# Federated Learning with Common Representation Learning Criterion and Personalized Predictor

Wenzhong Wang*†, Zaipeng Xie✉*†, Bingzhe Yu†, Zhihao Qu*†, Yufeng Zhang†, and Hongli Cao‡
*Key Laboratory of Water Big Data Technology of Ministry of Water Resources, Hohai University, Nanjing, China
†College of Computer and Information, Hohai University, Nanjing, China
‡Key Laboratory of Underwater Acoustic Signal Processing of Ministry of Education, Southeast University, Nanjing, China
Email: {hhu_wangwenzhong, zaipengxie, hhu_ybz, quzhihao, yufengzhang}@hhu.edu.cn, honglicao@seu.edu.cn

***Abstract*—Federated learning (FL) enables model training on decentralized devices while preserving data privacy. However, data heterogeneity poses a significant challenge to FL, and various approaches have been proposed to address it. Existing research has mainly focused on either enhancing global models or customizing personalized models for clients. This paper proposes a novel approach, FedCRC, that decouples the machine learning model into a representation extractor and predictor. This enables us to enhance both generalization and personalization, thereby addressing the challenge of data heterogeneity in FL. The approach employs a stable global predictor to unify the representation learning criterion during the training of the representation extractor. Additionally, a personalized predictor is trained for each client to achieve a personalized model tailored to the local data distribution. Our FedCRC algorithm was evaluated on multiple benchmark datasets with varying distributions, covering diverse settings. Extensive experimental results demonstrate the effectiveness of our method.**

***Index Terms*—Federated Learning, Data Heterogeneity, Representation Learning, Personalized Predictor**

## I. Introduction

Federated learning has emerged as a distributed machine learning paradigm to address the challenges of decentralized data [1]. FL enables decentralized clients to train models using their local data collaboratively, with the protection of data privacy. In a standard FL setup, a central server manages the global model and selects the clients participating in the training process. The clients then perform local updates based on the global model and send the updated model back to the server. However, when faced with data heterogeneity, standard FL algorithms may struggle to learn well-performing models from clients with different data distributions, presenting a significant obstacle to the success of FL. To address this challenge, novel approaches must be developed to improve the performance of FL algorithms in data heterogeneous settings.

Two primary paradigms have emerged to address the problem of data heterogeneity. The initial paradigm concentrates on improving the performance of the global model, as exemplified by [2]–[4], and [5]. In contrast, the second paradigm delves into personalized federated learning (PFL), which endeavors to develop models tailored to local data for each client, as illustrated by [6]–[8]. These approaches diverge in their motivations: FL algorithms for global models seek to bolster the model's generalization performance, while PFL algorithms aim to create personalized models suited for client-specific data. However, only a handful of efforts have simultaneously focused on enhancing the global model's generalization capability and developing personalized models for individual clients.

Several studies have sought to tackle these two objectives independently by segregating the model into separate components [3], [7], [8]. Deep neural network-based models can be split into a representation extractor associated with representation learning and a predictor connected to specific tasks [9]. The success of multi-task deep learning suggests that the representation extractor is responsible for extracting common representations, whereas the predictor is closely connected to task-specific aspects [10]. During the model training phase, the predictor can be regarded as a criterion for representation learning, as it transmutes the input data's representation from the representation space to the data label space [3]. However, the predictor's susceptibility to data distribution leads to disparate representation learning criteria for different clients in data heterogeneity FL [11].

Figure 1 illustrates the effect of data distribution on predictors using the L2 distance of predictor parameters. The disparity between predictors increases with the number of

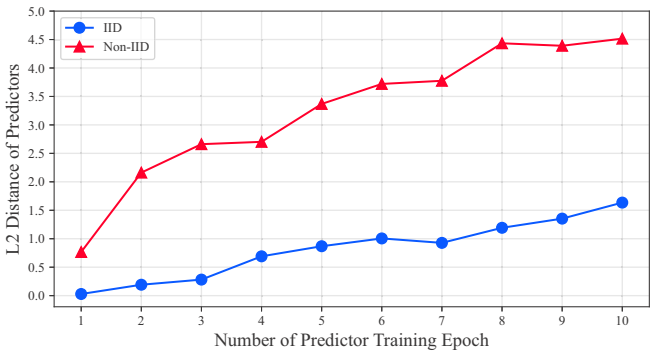


Fig. 1. L2 distance of predictors in data heterogeneous environments

This work is supported by The Belt and Road Special Foundation of the State Key Laboratory of Hydrology-Water Resources and Hydraulic Engineering under Grant No. 2021490811, the National Natural Science Foundation of China under Grant 62102131, and Natural Science Foundation of Jiangsu Province BK20210361.

updates in the data heterogeneous setting. This observation indicates that, in FL with data heterogeneity, the model's predictor can undergo significant alterations due to differences in data distribution. In this scenario, two clients train predictors based on the same pre-trained model but with different data distributions. This experiment uses a CNN model with four convolutional layers and one fully connected layer alongside the MNIST dataset. Regarding data distribution settings, IID denotes that client data are independently and identically distributed, while Non-IID indicates that client data are sampled from a Dirichlet distribution with $\beta = 0.1$. This study designates the predictor as the output layer of the model.

Building on the insights from the aforementioned experimental observations, this study sets forth a novel approach to federated learning, known as FedCRC (Federated Learning with Common Representation Learning Criterion and Personalized Predictor). FedCRC seeks to elevate the performance of both the global and personalized models. To achieve this objective, this study adopts a stable global predictor to ensure consistency in the representation learning criterion during the training of the representation extractor. Each client is trained with a personalized predictor for a tailored model. The global predictor parameters are held constant during the training of the representation extractor, and they remain fixed during the joint training of the personalized and global predictors. Additionally, an exponential moving average is applied to the global predictor to ensure a smooth evolution of the representation learning criterion over time and to prevent abrupt shifts. The main contributions of our work are as follows:

- We introduce a novel FL algorithm, FedCRC, which leverages a common representation learning criterion to train the model's representation extractor while concurrently training a personalized predictor for each client to achieve personalized models.
- Our algorithm effectively constructs personalized models without compromising the performance of the global model and enables seamless integration between the personalized and global predictors.
- Through extensive experiments on MNIST, CIFAR-10, and CIFAR-100 datasets with different levels of data heterogeneity, we demonstrate that FedCRC outperforms five cutting-edge algorithms in prediction accuracy and generalization performance.

## II. Related Work

Many methods have been proposed to enhance the global model's performance in FL with data heterogeneity. In order to diminish the model update bias induced by data heterogeneity, some methods incorporate regularization terms within the local learning process. For instance, FedProx [2] introduces a model parameter distance regularization term in the local learning objective, while SCAFFOLD [12] employs a control variable to rectify the local update gradient. However, these methods may not fully utilize the knowledge of local models due to the regularization terms. FedProto [4] proposes a prototype-based FL algorithm that employs the prototype representation of each class to guide local model training. MOON [5] utilizes model contrastive learning to enable the local model to learn the same representation as the global model but with an increased computational load. FedBABU [3] only updates the extractor of the model and neglects the collaboration of different model components. Loss-weighted FL algorithms use carefully designed loss functions to mitigate the impact of class imbalance on the model [13], [14]. Methods based on shared data, such as retraining the global model using public data [15], tend to violate privacy protection principles. It is desired to develop an approach that capitalizes on the full potential of local model knowledge without violating federated learning principles or increasing the computational overhead.

Personalized federated learning aims to train models for each client to accommodate local data distribution. FedMTL [6] treats model training for each client as a distinct optimization task, while Ditto [16] applies adaptive model dissimilarity penalization during the training process. Some decoupling-based methods group clients with similar data distributions and learn intra-group global models for each client group to obtain personalized global models [17]–[19]. However, personalized methods have to deal with the challenge of losing the global model's generalization capability. Decoupling-based methods, such as FedPer [7], divide the model into shared and personalized layers, where each client trains the model using its own personalized layers and aggregates only the shared layers. FedRep [8] ensures all clients share the representation extractor but have distinct predictors and concentrate on predictor training during local updates, resulting in the method lacking an extractor with strong generalization performance. However, methods that can simultaneously achieve strong generalization performance and personalized models for each client have yet to be fully realized, presenting an exciting opportunity for further research and advancement.

## III. Methodology

### A. Problem Formulation

In the FL scenario, let us assume there are $M$ clients. The data $D_i = (x_j, y_j)_{j=1}^{|D_i|}$ of client $i$ stems from the data distribution $P_i$, where $x_j$ and $y_j$ represent the input data and the corresponding category label of the $j$-th sample, respectively. The optimization objective of standard FL [1] can be expressed as follows:

$$\min_{\omega} \mathcal{L}(\omega) = \sum_{i=1}^{M} \frac{|D_i|}{N} L_i(\omega) \tag{1}$$

where $\omega$ represents the machine learning model, $N = \sum_i |D_i|$ represents the sum of client sample sizes. $L_i(\omega)$ is the empirical risk loss of client $i$, defined as follows:

$$L_i(\omega) = \mathbb{E}_{(x,y)\sim P_i}[\ell(\omega; x, y)] \tag{2}$$

where $\ell$ is the loss function for each data instance. After the local update is completed, the server collects the client

models $\omega_i$ and performs model averaging aggregation. The aggregation equation is as follows:

$$\omega_g = \sum_{i=1}^{m} \frac{|D_i|}{N} \omega_i \tag{3}$$

where $\omega_g$ represents the global model and $m$ denotes the number of models.

In FL with heterogeneous datasets, disparate data distributions can cause deviations in the client's local learning objective. Since clients employ diverse optimization objectives to update their local models, inconsistencies among these models emerge, and it is essential to address these discrepancies adequately. We propose the FedCRC algorithm, which handles the model's representation extractor and predictor independently. Furthermore, a local predictor is utilized for each client to obtain a personalized model. The global predictor is also updated to make the parts of the model fit each other.

### *B. Local Update Process of FedCRC*

FedCRC employs a distinct approach of updating different components of the model separately, as demonstrated in Fig. 2 which illustrates the local update process of FedCRC. Different with other model decoupling approaches, our approach trains the global model and the personalized model jointly.

*1) The Updating Process of Representation Extractor:* Deep neural network models can be divided into two parts: an extractor that obtains the representation of input data and a predictor that outputs the prediction results. Let $\omega = \{f(\phi), h(\theta)\}$ denote a model, where $f(\phi)$ is the representation extractor with parameter $\phi$, and for input data $x$, $z = f(x, \phi)$ is the representation vector of $x$. $h(\theta)$ is the predictor with parameter $\theta$, and for a given representation vector $z$, $\tilde{y} = h(z, \theta)$ is the prediction result of $z$. For ease of presentation, we use $f$ and $h$ to denote $f(\phi)$ and $h(\theta)$, respectively.

During the model training process, the predictor computes the representation $z$ and outputs a predicted label for each data instance, and the model is updated based on the loss between the predicted label and the true label. However, data heterogeneity can cause different clients to have different predictors, thus different representation learning criteria. We use an identical global predictor $h_g$ for all client models during the local training process to unify the representation learning criterion across clients. In client $i$, the representation extractor is optimized as follows:

$$f_g' = \arg\min_{f_g} L_i(f_g, h_g) \tag{4}$$

where $L_i$ represents the empirical risk loss of client $i$.

*2) The Updating Process of Local Predictor:* While the global model with a shared representation extractor can achieve good generalization performance, it may not be optimal for each client's local data distribution. Personalized models usually perform better on local data than global models. Our algorithm involves jointly training all clients' shared representation extractor $f_g$ to form a global model with a global predictor $h_g$. This representation extractor can effectively extract features from different data distributions and perform better generalization. To take advantage of the global representation extractor during the training of the personalized model, we also train a local predictor $h_i$ for each client using the fixed global representation extractor. By combining the local predictor and representation extractor in this way, our algorithm can get a personalized model for each client without affecting the generalized performance of the global model. Specifically, the local predictor is optimized as

$$h_i' = \arg\min_{h_i} L_i(f_g, h_i) \tag{5}$$

where $L_i$ represents the empirical risk loss of client $i$.

*3) The Updating Process of Global Predictor:* In centralized machine learning settings, it is common practice to update the complete model to enhance its performance. This highlights the significance of cooperation among different parts of the model. In addition to optimizing the representation extractor, training a global predictor that complements the updated extractor is crucial. Nonetheless, excessive updating of the global predictor, which acts as the benchmark for representation learning, can result in the failure of the representation extractor to converge. Hence, it is crucial to update the global predictor efficiently and stably.

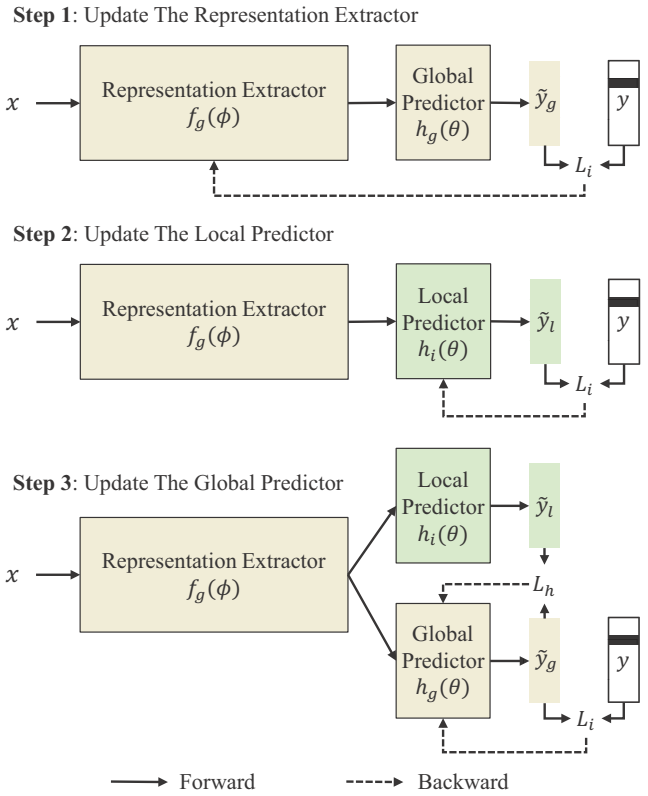


Fig. 2. The local update steps of FedCRC. Step 1: Train the global extractor using the fixed global predictor. Step 2: Train the local predictor using the fixed representation extractor. Step3: Train global predictor using the fixed representation extractor and local predictor.

Since personalized models often outperform global models on individual clients, we aim to leverage the knowledge of personalized models in our approach. To achieve this, we use the local predictor to guide the training of the global predictor. Specifically, we optimize the global predictor based on two objectives: the local empirical loss and the Kullback-Leibler (KL) divergence between the output of the local predictor and the global predictor. Although both predictors share the same representation extractor, they may produce different prediction results for the same feature vector $z$, with the local predictor typically having better performance. Therefore, we use the KL divergence to quantify the difference between the two output results and strive to make the global predictor imitate the output of the local predictor. The optimization objective guided by the local predictor is defined as follows:

$$L_h = KLoss(h_g(f_g(x)), h_i(f_g(x))) \tag{6}$$

The optimization of the global predictor is defined as follows:

$$h'_g = \arg\min_{h_g}[L_i(f_g, h_g) + L_h(f_g, h_g, h_i)] \tag{7}$$

When aggregating the global predictor from different clients, we need to consider the impact of drastic changes in the global predictor. To ensure the stability of the global representation learning criterion, we employ the exponential moving average technique to aggregate the global predictor. The exponential moving average factor $\tau$ prevents the global predictor to forget the previous representation learning criterion completely. We first compute the temporary global predictor $h'_g$ as follows:

$$h'_g = \sum_{i=1}^{m} \frac{|D_i|}{N} h_{gi} \tag{8}$$

The global predictor is then aggregated as follows:

$$h_g^{t+1} = \tau h_g^t + (1-\tau) h'_g \tag{9}$$

where the $t$ is the number of communication rounds, $\tau \in (0, 1)$.

### C. FedCRC Algorithm

Algorithm 1 presents the complete FedCRC algorithm. During the client's local update process, the representation extractor and predictor are updated independently. Initially, the representation extractor is updated using a stable global predictor to ensure that all clients share the same representation learning criterion. Subsequently, the extractor is used to train the local predictor, which yields the personalized model. Finally, the personalized model is employed to guide the training of the global predictor, allowing efficient updates to the global predictor due to the knowledge provided by the personalized model. When the server aggregates the client models, average aggregation is used for the representation extractor to enhance its generalization ability. To prevent sudden changes in the representation learning criterion, the global predictor is aggregated using an exponential moving average for the next round.

**Algorithm 1:** FedCRC Algorithm

**Input:** Client participation rate $\sigma$, communication rounds $T$, number of local updates $E$, exponential moving average factor $\tau$;
**Output:** Global model $\omega_g$, personal model $\omega_1$,...,$\omega_M$;
Initialize $\omega_g^0 \leftarrow \{f_g^0, h_g^0\}$, and $h_1^0, \cdots, h_M^0$;
Let $m \leftarrow max(\lfloor \sigma M \rfloor, 1)$;
**for** $t = 1, 2, \cdots, T$ **do**
  $\mathcal{S}^t \leftarrow$ randomly sample $m$ clients;
  **for** *client* $i$ *in* $\mathcal{S}^t$ **do**
    Client $i$ downloads $f_g^{t-1}, h_g^{t-1}$;
    Fix the parameters of $h_g^{t-1}$;
    $f_{g,i}^t \leftarrow$ update $f_g^{t-1}$ with (4) for $E$ epoch;
    Fix the parameters of $f_{g,i}^t$ and unfix the $h_g^{t-1}$;
    $h_i^t \leftarrow$ update $h_i^{t-1}$ with (5) for $E$ epoch;
    $h_{g,i}^t \leftarrow$ update $h_g^{t-1}$ with (7) for 1 epoch;
    Client $i$ uploads $f_{g,i}^t$ and $h_{g,i}^t$ to the server;
  **end**
  The server aggregate $f_{g,i}^t$;
  $f_g^t \leftarrow \frac{1}{\sum_i |D_i|} \sum_{i=1}^m |D_i| f_{g,i}^t$;
  Compute temporary variable $h'_g$;
  $h'_g \leftarrow \frac{1}{\sum_i |D_i|} \sum_{i=1}^m |D_i| h_{g,i}^t$;
  Use exponential moving average to get $h_g^t$;
  $h_g^t \leftarrow \tau h_g^{t-1} + (1-\tau) h'_g$;
**end**

## IV. Experimental Results

We perform experiments on multiple datasets with varying levels of data heterogeneity to evaluate the effectiveness of our algorithm. Our approach is compared with several other algorithms, including FedAvg [1], FedProx [2], and FedBabu [3], which concentrate on learning a global model. Additionally, we compared our method with personalized approaches, including FedPer [7] and FedRep [8].

### A. Experiment Settings

*1) Datasets and Models:* Our experiments are conducted on three datasets: MNIST, CIFAR-10, and CIFAR-100. For MNIST and CIFAR-10, we are using a CNN with four convolutional layers and one fully connected layer as the training model. For CIFAR-100, we use ResNet18 as the training model. For each dataset, all methods employ this model to conduct the experiments.

*2) Data Distribution:* To simulate the data heterogeneity environment, we used the Dirichlet distribution to sample data for each client. An allocation vector $q$ is obtained from the Dirichlet distribution, $q \sim Dir(\beta)$, where the $k$-th element of $q$ represents the proportion of the $k$-th class of data to all client data. When $\beta \to \infty$, it indicated a balanced distribution of data for each category among clients. When $\beta \to 0$, it meant increased heterogeneous distribution of data. We randomly split 80% of the client data as a training set

TABLE I
TEST ACCURACY ON MNIST, CIFAR-10, AND CIFAR-100 WITH DIFFERENT LEVELS OF DATA HETEROGENEITY

| Method | MNIST | | | | CIFAR-10 | | | | CIFAR-100 | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | $\beta = 0.1$ | $\beta = 0.5$ | $\beta = 1$ | IID | $\beta = 0.1$ | $\beta = 0.5$ | $\beta = 1$ | IID | $\beta = 0.1$ | $\beta = 0.5$ | $\beta = 1$ | IID |
| FedAvg [1] | 98.49 | 98.74 | 98.81 | 99.12 | 56.69 | 68.64 | 71.58 | 75.66 | 32.89 | 38.84 | 40.86 | 44.04 |
| FedProx [2] | 98.05 | 98.49 | 98.66 | 99.06 | 55.99 | 66.15 | 68.69 | 72.83 | 28.66 | 30.95 | 32.27 | 32.35 |
| FedBabu [3] | 98.22 | 98.52 | 98.72 | 99.13 | 63.52 | 71.27 | 72.69 | 76.09 | 39.46 | 43.53 | **45.69** | 45.99 |
| FedPer [7] | 97.05 | 98.47 | 98.85 | 99.14 | 12.57 | 63.29 | 70.64 | 75.03 | 26.25 | 35.79 | 38.71 | 41.5 |
| FedRep [8] | 98.47 | 98.10 | 98.08 | 97.76 | **85.57** | 70.60 | 65.21 | 55.9 | 47.76 | 26.26 | 20.70 | 9.43 |
| FedCRC | 98.37 | 98.58 | 98.79 | **99.17** | 64.61 | 70.75 | 72.77 | **76.41** | 40.00 | 43.30 | 44.73 | **47.66** |
| FedCRC-Per | **99.24** | **99.12** | **99.10** | 99.06 | 79.40 | **77.01** | **76.02** | 75.88 | **48.45** | **44.56** | 43.94 | 45.86 |

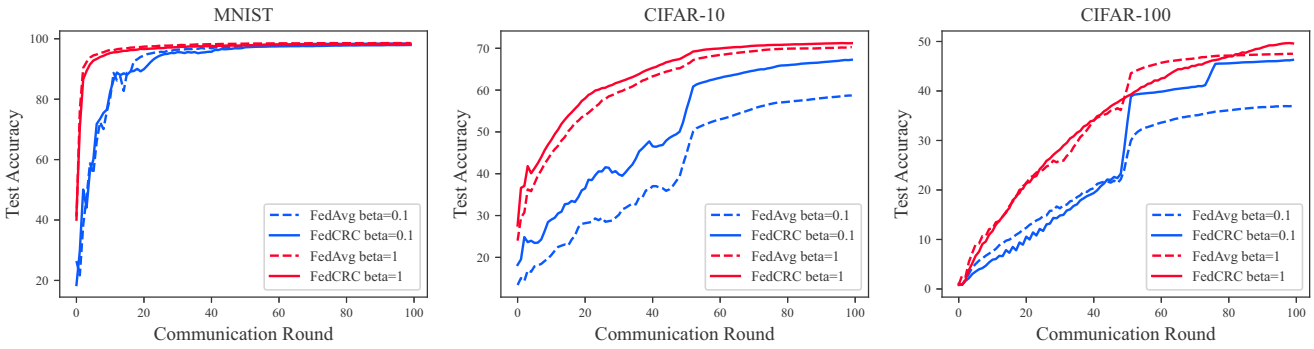


Fig. 3. Test accuracy comparison between FedCRC and FedAvg.

and the remaining 20% as a testing set. We created three data heterogeneity distributions with $\beta$ values of 0.1, 0.5, and 1 respectively, and IID data distribution.

*3) Federated Learning Setting:* We establish an FL environment consisting of 100 clients, with a participation probability of 0.1 for each client in every training round. We set the number of global communication rounds to 100, and each client performed 5 local update epochs. SGD with momentum is used as the model optimizer for all compared algorithms, with a momentum of 0.9 and an initial learning rate of 0.05. The learning rate decreases to 0.01 and 0.001 at the 50th and 75th communication rounds, respectively. The regular term weight in FedProx is set to 0.1, and the exponential moving average factor in our algorithm is set to 0.99 like the target network update setting in the self-supervised learning method [20]. Unless otherwise stated, these parameter settings are used for all experiments.

## B. Performance Analysis

We assess FedCRC's accuracy performance and compared it to other algorithms across multiple levels of data heterogeneity. The reported results in Table I show the average top-1 test accuracy across all clients. For methods that learn a single global model, we report the test results of the global model, while for personalized methods, we report the test results of the personalized model. FedCRC and FedCRC-Per respectively denote the test results of the global model and personalized model of our algorithm.

Our algorithm demonstrates superior accuracy compared to other algorithms across diverse datasets with varying data distributions. Moreover, both the global and personalized models in our algorithm exhibit excellent performance, indicating that training the personalized model does not compromise the performance of the global model. At the same time, the minimal cost is required to train two models simultaneously in our approach, specifically when training the personalized predictor.

## C. Convergence Speed

We assess the convergence speed of the FedCRC algorithm during the training process by measuring the test accuracy of FedCRC and FedAvg after each round of communication under two distinct data distributions. For this experiment, we set the client participation rate and the number of local updates to 1. Due to the limited space of the paper, we only present the experiment results of two algorithms. Our results, as shown in Fig. 3, indicate that FedCRC exhibits a faster improvement in accuracy than FedAvg under all data distributions, suggesting that our algorithm possesses superior convergence speed. It is worth noting that the sharp increase in accuracy in some specific rounds is due to the change in the learning rate.

## D. Generalization to New Clients

To evaluate the generalization performance of the FedCRC algorithm on new clients, we conduct an experiment to report the test accuracy of both FedCRC and the comparison algorithm on 40 clients who are not participating in the federated learning process. For personalized methods that do not train the global model, we aggregate the personalized models of the clients as the global model. We select 60 clients to participate in the federated learning process, and after the training is completed, we test the performance of the global model on the remaining 40 clients out of the total 100 clients.

TABLE II
CIFAR-10 TEST ACCURACY OF THE GLOBAL MODEL ON NEW CLIENTS

| Method | $\beta = 0.1$ | $\beta = 0.5$ | $\beta = 1$ |
|---|---|---|---|
| FedAvg | 45.47 | 61.58 | 61.74 |
| FedProx | 41.28 | 50.91 | 49.21 |
| FedBabu | 59.15 | **65.01** | 64.46 |
| FedPer | 14.25 | 55.85 | 58.30 |
| FedRep | 21.80 | 27.56 | 31.31 |
| FedCRC | **60.43** | 64.82 | **64.62** |

Table II presents the test results, highlighting the superior generalization performance of FedCRC compared to other algorithms. Conversely, the personalized method exhibits weaker generalization performance in comparison to training a global model. Notably, the level of data heterogeneity may also impact the generalization performance of the global model.

### *E. Effect of Exponential Moving Average Aggregation Factor*

To assess the impact of the exponential moving average factor on FedCRC, we examine the performance with various values of $\tau$. Since the exponential moving average process is only applicable to the global model, we conduct this experiment exclusively on the global model. Table III displays the effects of variations in the performance of our algorithm concerning the value of $\tau$. Notably, the performance of FedCRC progressively declines as $\tau$ decreases. To maintain a stable balance between global predictor updates and avoid abrupt changes, we adopt $\tau = 0.99$ for the aggregation of the global predictor, following a self-supervised learning approach [20].

TABLE III
CIFAR-10 TEST ACCURACY WITH DIFFERENT EXPONENTIAL MOVING AVERAGE FACTORS

| Weighting Factor | $\beta = 0.1$ | $\beta = 0.5$ | $\beta = 1$ | IID |
|---|---|---|---|---|
| $\tau = 0.99$ | **64.61** | **70.75** | **72.77** | **76.41** |
| $\tau = 0.9$ | 64.31 | 70.61 | 72.76 | 76.20 |
| $\tau = 0.5$ | 60.25 | 69.79 | 72.52 | 75.77 |
| $\tau = 0.2$ | 60.31 | 69.27 | 72.68 | 75.82 |

## V. CONCLUSIONS

This study presents FedCRC, a novel federated learning algorithm that tackles the challenge of data heterogeneity by decoupling the model into separate components. Our key insight is that the predictor can serve as a representation learning criterion for the extractor and is sensitive to varying data distributions. Based on this observation, we utilize a stable global predictor to train the representation extractor and a local predictor for each client to achieve personalized models. We also update the global predictor to ensure tight integration of the different model parts. Our experiments on multiple datasets with diverse levels of data heterogeneity demonstrate that FedCRC outperforms the compared algorithms. Future directions include improving our algorithms to cope with clients with dynamic data distributions and exploring the optimal division of the model.

## REFERENCES


[1] B. McMahan, E. Moore, D. Ramage *et al.*, "Communication-efficient learning of deep networks from decentralized data," in *Proceedings of Artificial Intelligence and Statistics (AISTATS)*, 2017, pp. 1273–1282.

[2] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated optimization in heterogeneous networks," *in Proceedings of Machine Learning and Systems (MLSys)*, vol. 2, pp. 429–450, 2020.

[3] J. Oh, S. Kim, and S.-Y. Yun, "FedBABU: Toward enhanced representation for federated image classification," in *Proceedings of International Conference on Learning Representations (ICLR)*, 2022.

[4] Y. Tan, G. Long, L. Liu, T. Zhou, Q. Lu, J. Jiang, and C. Zhang, "FedProto: Federated prototype learning across heterogeneous clients," in *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, vol. 36, no. 8, 2022, pp. 8432–8440.

[5] Q. Li, B. He, and D. Song, "Model-contrastive federated learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 10 713–10 722.

[6] V. Smith, C.-K. Chiang, M. Sanjabi, and A. S. Talwalkar, "Federated multi-task learning," *Advances in Neural Information Processing Systems*, vol. 30, 2017.

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

[8] L. Collins, H. Hassani, A. Mokhtari, and S. Shakkottai, "Exploiting shared representations for personalized federated learning," in *Proceedings of International Conference on Machine Learning (ICML)*. PMLR, 2021, pp. 2089–2099.

[9] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, pp. 1798–1828, 2013.

[10] A. Kendall, Y. Gal, and R. Cipolla, "Multi-task learning using uncertainty to weigh losses for scene geometry and semantics," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 7482–7491.

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

[12] S. P. Karimireddy, S. Kale, M. Mohri, S. Reddi, S. Stich, and A. T. Suresh, "Scaffold: Stochastic controlled averaging for federated learning," in *Proceedings of International Conference on Machine Learning (ICML)*. PMLR, 2020, pp. 5132–5143.

[13] X.-C. Li and D.-C. Zhan, "FedRS: Federated learning with restricted softmax for label distribution non-iid data," in *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, 2021, pp. 995–1005.

[14] L. Wang, S. Xu, X. Wang, and Q. Zhu, "Addressing class imbalance in federated learning," in *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, vol. 35, no. 11, 2021, pp. 10 165–10 173.

[15] M. Luo, F. Chen, D. Hu *et al.*, "No fear of heterogeneity: Classifier calibration for federated learning with Non-IID data," *Advances in Neural Information Processing Systems*, vol. 34, pp. 5972–5984, 2021.

[16] T. Li, S. Hu, A. Beirami, and V. Smith, "Ditto: Fair and robust federated learning through personalization," in *Proceedings of International Conference on Machine Learning (ICML)*, 2021, pp. 6357–6368.

[17] A. Ghosh, J. Chung, D. Yin, and K. Ramchandran, "An efficient framework for clustered federated learning," *Advances in Neural Information Processing Systems*, vol. 33, pp. 19 586–19 597, 2020.

[18] F. Sattler, K.-R. Müller, and W. Samek, "Clustered federated learning: Model-agnostic distributed multitask optimization under privacy constraints," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 8, pp. 3710–3722, 2020.

[19] Y. Ruan and C. Joe-Wong, "FedSoft: Soft clustered federated learning with proximal local updating," in *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, vol. 36, no. 7, 2022, pp. 8124–8131.

[20] J.-B. Grill, F. Strub, F. Altché *et al.*, "Bootstrap your own latent: A new approach to self-supervised learning," *Advances in Neural Information Processing Systems*, vol. 33, pp. 21 271–21 284, 2020.