Federated Learning with Nonvacuous Generalisation Bounds

Anonymous authors
Paper under double-blind review

Abstract

We introduce a novel strategy to train randomised predictors in federated learning, where each node of the network aims at preserving its privacy by releasing a local predictor but keeping secret its training dataset with respect to the other nodes. We then build a global randomised predictor which inherits the properties of the local private predictors in the sense of a PAC-Bayesian generalisation bound. We consider the synchronous case where all nodes share the same training objective (derived from a generalisation bound), and the heterogenous and homogenous cases where each node may have its own personalised training objective. We show through a series of numerical experiments that our approach achieves a comparable predictive performance to that of the batch approach where all datasets are shared across nodes. Moreover the predictors are supported by numerically nonvacuous generalisation bounds while preserving privacy for each node. We explicitly compute the increment on predictive performance and generalisation bounds for our two federated settings, highlighting the price to pay to preserve privacy.

1 Introduction

In the federated learning (FL) paradigm, a group of *users* (or nodes) is learning in parallel, and typically aims at preserving their personal datasets while sharing a common predictor. While maintaining the privacy of their own data, users mutually share information through a central *server*. There has been a significant surge of interest in federated learning in the past decade (Konečný et al., 2016b), with clear applications in healthcare, transportation and retail, where it is typically of the utmost interest to avoid the leak of private information to other organisations or devices, for ethical or business motivations. The existing literature essentially categorises *horizontal* and *vertical* FL, depending on whether users' datasets share many features or individuals. These two streams have generated various contributions such as the design of efficient communication strategies (Konečný et al., 2016a;b; Suresh et al., 2017), the preservation of privacy through differentially-private distributed optimisation methods (Agarwal et al., 2018), the enforcement of fairness (as in many cases, post-training learning models may be biased or unfair and may discriminate against some protected groups – Hardt et al., 2016; Mohri et al., 2019). We refer to Zhang et al. (2021); Mammen (2021); Kairouz et al. (2021) for recent surveys on FL.

Consider a simple federated learning framework (Bonawitz et al., 2017; McMahan et al., 2017b). In each round, the server first provides the initial model to each user, then each user updates the initial model with its personal data. Finally, the server aggregates the collected local models into a single global model, which is used as next round's initialisation if needed. Hereafter we will refer to this learning problem as *FL-SOB* (Federated Learning with Synchronous OBjectives). This is especially relevant when all users share a common learning goal (*e.g.*, hospitals learning from different datasets to identify or predict a specific single pathology). Deep neural networks have been used to develop powerful federated algorithms (McMahan et al., 2017a). A more complex scenario consists in *personalised FL* (*PFL*, Tan et al., 2022) where users may have their own distinct learning goal but still want to share joint information as these goals share some level of similarity. This corresponds, for instance, to *transfer learning* (see *e.g.*, Zhuang et al., 2021) situations where one wants to extract some information of a learning problem (*e.g.*, detecting tigers in images) to perform better on another one, sharing some similarities (*e.g.*, detecting cats).

Towards a unified framework. The recent PAC-FL framework of Zhang et al. (2023b) proposes a unified framework to formalise FL, intricating the notion of generalisation ability (designed as utility) alongside privacy, and quantifying how much data are protected (*i.e.* impossible to retrieve) while transmitting partial information to the server. This framework builds from the work of Zhang et al. (2019) investigating the tradeoffs between privacy, utility and efficiency. The question of an optimal trade-off is crucial to deploy the FL framework in practice (Tsipras et al., 2019).

On the place of generalisation in FL. Using their PAC-FL framework, Zhang et al. (2023b) proposed generalisation bounds involving the dimension of the predictor space. The question of generalisation in FL is central: Mohri et al. (2019); Zhang et al. (2023a) established Rademacher-based generalisation bounds, Yagli et al. (2020) provided bounds based on mutual information to explain both generalisation ability and privacy leakage per user. Bayesian methods have also been considered in FL-SOB (Yurochkin et al., 2019; Chen & Chao, 2021; Zhang et al., 2022) as well as in PFL (Kotelevskii et al., 2022).

PAC-Bayes learning in FL. Beyond Bayesian methods, PAC-Bayes learning (see the seminal works of Shawe-Taylor & Williamson, 1997; McAllester, 1998; 2003; Maurer, 2004 – we refer to the surveys of Guedj, 2019; Alquier, 2021, and to the recent monograph of Hellström et al., 2023) has recently re-emerged as a powerful framework in batch learning to explain the generalisation ability of neural nets by providing non-vacuous generalisation bounds (Dziugaite & Roy, 2017; Letarte et al., 2019; Pérez-Ortiz et al., 2021; Biggs & Guedj, 2021; 2022). PAC-Bayes combines information-theoretic tools with the Bayesian paradigm of generating a data-dependent *posterior* distribution over a predictor space from a *prior* distribution (or reference measure), usually data-independent. The flexibility of the PAC-Bayes framework makes it useful to explain generalisation in many learning settings. In particular, theoretical results and practical algorithms have been derived for various learning problems such as reinforcement learning (Fard & Pineau, 2010), online learning (Li et al., 2018; Haddouche & Guedj, 2022), constrative learning (Nozawa et al., 2020), generative models (Chérief-Abdellatif et al., 2022), multi-armed bandits (Seldin et al., 2011; 2012; Sakhi et al., 2022), meta-learning (Amit & Meir, 2018; Farid & Majumdar, 2021; Rothfuss et al., 2021; 2022; Ding et al., 2021), majority votes (Zantedeschi et al., 2021; Biggs et al., 2022) to name but a few.

Recently, some works have used the PAC-Bayes framework in FL: Reisizadeh et al. (2020) and Achituve et al. (2021) have evaluated the post-training predictor shared by all users through a PAC-Bayes bound. Rather than exploiting existing bounds, new PAC-Bayes results, tailored for personalised FL, recently emerged with the aim to explain the efficiency of learning procedures (Scott et al., 2023; Sefidgaran et al., 2023), although the PAC-Bayes bound is not minimised by the algorithm. Finally, recent works showed that the Bayesian procedure ELBO, adapted to FL, is exactly the minimisation of a PAC-Bayes upper bound (Kim & Hospedales, 2023; Vedadi et al., 2023). Thus, they show that those methods are well incorporated in a theoretical framework explaining their good generalisation ability.

Our contributions. Beyond being a safety check for generalisation, PAC-Bayes theory provides state-of-the art learning algorithms with tight generalisation guarantees in the batch setting. We adapt those algorithms to the FL-SOB and PFL settings. We propose GenFL (standing for Generalisation-driven Federated Learning), an algorithm in which users optimise local PAC-Bayes objectives (bounds from Dziugaite & Roy, 2017; Pérez-Ortiz et al., 2021). We show a global generalisation bound for all users in FL-SOB, and local ones in PFL. Finally, we show in numerical experiments that our procedure is competitive with the state-of-the-art and we bring nonvacuous generalisation guarantees to practitioners of federated learning.

Outline. We describe our notation in Section 2 and introduce in Section 3 a novel algorithm called GenFL, alongside two instantiations to FL-SOB and PFL. We present numerical experiments to support our methods. in Section 4. Our algorithms and the code used to generate figures in this paper is available at `https://anonymous.4open.science/r/GenFL-0147/README.md`. The paper closes with a discussion in Section 5. In Appendix A, we comment on strategies to compute PAC-Bayesian bounds, Appendix B contains a comprehensive description of our procedure in the PFL setting, and Appendix C provides additional experiments.

2 Background

Federated learning. We consider a predictor set $\mathcal{H}$, a data space $\mathcal{Z}$ and denote the space of distributions over $\mathcal{H}$, $\mathcal{M}(\mathcal{H})$. We let $\ell\colon \mathcal{H}\times\mathcal{Z}\to[0,1]$ denote a loss function. In FL, we consider an ensemble of $K\in\mathbb{N}^*$ users, and for each user $1\leq i\leq K$, we denote by $\mathcal{S}_i=(\mathbf{z}_{i,j})_{j=1\cdots m_i}$ its associated dataset of size m_i. We define $\mathcal{S}$, of size $m=\sum_{i=1}^{K}m_i$, to be the union of all $\mathcal{S}_i$. We assume that each $\mathcal{S}_i$ is *i.i.d.* with associated distribution $\mathcal{D}_i$. Each user $1\leq i\leq K$ aims to jointly learn a predictor $h\in\mathcal{H}$ while keeping private their training dataset $\mathcal{S}_i$.

Learning theory. In PAC-Bayes learning, instead of directly crafting a predictor $h\in\mathcal{H}$, we design a data-driven posterior distribution $\mathrm{Q}\in\mathcal{M}(\mathcal{H})$ with respect to a prior distribution P. To assess the generalisation ability of a predictor $h\in\mathcal{H}$, we define for each user k the *risk* to be $\mathrm{R}_{\mathcal{D}_i}:=\mathbb{E}_{\mathbf{z}\sim\mu}[\ell(h,\mathbf{z})]$ and its empirical counterpart $\hat{\mathrm{R}}_{\mathcal{S}_i}:=\frac{1}{m}\sum_{j=1}^{m_i}\ell(h,\mathbf{z}_{i,j})$. As PAC-Bayes focuses on elements of $\mathcal{M}(\mathcal{H})$, we also define the expected risk and empirical risks for $Q\in\mathcal{M}(\mathcal{H})$ as $\mathrm{R}_{\mathcal{D}_i}(\mathrm{Q}):=\mathbb{E}_{h\sim\mathrm{Q}}[\mathrm{R}_{\mathcal{D}_i}(h)]$ and $\hat{\mathrm{R}}_{\mathcal{S}_i}(\mathrm{Q}):=\mathbb{E}_{h\sim\mathrm{Q}}[\hat{\mathrm{R}}_{\mathcal{S}_i}(h)]$.

Background on PAC-Bayes learning. In a batch setting, we only consider the dataset $\mathcal{S}$ (this can be seen as the case where there is only one user) and we assume that all data are *i.i.d.* with distribution $\mathcal{D}$. For two probability measures P, Q we define the *Kullback-Leibler divergence* to be $\mathrm{KL}(\mathrm{Q},\mathrm{P})=\mathbb{E}_{h\sim\mathrm{P}}\left[\frac{d\mathrm{Q}}{d\mathrm{P}}(h)\right]$ where $\frac{d\mathrm{Q}}{d\mathrm{P}}$ is the Radon-Nikodym derivative. We also denote by kl the KL divergence between two Bernoulli distributions.

Generalisation bounds. We recall the following bound, due to McAllester (2003); Maurer (2004), which holds for bounded losses.

Theorem 1 (McAllester's bound)**.** *For any data-free prior distribution* $\mathrm{P}\in\mathcal{M}(\mathcal{H})$*, any* $\delta\in[0,1]$*, with probability at least* $1-\delta$*, for any posterior distribution* $\mathrm{Q}\in\mathcal{M}(\mathcal{H})$*,*

$$\mathrm{kl}\left(R_{\mathcal{D}}(\mathrm{Q}),R_{\mathcal{S}}(\mathrm{Q})\right)\leq\frac{\mathrm{KL}(\mathrm{Q}\|\mathrm{P})+\ln\frac{2\sqrt{m}}{\delta}}{m},\tag{1}$$

which leads to the following upper bound on the risk

$$R_{\mathcal{D}}(\mathrm{Q})\leq\mathrm{kl}^{-1}\left(\hat{R}_{\mathcal{S}}(\mathrm{Q})\middle\|\frac{\mathrm{KL}(\mathrm{Q}\|\mathrm{P})+\ln\frac{2\sqrt{m}}{\delta}}{m}\right),\tag{2}$$

where $\mathrm{kl}^{-1}(x,b)=\sup\{y\in[x,1]\mid kl(x,y)\leq b\}$.

Note that by the definition of kl^{-1}, (2) is the tightest upper bound on $\mathrm{R}_{\mathcal{D}}(\mathrm{Q})$ that we can obtain starting from (1). While kl^{-1} has no closed form, it is possible to approximate it efficiently via root-finding techniques (see, *e.g.*, Dziugaite & Roy, 2017, Appendix A). However, this function is hard to evaluate, and even harder to optimise. We need to rely on looser relaxations of (1) to design tractable optimisation procedures.

Relaxations of McAllester's bound. The most classical relaxation of (1) relies on Pinsker's inequality $\mathrm{kl}(q\|p)\geq\frac{(p-q)^2}{2}$ and leads to the following high-probability bound, valid under the same assumptions as Theorem 1:

$$\mathrm{R}_{\mathcal{D}}(\mathrm{Q})\leq\hat{\mathrm{R}}_{\mathcal{S}}(\mathrm{Q})+\sqrt{\frac{\mathrm{KL}(\mathrm{Q}\|\mathrm{P})+\ln\frac{2\sqrt{m}}{\delta}}{2m}}.\tag{3}$$

While (3) is well known and already appears in McAllester (2003), novel relaxations, exploiting refined Pinsker's inequality (see, *e.g.*, Boucheron et al., 2013, Lemma 8.4) have been exploited to obtain PAC-Bayes Bernstein bounds (Tolstikhin & Seldin, 2013; Mhammedi et al., 2019). Building on this inequality, Rivasplata et al. (2019); Pérez-Ortiz et al. (2021) proposed a *PAC-Bayes quadratic bound* recalled below, valid under the assumptions of Theorem 1

$$\mathrm{R}_{\mathcal{D}}(\mathrm{Q}) \leq \left(\sqrt{\hat{\mathrm{R}}_{\mathcal{S}}(\mathrm{Q}) + \frac{\mathrm{KL}\left(\mathrm{Q}\|\mathrm{P}\right) + \log\left(\frac{2\sqrt{m}}{\delta}\right)}{2m}} + \sqrt{\frac{\mathrm{KL}\left(\mathrm{Q}\|\mathrm{P}\right) + \log\left(\frac{2\sqrt{m}}{\delta}\right)}{2m}} \right)^2 . \tag{4}$$

Note that (3) and (4) are easier to optimise than (2), making them more relevant for practical learning algorithms.

Generalisation-driven learning algorithms. Most of the PAC-Bayesian bounds in the literature are fully empirical. This paves the way to use the bound as a training objectives and leads to generalisation-driven learning algorithms. A classical PAC-Bayesian algorithm is derived from Catoni's bound (see, *e.g.*, Catoni, 2007, Alquier et al., 2016, Theorem 4.1):

$$\underset{\mathrm{Q}\in\mathcal{M}(\mathcal{H})}{\operatorname{argmin}}\ \mathrm{R}_{\mathcal{S}}(\mathrm{Q}) + \frac{\mathrm{KL}(\mathrm{Q},\mathrm{P})}{\lambda}. \tag{5}$$

In (5) an *inverse temperature* $\lambda > 0$ appears and acts as a learning rate in gradient descent. Similarly, it is possible to derive batch learning algorithms from (3) (4) (Dziugaite & Roy, 2017; Pérez-Ortiz et al., 2021). Then, we have access to a theoretical upper bound which requires to approximate expectations over Q. We next discuss how to mitigate this.

Computing generalisation guarantees. In practice, the tightest McAllester's bound is computed,*i.e.*, (1). First, note that the KL divergence is easy to compute in the Gaussian case as it has a closed form (see, *e.g.*, Duchi, 2007, Section 9). Then, it remains to estimate the expected empirical risk over Q, which is costly in practice as it involves Monte Carlo approximations. To alleviate this issue, we leverage the trick from Dziugaite & Roy (2017, Section 3.3), which exploit a high probability upper bound of $\hat{\mathrm{R}}_{\mathcal{S}}(\mathrm{Q})$ with confidence level δ'

$$\hat{\mathrm{R}}_{\mathcal{S}}(Q) \leq \mathrm{kl}^{-1}\left(\hat{\mathrm{R}}_{\mathcal{S}}(\hat{Q}_n) \,\middle\|\, \frac{1}{n}\ln\left(\frac{2}{\delta'}\right)\right),$$

where $\hat{\mathrm{R}}_{\mathcal{S}}(\hat{Q}_n) = \frac{1}{n}\sum_{i=1}^{n}\ell(W_i, z_i), \forall i, W_i \sim \mathrm{Q}$. Then incorporating this upper bound in (1) gives the final bound we use, with probability at least $1-\delta-\delta'$:

$$\mathrm{R}_{\mathcal{D}}(\mathrm{Q}) \leq \mathrm{kl}^{-1}\left(\hat{\mathrm{R}}^n_{\mathcal{S}}(\mathrm{Q}) \,\middle\|\, \frac{\mathrm{KL}(\mathrm{Q}\|\mathrm{P}) + \ln\frac{2\sqrt{m}}{\delta}}{m}\right), \tag{6}$$

where $\hat{\mathrm{R}}^n_{\mathcal{S}}(\mathrm{Q}) = \mathrm{kl}^{-1}\left(\hat{\mathrm{R}}_{\mathcal{S}}(\hat{Q}_n) \,\middle\|\, \frac{1}{n}\ln\left(\frac{2}{\delta'}\right)\right)$.

To compute $\mathrm{kl}^{-1}(p, c)$ for any p, c, we leverage Dziugaite & Roy (2017, Appendix A) described in appendix A.

3 Generalisation-driven Federated Learning

In this section we introduce our algorithm GenFL.

From batch to federated PAC-Bayes algorithms. When training stochastic neural nets (SNNs) with PAC-Bayes objectives, it is common to assume that each weight follows a Gaussian distribution. For conciseness, we identify a SNN to the Gaussian distribution of all its weights $\mathcal{N}(\mu, Diag(\sigma_i))$. The works of Dziugaite & Roy (2017); Rivasplata et al. (2019); Biggs & Guedj (2021); Pérez-Ortiz et al. (2021); Perez-Ortiz et al. (2021); Biggs & Guedj (2022) proposed successful PAC-Bayesian training algorithms for SNNs which ensure generalisation guarantees. All these methods operate in a batch setting, *i.e.*, the optimiser has access to all data simultaneously. Thus, building on the work of Rivasplata et al. (2019); Pérez-Ortiz et al.

(2021), we propose Alg. 1, a new learning algorithm called GenFL (Generalisation-driven Federated Learning), casting PAC-Bayes into FL. We stress that GenFL benefits from nonvacuous generalisation guarantees (Section 4).

Algorithm 1 GenFL. users are indexed by k; B is the local minibatch size, E is the number of local epochs, η is the learning rate, f the PAC-Bayes objective. The prior is $\mathcal{N}(\mu_{\text{prior}}\sigma_{\text{prior}})$ with parameter δ.

```
Server executes:
$m \leftarrow \sum_{k=1}^{K} m_k$                                          ▷ Total dataset size
$w_1 \leftarrow (\mu_{\text{prior}}, \sigma_{\text{prior}})$
for each round $t$ do
    $S_t \leftarrow$ random set of $\max(C \cdot K, 1)$ users
    for each user $k \in S_t$ in parallel do
        $w_{t+1}^k \leftarrow$ userUpdate($k, w_t, m$)
    end for
    $w_{t+1} \leftarrow \sum_{k=1}^{K} \frac{m_k}{m} w_{t+1}^k$
end for

userUpdate(k, w, m):
$\mathcal{B} \leftarrow$ (split $\mathcal{S}_k$ into batches of size B)
for each local epoch $e = 1, 2, \cdots, E$ do
    for each local minibatch $b \in B$ do
        $w_s^k \leftarrow \mu^k + \sigma^k \odot \mathcal{N}(0,1)$                ▷ Reparam. trick
        $w^k \leftarrow w^k - \eta \nabla_w f_{m,\delta,\mu_{\text{prior}},\sigma_{\text{prior}}}(w_s^k; b)$
    end for
end for
return $w^k$
Ensure: Global model distribution $w_T$ mapped to $\mathcal{N}(\mu_T, \sigma_T)$
```

A generalisation-driven FL algorithm. GenFL combines a federated learning protocol (*i.e.*, FedSGD, FedAvg, McMahan et al. (2017a)) with a PAC-Bayes objective f. Its starts from the vector $w_1 = (\mu_{prior}, \sigma_{prior})$ corresponding to the initial distribution $\mathrm{P} = \mathcal{N}(\mu_{prior}, \sigma_{prior})$ and outputs after T rounds $w_T = (\mu_T, \sigma_T)$ corresponding to the posterior $Q = \mathcal{N}(\mu_T, \sigma_T)$. Hence the learning procedure is divided in rounds, where subsets of users are sampled. Sampled users are requested to perform local updates following a PAC-Bayes training procedure designed to ensure a good generalisation of the posterior distribution. Because users train SNNs, they sample weights from the posterior. In order to learn the variance parameter of the posterior (Gaussian), we use the well-known reparameterisation trick (Kingma & Welling, 2014): instead of directly sampling from the distribution, we sample from a standard Gaussian distribution and then apply a transformation to obtain the sampled weights: $W = \mu + \sigma V$ where $V \sim \mathcal{N}(0, \mathrm{Id})$, this allows to compute with respect to σ. When the round ends, the global model is computed from the local updates, thanks to the aggregation function from the FL protocol (weighted mean, median).

Next we show that with a PAC-Bayes objective f, we adapt GenFL to FL-SOB, where $\mathcal{S}$ is fully *i.i.d.*, *i.e.*, all user datasets have the same distribution, and to PFL where each dataset $\mathcal{S}_i$ is *i.i.d.* but any two dataset can have distinct distributions.

3.1 GenFL for FL-SOB

PAC-Bayesian objectives. We assume that for any i, $\mathcal{D}_i = \mathcal{D}$. Thus, $\mathcal{S}$ is a *i.i.d.* dataset of m points. We then consider the true and empirical risks on all $\mathcal{S}$ $\mathrm{R}_{\mathcal{D}}, \mathrm{R}_{\mathcal{S}}$. In a batch setting, it would be natural to optimise the bounds (3), (4). However, the user i only has access to its personal dataset $\mathcal{S}_i$ (of size m_i) to optimise its model *while knowing other datasets are involved.* We then derive accordingly from (3), (4) two PAC-Bayesian learning algorithms, valid for any user, namely f_1, f_2. Note that f_1 is adapted from the

$f_{classic}$ objective and f_2 is adapted from f_{quad} of Pérez-Ortiz et al. (2021):

$$f_1(\mathcal{S}_i) = \hat{\mathrm{R}}_{\mathcal{S}_i}(Q) + \sqrt{\frac{\mathrm{KL}(Q\|P) + \ln\frac{2\sqrt{m}}{\delta}}{2m}}. \tag{7}$$

$$f_2(\mathcal{S}_i) = \left(\sqrt{\hat{\mathrm{R}}_{\mathcal{S}_i}(Q) + \frac{\mathrm{KL}\left(Q\|P\right) + \log\left(\frac{2\sqrt{m}}{\delta}\right)}{2m}} + \sqrt{\frac{\mathrm{KL}\left(Q\|P\right) + \log\left(\frac{2\sqrt{m}}{\delta}\right)}{2m}}\right)^2. \tag{8}$$

Note that (7), (8) can be seen as proxys of (3), (4). Indeed, every user has access to the total number of data points m (as long as it is transmitted to the server), so the regularisation term (containing the KL divergence) is fully available, contrary to the empirical risk $\hat{\mathrm{R}}_{\mathcal{S}}$ which is then replaced by $\hat{\mathrm{R}}_{\mathcal{S}_i}$. Note that in this case, the KL divergence is divided by m instead of m_i (which would be natural if we were optimising (3) (4) for $\mathcal{S}_i$ instead of $\mathcal{S}$). This suggests that each user has to give more weight, during the optimisation phase, to its data than to the regularisation. The reason behind this is that the server, by aggregating predictors, performs a regularisation step on a global level, hence the need to prioritise data on a local one.

A global generalisation guarantee. A major interest of the *i.i.d.* assumption on $\mathcal{S}$ is that, as long as users all exploit the same posterior distribution Q, and they transmit their empirical scores $\hat{\mathrm{R}}_{\mathcal{S}_i}(Q)$, every user is able to compute the global generalisation guarantee of (2). This allows to maintain the bound of the batch setting (involving the total number of data points m), despite being in FL. This is empirically shown in Section 4. We present in Algorithm 2 FedBound, the algorithm we use to compute the global bound (6), valid for all users simultaneously.

Algorithm 2 FedBound. The K users are indexed by k; f PAC-Bayes objective, prior $\mathcal{N}(\mu_{\text{prior}},\ \sigma_{\text{prior}})$; posterior $\mathcal{N}(\mu_T,\ \sigma_T)$, δ, $\delta^{'}$ parameters, n number of Monte Carlo sampling

```
Server executes:
$m \leftarrow \sum_{k=1}^{K} m_k$                                   ▷ Total dataset size
$\mathrm{P} = \mathcal{N}(\mu_{\text{prior}}, \sigma_{\text{prior}})$            ▷ Prior (learned or random)
$\mathrm{Q} = \mathcal{N}(\mu_T, \sigma_T)$                             ▷ Posterior (learned)
for each user $k \in K$ in parallel do
    $error^k \leftarrow$ userMCSampling$(k, w_t, m)$
end for
$error \leftarrow \sum_{k=1}^{K} \frac{m_k}{m} error_k$
$KL\_inv \leftarrow \mathrm{kl}^{-1}\left(error \mid \frac{1}{n}\ln(\frac{2}{\delta^{'}})\right)$
Up-bound $\leftarrow \mathrm{kl}^{-1}\left(KL\_inv \mid \frac{\mathrm{KL}(Q\|P)+\ln\frac{2\sqrt{m}}{\delta}}{m}\right)$

userMCSampling(k, w, m):
for each MC sampling $i = 1, 2, \cdots, n$ do
    $W_i^k \sim \mathrm{Q}$                                         ▷ Sample weights from the posterior
    $error_i^k \leftarrow \hat{\mathrm{R}}_{\mathcal{S}_k}(W_i^k)$                         ▷ local empirical risk
end for
$error^k \leftarrow= \frac{1}{n}\sum_{i=1}^{n} error_i^k$
return $error^k$
Ensure: Up-bound holding with probability $1 - \delta - \delta^{'}$
```

3.2 GenFL for personalised federated learning

A general training for the prior distribution. In PFL, the learning objective of each user may differ, while sharing some similarities that can be learned and transferred from one user to another. This framework requires adjustments of our learning objectives. Indeed, contrary to Section 3.1, there is no clear global generalisation guarantee, so each user has then to optimise its own personal learning objective from a commonly shared prior. Using either a random prior or a learnt one on a fraction of users data, we run GenFL similarly to Section 3.1 with our PAC-Bayesian objective of interest f. The output distribution of GenFL is then considered as a common prior for all users which then needs to be personalised.

A personalisation step. Once a common prior distribution has been obtained from the federated training, it is necessary for each user to personalise it to its own problem. To do so, we apply PAC-Bayesian objectives similar to those in Section 3.1, namely f_1 (7) and f_2 (8), where the batch size is modified from m to m_i for the i-th user. This reflect that each user now optimises its local goal instead of the global one. Each user ends up with its own personalised posterior distribution. The way personalised bounds are implemented is similar to Algorithm 2, but without the aggregation step. We refer to Appendix B for additional details.

4 Experiments

In this section we provide practical instantiations of GenFL. We first consider in Section 4.1 the case of classification on MNIST in a federated setting using basic neural architectures. We then extend our experimental framework in Section 4.2 to the more challenging case of classification on CIFAR-10 with more sophisticated neural networks.

4.1 Classification on MNIST

4.1.1 Experimental framework

Our experimental framework is inspired from Pérez-Ortiz et al. (2021) combined with the FedAvg protocol McMahan et al. (2017a). We use the following libraries: Pytorch (Paszke et al., 2019) for deep learning, Flower (Beutel et al., 2020) for federated learning, Slurm (Yoo et al., 2003) for cluster experiments, and Hydra (Yadan, 2019) for overall experiment management. The cluster nodes we use have 48 SKYLAKE 3GHz CPUS. We do not use GPUs.

Prior distribution over weights. We propose two types of priors: data-free (random) prior chosen randomly around $\mathcal{N}(0, \mathrm{Id})$ (as in Dziugaite & Roy, 2017) and a data-dependent (learnt) prior. The latter is powerful to attenuate the KL divergence term, leading to sharper generalisation bounds and better accuracy. Such a data-driven prior implies to use a fraction of the dataset from the training data to optimise the prior. The bound computation is then realised with a reduced dataset size (divided by 2 in practice). However, the prior has gained efficiency (lower empirical risk) and the PAC-Bayes optimisation starts from a relevant point.

We use Gaussian distribution for both prior and posterior over the weights of a neural network. When data-free, the prior is $\mathrm{P} = \bigotimes_{l\in\text{layers}} \mathcal{N}(\text{truncated}(\mu^l_{\text{rand}}), \text{Diag}(\sigma_{\text{prior}}))$ with $\mu^l_{\text{rand}} \sim \mathcal{N}(0, \frac{1}{\sqrt{n^l_{in}}})$, and $\sigma_{\text{prior}} \in \mathbb{R}^{+*}$. The truncature is done at $\pm\frac{2}{\sqrt{n^l_{in}}}$ where n^l_{in} is the dimension of the inputs of the layer l. In the case of data-dependent prior, we have $\mathrm{P} = \bigotimes_{l\in\text{layers}} \mathcal{N}(\mu^l_{\text{learnt}}, \text{Diag}(\sigma_{\text{prior}}))$, where μ_{learnt}. It is obtained via ERM on the prior set on half of the training set, the other half being used for bound computation and posterior optimisation.

Dataset partition. To build a *i.i.d.* FL setup, we consider the case where each user has exactly the same number of samples per class. We partition MNIST as follows: we fix the number of users to 100. Then, each user receives a dataset size of 540, each class having 54 images. In the case of the learnt prior, we split the training set of each user in half, the first one being used to train the prior, the second exploited by our learning algorithms.

In the case of non-*i.i.d.* FL setup, we follow McMahan et al. (2017a). First we sort MNIST by label, then we partition the dataset into chunks of 300 contiguous samples each (thus containing at most 2 labels, because it is sorted). Again we split each user dataset in several parts. When the prior is random, we save 10% of the dataset to create a validation set. Remaining data is exploited for optimisation. When considering learnt priors, we save again 10% of the dataset as a validation set, we exploit 40% of the dataset to train the prior (respecting the proportion of each class), the remaining 50% being used by our learning algorithms.

Bound parameters. We used $\delta = 0.05$, $\delta^{'} = 0.01$, $n = 150000$ Monte Carlo samples.

Optimisation hyperparameters. The prior distribution scale σ_{prior} is set to $2,5 \times 10^{-2}$, the learning rate is 5×10^{-3} for 100 users. In order to compare with the batch learning setting, we compute our algorithms with 1 user. In this case, we use a learning rate of 5×10^{-4} to reach better performances. The momentum is 0.95 for posterior optimisation and 0.99 in prior optimisation. During prior optimisation, we used a dropout rate of 0.2 to avoid overfitting. As theoretical results of Section 2 require a loss function in $[0, 1]$, we use the bounded cross-entropy as in Pérez-Ortiz et al. (2021), *i.e.*, $\ell(x, y) = \frac{1}{\ln(p_{\min})} \cdot \ln(\tilde{\sigma}(x)_y)$ with $\tilde{\sigma}(x)_y = \max(\text{sigmoid}(x)_y, p_{\min})$. We took $p_{\min} = 10^{-4}$.

KL penalty. For stability reasons, we penalise the KL term during posterior optimisation (similarly to Pérez-Ortiz et al., 2021), thus we give more impact to the empirical risk during optimisation. Such a penalty helps performance and stability during training when random priors are involved. In this case, we use a penalty of 0.1.

Federated Learning hyperparameters. Starting from a random prior, we perform our algorithm during 200 rounds to make the SNNs converge. When learnt priors are considered, they are trained with a run of 100 rounds with 5 local epochs (convergence around 50 epochs). We then perform 10 additional rounds with 5 local epochs to train the posterior. In both cases, we select 10% of the users each round to participate in the training. As the dataset size of each user is small, we use a batch size of 25 (compared to 250 in the work of Pérez-Ortiz et al. (2021)).

Neural Architecture. We consider a stochastic 2 hidden layer MLP with 600 units each, resulting in 1,198,210 number of parameters for the prior (with fixed covariance matrix) and doubled for the SNN (as we consider diagonal covariance matrices).

Positive variance prior. To have a constrained positive standard deviation σ when sampling weights, we use the following transformation: $\sigma = \ln(1 + \exp(\rho))$. It makes σ always positive, and ρ can be any real number that is optimised during training procedure.

4.1.2 Results

Note that in a classification problem, the generalisation error translates a positive influence of the learning phase as long as it is smaller than 1 (which is what we refer to with the term nonvacuous). Indeed, a bound below this threshold shows that the posterior will not fail at each try. However, we focus on posteriors with generalisation bounds or test error smaller than 50% The reason is that, for a classification task, this threshold is the generalisation error of a randomised predictor with associated distribution Bernoulli(0.5). Thus, having results below this threshold provably show we generalise better than a naive strategy.

FL-SOB setting.

Table 1 gathers our results for GenFL applied with f_1, f_2 alongside FedBound. We compared our results with 100 clients with the output of our algorithms for 1 client, corresponding to the batch learning case. Our FL algorithms benefit from nonvacuous theoretical guarantees and test errors. In the case of data-dependent priors, test errors of GenFL nearly reach for both f_1 and f_2, the precision of their batch counterpart (3% in FL and 2% in batch). In the case of random prior, the KL penalty has a strong positive influence on the test errors. The generalisation bounds of our algorithms are uniformly deteriorated compared to the batch setting, *while being of the same magnitude*. This is important to notice as this is the price to pay to adapt batch bounds to a federated setting. Indeed, as each user only optimised a proxy of the common generalisation bound, it is legitimate to retrieve in our results a short discrepancy comparing to the batch case.

While f_2 is consistently achieving better generalisation upper bounds in Pérez-Ortiz et al. (2021), it is outperformed by f_1 in the FL setting. However, notice that f_2 provides uniformly better test errors than f_1, similarly to Pérez-Ortiz et al. (2021). Note that better results are achieved if one considers the KL penalty trick with data-free prior and no KL penalty trick with data-dependent prior. We interpret this fact as follows: given that the data-dependent prior is already performing well on training data, allowing the posterior optimisation to be unconstrained is not an issue as we found an area close from a local minimiser.

Table 1: Results for the FL-SOB scenario. ℓ^{0-1} corresponds to the 0-1 loss. The test error column is made on the test set of MNIST. The Bound column corresponds to the generalisation bounds, computed with Alg. 2. The KL/m column corresponds to the KL divergence term in the bound divided by $m = 60000$ in data-free prior or $m = 30000$ data-dependent prior.

Setup		Bound	Test Err.	KL div
Prior	Obj.	ℓ^{0-1}	ℓ^{0-1}	KL/m
Pérez-Ortiz et al. (2021): Random (1 client)	f_1	0.330	0,141	0,081
	f_2	0.316	0,092	0,138
Pérez-Ortiz et al. (2021): Learnt (1 client)	f_1	0.028	0,023	<0,001
	f_2	0.028	0,020	0,001
100 users - GenFL - KL Penalty=0.1				
Random (us)	f_1	0.333	0,123	0,107
	f_2	0.342	0,090	0,163
Learnt (us)	f_1	0,061	0,030	<0,001
	f_2	0,088	0,029	0,002
100 users - GenFL - No KL Penalty				
Random (us)	f_1	0,415	0,256	0,039
	f_2	0,408	0,251	0,041
Learnt (us)	f_1	0.039	0,030	<0,001
	f_2	0.040	0,030	<0,001

Table 2: Results for the PFL scenario. ℓ^{0-1} corresponds to the 0-1 loss. The test error column is made on the test set of each user (10% of local dataset).The Gen. Bound column gathers generalisation bounds. Each user bound is computed locally with $m_i = 300$ for learnt prior, while $m_i = 540$ for random prior.

Setup		Gen. Bound ℓ^{0-1}		
Prior	Obj.	min	mean	max
Random (us)	f_1	0,063	0,680	0,847
	f_2	0,075	0,713	0,893
Learnt (us)	f_1	0,054	0,112	0,222
	f_2	0,052	0,111	0,220
		Test Error ℓ^{0-1}		
Random (us)	f_1	0	0,552	0,767
	f_2	0	0,588	0,833
Learnt (us)	f_1	0	0,050	0,183
	f_2	0	0,044	0,150

However, as the random prior is not necessarily efficient, we need to move far from it to reach good empirical performances. However, moving freely from the prior distribution leads to a large KL divergence, hence the need to constrain the posterior optimisation to obtain both better bounds and test errors. A take-home message is that adapting PAC-Bayes algorithms to FL is effective: it gives nonvacuous results close to the batch setting.

PFL setting.

Table 2 provides an overview of our results in the non-*i.i.d.* case. It gathers, for both generalisation bounds and test error, the minimum, mean and maximum performance of all 100 users. The averaged performances are deteriorated compared to Table 1 for all settings as we consider a harder problem. It is worth noticing that our bounds are nonvacuous and that our algorithms with learnt priors benefit from bounds and test errors lower than 50%. Indeed, if we do not learn the prior, we see from the distribution of errors in Figure 1 that most users have a deteriorated bound and test errors. For learnt priors, the error distribution shows that all users enjoy a meaningful bound as well as sound performance. An interesting point is that the

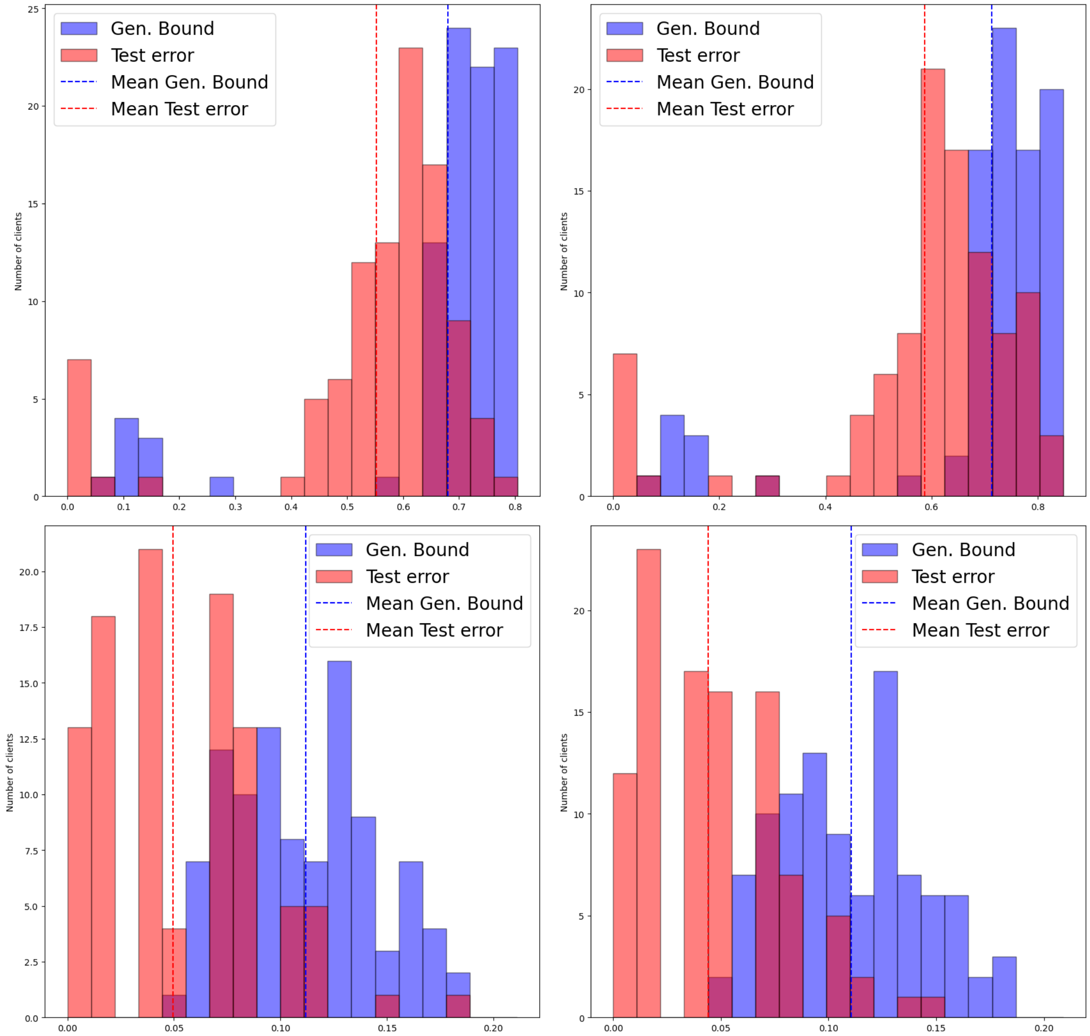

Figure 1: Histograms gathering test errors (red) and bound (blue) of all 100 users of the PFL setting. In order from top to botton: Random prior$-f_1$, Random prior$-f_2$, Learnt prior$-f_1$, Learnt prior$-f_2$.

common prior distribution does not support all users uniformly as we can see in Figure 1. Indeed, our algorithms with random priors suffer from deteriorated bounds and test errors on average and the worst case, but approximately 15% enjoy good test errors and 9% benefit from theoretical guarantees lower than 40%. Also, our algorithms with learnt priors enjoy test errors and generalisation guarantees lower than 20% for all users. Furthermore, approximately half of the users benefit from test errors lower than 5%. This highlights the importance of the prior in this non-*i.i.d.* setting. As the test set of each user is small (60 images), some users achieve a 0% test error.

4.2 Classification on CIFAR-10

To evaluate the effectiveness of our algorithm in more challenging scenarios, we conduct experiments on the CIFAR-10 dataset. We followed a setup akin to Section 4.1.1, with minor adjustments. Specifically, we exploit a learned prior trained with 70% of the dataset, the remaining being used for posterior estimation. Given the heightened complexity of the CIFAR-10 dataset, we involve Convolutional Neural Networks (CNNs) with 4 and 9 layers, denoted as CNNet4l and CNNet9l, respectively (these architectures also appear in Pérez-Ortiz et al., 2021, exploited as a baseline). To ensure enhanced performances, we also perform fine-tuning of hyperparameters related to FL such as local batch size and the number of local epochs, while keeping those associated with PAC-Bayes fixed to ensure relevant comparisons with the baseline.

We conducted a comprehensive grid search across a spectrum of hyperparameters to approximate the optimal model configurations. Specifically, we explored various combinations of local batch sizes $(1, 5, 10, 50)$ and local epoch counts $(1, 5, 10)$. Recognising that 100 rounds of training were inadequate for convergence, we extended the training duration to 300 rounds. Additionally, we adjusted the learning rate by a factor of 10 at round 200 to ease optimisation. From the resulting best configurations, we selected the most promising priors for further investigation. Subsequently, we fine-tuned the learning rate by exploring values between 5×10^{-4} and 1×10^{-3}. Remarkably, we discovered that the optimal learning rate, as reported in Pérez-Ortiz et al. (2021), remains consistent between the centralised and federated settings. Furthermore, we conducted experiments to tune the dropout rate, revealing that a dropout rate of 0.2 yielded the best performance on the CIFAR-10 dataset, consistent with findings in Pérez-Ortiz et al. (2021). This reaffirms the applicability of centralised PAC-Bayes hyperparameters to the decentralised PAC-Bayes paradigm. Detailed results of these experiments, showing accuracies for each experiments on the batch size, epoch counts, learning rate and dropout rate are provided in Appendix D.

Our analysis revealed that the most effective hyperparameter settings were a local batch size of 5 for CNNet9l and 10 for CNNet4l, with a corresponding local epoch count of 1 for both architectures. Notably, increasing the number of epochs led to quicker convergence but marginally reduced accuracy. A learning rate of 5×10^{-3} was optimal for both architectures, while a dropout rate of 0.2 was found to be the most effective.

We selected the priors with the highest accuracy from our grid search results (refer to the appendix for accuracy details). Subsequently, the posteriors were trained using identical hyperparameters as their corresponding priors. Additionally, we experimented with the KL penalty technique. Our findings are summarized in Table 3 for comparison with the baseline results presented in the first row. Notably, both the prior and posterior performances are slightly inferior to the baseline. This discrepancy can be attributed to the inherent difficulty of federated learning compared to the batch setting. However, it is crucial to note that the generalisation bounds remain non-vacuous.

Among the configurations tested, the most promising outcome was observed with CNNet9l using a KL penalty of 1.0 in conjunction with f_2. This configuration achieved a generalisation bound and a test error of 34.5% and 30.5%, respectively, which is 10% higher than the baseline for both metrics. This a similar conclusion than in Section 4.1, highlighting the price to pay to switch from batch to FL. Surprisingly, the KL penalty trick did not yield an improvement in the generalisation bound, contrary to Section 4.1. This is possibly linked to the intrinsic complexity of CIFAR-10. In particular, the inadequacy of the prior may necessitate further optimisation during posterior training, potentially causing the posterior to diverge significantly from the prior distribution. Consequently, the KL term increases post-training, warranting a more substantial penalty.

5 Discussion

In this work, we propose a novel algorithm for FL in two different settings: FL-SOB, which allows to exploit a global generalisation guarantee while keeping data separated; as well as PFL, which only involves an *i.i.d.* assumption for each user's dataset. Our work raises two questions: (a) is it possible to remove the *i.i.d.* assumption?(b) is it possible to maintain a global generalisation guarantee, even in the personalised setting? To answer (a), a line of work first initiated by Kuzborskij & Szepesvári (2019) (for *i.i.d.* data) and continued by Haddouche & Guedj (2023); Chugg et al. (2023); Jang et al. (2023) (for non-*i.i.d.* ones) focuses on PAC-Bayes bounds valid for data distribution with bounded variances. In the PFL setting, this could provide novel generalisation bounds without assuming each user possesses an *i.i.d.* dataset. About (b), the recent work of Sefidgaran et al. (2023) provides elements of answer: they derive a general PAC-Bayesian bound holding for the classical FL setting *for all users simultaneously* involving explicitly the number of users and rounds. This allows fruitful theoretical interpretations (especially on the number of rounds involved during the FL training), but leads to vacuous generalisation guarantees for classification task with a federated SVM. Following another route, based on PAC-Bayes methods for meta-learning, Boroujeni et al. (2024) provides a novel FL algorithm for PFL derived from an original theoretical result with strong performances. However,

Model	Obj.	β	ϵ	rounds	Bound	Test Error	KL Div	Prior Test Error
Pérez-Ortiz et al. (2021)								
CNNet9l	f_1	250	100	"1"	0.237	0.216	<0.0001	0.217
(baseline)	f_2	250	100	"1"	0.250	0.214	0.003	0.217
100 users - GenFL - KL Penalty=0.1								
CNNet4l	f_1	10	1	300	0.471	0.388	0.012	0.329
(us)	f_2	10	1	300	0.469	0.391	0.011	0.329
CNNet9l	f_1	5	1	300	0.393	0.303	0.019	0.274
(us)	f_2	5	1	300	0.396	0.304	0.018	0.274
100 users - GenFL - KL Penalty=1.0								
CNNet4l	f_1	10	1	300	0.466	0.390	0.011	0.329
(us)	f_2	10	1	300	0.458	0.391	0.009	0.329
CNNet9l	f_1	5	1	300	0.386	0.302	0.014	0.274
(us)	f_2	5	1	300	0.345	0.305	0.003	0.274

Table 3: Table displaying the results for the CIFAR-10 dataset alongside the baseline (batch setting) presented in the first row. 'Prior Test Error' represents the 0-1 error of the prior on the test set. The symbol β denotes the batch size of clients, while ϵ indicates the number of local epochs on each round.

their approach involves distributions on distributions spaces, giving their method a potentially high time complexity, they are also unable to compute nonvacuous generalisation guarantees. This constrasts with our results, even for the personalised setting, at the cost of considering generic PAC-Bayesian bounds, not explicitly tailored for federated learning. Establishing a PAC-Bayes bound designed for FL and leading to a non-vacuous generalisation guarantees remains an open challenge that we aim to address in a future work.

References

Idan Achituve, Aviv Shamsian, Aviv Navon, Gal Chechik, and Ethan Fetaya. Personalized Federated Learning With Gaussian Processes. In Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pp. 8392–8406, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/46d0671dd4117ea366031f87f3aa0093-Abstract.html.

Naman Agarwal, Ananda Theertha Suresh, Felix X. Yu, Sanjiv Kumar, and Brendan McMahan. cpSGD: Communication-efficient and differentially-private distributed SGD. In Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pp. 7575–7586, 2018. URL https://proceedings.neurips.cc/paper/2018/hash/21ce689121e39821d07d04faab328370-Abstract.html.

Pierre Alquier. User-friendly introduction to PAC-Bayes bounds. *arXiv*, abs/2110.11216, 2021. URL https://arxiv.org/abs/2110.11216.

Pierre Alquier, James Ridgway, and Nicolas Chopin. On the properties of variational approximations of Gibbs posteriors. *Journal of Machine Learning Research*, 2016.

Ron Amit and Ron Meir. Meta-Learning by Adjusting Priors Based on Extended PAC-Bayes Theory. In *International Conference on Machine Learning (ICML)*, 2018.

Daniel J Beutel, Taner Topal, Akhil Mathur, Xinchi Qiu, Javier Fernandez-Marques, Yan Gao, Lorenzo Sani, Hei Li Kwing, Titouan Parcollet, Pedro PB de Gusmão, and Nicholas D Lane. Flower: A Friendly Federated Learning Research Framework. *arXiv preprint arXiv:2007.14390*, 2020.

Felix Biggs and Benjamin Guedj. Differentiable PAC-Bayes objectives with partially aggregated neural networks. *Entropy*, 23(10), 2021. ISSN 1099-4300. URL https://www.mdpi.com/1099-4300/23/10/1280.

Felix Biggs and Benjamin Guedj. Non-vacuous generalisation bounds for shallow neural networks. In *Proceedings of the 39th International Conference on Machine Learning [ICML]*, volume 162 of *Proceedings of Machine Learning Research*, pp. 1963–1981. PMLR, July 2022. URL https://proceedings.mlr.press/v162/biggs22a.html.

Felix Biggs, Valentina Zantedeschi, and Benjamin Guedj. On margins and generalisation for voting classifiers. In *NeurIPS*, 2022. URL https://arxiv.org/abs/2206.04607.

Keith Bonawitz, Vladimir Ivanov, Ben Kreuter, Antonio Marcedone, H Brendan McMahan, Sarvar Patel, Daniel Ramage, Aaron Segal, and Karn Seth. Practical secure aggregation for privacy-preserving machine learning. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pp. 1175–1191, 2017.

Mahrokh Ghoddousi Boroujeni, Andreas Krause, and Giancarlo Ferrari Trecate. Personalized federated learning of probabilistic models: A pac-bayesian approach. *arXiv preprint arXiv:2401.08351*, 2024.

Stéphane Boucheron, Gábor Lugosi, and Pascal Massart. *Concentration Inequalities - A Nonasymptotic Theory of Independence*. Oxford University Press, 2013. ISBN 978-0-19-953525-5. doi: 10.1093/acprof:oso/9780199535255.001.0001. URL https://doi.org/10.1093/acprof:oso/9780199535255.001.0001.

Olivier Catoni. *PAC-Bayesian supervised classification: the thermodynamics of statistical learning*. Institute of Mathematical Statistics, 2007.

Hong-You Chen and Wei-Lun Chao. FedBE: Making Bayesian Model Ensemble Applicable to Federated Learning. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. URL https://openreview.net/forum?id=dgtpE6gKjHn.

Badr-Eddine Chérief-Abdellatif, Yuyang Shi, Arnaud Doucet, and Benjamin Guedj. On PAC-Bayesian reconstruction guarantees for VAEs. In *Proceedings of The 25th International Conference on Artificial Intelligence and Statistics [AISTATS]*, volume 151 of *Proceedings of Machine Learning Research*, pp. 3066–3079. PMLR, 28–30 Mar 2022. URL https://proceedings.mlr.press/v151/cherief-abdellatif22a.html.

Ben Chugg, Hongjian Wang, and Aaditya Ramdas. A unified recipe for deriving (time-uniform) PAC-Bayes bounds. *arXiv*, abs/2302.03421, 2023.

Nan Ding, Xi Chen, Tomer Levinboim, Sebastian Goodman, and Radu Soricut. Bridging the Gap Between Practice and PAC-Bayes Theory in Few-Shot Meta-Learning. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2021.

John Duchi. Derivations for linear algebra and optimization. *Berkeley, California*, 3(1):2325–5870, 2007.

Gintare Karolina Dziugaite and Daniel Roy. Computing Nonvacuous Generalization Bounds for Deep (Stochastic) Neural Networks with Many More Parameters than Training Data. In *Conference on Uncertainty in Artificial Intelligence (UAI)*, 2017.

Mahdi Milani Fard and Joelle Pineau. PAC-Bayesian model selection for reinforcement learning. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2010.

Alec Farid and Anirudha Majumdar. Generalization Bounds for Meta-Learning via PAC-Bayes and Uniform Stability. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2021.

Benjamin Guedj. A Primer on PAC-Bayesian Learning. In *Proceedings of the second congress of the French Mathematical Society*, volume 33, 2019. URL https://arxiv.org/abs/1901.05353.

Maxime Haddouche and Benjamin Guedj. Online PAC-Bayes Learning. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2022.

Maxime Haddouche and Benjamin Guedj. PAC-Bayes Generalisation Bounds for Heavy-Tailed Losses through Supermartingales. *Transactions on Machine Learning Research*, 2023.

Moritz Hardt, Eric Price, and Nati Srebro. Equality of Opportunity in Supervised Learning. In Daniel D. Lee, Masashi Sugiyama, Ulrike von Luxburg, Isabelle Guyon, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pp. 3315–3323, 2016. URL https://proceedings.neurips.cc/paper/2016/hash/9d2682367c3935defcb1f9e247a97c0d-Abstract.html.

Fredrik Hellström, Giuseppe Durisi, Benjamin Guedj, and Maxim Raginsky. Generalization Bounds: Perspectives from Information Theory and PAC-Bayes, 9 2023. *arXiv*.

Kyoungseok Jang, Kwang-Sung Jun, Ilja Kuzborskij, and Francesco Orabona. Tighter PAC-Bayes Bounds Through Coin-Betting. *arXiv*, abs/2302.05829, 2023.

Peter Kairouz, H. Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kallista Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, Rafael G. L. D'Oliveira, Hubert Eichner, Salim El Rouayheb, David Evans, Josh Gardner, Zachary Garrett, Adrià Gascón, Badih Ghazi, Phillip B. Gibbons, Marco Gruteser, Zaid Harchaoui, Chaoyang He, Lie He, Zhouyuan Huo, Ben Hutchinson, Justin Hsu, Martin Jaggi, Tara Javidi, Gauri Joshi, Mikhail Khodak, Jakub Konecný, Aleksandra Korolova, Farinaz Koushanfar, Sanmi Koyejo, Tancrède Lepoint, Yang Liu, Prateek Mittal, Mehryar Mohri, Richard Nock, Ayfer Özgür, Rasmus Pagh, Hang Qi, Daniel Ramage, Ramesh Raskar, Mariana Raykova, Dawn Song, Weikang Song, Sebastian U. Stich, Ziteng Sun, Ananda Theertha Suresh, Florian Tramèr, Praneeth Vepakomma, Jianyu Wang, Li Xiong, Zheng Xu, Qiang Yang, Felix X. Yu, Han Yu, and Sen Zhao. Advances and Open Problems in Federated Learning. *Foundations and Trends® in Machine Learning*, 14(1–2):1–210, 2021. ISSN 1935-8237. doi: 10.1561/2200000083. URL http://dx.doi.org/10.1561/2200000083.

Minyoung Kim and Timothy M. Hospedales. FedHB: Hierarchical Bayesian Federated Learning. *arXiv*, abs/2305.04979, 2023. doi: 10.48550/arXiv.2305.04979. URL https://doi.org/10.48550/arXiv.2305.04979.

Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In Yoshua Bengio and Yann LeCun (eds.), *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014. URL http://arxiv.org/abs/1312.6114.

Jakub Konečný, H. Brendan McMahan, Daniel Ramage, and Peter Richtárik. Federated Optimization: Distributed Machine Learning for On-Device Intelligence. *arXiv*, abs/1610.02527, 2016a. URL http://arxiv.org/abs/1610.02527.

Jakub Konečný, H. Brendan McMahan, Felix X. Yu, Peter Richtárik, Ananda Theertha Suresh, and Dave Bacon. Federated Learning: Strategies for Improving Communication Efficiency. *arXiv*, abs/1610.05492, 2016b. URL http://arxiv.org/abs/1610.05492.

Nikita Kotelevskii, Maxime Vono, Alain Durmus, and Eric Moulines. FedPop: A Bayesian Approach for Personalised Federated Learning. In *NeurIPS*, 2022. URL http://papers.nips.cc/paper_files/paper/2022/hash/395409679270591fd2a70abc694cf5a1-Abstract-Conference.html.

Ilja Kuzborskij and Csaba Szepesvári. Efron-Stein PAC-Bayesian Inequalities. arXiv:1909.01931, 2019. URL https://arxiv.org/abs/1909.01931.

Gaël Letarte, Pascal Germain, Benjamin Guedj, and François Laviolette. Dichotomize and generalize: PAC-Bayesian binary activated deep neural networks. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems [NeurIPS] 2019, 8-14 December 2019, Vancouver, BC, Canada*, pp. 6869–6879, 2019. URL https://papers.nips.cc/paper/8911-dichotomize-and-generalize-pac-bayesian-binary-activated-deep-neural-networks.

Le Li, Benjamin Guedj, and Sébastien Loustau. A quasi-Bayesian perspective to online clustering. *Electron. J. Statist.*, 12(2):3071–3113, 2018. doi: 10.1214/18-EJS1479. URL https://doi.org/10.1214/18-EJS1479.

Priyanka Mary Mammen. Federated Learning: Opportunities and Challenges, 2021.

Andreas Maurer. A note on the PAC-Bayesian theorem. *arXiv*, cs/0411099, 2004.

David A McAllester. Some PAC-Bayesian theorems. In *Proceedings of the eleventh annual conference on Computational Learning Theory*, pp. 230–234. ACM, 1998.

David A McAllester. PAC-Bayesian stochastic model selection. *Machine Learning*, 51(1):5–21, 2003.

Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-Efficient Learning of Deep Networks from Decentralized Data. In Aarti Singh and Jerry Zhu (eds.), *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, volume 54 of *Proceedings of Machine Learning Research*, pp. 1273–1282. PMLR, 20–22 Apr 2017a. URL https://proceedings.mlr.press/v54/mcmahan17a.html.

H. Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera y Arcas. Communication-Efficient Learning of Deep Networks from Decentralized Data, 2017b.

Zakaria Mhammedi, Peter Grünwald, and Benjamin Guedj. PAC-Bayes un-expected Bernstein inequality. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems [NeurIPS] 2019, 8-14 December 2019, Vancouver, BC, Canada*, pp. 12180–12191, 2019. URL http://papers.nips.cc/paper/9387-pac-bayes-un-expected-bernstein-inequality.

Mehryar Mohri, Gary Sivek, and Ananda Theertha Suresh. Agnostic Federated Learning. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 4615–4625. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/mohri19a.html.

Kento Nozawa, Pascal Germain, and Benjamin Guedj. PAC-Bayesian contrastive unsupervised representation learning. In *Conference on Uncertainty in Artificial Intelligence [UAI]*, 2020. URL https://proceedings.mlr.press/v124/nozawa20a.html.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems 32*, pp. 8024–8035. Curran Associates, Inc., 2019. URL http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.

Maria Perez-Ortiz, Omar Rivasplata, Emilio Parrado-Hernandez, Benjamin Guedj, and John Shawe-Taylor. Progress in Self-Certified Neural Networks. In *NeurIPS 2021 Workshop on Bayesian Deep Learning*, 2021.

María Pérez-Ortiz, Omar Rivasplata, John Shawe-Taylor, and Csaba Szepesvári. Tighter Risk Certificates for Neural Networks. *Journal of Machine Learning Research*, 22(227):1–40, 2021. URL http://jmlr.org/papers/v22/20-879.html.

Amirhossein Reisizadeh, Farzan Farnia, Ramtin Pedarsani, and Ali Jadbabaie. Robust Federated Learning: The Case of Affine Distribution Shifts. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin (eds.), *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/f5e536083a438cec5b64a4954abc17f1-Abstract.html.

Omar Rivasplata, Vikram M. Tankasali, and Csaba Szepesvári. PAC-Bayes with Backprop. *arXiv*, abs/1908.07380, 2019. URL http://arxiv.org/abs/1908.07380.

Jonas Rothfuss, Vincent Fortuin, Martin Josifoski, and Andreas Krause. PACOH: Bayes-optimal meta-learning with PAC-guarantees. In *International Conference on Machine Learning (ICML)*, 2021.

Jonas Rothfuss, Martin Josifoski, Vincent Fortuin, and Andreas Krause. PAC-Bayesian Meta-Learning: From Theory to Practice. *arXiv*, abs/2211.07206, 2022.

Otmane Sakhi, Nicolas Chopin, and Pierre Alquier. PAC-Bayesian Offline Contextual Bandits With Guarantees. *arXiv*, abs/2210.13132, 2022. To appear in ICML 2023.

Jonathan Scott, Hossein Zakerinia, and Christoph H. Lampert. PeFLL: A Lifelong Learning Approach to Personalized Federated Learning. *arXiv*, abs/2306.05515, 2023. doi: 10.48550/arXiv.2306.05515. URL https://doi.org/10.48550/arXiv.2306.05515.

Milad Sefidgaran, Romain Chor, Abdellatif Zaidi, and Yijun Wan. Federated Learning You May Communicate Less Often! *arXiv*, abs/2306.05862, 2023. doi: 10.48550/arXiv.2306.05862. URL https://doi.org/10.48550/arXiv.2306.05862.

Yevgeny Seldin, François Laviolette, John Shawe-Taylor, Jan Peters, and Peter Auer. PAC-Bayesian Analysis of Martingales and Multiarmed Bandits. *arXiv*, abs/1105.2416, 2011.

Yevgeny Seldin, François Laviolette, Nicolo Cesa-Bianchi, John Shawe-Taylor, and Peter Auer. PAC-Bayesian Inequalities for Martingales. *IEEE Transactions on Information Theory*, 2012.

J. Shawe-Taylor and R. C. Williamson. A PAC analysis of a Bayes estimator. In *Proceedings of the 10th annual conference on Computational Learning Theory*, pp. 2–9. ACM, 1997.

Ananda Theertha Suresh, Felix X. Yu, Sanjiv Kumar, and H. Brendan McMahan. Distributed Mean Estimation with Limited Communication. In Doina Precup and Yee Whye Teh (eds.), *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pp. 3329–3337. PMLR, 2017. URL http://proceedings.mlr.press/v70/suresh17a.html.

Alysa Ziying Tan, Han Yu, Lizhen Cui, and Qiang Yang. Towards Personalized Federated Learning. *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–17, 2022. doi: 10.1109/TNNLS.2022.3160699.

Ilya O. Tolstikhin and Yevgeny Seldin. PAC-Bayes-Empirical-Bernstein Inequality. In Christopher J. C. Burges, Léon Bottou, Zoubin Ghahramani, and Kilian Q. Weinberger (eds.), *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States*, pp. 109–117, 2013. URL https://proceedings.neurips.cc/paper/2013/hash/a97da629b098b75c294dffdc3e463904-Abstract.html.

Dimitris Tsipras, Shibani Santurkar, Logan Engstrom, Alexander Turner, and Aleksander Madry. Robustness May Be at Odds with Accuracy, 2019.

Elahe Vedadi, Joshua V. Dillon, Philip Andrew Mansfield, Karan Singhal, Arash Afkanpour, and Warren Richard Morningstar. Federated Variational Inference: Towards Improved Personalization and Generalization. *arXiv*, abs/2305.13672, 2023. doi: 10.48550/arXiv.2305.13672. URL https://doi.org/10.48550/arXiv.2305.13672.

Omry Yadan. Hydra - A framework for elegantly configuring complex applications. Github, 2019. URL https://github.com/facebookresearch/hydra.

Semih Yagli, Alex Dytso, and H. Vincent Poor. Information-Theoretic Bounds on the Generalization Error and Privacy Leakage in Federated Learning. In *2020 IEEE 21st International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, pp. 1–5, 2020. doi: 10.1109/SPAWC48557.2020.9154277.

Andy B. Yoo, Morris A. Jette, and Mark Grondona. Slurm: Simple linux utility for resource management. In Dror Feitelson, Larry Rudolph, and Uwe Schwiegelshohn (eds.), *Job Scheduling Strategies for Parallel Processing*, pp. 44–60, Berlin, Heidelberg, 2003. Springer Berlin Heidelberg. ISBN 978-3-540-39727-4.

Mikhail Yurochkin, Mayank Agarwal, Soumya Ghosh, Kristjan Greenewald, Nghia Hoang, and Yasaman Khazaeni. Bayesian Nonparametric Federated Learning of Neural Networks. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 7252–7261. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/yurochkin19a.html.

Valentina Zantedeschi, Paul Viallard, Emilie Morvant, Rémi Emonet, Amaury Habrard, Pascal Germain, and Benjamin Guedj. Learning Stochastic Majority Votes by Minimizing a PAC-Bayes Generalization Bound. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2021.

Chen Zhang, Yu Xie, Hang Bai, Bin Yu, Weihong Li, and Yuan Gao. A survey on federated learning. *Knowledge-Based Systems*, 216:106775, 2021.

Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric Xing, Laurent El Ghaoui, and Michael Jordan. Theoretically Principled Trade-off between Robustness and Accuracy. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 7472–7482. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/zhang19p.html.

Liling Zhang, Xinyu Lei, Yichun Shi, Hongyu Huang, and Chao Chen. Federated Learning for IoT Devices With Domain Generalization. *IEEE Internet Things J.*, 10(11):9622–9633, 2023a. doi: 10.1109/JIOT.2023.3234977. URL https://doi.org/10.1109/JIOT.2023.3234977.

Xiaojin Zhang, Anbu Huang, Lixin Fan, Kai Chen, and Qiang Yang. Probably Approximately Correct Federated Learning, 2023b.

Xu Zhang, Yinchuan Li, Wenpeng Li, Kaiyang Guo, and Yunfeng Shao. Personalized Federated Learning via Variational Bayesian Inference. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato (eds.), *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 26293–26310. PMLR, 17–23 Jul 2022. URL https://proceedings.mlr.press/v162/zhang22o.html.

Fuzhen Zhuang, Zhiyuan Qi, Keyu Duan, Dongbo Xi, Yongchun Zhu, Hengshu Zhu, Hui Xiong, and Qing He. A Comprehensive Survey on Transfer Learning. *Proc. IEEE*, 109(1):43–76, 2021. doi: 10.1109/JPROC.2020.3004555. URL https://doi.org/10.1109/JPROC.2020.3004555.