FedFRR: Federated Forgetting-Resistant Representation Learning

1st Hui Wang
School of Computer Science and Engineering
Beihang University
Beijing, China
whui@buaa.edu.cn

2nd Jie Sun
Zhongguancun Laboratory
Beijing, China
sunjie@zgclab.edu.cn

3rd Tianyu Wo*
School of Software
Beihang University
Zhongguancun Laboratory
Beijing, China
woty@buaa.edu.cn

4th Xudong Liu
School of Computer Science and Engineering
Beihang University
Zhongguancun Laboratory
Beijing, China
liuxd@buaa.edu.cn

2024 IEEE International Conference on Multimedia and Expo (ICME) | 979-8-3503-9015-5/24/$31.00 ©2024 IEEE | DOI: 10.1109/ICME57554.2024.10687881

***Abstract*—Continuous learning faces the challenge of catastrophic forgetting. Our research findings indicate that in unsupervised federated continual learning (UFCL), the limited model capacity and interference among participants are the key factors contributing to this problem. Specifically, the fixed capacity of the model restricts its ability to retain historical knowledge. Besides, the indiscriminate aggregation of weights from multiple participants can cause interference, damaging the model memory. To address these challenges, we propose FedFRR, a federated anti-forgetting representation learning approach. FedFRR fits the participants' data distribution through a weighted combination of primary network units (PNU) in the model and optimizes model memory by adjusting the structure of PNUs. Additionally, FedFRR addresses interference by truncating the PNU with less weight change, thus reducing the scope of weight aggregation. The experimental results demonstrate that FedFRR achieves state-of-the-art performance, significantly enhancing the model's anti-forgetting ability.**

***Index Terms*—Federated Learning (FL), UFCL, Anti-forgetting, Weight truncation**

I. Introduction

Federated Continual Learning is a novel paradigm that embeds Continual Learning [26] into the Federated Learning (FL) framework [1]. It allows a participant's model to continually learn from its local data as well as the accumulated knowledge acquired by other participants, which is highly compatible with scenarios such as Edge Computing (EC) and the Internet of Things (IoT). In practical EC and IoT scenarios, models often need to incorporate unsupervised learning techniques due to the scarcity of labeled training data, which poses an urgent need for unsupervised federated continual learning (UFCL). Nonetheless, there are still significant challenges in implementing UFCL in practice. One such challenge is catastrophic forgetting, which refers to the phenomenon where the model gradually shifts its learned parameters from previous data towards the new data during optimization, resulting in the loss of previously acquired knowledge [15]. Although catastrophic forgetting has drawn some attention recently, prior art (e.g., [18], [35]) has primarily focused on the supervised learning scenario, leaving the problem under UFCL largely unexplored.

This paper focuses on the challenges posed by catastrophic forgetting in the realm of UFCL. To conduct a comprehensive study, we perform an experiment to validate the effect of catastrophic forgetting under a specific UFCL scenario. The detailed experimental setup can be found in the appendix. The result indicates that forgetting can cause an accuracy drop of up to 38.9%. In fact, a fixed model structure leads to limited model capacity, and "free" parameters in the model tend to saturate as more data distributions are introduced from participants [4], [26]. This may be an essential reason for the decline in model memory. Additionally, we find that the gradient aggregation of participants in the FL may also interfere, accelerating the forgetting of the distributions that have already been fitted by the model.

In previous research [24], it has been suggested that incorporating assumptions about participants' data distribution can enhance the performance of the model. Building upon this understanding, we propose a novel approach that considers each participant's data distribution as a composite of multiple unidentified fundamental distributions (FD). Our approach further utilizes various primary network units (PNU) within the model framework to effectively model these FDs. By adjusting the structure and quantity of the PNUs with weight gradient truncation, we can directly optimize the performance of the model's memory. Although the PNU change may bring additional costs, for example, adding PNU will increase computational costs, in our method, a controllable number of PNUs and more straightforward PNU structures can reduce these costs. In addition, with the advancements in processor

* Corresponding author

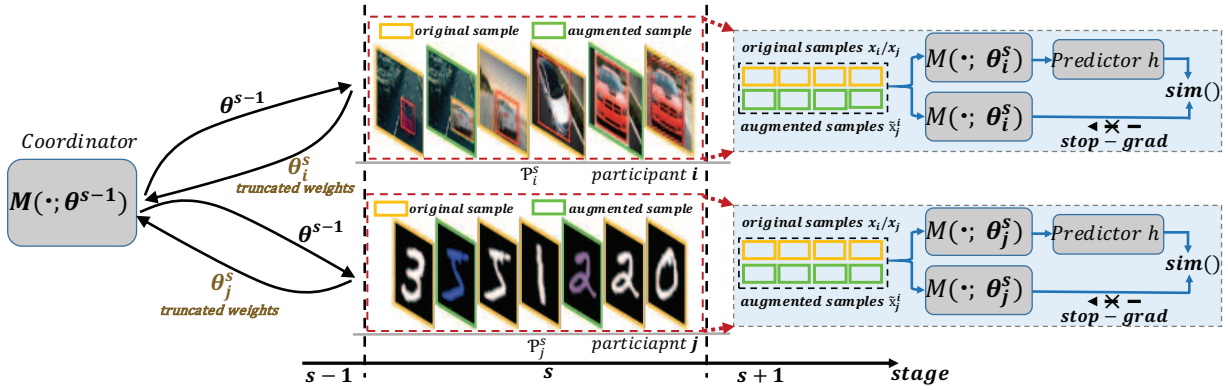


Fig. 1. FedFRR architecture. In stage s, the original and augmented sample pairs are processed by the model $\mathcal{M}(\cdot;\theta_p^s)$. Then a prediction MLP h is applied on one side, and a stop-gradient operation is applied on the other side. Model $\mathcal{M}(\cdot;\theta_p^s)$ maximizes the similarity between both sides.

technology and the availability of 5G networks, FL facilities' computation and communication capabilities have significantly improved [6]. Therefore, the costs introduced by FedFRR are acceptable in the current technological landscape. To the best of our knowledge, this paper is the first to focus on anti-forgetting in UFCL settings.

Contributions. (1) We conducted preliminary experiments, which revealed that forgetting can pose a threat to the performance of the UFCL model. **(2)** We propose a novel federated anti-forgetting representation learning method under UFCL settings, referred to as FedFRR. FedFRR utilizes distribution mixing and weight gradient truncation techniques to mitigate forgetting. **(3)** Through extensive experiments on both synthetic and real-world datasets, we demonstrate that FedFRR outperforms the state-of-the-art approaches. On average, FedFRR shows performance improvements of **7.8%** and **15.9%** in terms of model representation and anti-forgetting.

II. Related Work

Continual Learning (CL). Catastrophic Forgetting (CF) is one of the core challenges in CL, and existing CL methods can be divided into three categories based on technical details. The *regularization-based* methods [3], [25], [26] alleviate CF by adding constraints to the optimization process, such as regularizing objective or model weight. The *architecture-based* methods [4], [28], [29] often avoid CF problems by adjusting the model architecture (e.g., extending) or adding additional task-oriented parameters, and they have achieved significant results. The *rehearsal-based* methods [5], [27], [30] attempt to awaken the model's memory by replaying history or prompting mechanisms. However, the sources of CF in federated continual learning (FCL) are multifaceted and require multiple technologies to collaborate. For example, architecture-based methods can address the problem of limited model capacity, but the memory interference caused by weight aggregation in FCL is beyond the reach of the methods above.

Unsupervised Representation Learning (URL). URL methods can be divided into two categories. The *generative* methods [7] generate representations based on inputs. The *discriminative* methods [8], [9], [21] represented by contrastive learning learn representations by evaluating the similarity between positive and negative sample pairs. *SimSiam* [19] eliminates the need for large batches and negative samples in these methods and avoids model collapse by using a *Siamese* network that stops gradient operations. Meanwhile, such methods have been widely applied in anomaly detection [36], [37]. However, the representation performance of the above methods in FCL could be better due to limitations in sample size and optimization mode. We will incorporate a custom-designed, unsupervised semantic interpolation technique into *SimSiam* to compensate for its shortcomings.

III. Methodology

A. Problem Formulation

FedFRR focuses on a specific UFCL scenario that contains a global coordinator and P participants who continuously process streaming unlabeled samples $\{x\}_1^\infty, x \in \mathbb{R}^d$. We divide the sample streaming into different stages, the stage capacity recorded as C_p^s, the sample distribution of participant $p \in [1,...,P]$ in stage $s \in [1,...,S]$ is denoted as $\mathcal{P}_p^s$. To simplify the problem, we use a consistent stage capacity, i.e., $C_p^i = C_p^j, i,j \in [1,...,S]$. FedFRR uses contrastive learning [19] technique to extract the representation from participants' samples while improving the model anti-forgetting. Specifically, in stage s, let $\mathcal{M}(x;\theta_p^s)$: $\mathbb{R}^d \to \mathbb{R}^d$ parameterized by θ_p^s is the model of participant p mapping the $x \in \mathbb{R}^d$ to the representation $r \in \mathbb{R}^d$. We denote by R_{cl} the expected contrastive loss of participant p in s stage $R_{cl} = \mathbb{E}_{x\sim\mathcal{P}_p^s}[L_{cl}(\mathcal{M}(x;\theta_p^s))]$ and R_{af} the expected anti-forgetting loss $R_{af} = \mathbb{E}_{x\sim\mathcal{P}_p^s, x'\sim\mathcal{P}_p^k}\big[\sum_{k<s} L_{af}(\mathcal{M}(x;\theta_p^s), \mathcal{M}(x';\theta_p^k))\big]$. The optimization objective is formally defined as:

$$\arg\min_{\theta^S} \mathcal{L}(\theta^S) = \frac{1}{PS}\sum_{p=1}^{P}\sum_{s=1}^{S}\mathbb{E}_{x\sim\mathcal{P}_p^s, x'\sim\mathcal{P}_p^k}\big[L_{cl}(\mathcal{M}(x;\theta_p^s)) + \alpha\sum_{k<s} L_{af}(\mathcal{M}(x;\theta_p^s), \mathcal{M}(x';\theta_p^k))\big], \tag{2}$$

$$\frac{1}{PS}\sum_{p=1}^{P}\sum_{s=1}^{S}\left[\sum_{i,j,k=1}^{C}-\log\frac{\exp(sim(\mathcal{M}(x_i;\theta_p^s),\mathcal{M}(\tilde{x}_j^i;\theta_p^s))/\tau)}{\mathbb{1}_{[k\neq i]}\exp(sim(\mathcal{M}(x_i;\theta_p^s),\mathcal{M}(x_k;\theta_p^s))/\tau)+\exp(sim(\mathcal{M}(x_i;\theta_p^s),\mathcal{M}(\tilde{x}_j^i;\theta_p^s))/\tau)}+\alpha\sum_{m<s}KL\left(\mathcal{M}(\cdot;\theta_p^s),\ \mathcal{M}(\cdot;\theta_p^m)\right)\right] \tag{1}$$

where L_{cl} represents the unsupervised contrastive loss, L_{af} encourages the model $\mathcal{M}(\cdot;\theta_p^s)$ of participant p in the current stage s to have a similar performance to $\mathcal{M}(\cdot;\theta_p^k)$ in any historical stage k, i.e., maintaining memories in historical stages. α is the forgetting penalty coefficient.

B. Federated Model Architecture

FedFRR introduces a new formulation by assuming the distribution $\mathcal{P}_p^s$ of the participant p in stage s is a weighted mixture of U fundamental distributions (FD) $\{\tilde{\mathcal{P}}_u\}_{u=1}^U$, i.e., $\mathcal{P}_p^s=\sum_{u=1}^U\lambda_u\tilde{\mathcal{P}}_u$, where the mixing coefficients λ_u stand for the probabilities of sample in $\mathcal{P}_p^s$ coming from $\tilde{\mathcal{P}}_u$. Figure 2 shows the proposed model architecture. It comprises an input layer, an output layer, and multiple primary network units (PNU). All PNUs share the input, and the output is a weighted mixture of all PNUs' output. FedFRR does not predefine FDs but instead fits them through PNUs in optimization.

The multi-PNU architecture brings the following benefits: **(1)** The anti-forgetting ability of the model is positively correlated with its structure [15]. The more complex the structure (or more weights), the wider the distribution it can fit, i.e., the more knowledge it can learn. Compared to the regularization-based [25], rehearsal-based [35] methods, FedFRR can fundamentally improve the model memory by adding PNUs. **(2)** FedFRR can adjust the fitting ability of the PNU by changing its structure, thereby improving its representation performance. **(3)** FedFRR will truncate the PNU with small weight gradient changes in the optimization, reducing interference to the model memory.

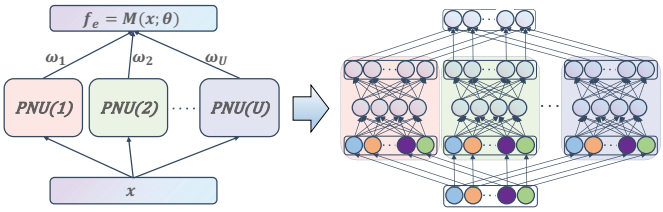


Fig. 2. Federated model architecture.

C. Contrastive Optimization of the Federated Model

As shown in Figure 1, the coordinator distributes the model $\mathcal{M}(\cdot;\theta^{s-1})$ to all participants at the beginning of stage s. Based on the *SimSiam* [19] method, participant p uses the local samples $\{x\}_1^{C_p^s}\sim\mathcal{P}_p^s$ to perform contrastive optimization on the local model $\mathcal{M}(\cdot;\theta_p^{s-1})$, and then uploads the weight gradient $\Delta\theta_p^{s-1}=\theta_p^{s-1}-\eta\mathcal{L}$ to the coordinator. The coordinator aggregates the gradients using the formula 3:

Algorithm 1 *FedFRR*

Input: P: participant number, S: stage number, C: stage capacity, T: weight truncated *PNU* number, A: augmented sample number in each stage, $\Gamma(\cdot)$: transform function, α: forgetting penalty coefficient, η: learning rate, $\mathcal{L}$: loss function.
Output: $\mathcal{M}(\theta^S)$.
Initializing the $\mathcal{M}(\theta^1)$, random $\lambda\sim Beta(8,8)$.

for $s=1$, ..., S **do**
  $\{\theta_p^s\}_{p=1}^P\leftarrow\theta^s$, distribute the model weights
  for $p=1$, ..., P **do**
    $\tilde{x}_j^i=\Gamma(\boldsymbol{BM}_\lambda\odot x_i)+(1-\boldsymbol{BM}_\lambda)\odot x_j,\ i,j\in[1...,C], i\neq j$, augmented samples in Formula 5
    optimizing θ_p^s with $\{x_i,x_j,\tilde{x}_j^i\}$, (Formula 1)
  end for
  $\theta^{s+1}\leftarrow\theta^s+\sum_{p=1}^P\Phi\left(\theta_p^s-\eta\mathcal{L}\right)$, (Formula 3)
end for
return θ^S

$$\theta^s\leftarrow\theta^{s-1}+\sum_{p=1}^{P}\Phi\left(\theta_p^{s-1}-\eta\mathcal{L}\right),\theta^s\triangleq\cup\left\{\varphi_u^s\mid u\in[1,...,U]\right\}, \tag{3}$$

where η and $\mathcal{L}$ denote the model learning rate and loss function. φ_u^s represents the weights of the *u*-th PNU in θ^s. $\Phi(\cdot)$ denotes an operation of zeroing the weights of the $T\in[0,...,U)$ PNUs with the slightest weight gradient change, i.e., weight truncation. FedFRR ignores the PNUs that have little effect on fitting the sample distribution through weight truncation, minimizing the scope of weight aggregation, as it may worsen the model memory. To accelerate the optimization, FedFRR uses semantic interpolation (see section III-D) to augment the sample in stage. Therefore, a participant in stage s includes original and augmented samples x and $\tilde{x}$, and the following criteria should be met to ensure the invariance [20] of the representation learned by the model $\mathcal{M}(\cdot;\theta_p^s)$.

$$P\left(\mathcal{P}_p^s\mid\mathcal{M}(\tilde{x}_j^i;\theta_p^s),\textit{SIAug}(x_i,x_j)\right)=P\left(\mathcal{P}_p^s\mid\mathcal{M}(x_i;\theta_p^s)\right)\cup P\left(\mathcal{P}_p^s\mid\mathcal{M}(x_j;\theta_p^s)\right),\forall x_i,x_j\sim\mathcal{P}_p^s, \tag{4}$$

where $\tilde{x}_j^i$ represents the augmented sample from x_i,x_j through operation $\textit{SIAug}(\cdot)$. FedFRR learns representation by minimizing the objective in Formula 1 over the samples x and $\tilde{x}$. P and S are the numbers of participants and stages. C is the stage capacity. $sim(\cdot)$ denotes the representation cosine similarity function, τ is the temperature and $\mathbb{1}_{[k\neq i]}=1$ if $k\neq i$. α is the weighting of the forgetting penalty. $KL(\cdot)$ is the Kullback-Leibler divergence [34]. In summary, our approach FedFRR can be summarized with Algorithm 1.

TABLE I
COMPARISON OF THE MODEL ACCURACY AND FORGETTING. WE USE THE MODEL COMPOSED OF RESNET-18 PNUS.

Method	MIXED		CMNIST		CCIFAR10		FFHQ		MiniImageNet	
	Acc	*Forgetting*	*Acc*	*Forgetting*	*Acc*	*Forgetting*	*Acc*	*Forgetting*	*Acc*	*Forgetting*
HLE	80.41 ± 0.59	10.30 ± 0.84	86.27 ± 0.39	09.12 ± 0.64	43.98 ± 0.91	06.14 ± 0.12	59.48 ± 0.76	08.20 ± 0.62	30.18 ± 0.26	06.20 ± 0.74
DER	79.49 ± 1.04	09.60 ± 1.14	84.48 ± 1.49	08.74 ± 1.62	48.84 ± 1.87	05.13 ± 0.35	58.91 ± 0.99	07.10 ± 0.23	29.38 ± 0.86	06.80 ± 1.72
SimSiam	78.73 ± 0.64	11.60 ± 1.26	88.84 ± 0.72	14.40 ± 1.65	50.81 ± 2.02	11.15 ± 2.15	60.28 ± 0.86	10.53 ± 1.53	33.68 ± 1.26	08.23 ± 2.73
RELIC	80.83 ± 1.38	12.30 ± 0.94	87.47 ± 0.86	13.01 ± 0.23	49.21 ± 1.12	09.14 ± 1.43	61.72 ± 0.16	09.21 ± 0.72	32.74 ± 0.73	09.11 ± 1.12
FedSimCLR	82.65 ± 2.04	13.20 ± 1.47	89.57 ± 0.61	11.12 ± 2.64	52.11 ± 1.32	08.61 ± 0.92	61.18 ± 1.76	11.33 ± 1.42	34.28 ± 0.66	10.22 ± 2.11
FedCA	79.63 ± 0.12	10.30 ± 0.96	88.74 ± 1.76	11.95 ± 1.23	51.71 ± 0.34	07.32 ± 1.23	60.48 ± 2.83	10.92 ± 1.53	35.28 ± 0.96	09.01 ± 1.34
FedWeIT	83.29 ± 1.39	07.20 ± 0.84	90.27 ± 1.03	07.14 ± 0.72	53.23 ± 1.87	04.82 ± 1.13	64.88 ± 0.96	05.73 ± 0.42	36.82 ± 1.76	05.11 ± 0.13
FedFRR	**86.79** ± 1.24	**04.90** ± 1.12	**91.62** ± 0.71	**04.22** ± 0.17	**54.78** ± 0.37	**04.01** ± 0.31	**66.08** ± 0.46	**05.43** ± 0.92	**39.52** ± 0.16	**03.32** ± 0.97

D. Semantic Interpolation Augmentation

The purer the distribution $\mathcal{P}_p^s$ of participant p in stage s, i.e., the more consistent the essential features of the samples in $\mathcal{P}_p^s$ are, the more beneficial it is to improve the fitting effect of PNU [24]. However, this may not be guaranteed effectively in an actual UFCL scenario. For instance, the images captured by a traffic camera at adjacent times may be vehicle and pedestrian. Therefore, FedFRR augments the samples in $\mathcal{P}_p^s$ using the semantic interpolation [31] technique. For the image samples, semantic interpolation captures the areas containing important information (i.e., areas strongly related to essential features) in the original samples through a Semantic Percent Map (SPM). Then it mixes the areas by cut-and-paste at symmetrical locations. The augmented sample retains the original sample's essential features. The semantically augmented sample $\tilde{x}_j^i$ from x_i and x_j can be expressed as:

$$\tilde{x}_j^i = \Gamma(\boldsymbol{BM}_\lambda \odot x_i) + (1 - \boldsymbol{BM}_\lambda) \odot x_j \ , \tag{5}$$

where $\odot$ denotes element-wise multiplication, the random value λ from a beta distribution Beta(β, β). The $\boldsymbol{BM}_\lambda$ denotes a binary mask containing a random square region whose area ratio to the original sample is λ. $\Gamma(\cdot)$ is a function that transforms (e.g., rotate, grayscale change) the cutout region of x_i and x_j to increase the diversity of the augmented sample $\tilde{x}_j^i$. FedFRR cuts out the high semantic value region $\boldsymbol{BM}_\lambda$ from x_i and pastes it into the same region of x_j, and vice versa. This technique strengthens the attention paid to the essential feature in the original sample. It avoids the noise caused by traditional augmentation methods, such as Mixup [32]. To improve the semantic information in the $\boldsymbol{BM}_\lambda$ region, FedFRR uses the following Modified-SPM (MSPM) to measure each original image pixel's semantic relatedness to the essential feature. For a given participant's image sample $x \in \mathbb{R}^{d \times h \times w}$, we denote $\boldsymbol{FM}_i^u(x)$ the i-th feature map in the last convolutional layer from the u-th PNU, and $\omega_u \in \mathbb{R}^d$ the u-th PNU weight corresponding to output feature in the Figure 2 (**Note:** Each element in the feature from the same PNU shares the weight connected to the output layer).

$$\boldsymbol{MSPM} = \Psi\left(\sum_{i=1}^{d}\left(\frac{1}{U}\sum_{u=1}^{U}\omega_u \boldsymbol{FM}_i^u(x)\right)\right) \ , \tag{6}$$

where $\Psi(\cdot)$ denotes an operation that upsamples the feature map to match dimensions with sample x and normalizes it. Unlike the original SPM [31], MSPM does not rely on supervisory signals, making it more suitable for UFCL scenarios.

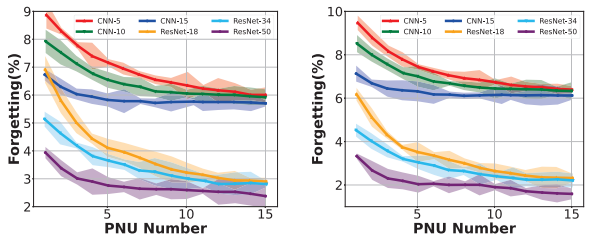
IV. EXPERIMENTS

Datasets and Baselines: We evaluate FedFRR using image classification datasets: MIXED [16], CMNIST [11], CCIFAR10 [14], FFHQ [12], and MiniImageNet [17]. Our baselines consist of the supervised group (HLE [35], DER [18]), unsupervised group (SimSiam [19], RELIC [20]), and federated group (FedSimCLR [21], FedCA [22], FedWeIT [23]). More details can be found in the appendix. **Implementation Details:** We use ResNet and multi-layer CNN for the PNU, the PNU number U = 10, participant number P = 100, the stage capacity C = 100, and the number of PNU with truncated weight $T = 0.1U$. In each stage, the augmented sample number $A = 0.5C$. The forgetting coefficient α = 0.5, and the learning rate η=0.01. The layout of PNUs includes: (1) *SI*: all PNUs have the same structure. (2) *MS*: The model's first and second half of PNUs are implemented in two different structures. (3) *MA*: two different structures alternately implement all PNUs. The metric ***Forgetting*** $= MAX\{ACC - ACC', 0\}$, ACC is the model accuracy after pre-training, and ACC' is the real-time accuracy in different stages.

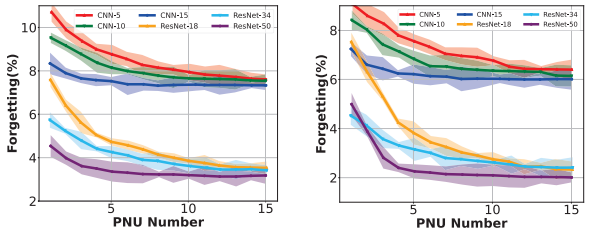
A. Quantitative Evaluation

Comparison of the Model Accuracy and Forgetting. The model accuracy and forgetting comparisons are shown in Table I. In general, FedFRR outperforms the three baseline groups by 8.6%, 8.8%, and 6.1% on average accuracy over all datasets, while forgetting has an average decrease of 11.6%, 20.8%, and 15.8%. The comparison indicates that under UFCL settings, distribution fitting and reducing the weight aggregation scope are effective ways to improve model representation and anti-forgetting performance simultaneously.

Evaluation on PNU Architecture. Figure 3 depicts the impact of the PNU architecture on model forgetting. For the ResNet-18/34/50 and CNN-5/10/15 PNUs, the more complex their structure, i.e., more parameters, the faster the forgetting decreases. Empirically, the ResNet or CNN PNUs may have a common forgetting lower bound. Figure 4 depicts the accuracy distribution of models composed of different PNUs. The

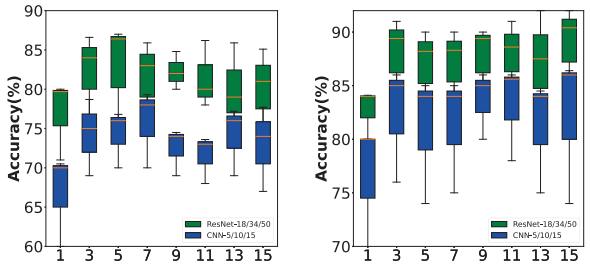


(a) MIXED (b) CMNIST

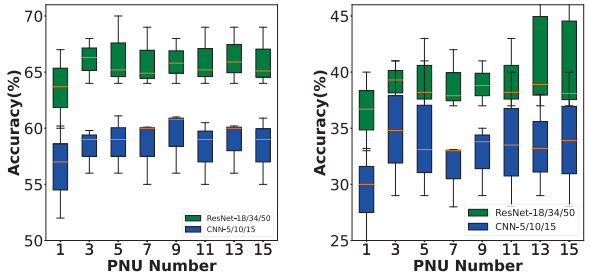


(c) FFHQ (d) MiniImageNet

Fig. 3. The impact of PNU on model forgetting.



(a) MIXED (b) CMNIST



(c) FFHQ (d) MiniImageNet

Fig. 4. The impact of PNU on model accuracy.

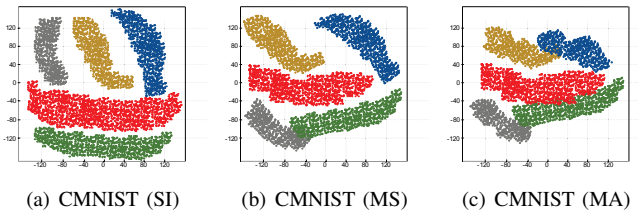
accuracy on the datasets increases by an average of 6.7%, 3.6%, 7.3%, and 4.5% as the PNU number from 1 to 3. The difference in accuracy between models composed of ResNet and CNN PNU indicates that ResNet PNU is more conducive to improving the model's representation ability. In addition, the accuracy distribution shows cohesion, obtaining a smaller Interquartile Range, indicating that the model's representation performance is not sensitive to structures within the same group, which increases the implementation space of PNU.

The Impact of Weight Truncation Ratio (WTR). Table II describes the impact of PNU layout and WTR on accuracy and forgetting. For different layouts, the accuracy is roughly the same, which means the accuracy is not sensitive to the PNU layout. With the increase of WTR, forgetting shows a decreasing trend and then increases, reaching its lowest at about WTR=0.1. The increase of WTR means the weight scope involved in aggregation decreases, reducing weight interference among participants. However, a further increase in WTR will result in the model losing more valuable weight information, leading to growth in the optimization stage. The results on more datasets can be found in the appendix.

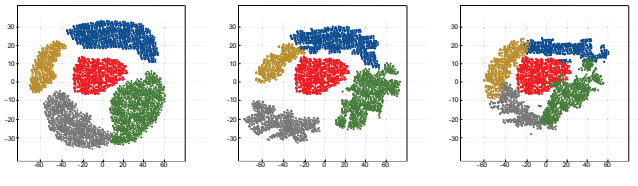
TABLE II
THE IMPACT OF WTR ON ACCURACY AND FORGETTING.

WTR	MIXED			CMNIST		
	Stage	*Acc*	*Forgetting*	*Stage*	*Acc*	*Forgetting*
			SI			
0.0	293	85.21 ± 1.91	07.12 ± 0.72	154	90.17 ± 1.74	10.41 ± 1.01
0.1	331	86.34 ± 0.13	**04.71** ± 1.51	195	91.46 ± 2.93	**05.74** ± 0.51
0.3	403	85.43 ± 1.81	08.71 ± 1.63	236	91.47 ± 1.72	07.37 ± 1.51
0.5	511	85.72 ± 0.71	07.72 ± 0.63	294	90.01 ± 0.47	11.41 ± 0.61
			MS			
0.0	302	86.31 ± 1.64	09.12 ± 1.71	172	89.71 ± 1.64	09.41 ± 2.72
0.1	356	86.71 ± 2.84	**03.91** ± 1.01	201	91.01 ± 2.01	**04.61** ± 1.31
0.3	420	85.11 ± 0.26	06.21 ± 1.41	224	90.45 ± 1.31	06.51 ± 2.31
0.5	541	84.81 ± 2.89	07.51 ± 0.41	284	89.81 ± 2.78	09.62 ± 0.73
			MA			
0.0	318	84.61 ± 0.12	07.61 ± 0.63	162	90.71 ± 1.41	09.74 ± 0.27
0.1	346	86.83 ± 1.71	**04.61** ± 1.31	185	89.01 ± 2.51	**05.81** ± 2.91
0.3	415	85.81 ± 1.61	05.64 ± 0.21	229	91.11 ± 0.13	08.76 ± 2.34
0.5	532	85.61 ± 2.10	06.73 ± 2.51	278	90.64 ± 1.14	12.61 ± 2.73

PNU Distribution Fitting Effect. Figure 5 depicts the decomposition effect of PNU on sample distribution through feature projection (using *t-SNE* algorithm [33]). The layout *SI* represents the model composed of four CNN-10 PNUs, while *MS* / *MA* represents two CNN-10 and two ResNet-18 PNUs. The output features from different PNUs in the model exhibit "high cohesion, low coupling", indicating that PNUs effectively decompose and fit the participant samples. In the *MA* layout, there is a slight overlap between the feature distributions, indicating a poor decoupling effect, which may be caused by the alternating arrangement of PNUs in the model.



(a) CMNIST (SI) (b) CMNIST (MS) (c) CMNIST (MA)



(d) FFHQ (SI) (e) FFHQ (MS) (f) FFHQ (MA)

Fig. 5. The distribution fitting effect of PNU. Red represents the output features of the model, while other colors represent the output features of different PNUs in the model.

Ablation Studies on Sample Augmentation. Table III describes the impact of different augmentation methods on the model accuracy. *Vanilla* represents random rotation or adding Gaussian noise to the sample. *Mixup* and *SIAug* denote the interpolation [10] and semantic augmentation for the random sample pairs of participants. Results show that the semantic augmentation for the essential features of the sample achieves state-of-the-art performance on all datasets.

TABLE III
ABLATION RESULTS OF SAMPLE AUGMENTATION METHODS.

Method	✗	*Vanilla*	*Mixup*	***SIAug(ours)***
MIXED	78.71 ± 1.21	80.12 ± 0.43	82.61 ± 1.91	**85.84** ± 0.71
CMNIST	84.61 ± 0.51	87.51 ± 1.43	86.71 ± 1.63	**91.01** ± 1.43
CCIFAR10	44.01 ± 2.72	47.71 ± 2.71	50.02 ± 1.82	**53.74** ± 1.34
FFHQ	57.51 ± 0.71	62.62 ± 1.94	60.71 ± 2.41	**64.71** ± 1.84
MiniImageNet	34.32 ± 2.51	36.91 ± 1.42	37.47 ± 1.72	**40.91** ± 2.03

V. CONCLUSION

We first formulate the model anti-forgetting representation learning problem under the UFCL setting and propose FedFRR, based on distribution fitting and weight gradient truncation techniques. The experiments show that FedFRR effectively improves the performance of model representation and anti-forgetting, outperforming the baseline methods.

VI. ACKNOWLEDGMENT

This work is supported by the National Science and Technology Major Project (2022ZD0120203).

REFERENCES

[1] B. McMahan, E. Moore, D. Ramage, and Hampson, "Communication-efficient learning of deep networks from decentralized data," in *Artificial intelligence and statistics*. PMLR, 2017, pp. 1273–1282.
[2] J. Lee, E. Kim, J. Lee, J. Lee, and J. Choo, "Learning debiased representation via disentangled feature augmentation," *Advances in Neural Information Processing Systems*, vol. 34, pp. 25 123–25 133, 2021.
[3] S.-W. Lee, J.-H. Kim, J. Jun *et al.*, "Overcoming catastrophic forgetting by incremental moment matching," *NeurIPS*, vol. 30, 2017.
[4] J. Yoon, E. Yang *et al.*, "Lifelong learning with dynamically expandable networks," in *ICLR*, 2018.
[5] M. Rostami, S. Kolouri, K. Kim, and E. Eaton, "Multi-agent distributed lifelong learning for collective knowledge acquisition," 2018.
[6] J. Pan and C.-C. Chang, "Towards collaborative intelligence: Routability estimation based on decentralized private data," in *Design Automation Conference*, 2022.
[7] A. Radford, "Unsupervised representation learning with deep convolutional gans," *Preprint*, 2015.
[8] S. Gidaris *et al.*, "Unsupervised representation learning by predicting image rotations," in *ICLR*, 2018.
[9] T. Wang and P. Isola, "Understanding contrastive representation learning through alignment and uniformity on the hypersphere," in *ICML*. PMLR, 2020, pp. 9929–9939.
[10] H. Zhang, M. Cisse *et al.*, "mixup: Beyond empirical risk minimization," in *ICLR*, 2018.
[11] Y. LeCun, L. Bottou *et al.*, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.
[12] T. Karras and S. Laine, "A style-based generator architecture for gans," in *CVPR*, 2019, pp. 4401–4410.
[13] A. Krizhevsky, "Learning multiple layers of features from tiny images," 2009.
[14] A. Krizhevsky, G. Hinton *et al.*, "Learning multiple layers of features from tiny images," 2009.
[15] J. Yoon, W. Jeong, G. Lee *et al.*, "Federated continual learning with weighted inter-client transfer," in *ICML*. PMLR, 2021, pp. 12 073–12 086.
[16] H.-W. Ng *et al.*, "A data-driven approach to cleaning large face datasets," in *ICIP*, 2014, pp. 343–347.
[17] J. Deng *et al.*, "Imagenet: A large-scale hierarchical image database," in *CVPR*. Ieee, 2009, pp. 248–255.
[18] P. Buzzega, M. Boschini *et al.*, "Dark experience for general continual learning: a strong, simple baseline," *NeurIPS*, vol. 33, pp. 15 920–15 930, 2020.
[19] X. Chen *et al.*, "Exploring simple siamese representation learning," in *CVPR*, 2021, pp. 15 750–15 758.
[20] J. Mitrovic *et al.*, "Representation learning via invariant causal mechanisms," in *ICLR*, 2020.
[21] T. Chen, S. Kornblith, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *ICML*. PMLR, 2020, pp. 1597–1607.
[22] F. Zhang *et al.*, "Federated unsupervised representation learning," *arXiv:2010.08982*, 2020.
[23] J. Yoon, W. Jeong, and S. J. Hwang, "Federated continual learning with weighted inter-client transfer," in *ICML*. PMLR, 2021, pp. 12 073–12 086.
[24] A. Zhong *et al.*, "Feddar: Federated domain-aware representation learning," *arXiv:2209.04007*, 2022.
[25] I. Mirzadeh and R. Pascanu, "Understanding the role of training regimes in continual learning," *NeurIPS*, vol. 33, pp. 7308–7320, 2020.
[26] J. Kirkpatrick, R. Pascanu, N. Rabinowitz *et al.*, "Overcoming catastrophic forgetting in neural networks," *Proceedings of the national academy of sciences*, vol. 114, no. 13, pp. 3521–3526, 2017.
[27] A. Chaudhry, M. Ranzato *et al.*, "Efficient lifelong learning with a-gem," in *ICLR*.
[28] J. Yoon and S. J. Hwang, "Scalable and order-robust continual learning with additive parameter decomposition," in *ICLR*. ICLR, 2020.
[29] J. Xu and Z. Zhu, "Reinforced continual learning," *NeurIIPS*, vol. 31, 2018.
[30] M. K. Titsias *et al.*, "Functional regularisation for continual learning with gaussian processes," in *ICLR*.
[31] S. Huang and X. Wang, "Snapmix: Semantically proportional mixing for augmenting fine-grained data," in *AAAI*, vol. 35, no. 2, 2021, pp. 1628–1636.
[32] H. Zhang, M. Cisse, and D. Lopez-Paz, "mixup: Beyond empirical risk minimization," in *ICLR*.
[33] L. Van der Maaten and G. Hinton, "Visualizing data using t-sne." *Journal of machine learning research*, vol. 9, no. 11, 2008.
[34] S. Kullback, *Information theory and statistics*. Courier Corporation, 1997.
[35] B. H. Lee *et al.*, "Online continual learning on hierarchical label expansion," in *ICCV*, 2023.
[36] R. Wang, C. Liu, X. Mou, K. Gao, X. Guo, P. Liu, T. Wo, and X. Liu, "Deep contrastive one-class time series anomaly detection," in *Proceedings of the 2023 SIAM International Conference on Data Mining (SDM)*. SIAM, 2023, pp. 694–702.
[37] X. Mou, R. Wang, T. Wang, J. Sun, B. Li, T. Wo, and X. Liu, "Deep autoencoding one-class time series anomaly detection," in *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023, pp. 1–5.