Confidence-aware Personalized Federated Learning via Variational Expectation Maximization

Junyi Zhu*
ESAT-PSI, KU Leuven
junyi.zhu@esat.kuleuven.be

Xingchen Ma*†
Amazon Web Services
xgchenma@amazon.de

Matthew B. Blaschko
ESAT-PSI, KU Leuven
matthew.blaschko@esat.kuleuven.be

Abstract

Federated Learning (FL) is a distributed learning scheme to train a shared model across clients. One common and fundamental challenge in FL is that the sets of data across clients could be non-identically distributed and have different sizes. Personalized Federated Learning (PFL) attempts to solve this challenge via locally adapted models. In this work, we present a novel framework for PFL based on hierarchical Bayesian modeling and variational inference. A global model is introduced as a latent variable to augment the joint distribution of clients' parameters and capture the common trends of different clients, optimization is derived based on the principle of maximizing the marginal likelihood and conducted using variational expectation maximization. Our algorithm gives rise to a closed-form estimation of a confidence value which comprises the uncertainty of clients' parameters and local model deviations from the global model. The confidence value is used to weigh clients' parameters in the aggregation stage and adjust the regularization effect of the global model. We evaluate our method through extensive empirical studies on multiple datasets. Experimental results show that our approach obtains competitive results under mild heterogeneous circumstances while significantly outperforming state-of-the-art PFL frameworks in highly heterogeneous settings.

1. Introduction

Federated learning (FL) is a distributed learning framework, in which clients optimize a shared model with their local data and send back parameters after training, and a central server aggregates locally updated models to obtain a global model that it re-distributes to clients [24]. FL is expected to address privacy concerns and to exploit the computational resources of a large number of edge devices. Despite these strengths, there are several challenges in the application of FL. One of them is the statistical heterogeneity of client data sets since in practice clients' data correlate with local environments and deviate from each other [13, 18, 19]. The most common types of heterogeneity are defined as:

Label distribution skew. Let J be the number of clients and the data distribution of client j be $P_j(x, y)$ and rewrite it as $P_j(x|y)P_j(y)$, two kinds of non-identical scenarios can be identified. One of them is *label distribution skew*, that is, the label distributions $\{P_j(y)\}_{j=1}^J$ are varying in different clients but the conditional generating distributions $\{P_j(x|y)\}_{j=1}^J$ are assumed to be the same. This could happen when certain types of data are underrepresented in the local environment.

Label concept drift. Another common type of non-IID scenario is *label concept drift*, in which the label distributions $\{P_j(y)\}_{j=1}^J$ are the same but the conditional generating distributions $\{P_j(x|y)\}_{j=1}^J$ are different across different clients. This could happen when features of the same type of data differ across clients and correlates with their environments, e.g. the Labrador Retriever (most popular dog in the United States) and the Border Collie (most popular dog in Europe) look different, thus the dog pictures taken by the clients in these two areas contain *label concept drift*.

Data quantity disparity. Additionally, clients may possess different amounts of data. Such *data quantity disparity* can lead to inconsistent uncertainties of the locally updated models and heterogeneity in the number of local updates. In practice, the amount of data could span a large range across clients, for example large hospitals usually have many more medical records than clinics. In particular, data quantity distributions often exhibit that large datasets are concentrated in a few locations, whereas a large amount of data is scattered across many locations with small dataset sizes [11, 32].

It has been proven that if federated averaging (`FedAvg` [24]) is applied, the aforementioned heterogeneity will slow down the convergence of the global model and in some cases leads to arbitrary deviation from the optimum [19, 33]. Several works have been proposed to alleviate this problem [4, 18, 33]. Another stream of work is personalized fed-

*Authors with equal contribution.
†Work was done at KU Leuven prior to joining Amazon.

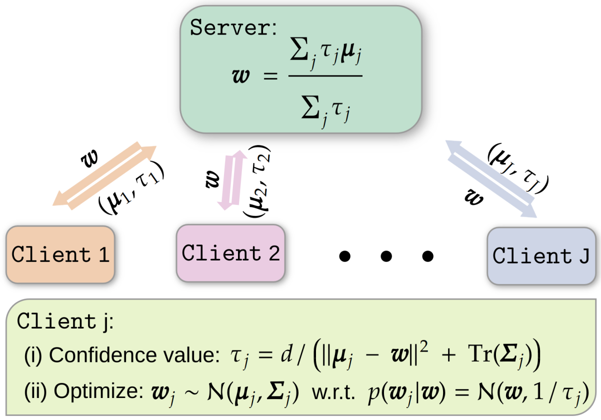


Figure 1. An overview of our confidence-aware PFL framework.

erated learning (PFL) [5, 7, 30, 31, 40], which trains multiple local models instead of a single global model. Most PFL frameworks still construct and optimize a shared model which serves as a constraint or initialization for the personalization of local models. In the theoretical analysis of most existing PFL frameworks, the un-weighted average of clients' model parameters is used as the shared model, however, to obtain better empirical results, a weighted average is usually used and the weights depend on the local data size. This inconsistency indicates a more principled method is needed to optimize the shared model.

To achieve this, we present a novel Bayesian framework through the lens of a hierarchical latent variable model. In this framework, a latent shared model manages to capture the common trend across local models, and local models adapt to individual environments based on the prior information provided by the shared model. In particular, we assume a conditional Gaussian distribution and use variational expectation maximization to optimize the Bayesian models, such that a closed-form solution for the shared model optimization w.r.t. clients' confidence values can be derived. The confidence value[1] is inversely proportional to the sum of the variance of variational approximation (uncertainty) and squared difference between the mean and the shared model (model deviation). Therefore a low confidence value indicates a high uncertainty or model deviation, and naturally the contribution of this local model to the shared model is considered to be low. An illustration of the proposed confidence-aware PFL framework is presented in Figure 1.

Additionally, most previous works are solely evaluated under the *label distribution skew* scenario, in this work we also investigate *label concept drift*. As far as we know, there are only few works [23, 27] considering *label concept drift* in the setting of PFL. We believe that this scenario is more challenging than *label distribution skew*, since the conditional data distribution from the same label in *label distribution skew* is the same across different clients. In *label concept drift* there can be high discrepancy between local data distributions, indicating higher heterogeneity.

[1]The confidence value is called precision in statistics.

Paper organization. Necessary background and notation that will be used in this paper are given in the next section. Section 3 presents the theoretical analysis of the proposed framework and the implementation of our algorithm is provided in Section 4. Experimental results are presented in Section 5. Related works are discussed in Section 6. Finally, we conclude in Section 7.

2. Problem Formulation

In FL a central server orchestrates clients to learn a global objective and the goal is to minimize:

$$\min_{\boldsymbol{w} \in \mathbb{R}^d} f(\boldsymbol{w}; \mathcal{D}) := \frac{1}{J} \sum_{j=1}^{J} f_j(\boldsymbol{w}; \mathcal{D}_j), \tag{1}$$

where J is the number of clients, $\mathcal{D}_j$ is the set of data available in client j, and $f_j(\boldsymbol{w}; \mathcal{D}_j)$ is the empirical average loss function of the j-th client:

$$f_j(\boldsymbol{w}; \mathcal{D}_j) = \frac{1}{n_j} \sum_{i=1}^{n_j} l(\boldsymbol{x}_i^{(j)}, y_i^{(j)}; \boldsymbol{w}), \tag{2}$$

where $(\boldsymbol{x}_i^{(j)}, y_i^{(j)}) \in \mathcal{D}_j$ is one data point of client j, $l(\cdot, \cdot; \boldsymbol{w})$ is the loss function using parameters $\boldsymbol{w}$ and $n_j := |\mathcal{D}_j|$ is the number of data points on the j-th client. Certainly, if $\{D_j\}_1^J$ are non-identically distributed, $\boldsymbol{w}$ cannot be optimal for all clients. Instead of using a single global model as in FL, in PFL we aim to solve the composed optimization problem:

$$\min_{\boldsymbol{w}_{1:J} \in \mathbb{R}^d} f(\boldsymbol{w}_{1:J}; \mathcal{D}) := \frac{1}{J} \sum_{j=1}^{J} f_j(\boldsymbol{w}_j; \mathcal{D}_j), \tag{3}$$

where $\boldsymbol{w}_{1:J}$ is a shorthand for the set of parameters $\{\boldsymbol{w}_1, \cdots, \boldsymbol{w}_J\}$ and $\boldsymbol{w}_j$ is the personalized parameter for the j-th client.

3. Confidence-aware Personalized Federated Learning

In this section, firstly we propose a general Bayesian framework for PFL. Then we derive the optimization methods based on Variational Expectation Maximization (VEM). We therefore name our proposed approach `pFedVEM`. We will see that our variational Bayes approach enables the clients to estimate confidence values for their local training results, which automatically adjust the weights in the model aggregation and the strengths of regularization.

3.1. Hierarchical Bayesian modeling

To develop a Bayesian framework, we need to obtain a posterior distribution for parameters which we are interested in. Once we have the posterior distribution, any deductive reasoning can be conducted. In the context of PFL, the target is the posterior distribution $p(\boldsymbol{w}_j|\mathcal{D}_j)$ of $\boldsymbol{w}_j$ for any client j. The most easy way to obtain $p(\boldsymbol{w}_j|\mathcal{D}_j)$ is by performing Bayesian inference locally. Given a weak prior $p(\boldsymbol{w}_j)$, the disadvantage of this approach is the variance of $p(\boldsymbol{w}_j|\mathcal{D}_j)$ could be high if the data quantity $|\mathcal{D}_j|$ on client j is limited. However in the context of Bayesian networks, a weak prior is almost unavoidable [34].

Another way to understand this is, since all clients are running similar tasks, $\mathcal{D}_{\{1,\cdots,J\}\backslash j}$ should be able to provide information to form the posterior of $\boldsymbol{w}_j$. In a distributed learning scheme like FL, the j-th client has no access to $\mathcal{D}_{\{1,\cdots,J\}\backslash j}$ and it is impossible to obtain the posterior $p(\boldsymbol{w}_j|\mathcal{D})$ directly. To overcome this restriction, we introduce a latent variable $\boldsymbol{w}$ such that all $\boldsymbol{w}_{1:J}$ depend on $\boldsymbol{w}$ and $\boldsymbol{w}$ captures the correlations between different clients. We slightly abuse the notation of $\boldsymbol{w}$ which denotes the global model in Section 2 as the latent variable can also act as a global model fitted on the complete data distribution in our approach. The relation between $\boldsymbol{w}$ and $\boldsymbol{w}_{1:J}$ implies conditional independence between clients:

$$p(\boldsymbol{w}_i|\boldsymbol{w})p(\boldsymbol{w}_j|\boldsymbol{w}) = p(\boldsymbol{w}_i, \boldsymbol{w}_j|\boldsymbol{w}). \tag{4}$$

The conditional distribution $p(\boldsymbol{w}_j|\boldsymbol{w})$ enables client's models $\boldsymbol{w}_{1:J}$ to be specialized in individual environments based on the common trend carried by the latent variable $\boldsymbol{w}$. We now turn to obtain the augmented posterior distribution of $\{\boldsymbol{w}, \boldsymbol{w}_{1:J}\}$. Using Bayes' rule, the posterior is proportional to the product of the prior and the likelihood function:

$$\begin{aligned} p(\boldsymbol{w}, \boldsymbol{w}_{1:J}|\mathcal{D}) &\propto p(\boldsymbol{w}, \boldsymbol{w}_{1:J})p(\mathcal{D}|\boldsymbol{w}, \boldsymbol{w}_{1:J}) \\ &\overset{4}{=} p(\boldsymbol{w}) \prod_{j=1}^{J} p(\boldsymbol{w}_j|\boldsymbol{w}) \exp\left(-n_j f_j(\boldsymbol{w}_j; \mathcal{D}_j)\right), \end{aligned}$$

where $p(\boldsymbol{w})$ is the prior distribution of the latent variable, $f_j(\boldsymbol{w}_j; \mathcal{D}_j)$ is defined in Equation (2) and is proportional to the negative of the data log-likelihood on client j. From the above augmented joint distribution, we see the introduction of the latent variable $\boldsymbol{w}$ enables complicated communication across clients. The marginalized joint distribution $p(\boldsymbol{w}_{1:J}|\mathcal{D}) = \int p(\boldsymbol{w}, \boldsymbol{w}_{1:J}|\mathcal{D})d\boldsymbol{w}$ can thus be flexible.

3.2. Variational expectation maximization

Before an update scheme for $\{\boldsymbol{w}, \boldsymbol{w}_{1:J}\}$ can be derived, it is necessary to specify the concrete forms for the conditional density $p(\boldsymbol{w}_j|\boldsymbol{w})$. In this work, we assume an isotropic Gaussian conditional prior $p(\boldsymbol{w}_j|\boldsymbol{w}) = \mathcal{N}(\boldsymbol{w}_j \mid \boldsymbol{w}, \rho_j^2\boldsymbol{I})$, where ρ_j^2 is the variance of this distribution. A Gaussian conditional implies all clients' parameters are close to this latent variable $\boldsymbol{w}$, which is a reasonable assumption since all clients are running similar tasks. Additionally, this enables a closed form for updating $\boldsymbol{w}$. In Section 5, it will be shown that the isotropic Gaussian assumption works well in practice.

Maximizing the marginal likelihood. One way to optimize the proposed Bayesian model is Maximum a Posterior Probability (MAP) which seeks a maximizer to the unnormalized posterior, the overall optimization is efficient and easy to implement. However, for MAP the assumption $p(\boldsymbol{w}_j|\boldsymbol{w}) = \mathcal{N}(\boldsymbol{w}_j \mid \boldsymbol{w}, \rho_j^2\boldsymbol{I})$ gives rise to a group of hyperparameters $\rho_{1:J}$, which is hard to set. The point estimation of $\boldsymbol{w}_j$ can also be unreliable. To address these issues, we introduce factorized variational approximation $q(\boldsymbol{w}_{1:J}) := \prod_{j=1}^{J} q_j(\boldsymbol{w}_j)$ to the true posterior distribution $p(\boldsymbol{w}_{1:J}|\mathcal{D})$. In this work, the axis-aligned multivariate Gaussian is used as the variational family, that is, $q_j(\boldsymbol{w}_j) = \mathcal{N}(\boldsymbol{w}_j \mid \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)$ and $\boldsymbol{\Sigma}$ is a diagonal matrix. To optimize these approximations $\{q_j(\boldsymbol{w}_j)\}_{j=1}^{J}$, we maximize the evidence lower bound (ELBO) of the marginal likelihood:

$$\begin{aligned} &\mathrm{ELBO}\left(q(\boldsymbol{w}_{1:J}), \rho_{1:J}^2, \boldsymbol{w}\right) \\ &= \sum_{j=1}^{J} \mathbb{E}_{q(\boldsymbol{w}_j)}[\log p(\mathcal{D}_j|\boldsymbol{w}_j)] - \mathrm{KL}[q(\boldsymbol{w}_j) \parallel p(\boldsymbol{w}_j|\boldsymbol{w}, \rho_j^2)]. \end{aligned} \tag{5}$$

The above ELBO can be maximized using VEM through blockwise coordinate descent. First, to obtain the variational approximations $q(\boldsymbol{w}_{1:J})$, for the j-th client we only need to use $\mathcal{D}_j$ and maximize:

$$\mathbb{E}_{q(\boldsymbol{w}_j)}[\log p(\mathcal{D}_j|\boldsymbol{w}_j)] - \mathrm{KL}[q(\boldsymbol{w}_j) \parallel p(\boldsymbol{w}_j|\boldsymbol{w}, \rho_j^2)] \tag{6}$$

Then after these local approximations have been formed, the server attempts to optimize the ELBO in Equation (5) by updating the latent variable $\boldsymbol{w}$ using the client's updated variational parameters. Simplifying Equation (5) w.r.t. $\boldsymbol{w}$ and $\rho_{1:J}$, we derive the objective function for server:

$$\begin{aligned} &\mathrm{ELBO}(\rho_{1:J}^2, \boldsymbol{w}) \\ &= \sum_{j=1}^{J} \mathbb{E}_{q(\boldsymbol{w}_j)}[\log p(\mathcal{D}_j, \boldsymbol{w}_j)] - \mathbb{E}_{q(\boldsymbol{w}_j)}[\log q(\boldsymbol{w}_j)] \\ &\propto \sum_{j=1}^{J} \mathbb{E}_{q(\boldsymbol{w}_j)}[\log p(\mathcal{D}_j|\boldsymbol{w}_j, \boldsymbol{w}) + \log p(\boldsymbol{w}_j|\boldsymbol{w}, \rho_j^2)] \\ &\propto \sum_{j=1}^{J} \mathbb{E}_{q(\boldsymbol{w}_j)}[\log p(\boldsymbol{w}_j|\boldsymbol{w}, \rho_j^2)]. \end{aligned} \tag{7}$$

The last line holds because $\log p(\mathcal{D}_j|\boldsymbol{w}_j, \boldsymbol{w}) = \log p(\mathcal{D}_j|\boldsymbol{w}_j)$ by assumption and $\log p(\mathcal{D}_j|\boldsymbol{w}_j)$ does not depend on $\boldsymbol{w}$ and $\rho_{1:J}^2$. Setting the first order derivative of Equation (7) w.r.t. $\boldsymbol{w}$ and $\rho_{1:J}^2$ to be zero, we derive the closed-form solutions for these parameters and we define the confidence value $\tau_j := 1/\rho_j^2$:

Confidence value:

$$\tau_j = d/(\overbrace{\mathrm{Tr}(\boldsymbol{\Sigma}_j)}^{\text{Uncertainty}} + \underbrace{\|\boldsymbol{\mu}_j - \boldsymbol{w}\|^2}_{\text{Model deviation}}), \tag{8}$$

Confidence-aware aggregation:

$$\boldsymbol{w}^* = \frac{\sum_{j=1}^{J} \tau_j \boldsymbol{\mu}_j}{\sum_{j=1}^{J} \tau_j}; \tag{9}$$

where d is the dimension of $\boldsymbol{w}$, $\mathrm{Tr}(\boldsymbol{\Sigma}_j)$ is the trace of the variational variance-covariance parameter, which represents the uncertainty of $\boldsymbol{\mu}_j$, and $\|\boldsymbol{\mu}_j - \boldsymbol{w}\|^2$ represents the model deviation induced by the heterogeneous data distribution.

3.3. Advantages

From Equation (8) we see the proposed variational Bayes approach `pFedVEM` enables the clients to estimate the confidence values over their local training results, which involve the derivation of a local model from the global model and the uncertainty of the trained parameters, such that the lower uncertainty and model deviation, the higher confidence. Then in the aggregation (cf. Equation (9)), the server will form a weighted average from the uploaded parameters w.r.t. the corresponding confidence values.

Such confidence-aware aggregation has two advantages: (i) a local model with lower uncertainty has a larger weight. For a similar purpose, many previous works assign weights based on the local data size. However, in the case that a client has a large amount of duplicated or highly correlated data, the uncertainty of our method would be less affected and more accurate. (ii) A local model is weighted less if it highly deviates from the global model. A large distance between the local model and global model indicates that the local data distribution differs a lot from the population distribution. Considering the model deviation will make the aggregation more robust to outliers (e.g. clients with data out of the bulk of the population distribution).

The confidence value also adjusts the regularization effect of the KL divergence term in Equation (6) during local training. Armed with the isotropic Gaussian assumption, we can now derive the closed form of that KL divergence regularizer. Simplifying w.r.t. $\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j$:

$$\begin{aligned}
&\mathrm{KL}[q(\boldsymbol{w}_j) \parallel p(\boldsymbol{w}_j|\boldsymbol{w}, \rho_j^2)] \\
&\propto -\sum_i \log \sigma_{j,i} + (\mathrm{Tr}(\boldsymbol{\Sigma}_j) + \|\boldsymbol{\mu}_j - \boldsymbol{w}\|^2)\tau_j/2 \\
&\geq -\frac{1}{2}\log \mathrm{Tr}(\boldsymbol{\Sigma}_j) + \frac{\tau_j}{2}(\mathrm{Tr}(\boldsymbol{\Sigma}_j) + \|\boldsymbol{\mu}_j - \boldsymbol{w}\|^2),
\end{aligned} \tag{10}$$

where $\sigma_{j,1:d}^2$ is the diagonal of $\boldsymbol{\Sigma}_j$ and the last line is taken according to Jensen's inequality and convexity of the negative log function.

Based on Equation (10), we observe that the gradient of $\boldsymbol{\mu}_j$ w.r.t. this KL divergence, i.e. $(\boldsymbol{\mu}_j - \boldsymbol{w})\tau_j$, is rescaled by the confidence value τ_j such that: (i) in case a client has a low uncertainty, e.g. the local data set is rich, the gradient arising from the log likelihood in Equation (6) will be large, while $(\boldsymbol{\mu}_j - \boldsymbol{w})\tau_j$ is also enlarged due to large τ_j, such that the information of the global model can be conveyed to the local model. (ii) If $\boldsymbol{\mu}_j$ tends to be highly deviated from $\boldsymbol{w}$, the regularization effect will not blow up due to reduced τ_j, therefore better personalization can be achieved if data are abundant and highly correlated to the local environments.

4. Algorithm Implementation

In this section, we discuss the implementation of our approach, especially the technical difficulties of optimizing Equation (6) and present the algorithm of `pFedVEM`.

Numerical stability and reparametrization. To guarantee the non-negativity of $\boldsymbol{\Sigma}$ and improve the numerical stability, we parameterize the Gaussian variational family of the clients with $(\boldsymbol{\mu}_{1:J}, \boldsymbol{\pi}_{1:J})$ such that the standard deviation of $q(\boldsymbol{w}_j)$ is $\mathrm{diag}(\log(1+\exp(\boldsymbol{\pi}_j)))$. Then in order to conduct gradient descent, we instead sample from the normal distribution and implement for any client j:

$$q(\boldsymbol{w}_j) = \boldsymbol{\mu}_j + \mathrm{diag}(\log(1+\exp(\boldsymbol{\pi}_j))) \cdot \mathcal{N}(0, \boldsymbol{I}_d). \tag{11}$$

Monte-Carlo approximation. Equation (6) contains the expectation $\mathbb{E}_{q(\boldsymbol{w}_j)}[\log p(\mathcal{D}_j|\boldsymbol{w}_j)]$ which rarely has a closed form. We therefore resort to Monte-Carlo (MC) estimation to approximate its value, for K times MC sampling, the objective becomes:

$$\frac{n_j}{K}\sum_{k=1}^{K} f_j(\mathcal{D}_j; \boldsymbol{w}_{j,k}) - \mathrm{KL}[q(\boldsymbol{w}_j) \parallel p(\boldsymbol{w}_j|\boldsymbol{w}, \rho_j^2)].$$

Head-base architecture. Empirically, we find the optimization of $q(\boldsymbol{w}_j)$ is more efficient using a head-base architecture design, which splits the entire network into a base model and a head model. The former outputs a representation of the data and the latter is a linear classifier layer following the base model. Such an architecture is also used in non-Bayesian PFL frameworks [3,5]. We personalize the head model with `pFedVEM` while letting the base model be trained via `FedAvg`. Additionally, the computational demand of `pFedVEM` is thus moderate compared with [40]. We do not exclude the possibility that using other federated optimization methods may obtain a better base model, but as we will show in Section 5, equipping `pFedVEM` with `FedAvg` already gives significantly improved results, so we leave other combinations for future work.

Algorithm 1 pFedVEM: PFL via Variational Expectation Maximization

Server input: T, $\boldsymbol{w}^0$, $\boldsymbol{\theta}^0$, s
Client input: $\boldsymbol{\mu}_{1:J}^0$, $\boldsymbol{\Sigma}_{1:J}^0$, R, K, η

for $t = 0$ to $T-1$ **do**
  Server executes:
  for $j = 1, \ldots, J$ **in parallel do**
    ClientUpdate($\boldsymbol{w}^t, \boldsymbol{\theta}^t$)
  Server selects a random subset of clients $\mathcal{S}_t$ from binomial distribution $B(J, s)$.
  Each client $j \in \mathcal{S}_t$ sends its updated variational parameters $\boldsymbol{\mu}_j^{t+1}$, τ_j^t and base model $\boldsymbol{\theta}_j^{t+1}$ to the server.
  ▷ *Server optimizes the latent variable* ◁
  $\boldsymbol{w}^{t+1} = \frac{\sum_{j \in \mathcal{S}_t} \tau_j^t \boldsymbol{\mu}_j^{t+1}}{\sum_{j \in \mathcal{S}_t} \tau_j^t}$
  ▷ *Server optimizes the base model* ◁
  $\boldsymbol{\theta}^{t+1} = \frac{\sum_{j \in \mathcal{S}_t} n_j \boldsymbol{\theta}_j^{t+1}}{\sum_{j \in \mathcal{S}_t} n_j}$
  ClientUpdate($\boldsymbol{w}^t, \boldsymbol{\theta}^t$)
  ▷ *Client optimizes* τ_j ◁
  $\tau_j^t = d/(\mathrm{Tr}(\boldsymbol{\Sigma}_j^{t+1}) + \|\boldsymbol{\mu}_j^{t+1} - \boldsymbol{w}^{t+1}\|^2)$
  ▷ *SGD on* $\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j$ *with* R *epochs,* K *sampling times and learning rate* η. ◁
  $(\boldsymbol{\mu}_j^{t+1}, \boldsymbol{\Sigma}_j^{t+1}) \in \arg\min_{(\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)} \mathbb{E}_{q(\boldsymbol{w}_j)}[\log p(\mathcal{D}_j|\boldsymbol{w}_j)] - \mathrm{KL}[q(\boldsymbol{w}_j) \,\|\, p(\boldsymbol{w}_j|\boldsymbol{w}^t, 1/\tau_j^t]$
  ▷ *SGD on the base model using the hyperparameters of FedAvg.* ◁
  $\boldsymbol{\theta}_j^{t+1} \in \arg\min_{\boldsymbol{\theta}} \mathbb{E}_{q(\boldsymbol{w}_j)}[f(\mathcal{D}_j; \boldsymbol{w}_j, \boldsymbol{\theta})]$

Following [31, 40], at each communication round t, the server broadcasts the latent variable $\boldsymbol{w}^t$ and base model $\boldsymbol{\theta}^t$ to all clients and receives the updated variational parameters and base models from a subset $\mathcal{S}_t$ of clients. The update of server parameters $(\boldsymbol{w}, \tau_{1:J})$ depends on each other (cf. Equation (8) and Equation (9)), we choose to update $\boldsymbol{w}$ first and then τ_j, thus τ_j can be updated at the client side after receiving the new $\boldsymbol{w}$. During the local training, clients first update the head model based on the latest base model and then optimize the base model w.r.t. the updated head model. It is worth noting that `pFedVEM` only adds one more scalar (τ_j) to the communication message besides the model parameters, *thus the communication cost is almost unchanged*. We summarize the optimization steps of `pFedVEM` in Algorithm 1.

5. Experiments

In this section we validate the performance of our approach `pFedVEM` when clients' data are statistically heterogeneous, i.e. *label distribution skew* and *label concept drift*. We also investigate *data quantity disparity* as introduced in Section 1. Additionally, we study a case of *feature distribution skew* (a mixture of *label distribution skew* and *label concept drift*), the results are given in Appendix E.3. We compare our approach with the following FL frameworks: (1) `FedAvg` [24], (2) `FedProx` [18], (3) `Scaffold` [14], PFL frameworks: (4) `FedPer` [3], (5) `FedRep` [5], (6) `IFCA` [9], (7) `PerFedAvg` [7], and PFL frameoworks that also output a global model: (8) `pFedME` [31], (9) `pFedBayes` [40], as well as the trivial local training scheme: (10) `Local`.

5.1. Experimental settings

To evaluate our method, we target image classification problems. Most previous works evaluate on Fashion-MNIST (FMNIST) [35] and CIFAR10 [15] datasets, with which *label distribution skew* can be modeled such that each client has a subset of all labels. We also consider this setting and let each client have 5 out of 10 labels randomly. Furthermore, we model *label concept drift* using CIFAR100 [15] and SUN397 [36] datasets. These two hierarchical datasets contain superclasses and subclasses (e.g. subclass couch belongs to the superclass household furniture). We set the classification task to be superclass prediction. CIFAR100 has 20 superclasses and each superclass has 5 subclasses. SUN397 has 3 superclasses and each superclass has 50 subclasses.[2] For each client, we first sample a random subclass from each superclass (1 out of 5 for CIFAR100 and 1 out of 50 for SUN397), then the client's local data is chosen from this subclass, hence *label concept drift* is induced. To model *data quantity disparity*, we randomly split the training set into partitions of different sizes by uniformly sampling slicing indices, and then distribute one partition to each client. The strength of splitting compared with sampling (without replacement) is that we can use the full training set and local data size can span over a wide range (the sampling range needs to be conservative, otherwise we may run into an out-of-index problem). A concrete example of such a data partition and the resulting local data size distribution is detailed in Appendix A.

When running the experiments the number of communication rounds is set to be 100. We evaluate in all settings the number of clients $J \in \{50, 100, 200\}$; the more clients the more scattered is the training data. To model stragglers, each client has a probability of 0.1 to send its parameters back to the server at each round of communication. Following [1, 31, 40], we consider a MLP with one hidden layer for FMNIST and LeNet-like CNN [16] for CIFAR10. The motivation of using small models in FL frameworks is that a single edge device usually has very limited computing power and memory. We devise a deeper CNN with 6 lay-

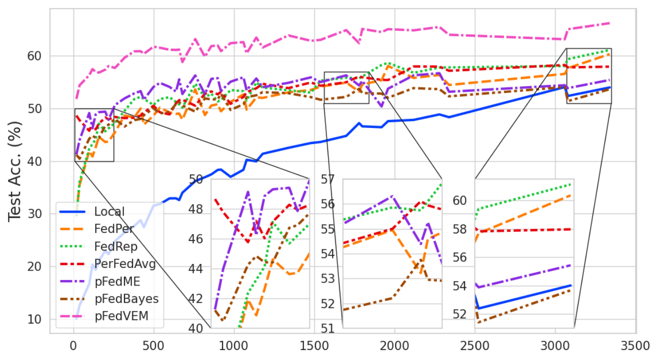
[2] SUN397 dataset is unbalanced so we sample 50 subclasses for each superclass and 100 data points for each subclass. Data are split randomly with 80% and 20% for training and testing. Images are resized to 64×64.

Dataset	Method	50 Clients		100 Clients		200 Clients	
		PM	GM	PM	GM	PM	GM
FMNIST	Local	89.2 ± 0.1	–	87.5 ± 0.1	–	85.7 ± 0.1	–
	FedAvg	–	83.5 ± 0.4	–	85.4 ± 0.3	–	85.9 ± 0.2
	FedProx	–	84.8 ± 0.5	–	$\mathbf{86.3 \pm 0.2}$	–	$\mathbf{86.5 \pm 0.1}$
	Scaffold	–	$\mathbf{85.6 \pm 0.2}$	–	85.4 ± 0.1	–	84.6 ± 0.0
	FedPer	91.4 ± 0.1	–	90.7 ± 0.1	–	89.7 ± 0.1	–
	FedRep	91.5 ± 0.1	–	90.7 ± 0.1	–	89.9 ± 0.1	–
	IFCA	84.1 ± 1.0	–	85.6 ± 0.2	–	86.1 ± 0.2	–
	PerFedavg	88.7 ± 0.2	–	88.6 ± 0.1	–	88.3 ± 0.2	–
	pFedME	$\mathbf{91.9 \pm 0.1}$	82.0 ± 0.7	$\mathbf{91.4 \pm 0.1}$	84.4 ± 0.6	90.6 ± 0.1	85.1 ± 0.1
	pFedBayes	$\mathbf{91.9 \pm 0.1}$	83.5 ± 0.3	91.3 ± 0.1	84.2 ± 0.3	90.5 ± 0.1	84.4 ± 0.1
	Ours	91.8 ± 0.1	83.9 ± 0.3	$\mathbf{91.4 \pm 0.1}$	85.6 ± 0.2	$\mathbf{90.7 \pm 0.1}$	86.2 ± 0.2
CIFAR10	Local	56.9 ± 0.1	–	52.1 ± 0.1	–	46.6 ± 0.1	–
	FedAvg	–	57.7 ± 0.9	–	59.4 ± 0.6	–	59.2 ± 0.3
	FedProx	–	58.0 ± 0.7	–	59.4 ± 0.5	–	59.1 ± 0.2
	Scaffold	–	$\mathbf{60.4 \pm 0.3}$	–	59.8 ± 0.2	–	55.4 ± 0.3
	FedPer	72.7 ± 0.3	–	68.4 ± 0.4	–	63.4 ± 0.3	–
	FedRep	71.4 ± 0.3	–	67.4 ± 0.4	–	62.8 ± 0.2	–
	IFCA	59.4 ± 0.8	–	60.1 ± 0.5	–	59.5 ± 0.5	–
	PerFedavg	62.9 ± 0.8	–	65.6 ± 0.8	–	64.2 ± 0.1	–
	pFedME	72.3 ± 0.1	56.6 ± 1.0	71.4 ± 0.2	$\mathbf{60.1 \pm 0.3}$	68.5 ± 0.2	58.7 ± 0.2
	pFedBayes	71.4 ± 0.3	52.0 ± 1.0	68.5 ± 0.3	53.2 ± 0.7	64.6 ± 0.2	51.4 ± 0.3
	Ours	$\mathbf{73.2 \pm 0.2}$	56.0 ± 0.4	$\mathbf{71.9 \pm 0.1}$	$\mathbf{60.1 \pm 0.2}$	$\mathbf{70.1 \pm 0.3}$	$\mathbf{59.4 \pm 0.3}$
CIFAR100	Local	34.3 ± 0.2	–	27.6 ± 0.3	–	22.2 ± 0.2	–
	FedAvg	–	51.7 ± 0.5	–	49.4 ± 0.7	–	44.7 ± 0.5
	FedProx	–	48.4 ± 0.6	–	45.5 ± 0.5	–	42.4 ± 0.3
	Scaffold	–	47.2 ± 0.4	–	41.4 ± 0.7	–	30.0 ± 0.1
	FedPer	49.7 ± 0.7	–	39.3 ± 0.7	–	30.6 ± 0.9	–
	FedRep	50.9 ± 0.9	–	41.2 ± 0.6	–	30.5 ± 0.6	–
	IFCA	51.9 ± 1.0	–	49.2 ± 0.7	–	44.9 ± 0.6	–
	PerFedavg	52.1 ± 0.4	–	48.3 ± 0.5	–	40.1 ± 0.3	–
	pFedME	52.5 ± 0.5	47.9 ± 0.5	47.6 ± 0.5	45.1 ± 0.3	41.6 ± 1.8	41.5 ± 1.6
	pFedBayes	49.6 ± 0.3	42.5 ± 0.5	46.5 ± 0.2	41.3 ± 0.3	40.1 ± 0.3	37.4 ± 0.3
	Ours	$\mathbf{61.0 \pm 0.4}$	$\mathbf{52.8 \pm 0.4}$	$\mathbf{56.2 \pm 0.4}$	$\mathbf{52.3 \pm 0.4}$	$\mathbf{51.1 \pm 0.6}$	$\mathbf{49.2 \pm 0.5}$
SUN397	Local	82.4 ± 0.9	–	72.0 ± 2.2	–	67.4 ± 1.4	–
	FedAvg	–	73.2 ± 0.1	–	72.6 ± 0.1	–	72.7 ± 0.4
	FedProx	–	$\mathbf{73.7 \pm 0.2}$	–	73.3 ± 0.4	–	70.8 ± 0.3
	Scaffold	–	69.5 ± 0.4	–	65.5 ± 0.4	–	59.9 ± 0.6
	FedPer	88.4 ± 0.4	–	82.3 ± 0.2	–	80.0 ± 0.1	–
	FedRep	87.8 ± 0.3	–	82.1 ± 1.2	–	79.6 ± 0.4	–
	IFCA	72.5 ± 0.5	–	71.5 ± 0.5	–	68.0 ± 0.5	–
	PerFedavg	76.5 ± 0.7	–	73.5 ± 0.6	–	72.4 ± 0.7	–
	pFedME	89.6 ± 0.7	72.2 ± 0.7	82.8 ± 2.0	72.3 ± 0.6	82.9 ± 1.1	73.0 ± 1.5
	pFedBayes	83.7 ± 0.7	66.1 ± 1.0	77.4 ± 2.0	65.4 ± 0.6	74.6 ± 0.3	64.2 ± 0.4
	Ours	$\mathbf{91.1 \pm 0.2}$	73.3 ± 0.4	$\mathbf{86.6 \pm 1.2}$	$\mathbf{74.1 \pm 0.7}$	$\mathbf{84.5 \pm 0.5}$	$\mathbf{74.3 \pm 0.8}$

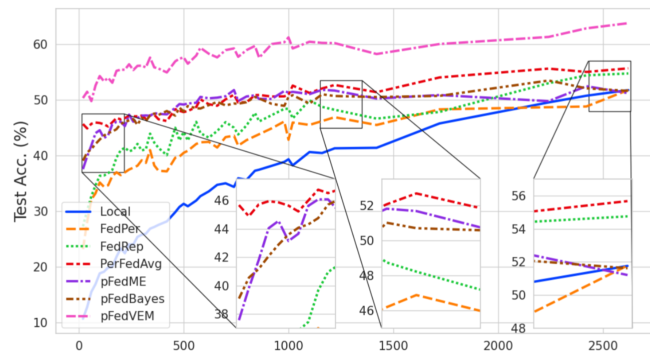
Table 1. Average test accuracy of PMs and test accuracy of GM (% $\pm$ SEM) over 50, 100, 200 clients on FMNIST, CIFAR10, CIFAR100 and SUN397. Best result is in bold.

ers for CIFAR100 and SUN397 as the tasks on these two datasets are more complex. We illustrate the network architectures in Appendix B.

For experiments on each dataset, we search for hyperparameters with the number of clients $J = 100$ and use these hyperparameters for the other two cases $J = \{50, 200\}$. For `pFedVEM` we search for the learning rate $\eta \in \{0.01, 0.001, 0.0001\}$, initial variance $\rho^2_{1:J} \in$



(a) Client = 50



(b) Client = 100

Figure 2. Test accuracy vs. local data size over 50, 100 clients on CIFAR100.

$\{1, 0.1, 0.01\}$ and client training epochs $R \in \{5, 10, 20\}$, MC sampling is fixed to 5 times. The optimization method is set to full-batch gradient descent, so we do not need to tune on the batch-size. We extensively search for the baselines' hyperparameters including learning rate, epochs, batch-size and special factors depending on the frameworks. Tables with respective hyperparameters and corresponding searching ranges are presented in Appendix C.

We evaluate both a personalized model (PM) and global model (GM). PMs are evaluated with test data corresponding to the respective labels (for *label distribution skew*) or subclasses (for *label concept drift*) the clients have, while GM is evaluated on the complete test set. All experiments have been repeated for five times using the same group of five random seeds which are used for data generation, parameter initialization and client sampling. We report the mean and its standard error (SEM). All experiments are conducted on a cluster within the same container environment.

5.2. Results

Overall performance. We first present the average of PMs' test accuracy along with the GM's test accuracy which are two typical evaluation values of PFL and FL frameworks. As shown in Table 1, `pFedVEM` is competitive on FMNIST and obtains better PMs on CIFAR10 and SUN397, while it's PMs and GM on CIFAR100 significantly outperform the baselines. Based on the model statistics estimated by `pFedVEM` (see Appendix D) we observe that when trained on CIFAR100 local models and confidence values are more scattered, which indicates that `pFedVEM` is more robust and capable at handling high statistical heterogeneity. We also presents the plots of accuracy vs. communication rounds to compare the convergence rate of different approaches in Appendix E.1.

Additionally, Bayesian neural networks (BNNs) are known for their exceptional performance when data is scarce [40]. This is because BNNs deal with probability distributions instead of point estimates, and the prior distribution over the weights serves as a natural regularizer. So far, we have compared `pFedVEM` to baselines using the full training datasets. To demonstrate the advantage of `pFedVEM` with limited data, we examine two cases: (i) the accuracy of the 10% of clients with the smallest local datasets, (ii) a smaller total number of training samples $|D|$ over all clients. The results are provided in Appendix E.2.

Accuracy vs. local data quantity. FL is a collaboration framework. To attract more clients joining in the collaboration, we need to provide sufficient incentive. Utility gain is a major motivation, which we define as the gap of local model performance between local and federated training. Although average accuracy of PMs in Table 1 reflects the overall utility gain of a federated group, it is deficient in characterizing the utility gain of individual clients, especially considering the utility gain could vary for clients with relatively more or less local data in a federated group. To investigate this, we plot the accuracy over individual data size on CIFAR100 with 50 and 100 clients (see Figure 2). Comparing PFL frameworks with `Local`, we observe: (1) Generally, clients with relatively fewer data can gain more by joining a collaboration. (2) `FedRep` is good at supporting clients with relatively larger amounts of data, but tends to ignore small clients. (3) `pFedBayes` and `pFedME` can train small clients well, but big clients may not gain or even lose performance by joining these two frameworks, perhaps due to the strong constraint to the global model for a better overall performance. (4) Clients with different data sizes benefit from our confidence-aware PFL framework `pFedVEM`.

Ablation study. We also conduct ablation studies to understand the two terms uncertainty and model deviation in the confidence value of Equation (8) by retaining only the

Method	Hetero.	Homo.
pFedVEM	61.0 ± 0.4	49.4 ± 0.2
Uncertainty	59.9 ± 0.2	49.2 ± 0.2
Mean Diff. (%)	$\mathbf{-1.1}$	-0.2

Table 2. Test accuracy (% $\pm$ SEM) of pFedVEM and Uncertainty over 50 clients on CIFAR100. Each client has data of 1 out of 5 subclass (Hetero.) or all 5 subclasses (Homo.) per superclass.

Method	Random	Equal
pFedVEM	61.0 ± 0.4	60.7 ± 0.3
Model deviation	60.0 ± 0.4	60.4 ± 0.2
Mean Diff. (%)	$\mathbf{-1.0}$	-0.3

Table 3. Test accuracy (% $\pm$ SEM) of pFedVEM and Model deviation over 50 clients on CIFAR100. Clients have different local data sizes (Random) or the same amount of local data (Equal).

uncertainty term or the model deviation term and evaluate the resulting methods. Based on the results, we see the Uncertainty only method loses more performance when data are non-indentically distributed, indicating that model deviation is helpful when local models tend to deviate from each other (see Table 2). The Model deviation only method loses more performance when clients have different data sizes, indicating that uncertainty estimation is important under *data quantity disparity* (see Table 3). Nevertheless, both ablation methods perform worse than `pFedVEM` under different circumstances.

6. Related Works

Federated learning. Since the introduction of the first FL framework `FedAvg` [24] which optimizes the global model by weighted averaging client updates that come from local SGD, many methods [10, 18, 26, 33] have been proposed to improve it from different perspectives. `FedNova` [33] normalizes the client updates before aggregation to address the objective inconsistency induced by heterogeneous number of updates across clients. `FedProx` [18] adds a proximal term to the local objective to alleviate the problem of both systems and statistical heterogeneity. `Scaffold` [14] uses variance reduction to correct for the client-drift in local updates. `IFCA` [9] partitioned clients into clusters. `FedBE` [4] and `BNFed` [37] take the Bayesian inference perspective to make the model aggregation more effective or communication efficient. In particular, a Gaussian distribution is empirically proven to work well on fitting the local model distribution [4]. `FedPA` [2] shows there is an equivalence between the Bayesian inference of the posterior mode and the federated optimization under the uniform prior. [21] views the federated optimization as a hierarchical latent model and shows that `FedAvg` is a specific instance under this viewpoint. Both works indicate a Bayesian framework is general in modeling federated optimization problems.

Personalized federated learning. One fundamental challenge in FL is statistical heterogeneity of clients' data [13]. Many of the FL frameworks described above are developed to prevent the global model from diverging under this problem, while another way to cope with this issue is to learn a personalized model per client [1, 3, 6, 17, 22, 29–31]. `FedPer` [3] introduces a personalization layer (head model) for each client, while all clients share a base model to learn a representation. `FedLG` [20] and `FedRep` [5] refine such head-base architecture by optimizing the representation learning. Inspired by the Model-Agnostic Meta-Learning (MAML) framework, `PerFedAvg` [7] propose to learn a initial shared model such that clients can easily adapt to their local data with a few steps of SGD. `pFedHN` [29] use a hypernetwork to generate a set of personalized models. Several works attempt to find the clients with higher correlation and strengthen their collaboration [23, 30, 38, 39]. `pFedMe` [31] introduces bi-level optimization by using the Moreau envelope to regularize a client's loss function. Similar to our work, `pFedBayes` [40] uses Bayesian variational inference and also assumes a Gaussian distribution. However, they develop a framework with a fixed variance for all the local models' prior distribution and therefore do not obtain the personalized confidence value involving the model deviation and uncertainty as `pFedVEM` does.

7. Conclusion

In this paper, we addressed the problem of personalized federated learning under different types of heterogeneity, including label distribution skew as well as label concept drift and proposed a general framework for PFL via hierarchical Bayesian modeling. To optimize the global model, our method presents a principled way to aggregate the updated local models via variational expectation maximization. Our framework optimizes the local model using variational inference and the KL divergence acts as a regularizer to prevent the local model diverge too far away from the global model. Through extensive experiments under different heterogeneous settings, we show our proposed method `pFedVEM` yields consistently superior performance to main competing frameworks on a range of different datasets.

Acknowledgements

This research received funding from the Flemish Government (AI Research Program) and the Research Foundation - Flanders (FWO) through project number G0G2921N.

References

[1] Idan Achituve, Aviv Shamsian, Aviv Navon, Gal Chechik, and Ethan Fetaya. Personalized federated learning with gaussian processes. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021. 5, 8, 11

[2] Maruan Al-Shedivat, Jennifer Gillenwater, Eric Xing, and Afshin Rostamizadeh. Federated Learning via Posterior Averaging: A New Perspective and Practical Algorithms. *arXiv:2010.05273 [cs, stat]*, Jan. 2021. 8

[3] Manoj Ghuhan Arivazhagan, Vinay Aggarwal, Aaditya Kumar Singh, and Sunav Choudhary. Federated learning with personalization layers. *arXiv preprint arXiv:1912.00818*, 2019. 4, 5, 8

[4] Hong-You Chen and Wei-Lun Chao. FedBE: Making bayesian model ensemble applicable to federated learning. In *International Conference on Learning Representations*, 2021. 1, 8

[5] Liam Collins, Hamed Hassani, Aryan Mokhtari, and Sanjay Shakkottai. Exploiting shared representations for personalized federated learning. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139, pages 2089–2099, 18–24 Jul 2021. 2, 4, 5, 8

[6] Luca Corinzia, Ami Beuret, and Joachim M. Buhmann. Variational Federated Multi-Task Learning. *arXiv:1906.06268 [cs, stat]*, Feb. 2021. 8

[7] Alireza Fallah, Aryan Mokhtari, and Asuman Ozdaglar. Personalized federated learning with theoretical guarantees: A model-agnostic meta-learning approach. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 3557–3568, 2020. 2, 5, 8, 11

[8] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario March, and Victor Lempitsky. Domain-adversarial training of neural networks. *Journal of Machine Learning Research*, 17(59):1–35, 2016. 15

[9] Avishek Ghosh, Jichan Chung, Dong Yin, and Kannan Ramchandran. An efficient framework for clustered federated learning. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 19586–19597. Curran Associates, Inc., 2020. 5, 8, 13

[10] Filip Hanzely and Peter Richtárik. Federated Learning of a Mixture of Global and Local Models. *arXiv:2002.05516 [cs, math, stat]*, Feb. 2021. 8

[11] Patrick Heidorn. Shedding light on the dark data in the long tail of science. *Library Trends*, 57:280–299, 09 2008. 1

[12] J.J. Hull. A database for handwritten text recognition research. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(5):550–554, 1994. 15

[13] 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 Konečný, Aleksandra Korolova, Farinaz Koushanfar, Sanmi Koyejo, Tancrède Lepoint, Yang Liu, Prateek Mittal, Mehryar Mohri, Richard Nock, Ayfer Özgür, Rasmus Pagh, Mariana Raykova, Hang Qi, Daniel Ramage, Ramesh Raskar, 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, 2019. 1, 8

[14] Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank Reddi, Sebastian Stich, and Ananda Theertha Suresh. SCAFFOLD: Stochastic controlled averaging for federated learning. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 5132–5143. PMLR, 13–18 Jul 2020. 5, 8, 13

[15] Alex Krizhevsky. Learning multiple layers of features from tiny images. *University of Toronto*, 2012. 5, 11

[16] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. 5

[17] Qinbin Li, Bingsheng He, and Dawn Song. Model-contrastive federated learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021. 8

[18] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. In *Proceedings of Machine Learning and Systems*, volume 2, pages 429–450, 2020. 1, 5, 8

[19] Xiang Li, Kaixuan Huang, Wenhao Yang, Shusen Wang, and Zhihua Zhang. On the convergence of fedavg on non-iid data. In *International Conference on Learning Representations*, 2020. 1

[20] Paul Pu Liang, Terrance Liu, Liu Ziyin, Nicholas B. Allen, Randy P. Auerbach, David Brent, Ruslan Salakhutdinov, and Louis-Philippe Morency. Think locally, act globally: Federated learning with local and global representations, 2020. 8

[21] Christos Louizos, Matthias Reisser, Joseph Soriaga, and Max Welling. An Expectation-Maximization Perspective on Federated Learning. *arXiv:2111.10192 [cs, stat]*, Nov. 2021. 8

[22] Yishay Mansour, Mehryar Mohri, Jae Ro, and Ananda Theertha Suresh. Three Approaches for Personalization with Applications to Federated Learning. *arXiv:2002.10619 [cs, stat]*, July 2020. 8

[23] Othmane Marfoq, Giovanni Neglia, Richard Vidal, and Laetitia Kameni. Personalized federated learning through local memorization. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 15070–15092. PMLR, 17–23 Jul 2022. 2, 8

[24] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-Efficient Learning of Deep Networks from Decentralized Data. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, volume 54, pages 1273–1282, 2017. 1, 5, 8

[25] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y. Ng. Reading digits in natural images with unsupervised feature learning. In *NIPS Workshop on Deep Learning and Unsupervised Feature Learning 2011*, 2011. 15

[26] Sotirios Nikoloutsopoulos, Iordanis Koutsopoulos, and Michalis K. Titsias. Personalized Federated Learning with Exact Stochastic Gradient Descent. *arXiv:2202.09848 [cs]*, Feb. 2022. 8

[27] Sashank J. Reddi, Zachary Charles, Manzil Zaheer, Zachary Garrett, Keith Rush, Jakub Konečný, Sanjiv Kumar, and Hugh Brendan McMahan. Adaptive federated optimization. In *International Conference on Learning Representations*, 2021. 2

[28] Prasun Roy, Subhankar Ghosh, Saumik Bhattacharya, and Umapada Pal. Effects of degradations on deep neural network architectures. *arXiv preprint arXiv:1807.10108*, 2018. 15

[29] Aviv Shamsian, Aviv Navon, Ethan Fetaya, and Gal Chechik. Personalized Federated Learning using Hypernetworks. *arXiv:2103.04628 [cs]*, Mar. 2021. 8, 11

[30] Virginia Smith, Chao-Kai Chiang, Maziar Sanjabi, and Ameet S Talwalkar. Federated multi-task learning. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30, 2017. 2, 8

[31] Canh T. Dinh, Nguyen Tran, and Josh Nguyen. Personalized federated learning with moreau envelopes. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 21394–21405, 2020. 2, 5, 8, 11, 13

[32] Anne E. Thessen, Hong Cui, and Dmitry Mozzherin. Applications of natural language processing in biodiversity science. *Advances in Bioinformatics*, 2012, May 2012. 1

[33] Jianyu Wang, Qinghua Liu, Hao Liang, Gauri Joshi, and H. Vincent Poor. Tackling the objective inconsistency problem in heterogeneous federated optimization. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 7611–7623, 2020. 1, 8

[34] Florian Wenzel, Kevin Roth, Bastiaan Veeling, Jakub Swiatkowski, Linh Tran, Stephan Mandt, Jasper Snoek, Tim Salimans, Rodolphe Jenatton, and Sebastian Nowozin. How good is the Bayes posterior in deep neural networks really? In *Proceedings of the 37th International Conference on Machine Learning*, volume 119, pages 10248–10259, 13–18 Jul 2020. 3

[35] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms, 2017. 5, 11

[36] Jianxiong Xiao, James Hays, Krista A. Ehinger, Aude Oliva, and Antonio Torralba. SUN database: Large-scale scene recognition from abbey to zoo. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2010. 5, 11

[37] 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, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 7252–7261. PMLR, 09–15 Jun 2019. 8, 11

[38] Jie Zhang, Song Guo, Xiaosong Ma, Haozhao Wang, Wenchao Xu, and Feijie Wu. Parameterized knowledge transfer for personalized federated learning. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 10092–10104. Curran Associates, Inc., 2021. 8

[39] Michael Zhang, Karan Sapra, Sanja Fidler, Serena Yeung, and Jose M. Alvarez. Personalized federated learning with first order model optimization. In *International Conference on Learning Representations*, 2021. 8

[40] Xu Zhang, Yinchuan Li, Wenpeng Li, Kaiyang Guo, and Yunfeng Shao. Personalized federated learning via variational Bayesian inference. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162, pages 26293–26310, 17–23 Jul 2022. 2, 4, 5, 7, 8, 11