J. Lance, Eegnet: a compact convolutional neural network for eeg-based brain–computer interfaces, Journal of neural engineering 15 (5) (2018) 056013.

[37] S. Bonnabel, Stochastic gradient descent on riemannian manifolds, IEEE Transactions on Automatic Control 58 (9) (2013) 2217–2229. doi:10.1109/tac.2013.2254619.
URL http://dx.doi.org/10.1109/TAC.2013.2254619

[38] H. Zhang, S. J Reddi, S. Sra, Riemannian svrg: Fast stochastic optimization on riemannian manifolds, Advances in Neural Information Processing Systems 29 (2016).

[39] H. Yang, M. Fang, J. Liu, Achieving linear speedup with partial worker participation in non-iid federated learning, arXiv preprint arXiv:2101.11203 (2021).

[40] F. Lotte, L. Bougrain, A. Cichocki, M. Clerc, M. Congedo, A. Rakotomamonjy, F. Yger, A review of classification algorithms for EEG-based brain–computer interfaces: a 10 year update, Journal of neural engineering 15 (3) (2018) 031005.

[41] I. Carrara, B. Aristimunha, M.-C. Corsi, R. Y. de Camargo, S. Chevallier, T. Papadopoulo, Geometric neural network based on phase space for bci-eeg decoding, Journal of Neural Engineering 22 (1) (2025) 016049.

[42] W. Yi, S. Qiu, K. Wang, H. Qi, L. Zhang, P. Zhou, F. He, D. Ming, Evaluation of eeg oscillatory patterns and cognitive process during simple and compound limb motor imagery, PloS one 9 (12) (2014) e114853.

[43] R. T. Schirrmeister, J. T. Springenberg, L. D. J. Fiederer, M. Glasstetter, K. Eggensperger, M. Tangermann, F. Hutter, W. Burgard, T. Ball, Deep learning with convolutional neural networks for eeg decoding and visualization, Human brain mapping 38 (11) (2017) 5391–5420.

[44] A. L. Goldberger, L. A. Amaral, L. Glass, J. M. Hausdorff, P. C. Ivanov, R. G. Mark, J. E. Mietus, G. B. Moody, C.-K. Peng, H. E. Stanley, Physiobank, physiotoolkit, and physionet: components of a new research resource for complex physiologic signals, circulation 101 (23) (2000) e215–e220.

[45] P. Ablin, S. Vary, B. Gao, P.-A. Absil, Infeasible deterministic, stochastic, and variance-reduction algorithms for optimization under orthogonality constraints, Journal of Machine Learning Research 25 (389) (2024) 1–38.

[46] C. Ionescu, O. Vantzos, C. Sminchisescu, Matrix backpropagation for deep networks with structured layers, in: Proceedings of the IEEE international conference on computer vision, 2015, pp. 2965–2973.

[47] L. Prechelt, Early stopping—but when?, in: Neural Networks: Tricks of the trade, Springer, 1998, pp. 55–69.

[48] X. Li, M. Jiang, X. Zhang, M. Kamp, Q. Dou, Fedbn: Federated learning on non-iid features via local batch normalization, arXiv preprint arXiv:2102.07623 (2021).