FedCR: Personalized Federated Learning Based on Across-Client Common Representation with Conditional Mutual Information Regularization

Hao Zhang[1] **Chenglin Li**[1] **Wenrui Dai**[1] **Junni Zou**[1] **Hongkai Xiong**[1]

Abstract

In personalized federated learning (PFL), multiple clients train customized models to fulfill their personal objectives, which, however, are prone to overfitting to local data due to the heterogeneity and scarcity of local data. To address this, we propose from the information-theoretic perspective a personalized federated learning framework based on the common representation learned across clients, named FedCR. Specifically, we introduce to the local client update a regularizer that aims at minimizing the discrepancy between local and global conditional mutual information (CMI), such that clients are encouraged to learn and exploit the common representation. Upon this, each client learns individually a customized predictor (head), while the extractor (body) remains to be aggregated by the server. Our CMI regularizer leads to a theoretically sound alignment between the local and global stochastic feature distributions in terms of their Kullback-Leibler (KL) divergence. More importantly, by modeling the global joint feature distribution as a product of multiple local feature distributions, clients can efficiently extract diverse information from the global data but without need of the raw data from other clients. We further show that noise injection via feature alignment and ensemble of local predictors in FedCR would help enhance its generalization capability. Experiments on benchmark datasets demonstrate a consistent performance gain and better generalization behavior of FedCR.

1. Introduction

Federated learning (FL) (McMahan et al., 2017) has emerged as a new distributed learning framework in which a group of clients, under coordination of the central server, train collaboratively a single global machine learning model without sharing or exchanging the private data. These private data generated and collected at clients are usually non-independent and identical distributed (non-iid). In such a setting, when clients wish to fulfill their personal objectives and tasks, direct share of the universally global model may lead to a poor generalization performance at local clients because of the mismatch between global and local data distributions. To address this, personalized federated learning (PFL) (Kairouz et al., 2021) has been proposed to enable each client to train an individual model fitted to the local data distribution. Nevertheless, in addition to the statistical heterogeneity, source data at local clients are usually scarce and limited, where some labels may even have few data points. Thus, conventional local training at clients towards the local optimum still easily leads to a model overfitting.

Recent advance in multi-task learning and representation learning has already demonstrated that decoupling the learning procedure into representation and prediction would boost the performance (Yu et al., 2020; Kang et al., 2019). Motivated by this, a feasible solution is to decouple the PFL objectives into global representation learning and local prediction. For the global representation learning, body of the model (i.e., feature extractor) extracts the low-dimensional common features across clients to avoid overfitting. While in the local prediction, upon the learned common representation, head of the model (i.e., predictor) makes a task-oriented and personalized decision. Along this direction, many works (Collins et al., 2021; Arivazhagan et al., 2019; Oh et al., 2021) have attempted to simply average the body of model parameters from heterogeneous clients to exploit the common representation. Despite their progress, the shared feature extraction at the model-parameter level is naturally insufficient for capturing effective common features from heterogeneous data belonging to multiple classes.

Aiming to learn the common representation directly from the feature level, a very recent work, FedPAC (Xu et al., 2023), proposes a class-wise feature alignment, which how-

[1] The authors are all with the School of Electronic Information and Electrical Engineering, Shanghai Jiao Tong University, Shanghai 200240, China. Correspondence to: Chenglin Li <LCL1985@sjtu.edu.cn>, Junni Zou <zoujunni@sjtu.edu.cn>.

Proceedings of the 40^{th} International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023. Copyright 2023 by the author(s).

ever can only be used for the label distribution shift scenario that requires the same number of classes assigned across clients. While there is still an alternative direction of efficient representation learning unexplored for PFL, i.e., from the information-theoretic perspective. In centralized learning, for example, the information bottleneck (Shwartz-Ziv & Tishby, 2017) imposes mutual information (MI) between the input data and learned features as constraints on the features, to regularize the amount of information contained and thus remove redundancy. However, direct incorporation of existing MI-based methods into the local client update cannot address the distinct challenges of PFL, which may still suffer a significant local overfitting due to data scarcity.

To cope with the data heterogeneity and scarcity challenge in PFL settings from the information-theoretic perspective, we propose in this paper a personalized federated learning framework, named FedCR, by effectively exploiting the common representation learned across clients. FedCR is based on the insight that conditional mutual information (CMI) can help compress superfluous information of representation while at the same time providing the prediction (Fischer, 2020). Different from FedSR (Nguyen et al., 2022) that simply minimizes the local CMI as a regularizer for local training, we consider minimizing the discrepancy between local CMI and global CMI instead, to learn shared and invariant features across clients. Furthermore, our CMI regularizer leads to an alignment between local and global feature distributions by regularizing their Kullback-Leibler (KL) divergence. Specifically, we model the global feature distribution in the KL divergence as a product-of-experts (PoE) over the marginal distributions of clients, rather than through an additional generative network (Zhu et al., 2021) which may be unstable and expensive. Thus, client models can be trained easily and stably in practice with negligibly additional computation complexity. With a deeper look, we show that our CMI regularizer implicitly transfers the global information into clients through noise injection, which has been proved empirically to prevent overfitting. We also demonstrate that the uncertainty in the proposed stochastic features of FedCR can further improve the client's classification calibration with the ensemble of local predictors. Our main contributions can be summarized as follows.

- We propose an information-theoretic method for PFL to transfer global information to local clients by introducing the CMI regularizer, which leads to a theoretically sound alignment between global and local feature distributions.
- We provide key insights to understanding our FedCR. The noise injection via feature alignment and ensemble of local predictors in FedCR enhance generalization. We also show a theoretical connection from the shared common representation learned by FedCR to an improved generalization bound.
- We evaluate empirically the effectiveness of our FedCR on a wide range of benchmark datasets, showing that FedCR outperforms various existing model architectures that aim to handle data distribution shift and model overfitting.

2. Related Work

Personalized federated learning (PFL). There is a large body of existing works in PFL. Recent literature has utilized a variety of techniques to address the challenges of PFL and train customized models, including local fine-tuning (Sim et al., 2019), hyper-networks (Shamsian et al., 2021), meta-learning (Fallah et al., 2020), model-parameter regularization (Ditto) (Li et al., 2021), etc. One promising direction is to decouple the PFL model into extractor (body of the model) and predictor (head of the model), as inspired by multi-task learning and representation learning. LG-FedAvg (Liang et al., 2020) learns the entire local model at each client and only aggregates the predictors at server. Conversely, FedPer (Arivazhagan et al., 2019) still trains the entire local client model but aggregates the feature extractors at the server. Furthermore, FedRep (Collins et al., 2021) learns the entire model sequentially at local training stage and only aggregates the extractors, while during local update, each client first learns the predictor with the aggregated extractor. FedBABU (Oh et al., 2021) only updates the extractor with the randomly initialized and never updated predictor during local training, and aggregates only the extractors. Moreover, the very recent FedPAC (Xu et al., 2023) attempts to align the local and global feature embeddings directly to exploit shared representation from the feature level, which, however, can only apply to label distribution shift scenarios. From an information-theoretic perspective, we propose a different conditional mutual information-based method to learn the shared and invariant representations across clients.

Mutual information (MI)-based representation learning. Supervised representation learning has long been an important topic in machine learning, while the mutual information (MI)-based approaches have been widely applied within this area. Its key idea is to quantify the information contained in the representation by MI, which measures the dependency between two random variables. A typical approach is the information bottleneck (Shwartz-Ziv & Tishby, 2017; Alemi et al., 2016), which maximizes the MI between representations and labels to retain information related to prediction while minimizing the MI between input data and representations to discard irrelevant information. Furthermore, conditional entropy bottleneck (Fischer, 2020) directly uses one CMI term to compress redundant information of representations while performing prediction, instead of the two individual MI terms in the information bottleneck. Indeed, some works have already attempted to address the challenges of FL from the perspective of information theory. For example, the pioneer work (Adilova et al., 2019) first empirically explores the impact of aggregation frequency at server on the information flow. MIFL (Uddin et al., 2020) provides a novel MI-driven federated optimization by considering MI between local and global models. FedSR (Nguyen et al., 2022) learns a simple representation by minimizing the local CMI between local representations and input data. Different from them, our method considers learning a common representation to help avoid model overfitting to local data.

3. Problem Statement

In PFL, we aim to learn the personalized local model w_i for each client $i \in \mathcal{M}$ via the following optimization problem:

$$\min_{\{w_1,\cdots,w_m\}} \frac{1}{m}\sum_{i=1}^{m}\left[f_i\left(w_i\right) + \mathcal{R}\left(\Omega, w_1, \cdots, w_m\right)\right], \quad (1)$$

where $f_i\left(w_i\right)$ is the local objective function of the i-th client associated with data distribution $p(x_i, y_i)$ and a dataset including N_i data points $\{(x_i^{(n)}, y_i^{(n)})\}_{n=1}^{N_i}$ that belong to classes $\mathcal{C}_i$, i.e., $y_i^{(n)} \in \mathcal{C}_i$ represents the corresponding label of input data sample $x_i^{(n)}$. We further denote the global data from the entire set of all the m clients by a set of random variables $x = (x_1, ..., x_m)$, and the global label class set across clients by $\mathcal{C} = \{\mathcal{C}_1, ..., \mathcal{C}_m\}$. Moreover, $\mathcal{R}$ is a regularizer imposed to prevent overfitting due to local data heterogeneity and scarcity, where Ω represents a certain type of shared information introduced to relate clients.

Following the framework of representation learning, we decouple the local model into the feature extractor (body) and predictor (head), i.e. $w_i = [w_i^f; w_i^p]$. We first learn a low-dimensional representation z of the raw data x_i with a distribution $p(z|x_i; w_i^f)$ parameterized by w_i^f. Upon extraction of z, we then learn a predictor that predicts y_i given z with the predictive distribution $\hat{p}(y_i|z; w_i^p)$ parameterized by w_i^p. Hence, for both the regression or classification tasks where the loss function is usually chosen as the negative log predictive, the local objective of client i can be written as:

$$f_i(w_i) = \mathbb{E}_{p(x_i,y_i)}\left[-\log \mathbb{E}_{p(z|x_i)}[\hat{p}(y_i|z)]\right], \quad (2)$$

where we omit w_i^f and w_i^p for notation simplicity.Refer to Appendix A.3.1 for detailed derivation of Eq. (2). Note that the feature extractor in practice can be either deterministic or stochastic mappings. Without loss of generality, we consider a stochastic representation mapping with the form $p(z|x_i) = \mathcal{N}(z|\mu(x_i), \Sigma(x_i))$, where $\mathcal{N}$ is the normal distribution and w_i^f outputs both the mean μ and covariance matrix Σ of z. This can also be thought as a generalization of deterministic representation, as it reduces to the deterministic case when $\Sigma(x_i) \to 0$. By explicitly modeling the representation distribution, we can easily use the MI or KL divergence to constrain this known representation distribution.

Table 1. Summary of main notations.

T, t	number, index of communication rounds
K, k	number, index of local update steps
x_i, y_i	random variables denoting client i's raw data, label
w_i, w	client i's model, aggregated server model
$\mathcal{M}, m$	set of all clients with cardinality m
$\mathcal{P}_t, p$	set of sampled active clients with cardinality p

Obviously, the representation z here contains only the information of local data x_i, which may easily lead to model overfitting and thus a performance degradation. A natural solution is to add regularization terms to this local feature. For example, FedSR (Nguyen et al., 2022) considers minimizing the local CMI term $I_i(z, x_i|y_i)$ and L2-norm $\|z\|_2^2$ to constrain the representation z. By doing so, FedSR learns a simple representation of the data for a better generalization. This CMI regularizer in FedSR, though improving generalization performance, is still sub-optimal without explicitly injecting any global information of the raw data into local features. It thus remains an open problem in FPL: how can local clients utilize global information at the feature level.

4. Proposed FedCR

4.1. Global and Local CMI Constraint

To tackle this challenge, our basic idea is to learn a representation of the local data at each client, under the guidance of global information captured from the entire dataset across clients. Specifically, we impose to the local client update a constraint on the difference between each client's local CMI $I_i(z; x_i|y_i)$ and the global CMI $I(z; x|y_i)$, which quantifies the relevance between the local (or global) features and input data x_i (or x) given a specific label y_i, respectively. By incorporating this information-theoretic constraint, the original optimization problem in Eq. (1) is reformulated as:

$$\begin{aligned} \min_{\{w_1,\cdots,w_m\}} \quad & \frac{1}{m}\sum_{i=1}^{m} f_i\left(w_i\right) \\ \text{s.t.} \quad & \|I_i(z; x_i|y_i) - I(z; x|y_i)\| < I_c, \forall i \in \mathcal{M}, \end{aligned} \quad (3)$$

where the global data x is represented as a series of random variables $(x_1, ..., x_m)$, and I_c constrains the difference between the local and global CMI. Intuitively, this constraint specifies that for a given label y_i, the features learned from local input data x_i is also aligned with the common features captured from the global input data x within the same label class $c = y_i$. In the special case when $I_c = 0$, all the clients will be enforced to learn a stochastic mapping from input data to a latent space, in which the representations are consistent and invariant across clients. Compared to FedSR (Nguyen et al., 2022) that regularizes only the local CMI $I_i(z; x_i|y_i)$, we allow the extraction of local features to incorporate more diverse information from the global data, thus alleviating the local feature shift via learning an inter-client invariant representation. By further introducing a Lagrange multiplier $\beta \geq 0$, the constrained problem in Eq. (3) is converted to:

$$\min_{\{w_1,\cdots,w_m\}} \frac{1}{m}\sum_{i=1}^{m} f_i\left(w_i\right) + \beta \| I_i(z;x_i|y_i) - I(z;x|y_i) \|, \tag{4}$$

where the second term is our proposed CMI regularizer, corresponding to $\mathcal{R}$ in Eq (1), which is practically intractable. We thus propose an equivalent way to calculate it.

Lemma 4.1. *Let $y \rightarrow x \rightarrow z$ be a Markov chain, for $x_i \subseteq x$ and within the same label classes of client i, we have:*

$$\begin{aligned} &I(z;x|y_i) - I_i(z;x_i|y_i) \\ =&\mathbb{E}_{p(x_i,y_i)}\mathbb{E}_{p(x|x_i,y_i)}\left[\mathrm{KL}[p(z|x)||p(z|x_i)]\right], \end{aligned} \tag{5}$$

where $\mathrm{KL}[p(z|x)||p(z|x_i)]$ denotes the KL divergence between $p(z|x)$ and $p(z|x_i)$ given the label y_i, i.e. a class-wise feature alignment. Note that $p(x|x_i,y_i)$ indicates that the global data can be determined only given the clients actively participating in the training at a communication round, due to the partial client participation setting in PFL.

Proof. See Appendix A.3.2 for the detailed proof. □

Lemma 4.1 shows that the CMI regularizer in Eq. (3) actually constrains the KL divergence between $p(z|x)$ and $p(z|x_i)$, which aligns the stochastic representation of joint and single posterior within the label classes for each client. Since the KL divergence is always non-negative, this also implies $I_i(z;x_i|y_i) \leq I(z;x|y_i)$. Note that FedPAC (Xu et al., 2023) also performs a strict alignment of the deterministic features with an L2-norm regularizer, which can be considered as our special case. While noise and uncertainty incorporated in stochastic features can further improve generalization, as will be discussed later.

4.2. Estimation of Global Common Representation

One question then becomes outstanding: how can the server get the global common representation $p(z|x)$ for each class without the need of raw data at clients. Since clients do not wish to disclose their raw data, we introduce the following product-of-experts (PoE) (Hinton, 2002), which factorizes the joint posterior into a product of individual posteriors.

Lemma 4.2. *For marginal posteriors $p(z|x_i)(\forall i \in \mathcal{M})$, the joint posterior can be approximated by (Hinton, 2002):*

$$p(z|x) = p\left(z|x_1,\ldots,x_m\right) \propto \tau \cdot p(z)\prod_{i=1}^{m} p\left(z|x_i\right), \tag{6}$$

where $\tau \triangleq \frac{\prod_{i=1}^{M} p(x_i)}{p(x_1,\ldots,x_m)}$ represents the degree of independence between clients, and $p(z)$ is a prior distribution, usually the spherical Gaussian.

Proof. See Appendix A.3.3 for the detailed proof. □

With PoE, we can get a simple analytical solution when $p(z|x_i), \forall i \in \mathcal{M}$ is diagonal Gaussian. Specifically, a product of Gaussian experts is itself with the mean

$$\mu = \left(\mu_0\Sigma_0^{-1} + \sum_{i\in\mathcal{M}}\mu_i\Sigma_i^{-1}\right)\left(\Sigma_0^{-1} + \sum_{i\in\mathcal{M}}\Sigma_i^{-1}\right)^{-1}$$

and covariance $\Sigma = \left(\Sigma_0^{-1} + \sum_{i\in\mathcal{M}}\Sigma_i^{-1}\right)^{-1}$, where $p(z|x_i) \sim \mathcal{N}(\mu_i,\Sigma_i)$ and $p(z) \sim \mathcal{N}(\mu_0,\Sigma_0)$.

In fact, there are also other alternatives to model this joint posterior, including data-free generative model (Zhu et al., 2021), or mixture-of-experts (MoE), i.e., sum of Gaussian experts. However, the former one needs to retrain a generative model at server, which can be very unstable and expensive in practice. While for the latter, MoE is usually inefficient in high-dimensional spaces (Hinton, 2002), since the posterior distribution produced by MoE cannot be sharper than the individual experts (i.e., the marginal posteriors). In contrast, PoE (i.e., the joint posterior) can generate much sharper distributions than the single experts, which helps the global posterior contain not only common but complementary information across clients.

4.3. Implementation of FedCR

We then design and elaborate our FedCR with a summary shown in Algorithm 1 and an overview illustrated in Fig. 1.

Local training. For each client i that actively participates in the training at communication round t, it first receives the global representation $p(z^c|x) = \mathcal{N}(z|\mu_t^c,\Sigma_t^c)$ for each class $c \in \mathcal{C}$, and the global aggregated average feature extractor w^f, which will be then combined with the local predictor w_i^p as the initialized local model in Line 5. We then train its local model $w_{i,0} = [w^f, w_i^p]$ including the extractor and predictor, by minimizing the local objective function:

$$\begin{aligned} \mathcal{L}_i =& f_i\left(w_i\right) + \beta\|I_i(z;x_i|y_i) - I(z;x|y_i)\| \\ =&\mathbb{E}_{p(x_i,y_i)}\left[-\log\mathbb{E}_{p(z|x_i)}[\hat{p}(y_i|z)]\right] \\ &+ \beta\mathbb{E}_{p(x_i,y_i)}\mathbb{E}_{p(x|x_i,y_i)}\left[\mathrm{KL}[p(z|x)||p(z|x_i)]\right] \\ \approx&\frac{1}{N_i}\sum_{n=1}^{N_i}\left[-\log\mathbb{E}_{p(z|x_i^{(n)})}[\hat{p}(y_i^{(n)}|z)]\right] \\ &+ \beta\frac{1}{N_i}\sum_{n=1}^{N_i}\left[\mathrm{KL}[p(z^{c=y_i^{(n)}}|x)||p(z|x_i^{(n)})]\right], \end{aligned} \tag{7}$$

where we can use the re-parameterization trick (Kingma & Welling, 2013) to perform backpropagation in the stochastic network w_i^f. Specifically, for the diagonal Gaussian distribution $p(z|x_i^{(n)}) = \mathcal{N}(z|\mu(x_i^{(n)}),\Sigma(x_i^{(n)}))$, we can execute $z = \epsilon\cdot\Sigma(x_i^{(n)}) + \mu(x_i^{(n)})$ in practice, where $\epsilon \sim \mathcal{N}(0,1)$.

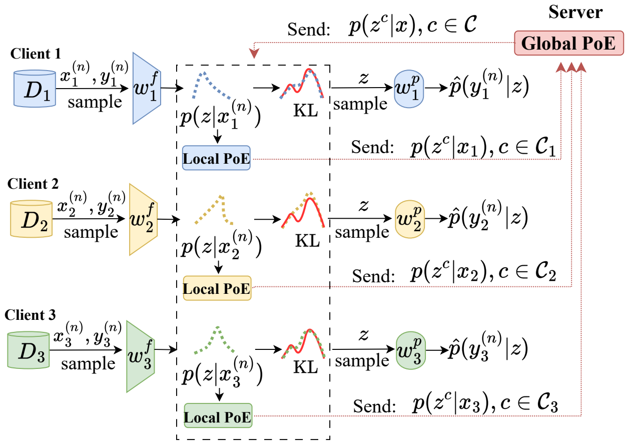


Figure 1. Overview of FedCR: during local training, the feature of sample $x_i^{(n)}$ is aligned with the global common representation of the same category $c = y_i^{(n)}$, where we assume that the samples at the three clients have the same label for ease of illustration. While the latest feature of each sample is retained, which is later multiplied through local PoE at the end of local update.

Further note that when setting $\beta = 0$ and $\Sigma(x_i^{(n)}) = 0$, this stochastic network w_i^f becomes deterministic. In this case, our method will degenerate to FedPer (Arivazhagan et al., 2019).

To reduce communication overhead, we do not need to directly upload the feature distribution of each sample point, but perform the following class-wise local PoE in advance:

$$p\left(z^c \mid x_i\right) = \prod_{n=1}^{N_i} p\left(z \mid x_i^{(n)}\right)^{\mathbf{1}\left(y_i^{(n)}=c\right)}, \forall c \in \mathcal{C}_i, \quad (8)$$

where $\tau = 1$ due to random sampling and the prior $p(z)$ is considered at server. To reduce the computation complexity, on the other hand, we directly use the latest $p(z|x_i^{(n)})$ generated when updating the network w_i^f, instead of an additional forward propagation of all input data to obtain the stochastic features after local training as used in FedPAC (Xu et al., 2023). This approach can be thought of as a queue where, upon sampling and forward propagation, the features retained from the previous update step are discarded, while only the features generated by the current update step are retained for that data sample. As a result, $p(z|x_i^{(n)}; w_{i,k}^f)$ may come from a different local iteration $k \in [K]$ when using SGD. Such a manner has important advantages of avoiding privacy leakage to a large extent, since the server can hardly infer meaningful information of raw data based on the local PoE $p(z^c|x_i)$ and updated local model $w_{i,K}^f$.

Global aggregation. At the server, we first aggregate the feature extractors to get w^f. Then, before aggregating the global representation, we need to consider that due to the data heterogeneity/scarcity at clients and partial client participation property in PFL, it is very likely that the entire data at all the clients that currently participate in this communication round may not contain a certain class. In this case, we only need to use the global features of the previous communication round for that class (Line 16). Next, we aggregate the global feature representation for each class belonging to these active clients, by:

$$p(z^c|x) = \prod_{i=1}^{p} \tau p(z) p(z^c|x_i), \forall c \in \mathcal{C}, \quad (9)$$

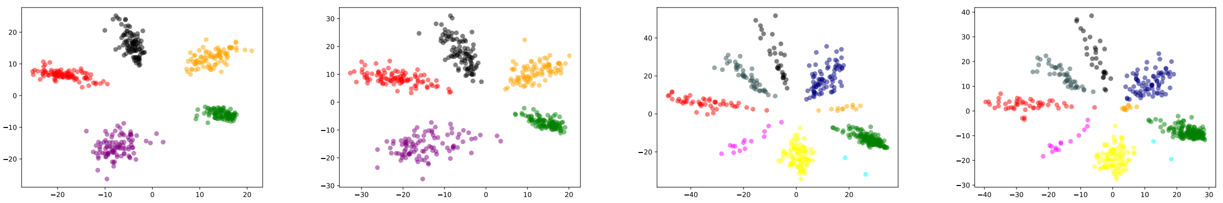
where hyper-parameter $\tau = 1$ if the clients participate in the training uniformly and randomly, and we set $p(z) \sim \mathcal{N}(0, 1)$. The aggregated feature extractor and global representation are then broadcast to clients, helping their local training via the common and invariant global representation.

4.4. Extension to Model Non-Aggregation Scenarios

Up to now, our FedCR has practically realized the global common representation shared across clients. It in essence can be applied to another more challenging PFL scenario, where each client designs their own unique model to meet distinct specifications. In this case, the server cannot directly average model parameters due to model heterogeneity. Moreover, clients may even be unable to share their models due to intellectual property, privacy or communication concerns. For such a more realistic and complex setting, FedCR is still effective by only exchanging the feature representation. In this case, the size of model of each client can be arbitrary, while we only require that the normal distributions output by feature extractors have the same dimension. Note that this model non-aggregation paradigm has an additional advantage of less vulnerable to privacy leakage. Empirical evaluations in Appendix A.2.1 show that local clients benefit from the global common representation, even without sharing the model. Moreover FedCR can generally outperform the "Local-Training" scheme in which clients separately train their own models without sharing anything.

4.5. Limitations

We then discuss possible limitations of our FedCR in terms of computation, communication, and privacy. First, we acknowledge that FedCR requires additional computation and transmission of local feature distributions. However, as stated in Section 4.3, we directly utilize the feature distribution generated when updating model parameters. Thus, there is almost no additional computational overhead. Meanwhile, a Gaussian distribution with a diagonal covariance matrix (i.e., we only transfer the diagonal elements of covariance) is much smaller than the model size. Finally, for privacy concerns, the server is hard to infer meaningful raw data information, since the low-dimensional features transmitted are the product of sample features generated by dif-



(a) β=0.005 (non-iid_1) (b) β=0 (non-iid_1) (c) β=0.005 (non-iid_2) (d) β=0 (non-iid_2)

Figure 2. Visualizing embeddings of about 200 local test images (MNIST) at the randomly picked client in two dimensional space. The horizontal and vertical axes are the mean of the two-dimensional Gaussian feature, and the images are colored based on their true class labels (non-iid_1: each client has only 5 classes; non-iid_2: the randomly picked client has 9 classes and the data on each class is not uniformly distributed). For non-iid_1 scenario, there is still a clear class-wise feature boundary even without the proposed CMI regularizer ($\beta = 0$), while it is slightly less concentrated than the counterpart that leverages CMI. However, for the more complex non-iid_2 scenario, without CMI regularization, the feature distribution represented in orange overlaps with other classes and becomes indistinguishable.

Algorithm 1 FedCR for personalized federated learning.

```
server: $\mu_0^c = \mathbf{0}$, $\Sigma_0^c = \mathbf{1}$; clients: learning rate $\eta$
for each round $t = 1, 2, 3, ..$ do
  sample clients $\mathcal{P}_t \subseteq \mathcal{M}$
  // local training:
  for each client $i \in \mathcal{P}_t$ in parallel do
    receive $\mu_t^c, \Sigma_t^c, w^f$ and initialize $w_{i,0} = [w^f, w_i^p]$
    for each local step $k = 1, 2, \dots, K$ do
      update the whole network $w_i$ by SGD as (7):
      $w_{i,k} = w_{i,k-1} - \eta \nabla_{w_{i,k-1}} \mathcal{L}_i$
    end for
    update the local class-wise feature $\mu_i^c, \Sigma_i^c$ as (8)
    set $w_i^p = w_{i,K}^p$; send $w_{i,K}^f$, $\mu_i^c, \Sigma_i^c$ to server
  end for
  // global aggregation at server:
  $w^f = \frac{1}{|\mathcal{P}_t|} \sum_{i \in \mathcal{P}_t} w_i^{f,K}$
  if $c \in \mathcal{C}$ and $c \notin \mathcal{C}_i (\forall i \in \mathcal{P}_t)$ then
    $\mu_t^c = \mu_{t-1}^c$; $\Sigma_t^c = \Sigma_{t-1}^c$
  end if
  update the global class-wise feature $\mu_t^c, \Sigma_t^c$ as (9)
  send $w^f, \mu_t^c, \Sigma_t^c$ to clients
end for
```

ferent local models. We can also apply privacy-preserving techniques to feature distribution, such as homomorphic encryption (Rivest et al., 1978). In fact, there is a trade-off between security, efficiency, and practicality in PFL. Our method mainly focuses on improving efficiency and practicality, which is also the main motivation for clients to participate in PFL instead of single-machine training, where they would share more information for better collaboration.

5. Discussion and Analysis

5.1. Noise Injection and Uncertainty for Generalization

FedCR is stochastic at the feature representation level by modeling the mean and covariance of local features, which can also be viewed as injecting noise into the features. Ample evidences (e.g., data augmentation (Zhang et al., 2021) and dropout (Srivastava et al., 2014)) have suggested that noise injection can prevent overfitting and enhance generalization. This local feature distribution is then aligned with the global feature distribution by KL divergence. In this sense, FedCR transfers global information into clients through the injection of noise. Such a noise injection by distribution alignment in FedCR increases diversity in the intra-class representation while maintaining sharp inter-class representation boundaries, as shown in Fig. 2. Under two non-iid settings on MNIST dataset with 100 clients and 10% participation rate (referring to Section 6 for details), we set the feature distribution to a 2-dimensional Gaussian (which may degrades performance), and randomly select a client after 20 communication rounds. At this selected client, we continue to train locally for 10 epochs, and then output the feature distribution on the test set of this client. Fig. 2 shows that with feature alignment, the feature boundaries between classes are more distinguishable, which leads the local predictors to being more robust when inferring data.

Moreover, FedCR can improve the ability of local classification calibration and generalization by quantifying the uncertainty (Alemi et al., 2018). Proper uncertainty is of crucial importance to avoid an overconfident, offensive or incorrect prediction. There is also a rich literature in improving the quantification of uncertainty of neural networks, such as temperature scaling (Guo et al., 2017). By explicitly modeling the representation distribution z, we then use Monte Carlo samples of z to predict labels (i.e., $p(y_i|x_i) = \frac{1}{S}\sum_{s=1}^{S} \hat{p}(y_i|z^s)$ for $z^s \sim p(z|x_i)$, where S is the number of Monte Carlo samples), which implicitly models the distribution of predictive labels. This stochasticity in feature representation induces an decent ensemble of local predictors, while the ensemble technique has been shown to yield well-calibrated uncertainty estimates (Lakshminarayanan et al., 2017).

Table 2. Averaged test accuracy (%) of local models at 100 clients with 10% participation rate (averaged over 3 random seeds).

Method	EMNIST-L		FMNIST		CIFAR10		CIFAR100	
	non-iid_1	non-iid_2	non-iid_1	non-iid_2	non-iid_1	non-iid_2	non-iid_1	non-iid_2
FedAvg (McMahan et al., 2017)	95.89	94.57	88.15	88.63	76.83	76.34	32.08	32.19
FedAvg-FT (Sim et al., 2019)	96.32	95.78	92.76	91.70	83.65	83.20	54.42	41.46
FedPer (Arivazhagan et al., 2019)	96.13	94.67	88.91	86.35	73.73	69.36	39.5	25.48
LG-FedAvg (Liang et al., 2020)	88.44	85.59	86.86	85.44	61.37	61.53	44.48	28.23
FedRep (Collins et al., 2021)	96.19	94.66	89.78	88.90	78.10	71.98	44.62	24.70
FedBABU (Oh et al., 2021)	96.22	94.88	89.21	89.18	73.60	69.37	44.00	26.15
Ditto (Li et al., 2021)	96.60	96.30	92.03	91.75	83.25	82.81	58.40	41.85
FedSR-FT (Nguyen et al., 2022)	86.22	84.71	85.55	85.45	61.47	60.96	40.82	24.56
FedPAC (Xu et al., 2023)	96.97	96.43	93.45	92.15	**85.03**	84.07	58.65	43.25
FedCR	**97.47**	**96.98**	**93.78**	**93.00**	84.74	**84.26**	**62.96**	**44.06**

5.2. Common Representation to Improve Generalization

Furthermore, we can also give an improved averaged generalization bound among clients from the viewpoint of domain adaptation (Ben-David et al., 2006; Zhu et al., 2021).

Theorem 5.1. *Let $\mathcal{H}$ ($h \in \mathcal{H} : \mathcal{Z} \to \mathcal{Y}$) be a hypothesis space of VC-dimension d and $\mathcal{X} \to \mathcal{Z}$ be a feature representation function shared across clients. Given a global meta-distribution $\mathcal{D}$ from which the active clients with a local distribution $\mathcal{D}_i$ are drawn, let $\hat{\epsilon}_{\mathcal{D}_i}(h_i)$ denote the empirical risk of hypothesis h_i on $\mathcal{D}_i$. Similarly, let $\epsilon_{\mathcal{D}}(h_i)$ denote the expected risk of hypothesis h_i on $\mathcal{D}$. Let $\tilde{\mathcal{D}}_i, \tilde{\mathcal{D}}_i'$ be the induced distribution of $\mathcal{D}_i$ by samples of size n for FedPer and FedCR, and $\tilde{\mathcal{D}}$ be the induced distribution of $\mathcal{D}$. Then with probability at least $1 - \delta$, we have*

$$\frac{1}{m}\sum_{i\in\mathcal{M}}\epsilon_{\mathcal{D}}(h_i) \le \frac{1}{m}\sum_{i\in\mathcal{M}}\hat{\epsilon}_{\mathcal{D}_i}(h_i) + \frac{1}{m}\sum_{i\in\mathcal{M}} d_{\mathcal{H}\Delta\mathcal{H}}\left(\tilde{\mathcal{D}}_i', \tilde{\mathcal{D}}\right) + \sqrt{\frac{4}{n}\left(d\log\frac{2en}{d} + \log\frac{4m}{\delta}\right)} + \frac{1}{m}\sum_{i\in\mathcal{M}}\lambda_i \qquad (10)$$

where e is the base of the natural logarithm, $\lambda_i = \min_h\left(\hat{\epsilon}_{\mathcal{D}_i}(h_i) + \epsilon_{\mathcal{D}}(h_i)\right)$ denotes the combined risk of the ideal hypothesis, and $d_{\mathcal{H}\Delta\mathcal{H}}(\tilde{\mathcal{D}}_i', \tilde{\mathcal{D}})$ denotes the distance of distribution with $d_{\mathcal{H}\Delta\mathcal{H}}(\tilde{\mathcal{D}}_i', \tilde{\mathcal{D}}) < d_{\mathcal{H}\Delta\mathcal{H}}(\tilde{\mathcal{D}}_i, \tilde{\mathcal{D}})$.

Proof. See Appendix A.3.4 for the detailed proof. □

Remark 5.2. By aligning the representation distribution, discrepancy between the global distribution and the local distribution of individual clients in latent space is much smaller than that of FedPer (Arivazhagan et al., 2019), which facilitates clients to learn shared and invariant representation. Such a feature alignment enables local predictors to generalize better to inferring data outside the local distribution, even with an extremely limited amount of local data at clients.

6. Experiment

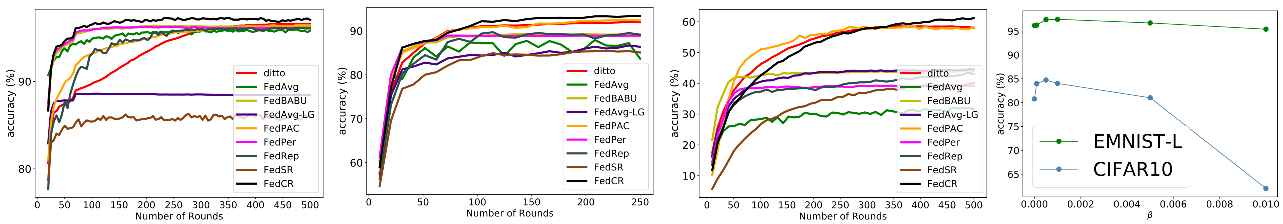
6.1. Experimental Setup

Datasets and models. We evaluate our FedCR on four benchmark datasets, EMNIST-L, Fashion-MNIST (FMNIST), CIFAR10 and CIFAR100 with the non-iid train/test splits. Specifically, we use two non-iid settings. **Non-iid_1:** each client is randomly assigned five classes for EMNIST-L, FMNIST, and CIFAR10 (while fifteen classes per client for CIFAR100) with the same amount of data in each class; **non-iid_2:** each client has an uncertain number of classes, and the data within each class varies widely by setting client sample labels according to the Dirichlet distribution. In the non-iid_2 setting, each client has about four classes that consume 80% of data, and misses one or two classes with Dirichlet parameter 0.5 for EMNIST-L, FMNIST and CIFAR10 (and Dirichlet parameter 0.3 for CIFAR100). All data is split into 70% training set and 30% test set. The test set and the training set have the same data distribution. Moreover, in all experiments, we use the same MLP on EMNIST-L and CNN on FMNIST, CIFAR10, CIFAR100 for all methods. For FedSR and FedCR, we additionally add a Gaussian layer and set the feature distribution dimension as a hyper-parameter in the same way as in DVIB (Alemi et al., 2016). When stochastic features become deterministic, our model will be the same as other methods. Please refer to Appendix A.1.2 for details.

Comparison methods. We compare validation (test) performance of FedCR[1] to other methods, including FedAvg and its locally fine-tuning version (FedAvg- FT), FedSR based on local fine-tuning (FedSR- FT)[2], FedPer, FedRep, LG-FedAvg, FedBABU, Ditto, and FedPAC (which also uses features of the previous round for the globally missing classes). In fact, since we just modify the feature extractor to output stochastic feactures and perform feature distri-

[1] Implementable codes for evaluation of our FedCR is available at: https://github.com/haozzh/FedCR.

[2] FedSR is proposed to train the global model, based on which we perform additional local fine-tuning for personalized learning.



(a) EMNIST-L (non-iid_1) (b) FMNIST (non-iid_1) (c) CIFAR100 (non-iid_1) (d) impact of β

Figure 3. Averaged test accuracy of clients vs. number of communication rounds on different methods and sensitivity analysis of β, with non-iid_1 setting. (a), (b), (c): test accuracy on EMNIST-L, FMNIST, CIFAR100; (d): test accuracy on EMNIST-L and CIFAR10.

bution alignment at clients, our FedCR can be seamlessly incorporated into most methods, such as FedFomo (Zhang et al., 2020) which calculates the aggregation weights based on the model and loss differences.

Implementation. We evaluate the performance of local models after 250 communication rounds for FMNIST and after 500 rounds for EMNIST-L, CIFAR10 and CIFAR100, at 100 clients with 10% participation rate. Note that we also evaluate local models on the test set of active clients after local updates at each round, followed by the global aggregation, as shown in Fig. 3. At the last round, the final average model accuracy is computed on test set, after locally updating the head of the fully trained global model for ten epochs at each client, which is the same strategy as used by FedRep. The client learning rate η and hyper-parameters of all approaches are individually tuned over a grid. Please refer to Appendix A.1.3 for additional setup details. We simply set τ in Eq. (9) to 1 due to random client participation in our experiments. For more complex client participation scenarios, we need to further tune value of τ. We also average the final accuracy reported over 3 random seeds by rerunning the experiments with different seeds.

6.2. Experimental Results

Performance evaluation. Experimental results of all methods under two different non-iid settings are shown in Table 2 and Figs. 3(a)-3(c). In most cases, our FedCR presents a superior performance than other algorithms on the four datasets with different data distributions. We attribute this to the CMI regularizer, which encourages all clients to learn their local models in alignment with the common and invariant global feature representation. For the more complex task (i.e., non-iid_1 and non-iid_2 on CIFAR100), most methods are ineffective, but FedCR still performs well. This is because even if there is only a small amount of local data within a certain classes, the client can still learn the diverse feature of data for these classes through the CMI regularizer.

Reasons for slow convergence at initial stage. Note that FedCR converges slightly slower at the initial stage as shown in Fig. 3(c).This might be due to the fact that the feature distribution uploaded by the client is generated from different local models during updating model parameters by SGD. At the beginning of training, the model changes rapidly, resulting in instability of the product of feature distributions. FedPAC obtains feature embeddings through an additional single pass on all data based on the final model after the local training, thus presenting a quicker start. But as stated in Section 4.3, by directly utilizing the features generated during the process of local update in FedCR, the computational overhead is reduced, while the server can also hardly infer meaningful information of raw data based only on the final model and the product of features generated from different local models at each client. Moreover, though converging slightly slower initially, FedCR can eventually converge to a higher accuracy. Thus, there is a trade-off between privacy /resource consumption and training efficiency.

Table 3. Averaged test accuracy (%) vs. Gaussian dimensions.

Dimensions	EMNIST-L	FMNIST	CIFAR10	CIFAR100
512	97.39	93.78	84.50	62.96
256	97.47	93.57	84.74	61.78
128	97.34	93.50	84.55	57.57
64	96.83	93.34	84.43	**54.35**
32	96.77	93.45	84.41	53.58
16	97.22	93.27	84.07	49.98
8	97.02	93.51	84.54	41.37
4	**95.82**	**92.07**	**82.13**	31.28
2	90.46	82.51	67.86	21.65

Choice of appropriate β. β is a critical hyper-parameter in FedCR. To test its sensitivity, we plot the validation performance for different choices of β on EMNIST-L and CIFAR10 datasets with non-iid_1 setting, as shown in Fig. 3(d). It can be see that there are many appropriate choices of β. Thus, FedCR is easy to tune in this setting, and similar results hold for other settings. For $\beta = 0$, our CMI regularizer cannot promote clients to learn global features, presenting a poorer performance. But when we enlarge β, the weight of prediction loss in the entire loss function in Eq. (7) decreases gradually, which may also lead to a decline in the model performance after β exceeds a certain value.

Effects of feature dimensions. The dimension of Gaussian features is an interesting hyper-parameter for stochastic neural networks. The Gaussian feature with less dimensions can greatly reduce the communication overhead. The feature extractor compresses high-dimensional raw data into low-dimensional and low-information feature distribution. Ideally, in stochastic neural networks, for 10-category tasks (EMNIST-L, FMNIST, CIFAR10), only 4-bits ($log_2(10) \approx 4$ bits) mutual information between x and z is required to predict labels. Similarly, for 100-category tasks (CIFAR100), 7-bits (($log_2(100) \approx 7$ bits) mutual information between x and z is required. So one may wonder how many dimensional features can accurately describe this mutual information in PFL? To anwer it, we demonstrate the average test accuracy over all clients with non-iid_1 setting for different Gaussian feature dimensions, as shown in Table 3. It can be found that roughly until Gaussian features are lower than 64 dimensions on CIFAR100 (4 dimensions on EMNIST-L, FMNIST, and CIFAR10), the feature representation can no longer represent 100 (10) different categories, which corresponds to a setting in which the mutual information between x and z is less than 7 bits for CIFAR100 (4 bits for EMNIST-L, FMNIST, and CIFAR10).

7. Conclusion

In this paper, we have proposed for PFL a FedCR method, which enforced all local clients to learn common and invariant feature representation by minimizing the discrepancy between local and global CMI. This CMI regularizer leaded to the theoretically sound alignment of global and local features. More importantly, for the global feature distribution, we used the PoE estimation to avoid leakage of local raw data. We provided theoretical generalization analysis, and also empirically showed the effectiveness of our FedCR. As our future works, we will focus on further improving the communication efficiency and addressing the fairness issues of FedCR. One possible research direction is to incorporate gradient quantization to alleviate the additional communication overhead incurred by FedCR. Besides, we did not consider the fairness issue in FedCR, which might be violated in some extreme cases where clients with insufficient data or a large distribution difference may be overlooked due to the CMI regularizer. Thus, we also intend to refine our FedCR from the perspective of fairness in the future.

8. Acknowledgements

This work was supported in part by the National Natural Science Foundation of China under Grant T2122024, Grant 62125109, Grant 62250055, Grant 61932022, Grant 61931023, Grant 61971285, Grant 61831018, Grant 62120106007, Grant 61972256, in part by the Program of Shanghai Science and Technology Innovation Project under Grant 20511100100, in part by the Key Research and Development Plan of Shandong Province under Grant 2020CXGC010701, and in part by the Shanghai Rising-Star Program under Grant 20QA1404600.

References

Acar, D. A. E., Zhao, Y., Navarro, R. M., Mattina, M., Whatmough, P. N., and Saligrama, V. Federated learning based on dynamic regularization. *arXiv preprint arXiv:2111.04263*, 2021.

Adilova, L., Rosenzweig, J., and Kamp, M. Information-theoretic perspective of federated learning. *arXiv preprint arXiv:1911.07652*, 2019.

Alemi, A. A., Fischer, I., Dillon, J. V., and Murphy, K. Deep variational information bottleneck. *arXiv preprint arXiv:1612.00410*, 2016.

Alemi, A. A., Fischer, I., and Dillon, J. V. Uncertainty in the variational information bottleneck. *arXiv preprint arXiv:1807.00906*, 2018.

Arivazhagan, M. G., Aggarwal, V., Singh, A. K., and Choudhary, S. Federated learning with personalization layers. *arXiv preprint arXiv:1912.00818*, 2019.

Ben-David, S., Blitzer, J., Crammer, K., and Pereira, F. Analysis of representations for domain adaptation. *Advances in neural information processing systems*, 19, 2006.

Collins, L., Hassani, H., Mokhtari, A., and Shakkottai, S. Exploiting shared representations for personalized federated learning. In *International Conference on Machine Learning*, pp. 2089–2099. PMLR, 2021.

Fallah, A., Mokhtari, A., and Ozdaglar, A. Personalized federated learning: A meta-learning approach. *arXiv preprint arXiv:2002.07948*, 2020.

Fischer, I. The conditional entropy bottleneck. *Entropy*, 22 (9):999, 2020.

Guo, C., Pleiss, G., Sun, Y., and Weinberger, K. Q. On calibration of modern neural networks. In *International conference on machine learning*, pp. 1321–1330. PMLR, 2017.

Hinton, G. E. Training products of experts by minimizing contrastive divergence. *Neural computation*, 14(8):1771–1800, 2002.

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

Kang, B., Xie, S., Rohrbach, M., Yan, Z., Gordo, A., Feng, J., and Kalantidis, Y. Decoupling representation and classifier for long-tailed recognition. *arXiv preprint arXiv:1910.09217*, 2019.

Kingma, D. P. and Welling, M. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Lakshminarayanan, B., Pritzel, A., and Blundell, C. Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in neural information processing systems*, 30, 2017.

Li, T., Hu, S., Beirami, A., and Smith, V. Ditto: Fair and robust federated learning through personalization. In *International Conference on Machine Learning*, pp. 6357–6368. PMLR, 2021.

Liang, P. P., Liu, T., Ziyin, L., Allen, N. B., Auerbach, R. P., Brent, D., Salakhutdinov, R., and Morency, L.-P. Think locally, act globally: Federated learning with local and global representations. *arXiv preprint arXiv:2001.01523*, 2020.

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

Nguyen, A. T., Torr, P., and Lim, S.-N. Fedsr: A simple and effective domain generalization method for federated learning. In *Advances in Neural Information Processing Systems*, 2022.

Oh, J., Kim, S., and Yun, S.-Y. Fedbabu: Towards enhanced representation for federated image classification. *arXiv preprint arXiv:2106.06042*, 2021.

Rivest, R. L., Adleman, L., Dertouzos, M. L., et al. On data banks and privacy homomorphisms. *Foundations of secure computation*, 4(11):169–180, 1978.

Shamsian, A., Navon, A., Fetaya, E., and Chechik, G. Personalized federated learning using hypernetworks. In *International Conference on Machine Learning*, pp. 9489–9502. PMLR, 2021.

Shwartz-Ziv, R. and Tishby, N. Opening the black box of deep neural networks via information. *arXiv preprint arXiv:1703.00810*, 2017.

Sim, K. C., Beaufays, F., Benard, A., Guliani, D., Kabel, A., Khare, N., Lucassen, T., Zadrazil, P., Zhang, H., Johnson, L., et al. Personalization of end-to-end speech recognition on mobile devices for named entities. In *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pp. 23–30. IEEE, 2019.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014.

Uddin, M. P., Xiang, Y., Lu, X., Yearwood, J., and Gao, L. Mutual information driven federated learning. *IEEE Transactions on Parallel and Distributed Systems*, 32(7): 1526–1538, 2020.

Wu, M. and Goodman, N. Multimodal generative models for scalable weakly-supervised learning. *Advances in Neural Information Processing Systems*, 31, 2018.

Xu, J., Tong, X., and Huang, S.-L. Personalized federated learning with feature alignment and classifier collaboration. In *The Eleventh International Conference on Learning Representations*, 2023.

Yu, H., Zhang, N., Deng, S., Yuan, Z., Jia, Y., and Chen, H. The devil is the classifier: Investigating long tail relation classification with decoupling analysis. *arXiv preprint arXiv:2009.07022*, 2020.

Zhang, C., Bengio, S., Hardt, M., Recht, B., and Vinyals, O. Understanding deep learning (still) requires rethinking generalization. *Communications of the ACM*, 64(3):107–115, 2021.

Zhang, M., Sapra, K., Fidler, S., Yeung, S., and Alvarez, J. M. Personalized federated learning with first order model optimization. *arXiv preprint arXiv:2012.08565*, 2020.

Zhu, Z., Hong, J., and Zhou, J. Data-free knowledge distillation for heterogeneous federated learning. In *International Conference on Machine Learning*, pp. 12878–12889. PMLR, 2021.

A. Appendix

A.1. Detailed Experiment Setup

A.1.1. DATASETS

We use the benchmark visual datasets EMNIST-L, FashionMNIST (FMNIST), CIFAR10, and CIFAR100, which consist of 10, 10, 10, and 100 different labels, respectively. Note that for EMNIST-L, we choose the first 10 letters of the letter section, which is similar to FedDyn (Acar et al., 2021). All data is split into 70% training set and 30% test set, and the train set and test set have the same distribution. The train and test splits for EMNIST-L, FMNIST, CIFAR-10 and CIFAR-100 are shown in Table 4.

Table 4. Train and test splits

Dataset	No. All data	No. Train per client (100 clients)	No. Test per client	Batch size	Rounds
EMNIST-L	56000 (48000+8000)	392	168	48	500
FMNIST	70000 (60000+10000)	490	210	48	250
CIFAR10	60000 (50000+10000)	420	180	48	500
CIFAR100	60000 (50000+10000)	420	180	48	500

To generate non-iid splits for the four datasets, we use two ways to divide training samples by classes and assign them to clients. For non-iid_1 case, we directly assign fixed classes to each client, and the amount of data in each class is the same. Specifically, for EMNIST-L, FMNIST, and CIFAR10, each client contains 5 classes, and for CIFAR100 each client contains 15 classes.

For non-iid_2 case, we use the similar approach as in FedDyn (Acar et al., 2021), where we apply Dirichlet distribution over the labels of dataset to create heterogeneous dataset. Specifically, we use Dirichlet distribution to produce a vector with a size equal to the number of classes for each client. These vectors correspond to the class priority for each client. Then, labels are sampled based on these vectors of each client, and the images are sampled based on the label without replacement. We repeat this process until all data are assigned to clients. Here the factor of Dirichlet distribution corresponds to the degree of data non-iid-ness. For Dirichlet parameter of 0.5 on EMNIST-L, FMNIST, and CIFAR10, each client has about 80% samples which belong to mostly four different classes. For CIFAR100, we set Dirichlet parameter to 0.3.

A.1.2. MODELS

For EMNIST-L, we use a simple MLP with cross entropy loss, as shown in Table 7.

For FMNIST, CIFAR-10 and CIFAR-100, we use the CNN model consisting of two convolutional layers with 64 5×5 filters, two 2×2 max pooling layers, two fully connected layers with 1024 neurons, and finally a softmax layer. A full description of the model is in Table 5. Our CNN model is similar to those used in FedAvg (McMahan et al., 2017) (without batch normalization layers) and FedDyn.

For the stochastic neural network in FedCR and FedSR, we additionally add a fully connected layer to generate $2V$ vectors as shown in Table 6 and 8. The first V vectors encodes μ of z, the remaining V outputs σ, i.e. diagonal elements of the covariance matrix Σ (after a softplus transform). In this way, we produce a decoupled V-dimensional Gaussian distribution, from which we sample a V-dimensional latent feature. Finally, we maps the V dimensional latent feature to the logits by fully connected layer. Note that when the feature becomes deterministic ($\sigma \to 0$), the two fully connected layers can be equivalent to one layer. That is, the model in this case is the same as other methods.

A.1.3. HYPER-PARAMETERS

All approachs are implemented in PyTorch 1.4.0 and CUDA 9.2, with GEFORCE GTX 1080 Ti throughout our experiments. We tune hyper-parameter over a grid to compare the performance of different methods. For local update in all methods, we tune the local learning rate over $\{0.5, 0.1, 0.05, 0.01, 0.005, 0.001\}$ and set up 10 epochs of local updates. For our proposed method FedCR, we average over 18 posterior samples, which seems to be sufficient to gain benefit from ensemble. Moreover, we tune the parameter β over $\{0.01, 0.005, 0.001, 0.0005, 0.0001\}$ and set it to 0.0005 for EMNIST-L, FMNIST, CIFAR10 and CIFAR100 (non-iid_1), and 0.001 for CIFAR100 (non-iid_2). For Ditto (Li et al., 2021), we tune the parameter λ over $\{1, 0.1, 0.01, 0.001\}$ and set it to 0.1 for EMNIST-L, 0.001 for FMNIST, and CIFAR100 (non-iid_1), and 0.01 for

Table 5. CNN Architecture for other methods

Layer Type	Size
Convolution + ReLu	5×5×64
Max Pooling	2×2
Convolution + ReLu	5×5×64
Max Pooling	2×2
Fully Connected + ReLU	1600×1024
Fully Connected + ReLU	1024×1024
Fully Connected	1024×10 & 1024×100

Table 6. CNN architecture for FedCR and FedSR

Layer Type	Size
Convolution + ReLu	5×5×64
Max Pooling	2×2
Convolution + ReLu	5×5×64
Max Pooling	2×2
Fully Connected + ReLU	1600×1024
Fully Connected + ReLU	1024×1024
Fully Connected	1024× $2V$ (to generate $\mathcal{N}(\mu, \sigma)$)
Fully Connected	V × 10 & V ×100

Table 7. MLP Architecture for other methods

Layer Type	Size
Fully Connected + ReLU	784×1024
Fully Connected + ReLU	1024×1024
Fully Connected	1024×10

Table 8. MLP architecture for FedCR and FedSR

Layer Type	Size
Fully Connected + ReLU	784×1024
Fully Connected + ReLU	1024×1024
Fully Connected	1024× $2V$ (to generate $\mathcal{N}(\mu, \sigma)$)
Fully Connected	V ×10

CIFAR100 (non-iid_2). For FedSR-FT, we tune the parameter α^{L2R} over $\{0.1, 0.01, 0.001\}$ and the parameter α^{CMI} over $\{0.1, 0.01, 0.001, 0.0001\}$ and set them to 0.01, 0.001 for EMNIST-L, FMNIST and CIFAR10, and 0.001, 0.001 for CIFAR100. For FedPAC, we tune the parameter λ over $\{10, 5, 1, 0.5, 0.1\}$ and set it to 1. Note that for non-iid_2 case, since the data distribution of each client is very different, the use of classifier combination in FedPAC will sometimes degrade the experimental performance. Therefore, we do not consider using predictor collaboration for this case.

A.2. Additional Experimental Results

A.2.1. Experimental Results on Model Non-Aggregation Scenarios

For the more challenging model non-aggregation scenarios, FedCR is ready to benefit local models by sharing only feature distribution yet not model parameters, which can further alleviate privacy and communication concerns. In order to explore this potential, we perform experiments on CIFAR100 (non-iid_1) with 100 clients and 10% participation rate over different number of classes at each client. We compare our method (non-aggregate-model) with Local-Training in which each client separately trains its own model without sharing anything, FedAvg (sharing the whole model), and FedPer (sharing the feature extractor). Moreover, in this experiment, we set the Gaussian dimensions to 256. This means tfat We only need to transmit $51,200$ (512×100) parameters, which is much smaller than the model's size of $2,794,852$ parameters.

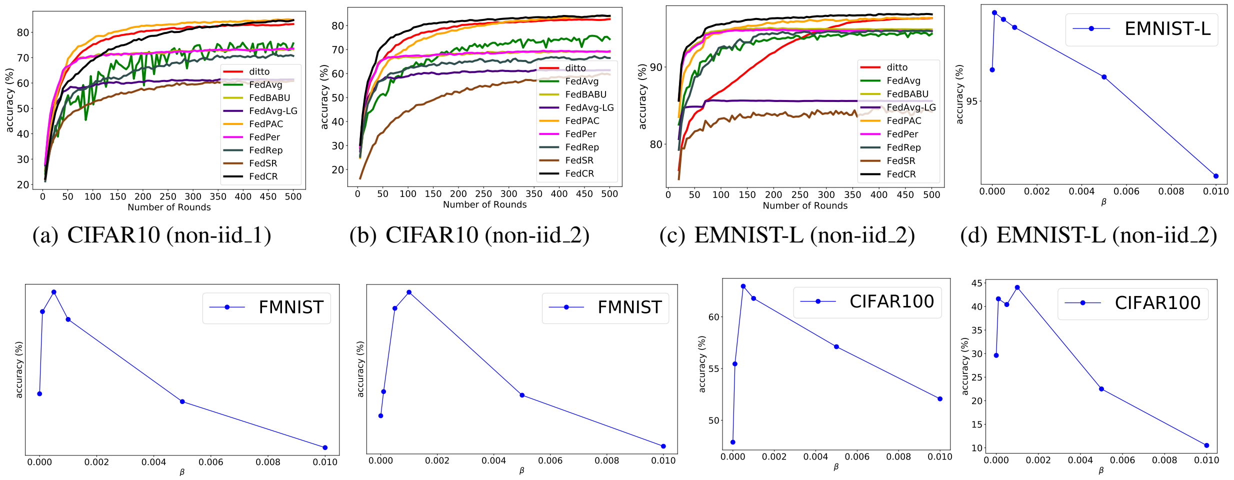
Table 9. Experimental Results for Model Non-aggregation Scenarios on CIFAR100 (non-iid_1)

Num_Class	Local-Training	FedAvg	FedPer	FedCR (non-aggregate-model)
10	52.44	35.51	54.98	**55.87**
15	40.99	34.08	39.5	**44.45**
20	28.33	31.89	37.03	**37.16**

The experimental results are shown in the Table 9. Our method still significantly benefits local model's performance, even when not sharing models but only features. For complex scenarios with more classes (20 classes), Local-Training cannot effectively complete the classification, but FedCR (non-aggregate-model) still greatly improves the model performance.

A.2.2. Convergence Curves and Sensitivity Analysis of β for Other Settings

We plot the averaged validation accuracy of models and sensitivity analysis for other settings, as shown in Fig. 4.



(a) CIFAR10 (non-iid_1) (b) CIFAR10 (non-iid_2) (c) EMNIST-L (non-iid_2) (d) EMNIST-L (non-iid_2)

(e) FMNIST (non-iid_1) (f) FMNIST (non-iid_2) (g) CIFAR100 (non-iid_1) (h) CIFAR100 (non-iid_2)

Figure 4. Averaged test accuracy of clients vs. number of communication rounds on different methods and sensitivity analysis of β.

A.2.3. Variance Measurements of Testt Accuracy with Different Seeds

We re-run experiments over different seeds to get the variance measurements of test accuracy to eliminate the impact of random seeds, as shown in Table 10.

Table 10. Average accuracy with the variance measurements (%) of local models at 100 clients with 10% participation rate .

Method	EMNIST-L		FMNIST		CIFAR10		CIFAR100	
	non-iid_1	non-iid_2	non-iid_1	non-iid_2	non-iid_1	non-iid_2	non-iid_1	non-iid_2
FedAvg	95.89±0.29	94.57±0.37	88.15±0.16	88.63±0.74	76.83±0.45	76.26±0.53	32.08±0.34	32.19±0.54
FedAvg-FT	96.32±0.51	95.78±0.43	92.76±0.63	91.70±0.36	83.65±0.37	83.20±0.81	54.42±0.19	41.46±0.61
FedPer	96.13±0.38	94.67±0.84	88.91±0.62	86.35±0.16	73.73±0.52	69.36±0.37	39.5±0.73	25.48±0.35
LG-FedAvg	88.44±0.63	85.59±0.61	86.86±0.84	85.44±0.37	61.37±0.47	61.53±0.23	44.48±0.73	28.23±0.15
FedRep	96.19±0.53	94.66±0.64	89.78±0.43	88.90±0.26	78.10±0.72	71.98±0.53	44.62±0.47	24.70±0.49
FedBABU	96.22±0.41	94.88±0.73	89.21±0.32	89.18±0.95	73.60±0.64	69.37±0.43	44.00±0.14	26.15±0.31
Ditto	96.60±0.50	96.30±0.19	92.03±0.62	91.75±0.46	83.25±0.15	82.81±0.36	58.40±0.81	41.85±0.72
FedSR-FT	86.22±0.30	84.71±0.34	85.55±0.52	85.45±0.73	61.47±0.46	60.96±0.65	40.82±0.34	24.56±0.26
FedPAC	96.97±0.47	96.43±0.75	93.45±0.24	92.15±0.32	**85.03**±0.29	84.07±0.64	58.65±0.63	43.25±0.47
FedCR	**97.47**±0.18	**96.98**±0.35	**93.78**±0.55	**93.00**±0.23	84.74±0.63	**84.26**±0.26	**62.96**±0.93	**44.06**±0.64

A.3. Proof

A.3.1. Proof of equation (2)

According to the Markov chain: $y_i \to x_i \to z \to \hat{y}_i$, the predictive distribution of the predicted label $\hat{y}_i$ given input data x_i of our model is:

$$p(\hat{y}_i|x_i) = \int p(\hat{y}_i, z|x_i)dz = \int \frac{p(\hat{y}_i, z, x_i)}{p(z, x_i)} \frac{p(z, x_i)}{p(x_i)} dz = \int p(z|x_i)p(\hat{y}_i|z, x_i)dz = \mathbb{E}_{p(z|x_i)}[\hat{p}(y_i|z)]. \tag{11}$$

The last inequality is based on the Markov chain $p(\hat{y}_i|z, x_i) = p(\hat{y}_i|z)$, where in addition we further denote $p(\hat{y}_i|x_i)$ as $\hat{p}(y_i|x_i)$. Hence, for both the regression or classification tasks where the loss function is usually chosen as the negative log predictive, i.e, $\ell(w_i; x_i, y_i) = -\log \mathbb{E}_{p(z|x_i)}[\hat{p}(y_i|z)]$, the local objective function of client i is $f_i(w_i) = \mathbb{E}_{p(x_i,y_i)}\left[-\log \mathbb{E}_{p(z|x_i)}[\hat{p}(y_i|z)]\right]$.

A.3.2. Proof of Lemma 4.1

Lemma A.1. *For $x_i \subseteq x$ and within the same label classes of client i, we have:*

$$I(z;x|y_i) - I_i(z;x_i|y_i) = \mathbb{E}_{p(x_i,y_i)}\mathbb{E}_{p(x|x_i,y_i)}\left[\mathrm{KL}[p(z|x)||p(z|x_i)]\right], \tag{12}$$

where $\mathrm{KL}[p(z|x)||p(z|x_i)]$ denotes the KL divergence between $p(z|x)$ and $p(z|x_i)$ given the label y_i, i.e. a class-wise feature alignment. Note that $p(x|x_i, y_i)$ indicates that the global data can be determined only given the clients actively participating in the training at a communication round, due to the partial client participation setting in PFL.

Proof. By setting $x = x_i \cup \neg x_i$ and then $p(x) = p(x_1, ..., x_m) = p(x_i, \neg x_i)$, we have

$$\begin{aligned}
&I_i(z;x_i|y_i) - I(z;x|y_i)\\
=&\iiint p(x_i,z,y_i)\log\frac{p(z|x_i,y_i)}{p(z|y_i)}dzdx_idy_i - \iiint p(x,z,y_i)\log\frac{p(z|x,y_i)}{p(z|y_i)}dzdxdy_i\\
=&\iiiint p(x_i,z,\neg x_i,y_i)\log\frac{p(z|x_i,y_i)}{p(z|y_i)}dzdx_id\neg x_idy_i - \iiiint p(\neg x_i,z,x_i,y_i)\log\frac{p(z|x,y_i)}{p(z|y_i)}dzd\neg x_idx_idy_i\\
=&\iiiint p(x_i,z,\neg x_i,y_i)\log\frac{p(z|x_i,y_i)}{p(z|x,y_i)}dzdx_id\neg x_idy_i\\
\overset{(A_1)}{=}&\iiiint p(x_i,y_i,z,\neg x_i)\log\frac{p(z|x_i)}{p(z|x)}dzdx_id\neg x_idy_i\\
=&\iiiint p(x_i,y_i)p(\neg x_i|x_i,y_i)p(z|\neg x_i,x_i)\log\frac{p(z|x_i)}{p(z|x)}dzdx_id\neg x_idy_i\\
\overset{(A_2)}{=}&\iiiint p(x_i,y_i)p(x|x_i,y_i)p(z|x)\log\frac{p(z|x_i)}{p(z|x)}dzdx_id\neg x_idy_i\\
=&-\mathbb{E}_{p(x_i,y_i)}\mathbb{E}_{p(x|x_i,y_i)}[\mathrm{KL}[p(z|x)|p(z|x_i)]],
\end{aligned} \tag{13}$$

where (A1) is due to the property of Markov chain $y_i \to x_i \to z$, and (A2) is from that $p(x|x_i, y_i) = \frac{p(x,x_i,y_i)}{p(x_i,y_i)} = \frac{p(x,y_i)}{p(x_i,y_i)} = \frac{p(\neg x_i,x_i,y_i)}{p(x_i,y_i)} = p(\neg x_i|x_i, y_i)$.

□

A.3.3. Proof of Lemma 4.2

Lemma A.2. *For marginal posteriors $p(z|x_i)(\forall i \in \mathcal{M})$, the joint posterior can be approximated as (Hinton, 2002):*

$$p(z|x) = p\left(z|x_1,\ldots,x_m\right) \propto \tau p(z)\prod_{i=1}^{m} p\left(z|x_i\right), \tag{14}$$

where $\tau \triangleq \frac{\prod_{i=1}^{M} p(x_i)}{p(x_1,\ldots,x_m)}$ represents the degree of independence between clients, and $p(z)$ is a prior distribution, usually the spherical Gaussian.

Proof. Throughout this proof, we utilize the similar techniques as in (Hinton, 2002; Wu & Goodman, 2018).

$$\begin{aligned}
p\left(z|x_1,\ldots,x_m\right) &= \frac{p\left(x_1,\ldots,x_m|z\right)p(z)}{p\left(x_1,\ldots,x_m\right)} \\
&\overset{(A_1)}{=} \frac{p(z)}{p\left(x_1,\ldots,x_m\right)}\prod_{i=1}^{m}p\left(x_i|z\right) \\
&= \frac{p(z)}{p\left(x_1,\ldots,x_m\right)}\prod_{i=1}^{m}\frac{p\left(z|x_i\right)p\left(x_i\right)}{p(z)} \\
&= \frac{\prod_{i=1}^{m}p\left(z|x_i\right)}{\prod_{i=1}^{m-1}p(z)}\cdot\frac{\prod_{i=1}^{m}p\left(x_i\right)}{p\left(x_1,\ldots,x_m\right)} \\
&\propto \tau\frac{\prod_{i=1}^{m}p\left(z|x_i\right)}{\prod_{i=1}^{m-1}p(z)},
\end{aligned} \tag{15}$$

where (A1) is from the conditional independence assumptions, i.e., $x_1,...,x_m$ are conditionally independent given the common latent feature z (Wu & Goodman, 2018). In the last equality, we define $\tau \triangleq \frac{\prod_{i=1}^{M}p(x_i)}{p(x_1,\ldots,x_m)}$, which represents the degree of independence between clients' data distribution.

Moreover, if we approximate $p\left(z|x_i\right)$ with $q\left(z|x_i\right) \equiv \tilde{q}\left(z|x_i;w_i^f\right)p(z)$, then Eq. (15) can be rewritten as:

$$p\left(z|x_1,\ldots,x_m\right) \propto \tau\frac{\prod_{i=1}^{m}p\left(z|x_i\right)}{\prod_{i=1}^{m-1}p(z)} = \tau\frac{\prod_{i=1}^{m}\left[\tilde{q}\left(z|x_i;w_i^f\right)p(z)\right]}{\prod_{i=1}^{m-1}p(z)} = \tau p(z)\prod_{i=1}^{m}\tilde{q}\left(z|x_i;w_i^f\right). \tag{16}$$

Replacing the symbols, we have:

$$p(z|x) = p\left(z|x_1,\ldots,x_m\right) \propto \tau p(z)\prod_{i=1}^{m}p\left(z|x_i\right), \tag{17}$$

In fact, here we are trying to abvoid the quotient term, and this trick in Eq. (16) is also used by the (Hinton, 2002; Wu & Goodman, 2018). Since $p(z)$ is an irrelevant prior , we can also directly absorb it as the previous hyperparameter τ. □

A.3.4. Proof of Theorem 5.1

Theorem A.3. *Let $\mathcal{H}$ be a hypothesis space of VC-dimension d and $\mathcal{X} \to \mathcal{Z}$ be a feature representation function shared across clients. Given a global meta-distribution $\mathcal{D}$ from which the active clients with a local distribution $\mathcal{D}_i$ are drawn, let $\hat{\epsilon}_{\mathcal{D}_i}(h_i)$ denote the empirical risk of hypothesis h_i on $\mathcal{D}_i$. Similarly, let $\epsilon_{\mathcal{D}}(h_i)$ denote the expected risk of hypothesis h_i on $\mathcal{D}$. Let $\tilde{\mathcal{D}}_i, \tilde{\mathcal{D}}_i'$ be the induced distribution of $\mathcal{D}_i$ by samples of size n for FedPer and FedCR, and $\tilde{\mathcal{D}}$ be the induced distribution of $\mathcal{D}$. Then with probability at least $1-\delta$,*

$$\frac{1}{m}\sum_{i\in\mathcal{M}}\epsilon_{\mathcal{D}}(h_i) \leq \frac{1}{m}\sum_{i\in\mathcal{M}}\hat{\epsilon}_{\mathcal{D}_i}(h_i) + \frac{1}{m}\sum_{i\in\mathcal{M}}d_{\mathcal{H}\Delta\mathcal{H}}\left(\tilde{\mathcal{D}}_i',\tilde{\mathcal{D}}\right) + \sqrt{\frac{4}{n}\left(d\log\frac{2en}{d}+\log\frac{4m}{\delta}\right)} + \frac{1}{m}\sum_{i\in\mathcal{M}}\lambda_i \tag{18}$$

where e is the base of the natural logarithm, $\lambda_i = \min_h\left(\hat{\epsilon}_{\mathcal{D}_i}(h_i)+\epsilon_{\mathcal{D}}(h_i)\right)$ denotes the combined risk of the ideal hypothesis, and $d_{\mathcal{H}\Delta\mathcal{H}}(\tilde{\mathcal{D}}_i',\tilde{\mathcal{D}})$ denotes the distance of distribution with $d_{\mathcal{H}\Delta\mathcal{H}}(\tilde{\mathcal{D}}_i',\tilde{\mathcal{D}}) < d_{\mathcal{H}\Delta\mathcal{H}}(\tilde{\mathcal{D}}_i,\tilde{\mathcal{D}})$.

Proof. Throughout this proof, we utilize the same techniques as in (Zhu et al., 2021; Ben-David et al., 2006). By treating local data $\mathcal{D}_i$ as the source and the global data as the target, we can get, $\forall\delta > 0$, with probability $1-\frac{\delta}{m}$:

$$\epsilon_{\mathcal{D}}(h_i) \leq \hat{\epsilon}_{\mathcal{D}_i}(h_i) + d_{\mathcal{H}\Delta\mathcal{H}}\left(\tilde{\mathcal{D}}_i',\tilde{\mathcal{D}}\right) + \lambda_i + \sqrt{\frac{4}{n}\left(d\log\frac{2en}{d}+\log\frac{4m}{\delta}\right)}. \tag{19}$$

Then we have:

$$
\begin{aligned}
&\Pr\left[\frac{1}{m}\sum_{i\in\mathcal{M}}\epsilon_{\mathcal{D}}(h_i) > \frac{1}{m}\sum_{i\in\mathcal{M}}\left(\hat{\epsilon}_{\mathcal{D}_i}(h_i) + \left(d_{\mathcal{H}\Delta\mathcal{H}}\left(\tilde{\mathcal{D}}_i',\tilde{\mathcal{D}}\right)+\lambda_i\right) + \sqrt{\frac{4}{n}\left(d\log\frac{2en}{d}+\log\frac{4m}{\delta}\right)}\right)\right]\\
\leq &\Pr\left[\bigvee_{i\in\mathcal{M}}\epsilon_{\mathcal{D}}(h_i) > \hat{\epsilon}_{\mathcal{D}_i}(h_i) + d_{\mathcal{H}\Delta\mathcal{H}}\left(\tilde{\mathcal{D}}_i',\tilde{\mathcal{D}}\right)+\lambda_i + \sqrt{\frac{4}{n}\left(d\log\frac{2en}{d}+\log\frac{4m}{\delta}\right)}\right]\\
\leq &\sum_{i\in\mathcal{M}}\frac{\delta}{m} = \delta.
\end{aligned}
\tag{20}
$$

For $d_{\mathcal{H}\Delta\mathcal{H}}(\tilde{\mathcal{D}}_i',\tilde{\mathcal{D}}) < d_{\mathcal{H}\Delta\mathcal{H}}(\tilde{\mathcal{D}}_i,\tilde{\mathcal{D}})$, intuitively, after feature distribution alignment, the discrepancy between the local distribution and the global distribution in latent space is smaller than the discrepancy between the unaligned individual and global distributions (Zhu et al., 2021).

Theorem 5.1 demonstrates that by explicitly aligning feature distribution, the local hypothesis can infer data outside the local distribution and generalize better, even with an extremely limited amount of local data at clients.

□

A.4. Detailed Version of Algorithm 1

Here, we provide a detailed version of the practical execution process of our FedCR algorithm, as outlined in Algorithm 2. This detailed version will allow us to demonstrate how our algorithm performs step by step in practice. In Line 11, for the KL divergence of two Gaussian distributions, it is calculated as follows:

$$
KL(p(\boldsymbol{x})\|q(\boldsymbol{x})) = \frac{1}{2}\left[\left(\boldsymbol{\mu}_p-\boldsymbol{\mu}_q\right)^{\top}\boldsymbol{\Sigma}_q^{-1}\left(\boldsymbol{\mu}_p-\boldsymbol{\mu}_q\right) - \log\det\left(\boldsymbol{\Sigma}_q^{-1}\boldsymbol{\Sigma}_p\right) + \operatorname{Tr}\left(\boldsymbol{\Sigma}_q^{-1}\boldsymbol{\Sigma}_p\right) - n\right], \tag{21}
$$

where $p(x) \sim \mathcal{N}(\mu_p, \Sigma_p)$, $q(x) \sim \mathcal{N}(\mu_q, \Sigma_q)$, and n is the dimension.

Algorithm 2 FedCR for personalized federated learning (detailed version).

```
**server**: $\mu_0^c = \mathbf{0}$, $\Sigma_0^c = \mathbf{1}$; **clients**: learning rate $\eta$
**for** each round $t = 1, 2, 3, ..$ **do**
    sample clients $\mathcal{P}_t \subseteq \mathcal{M}$
    // *local training*:
    **for** each client $i \in \mathcal{P}_t$ in parallel **do**
        receive $\mu_t^c, \Sigma_t^c, w^f$ and initialize $w_{i,0} = [w^f, w_i^p]$
        **for** each local step $k = 1, 2, \ldots, K$ **do**
            update the whole network $w_i$ by SGD as (7):
            $\mu_i^{y_i^{(n)}}, \Sigma_i^{y_i^{(n)}} = f\left(x_i^{(n)}; w_{i,k-1}^f\right),$
            $z_i \leftarrow \mu_i^{y_i^{(n)}} + \epsilon \cdot \Sigma_i^{y_i^{(n)}}$ where $\epsilon \sim \mathcal{N}(0, 1),$
            $\mathcal{L}_i = l\left(f\left(z_i; w_{i,k-1}^p\right), y_i^{(n)}\right) + \beta \cdot KL(\mathcal{N}(\mu_t^{c=y_i^{(n)}}, \Sigma_t^{c=y_i^{(n)}}) \| \mathcal{N}(\mu_i^{y_i^{(n)}}, \Sigma_i^{y_i^{(n)}})),$
            $w_{i,k} = w_{i,k-1} - \eta_L \nabla_w \mathcal{L}_i$
            save the most recent features $\mu_i^{y_i^{(n)}}, \Sigma_i^{y_i^{(n)}}$ for sample $(x_i^{(n)}, y_i^{(n)})$
        **end for**
        **for** each class $c$ in $\mathcal{C}_i$ **do**
            // *product of Gaussian distributions based on Line 209 and Eq. (8)*
            $\mu_i^c = \left(\sum_{n=1}^{N_i} \mathbf{1}(y_i^{(n)} = c)\mu_i^{y_i^{(n)}}(\Sigma_i^{y_i^{(n)}})^{-1}\right)\left(\sum_{n=1}^{N_i} \mathbf{1}(y_i^{(n)} = c)(\Sigma_i^{y_i^{(n)}})^{-1}\right)^{-1}$
            $\Sigma_i^c = \left(\sum_{n=1}^{N_i} \mathbf{1}(y_i^{(n)} = c)(\Sigma_i^{y_i^{(n)}})^{-1}\right)^{-1}$
        **end for**
        set $w_i^p = w_{i,K}^p$; send $w_{i,K}^f$, $\mu_i^c, \Sigma_i^c$ to server
    **end for**
    // *global aggregation at server*:
    $w^f = \frac{1}{|\mathcal{P}_t|} \sum_{i \in \mathcal{P}_t} w_i^{f,K}$
    **if** $c \in \mathcal{C}$ and $c \notin \mathcal{C}_i (\forall i \in \mathcal{P}_t)$ **then**
        $\mu_t^c = \mu_{t-1}^c$; $\Sigma_t^c = \Sigma_{t-1}^c$
    **end if**
    **for** each class $c$ in $\mathcal{C}$ **do**
        // update the global class-wise feature $\mu_t^c, \Sigma_t^c$ as Eq. (9) and $p(z) \sim \mathcal{N}(\mathbf{0}, \mathbf{1})$
        $\mu_t^c = \left(\mathbf{0} + \sum_{i \in \mathcal{P}_t} \mu_i^c (\Sigma_i^c)^{-1}\right)\left(\mathbf{1} + \sum_{i \in \mathcal{P}_t} (\Sigma_i^c)^{-1}\right)^{-1}$
        $\Sigma_t^c = \left(\mathbf{1} + \sum_{i \in \mathcal{P}_t} (\Sigma_i^c)^{-1}\right)^{-1}$
    **end for**
    send $w^f, \mu_t^c, \Sigma_t^c$ to clients
**end for**
```