Federated Learning with Manifold Regularization and Normalized Update Reaggregation

Xuming An[1] **Li Shen**[2*] **Han Hu**[1*] **Yong Luo**[3]
[1] School of Information and Electronics, Beijing Institute of Technology, China
[2] JD Explore Academy, China [3] School of Computer Science, Wuhan University, China
{anxuming,hhu}@bit.edu.cn, mathshenli@gmail.com, luoyong@whu.edu.cn

Abstract

Federated Learning (FL) is an emerging collaborative machine learning framework where multiple clients train the global model without sharing their own datasets. In FL, the model inconsistency caused by the local data heterogeneity across clients results in the near-orthogonality of client updates, which leads to the global update norm reduction and slows down the convergence. Most previous works focus on eliminating the difference of parameters (or gradients) between the local and global models, which may fail to reflect the model inconsistency due to the complex structure of the machine learning model and the Euclidean space's limitation in meaningful geometric representations. In this paper, we propose FedMRUR by adopting the manifold model fusion scheme and a new global optimizer to alleviate the negative impacts. Concretely, FedMRUR adopts a hyperbolic graph manifold regularizer enforcing the representations of the data in the local and global models are close to each other in a low-dimensional subspace. Because the machine learning model has the graph structure, the distance in hyperbolic space can reflect the model bias better than the Euclidean distance. In this way, FedMRUR exploits the manifold structures of the representations to significantly reduce the model inconsistency. FedMRUR also aggregates the client updates norms as the global update norm, which can appropriately enlarge each client's contribution to the global update, thereby mitigating the norm reduction introduced by the near-orthogonality of client updates. Furthermore, we theoretically prove that our algorithm can achieve a linear speedup property $\mathcal{O}(\frac{1}{\sqrt{SKT}})$ for non-convex setting under partial client participation, where S is the participated clients number, K is the local interval and T is the total number of communication rounds. Experiments demonstrate that FedMRUR can achieve a new state-of-the-art (SOTA) accuracy with less communication.

1 Introduction

FL is a collaborative distributed framework where multiple clients jointly train the model with their private datasets [27, 28]. To protect privacy, each client is unable to access the other dataset [2]. A centralized server receives the parameters or gradients from the clients and updates the global model[46]. Due to the limited communication resource, only part of the clients is involved in the collaborative learning process and train the local model in multiple intervals with their own datasets within one communication round [23]. Due to the data heterogeneity, clients' partial participation and multiple local training yield severe model inconsistency, which leads to the divergences between the directions of the local updates from the clients and thus reduces the magnitude of global updates [17]. Therefore, the model inconsistency is the major source of performance degradation in FL [40, 15].

*Corresponding authors: Li Shen and Han Hu

37th Conference on Neural Information Processing Systems (NeurIPS 2023).

So far, numerous works have focused on the issues of model inconsistency to improve the performance of FL. Many of them [20, 16, 42, 1] utilize the parameter (or gradient) difference between the local and global model to assist the local training. By incorporating the global model information into local training, the bias between the local and global objectives can be diminished at some level. However, the parameter (or gradient) difference may fail to characterize the model bias due to the complex structure of modern machine learning model and the Euclidean space has limitations in providing powerful and meaningful geometric representations [10]. Meanwhile, incorporating the difference introduces extra high computation and communication costs because of the high-dimensional model parameter, which is common in the modern machine learning area[25]. Some other works [22, 39, 44, 24] exploit the permutation invariance property of the neurons in the neural networks to align and aggregate the model parameters for handling the model inconsistency issues, but the extra computation required for neuron alignment may slow down the speed of FL. In addition, Charles *et al.* [3] demonstrate that after multiple rounds, the similarities between the client updates approach zero and the local update direction are almost orthogonal to each other in FL. If the server aggregates the local updates, each client's contribution is little, which reduces the global update step. Therefore, we need to reduce the model inconsistency and compensate for the global norm reduction introduced by the near-orthogonality of client updates.

In order to alleviate the model inconsistency and compensate for the global reduction, we propose a practical and novel algorithm, dubbed as **FedMRUR** (**Fed**erated learning with **M**anifold **R**egularization and Normalized **U**pdate **R**eaggregation). FedMRUR adopts two techniques to achieve SOTA performance. i) Firstly, FedMRUR adopts the hyperbolic graph fusion technique to reduce the model inconsistency between the client and server within local training. The intuition is that adding the manifold regularizer to the loss function to constrain the divergence between the local and global models. Unlike the Euclidean space, the hyperbolic space is a manifold structure with the constant negative curvature, which has the ability to produce minimal distortion embedding[8] with the low storage constraints[30] for graph data. And the neural network, the most prevail machine learning model, has a graph structure[34, 24], we map the representations to the hyperbolic space and compute their distance to indicate the model bias precisely. Considering the numerical stability[6], we select the Lorentz model to describe the hyperbolic space and the squared Lorentzian distance[19] to indicate the representations' proximity. By adopting the hyperbolic graph fusion technique, FedMRUR can constrain model inconsistency efficiently. ii) Secondly, FedMRUR aggregates the client's local updates in a novel normalized way to alleviate the global norm reduction. In the normalized aggregation scheme, the server aggregates the local update norms as the global norm and normalizes the sum of the local updates as the global direction. Compared with directly aggregating local updates, the new aggregation scheme enables each customer's contribution to be raised from its projection on the global direction to its own size. As a result, the size of the global update becomes larger and compensates for the norm reduction introduced by model inconsistency, which improves the convergence and generalization performance of the FL framework.

Theoretically, we prove that the proposed FedMRUR can achieve the convergence rate of $\mathcal{O}(\frac{1}{\sqrt{SKT}})$ on the non-convex and L-smooth objective functions with heterogeneous datasets. Extensive experiments on CIFAR-10/100 and TinyImagenet show that our proposed FedMRUR algorithm achieves faster convergence speed and higher test accuracy in training deep neural networks for FL than several baselines including FedAvg, FedProx, SCAFFOLD, FedCM, FedExp, and MoFedSAM. We also study the impact on the performance of adopting the manifold regularization scheme and normalized aggregation scheme. In summary, the main contributions are as follows:

- We propose a novel and practical FL algorithm, FedMRUR, which adopts the hyperbolic graph fusion technique to effectively reduce the model inconsistency introduced by data heterogeneity, and a normalized aggregation scheme to compensate the global norm reduction due to the *near-orthogonality* of client updates, which achieves fast convergence and generalizes better.
- We provide the upper bound of the convergence rate under the smooth and non-convex cases and prove that FedMRUR has a linear speedup property $\mathcal{O}(\frac{1}{\sqrt{SKT}})$.
- We conduct extensive numerical studies on the CIFAR-10/100 and TinyImagenet dataset to verify the performance of FedMRUR, which outperforms several classical baselines on different data heterogeneity.

2 Related Work

McMahan *et al.* [27] propose the FL framework and the well-known algorithm, FedAvg, which has been proved to achieve a linear speedup property [43]. Within the FL framework, clients train local models and the server aggregates them to update the global model. Due to the heterogeneity among the local dataset, there are two issues deteriorating the performance: the model biases across the local solutions at the clients [20] and the similarity between the client updates (which is also known as the *near-orthogonality* of client updates) [3], which needs a new aggregation scheme to solve. In this work, we focus on alleviating these two challenges to improve the convergence of the FL algorithms.

Model consistency. So far, numerous methods focus on dealing with the issue of model inconsistency in the FL framework. Li *et al.* [20] propose the FedProx algorithm utilizing the parameter difference between the local and global model as a prox-correction term to constrain the model bias during local training. Similar to [20], during local training, the dynamic regularizer in FedDyn [1] also utilizes the parameter difference to force the local solutions approaching the global solution. FedSMOO [36] utilizes a dynamic regularizer to make sure that the local optima approach the global objective. Karimireddy *et al.* [16] and Haddadpour *et al.* [9] mitigate the model inconsistency by tracking the gradient difference between the local and global side. Xu *et al.* [42] and Qu *et al.* [31] utilize a client-level momentum term incorporating global gradients to enhance the local training process. Sun *et al.* [37] estimates the global aggregation offset in the previous round and corrects the local drift through a momentum-like term to mitigate local over-fitting. Liu *et al.* [26] incorporate the weighted global gradient estimations as the inertial correction terms guiding the local training to enhance the model consistency. Charles *et al.* [4] demonstrate that the local learning rate decay scheme can achieve a balance between the model inconsistency and the convergence rate. Tan *et al.* [38] show that the local learning rate decay scheme is unable to reduce the model inconsistency when clients communicate with the server in an asynchronous way. Most methods alleviate the model inconsistency across the clients by making use of the parameter (or gradient) difference between the local and global model.

Aggregation scheme. There are numerous aggregation schemes applied on the server side for improving performance. Some works utilize classical optimization methods, such as SGD with momentum [45], and adaptive SGD [5], to design the new global optimizer for FL. For instance, FedAvgM [13, 35] and STEM [17] update the global model by combining the aggregated local updates and a momentum term. Reddi *et al.* [32] propose a federated optimization framework, where the server performs the adaptive SGD algorithms to update the global model. FedNova [41] normalizes the local updates and then aggregates them to eliminate the data and device heterogeneity. In addition, the permutation invariance property of the neurons in the neural networks is also applied for improving robustness to data heterogeneity. FedFTG [48] applies the data-free knowledge distillation method to fine-tune the global model in the server. FedMA [39] adopts the Bayesian optimization method to align and average the neurons in a layer-wise manner for a better global solution. Li *et al.* [22] propose Position-Aware Neurons (PANs) coupling neurons with their positions to align the neurons more precisely. Liu *et al.* [24] adopt the graph matching technique to perform model aggregation, which requires a large number of extra computing resources in the server. Many deep model fusion methods [21] are also applied in the research field of FL, such as model ensemble [47] and CVAE [12]. The aforementioned algorithms utilize the local parameters or gradients directly without considering the *near-orthogonality* of client updates, which may deteriorate the convergence performance of the FL framework.

The proposed method FedMRUR adopts the hyperbolic graph fusion technique to reduce the model inconsistency and a normalized update aggregation scheme to mitigate the norm reduction of the global update. Compared with the previous works, we utilize the squared Lorentzian distance of the features in the local and global model as the regularization term. This term can more precisely measure the model bias in the low-dimensional subspace. For the update aggregation at the server, FedMRUR averages the local updates norm as the global update norm, which achieves to alleviate the norm reduction introduced by the near-orthogonality of the client updates.

3 Methodology

In this section, we first formally describe the problem step for FL and then introduce the FedMRUR and the two novel hyperbolic graph fusion and normalized aggregation techniques in FedMRUR.

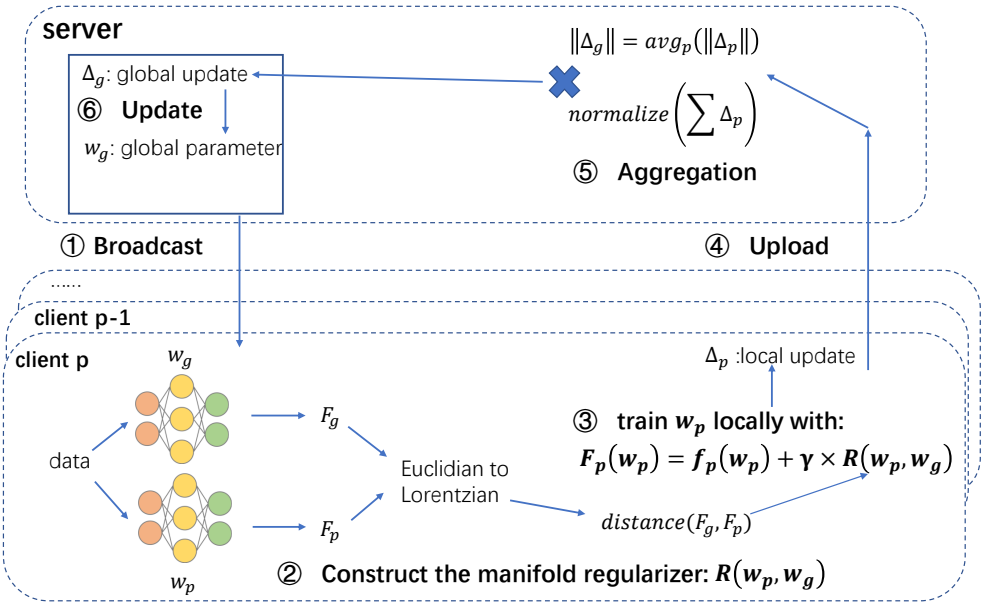


Figure 1: The workflow of FedMRUR. Once the global parameter x_0 is received, the client initializes the local model x_p and starts hyperbolic graph fusion. In the hyperbolic graph fusion, the client first takes the local and global model to get their representations and maps them into the hyperbolic space. Then, the client use their distances in the hyperbolic space as a regularizer to constrain model divergence. Next, the client performs local training and uploads the updates to the server. The server adopts the normalized scheme to aggregate the local updates and performs the global model update.

3.1 Problem setup

We consider collaboratively solving the stochastic non-convex optimization problem with P clients :

$$\min_w f(w) := \frac{1}{P}\sum_{p\in\mathcal{P}} f_p(w), \text{ with } f_p(w) := \mathbb{E}_{z\sim D_p}[l(w,z)], \tag{1}$$

where w is the machine learning model parameter and z is a data sample following the specified distribution D_p in client p; meanwhile $l(w,z)$ represents the model loss function evaluated by data z with parameter w. $f_p(w)$ and $f(w)$ indicate the local and global loss function, respectively. The loss function $l(w,z)$, $f_p(w)$, and $f(w)$ are non-convex due to the complex machine learning model. The heterogeneity of distribution D_p causes the model inconsistency across the clients, which may degrade the performance of the FL framework.

Notations. We define some notations to describe the proposed method conveniently. $\|\cdot\|$ denotes the spectral norm for a real symmetric matrix or L_2 norm for a vector. $\langle\cdot,\cdot\rangle$ denotes the inner product of two vectors. For any nature a, b, $a\wedge b$ and $a\vee b$ denote $\min\{a,b\}$ and $\max\{a,b\}$, respectively. The notation $O(\cdot)$, $\Theta(\cdot)$, and $\Omega(\cdot)$ are utilized to hide only absolute constants that don't depend on any problem parameter.

3.2 FedMRUR Algorithm

In this part, we describe our proposed FedMRUR algorithm (see Figure 1 and Algorithm 1) to mitigate the negative impacts of model inconsistency and improve performance. We add a manifold regularization term on the objective function to alleviate the model inconsistency. To eliminate the near-orthogonality of client updates, we design a new method to aggregate the local updates from the clients. Within one communication round, the server first broadcast the global model to the participating clients. During local training, the client takes the sampled data into the local and received global model and gets the representations. Then the client maps the representations into the hyperbolic space and computes their distance, which is used to measure the divergence between the local and global models. Next, the client adopts the distance as a manifold regular to constrain the model bias, achieving model fusion in the hyperbolic graph. After local training, the client uploads its local update to the server. The server aggregates the local update norms as the global update step

Algorithm 1 FedMRUR

Input: initial parameter w^0; momentum coefficient α; perturbation radius ρ; local interval K ; communication rounds T; set of selected clients S_t; global and local learning rate η_g, η_l.
Output: Global parameter $w^t, \forall t \in T$.
Initialization: Initialize $\triangle^0 = \mathbf{0}$ and w^0 as the global parameter at the server.
For $t = 0, 1, ..., T-1$ do:
  The server broadcasts parameter w^t and global update $\triangle^t$ to the selected clients S_t.
  For client $p \in S_t$ in parallel do:
  client p initialize the local parameter as $w_p^{t,0} = w^t$.
    For $k = 0, ..., K-1$ do:
      $\widetilde{w}_p^{t,k} = w_p^{t,k} + \rho \frac{\nabla F_p(w_p^{t,k})}{\|\nabla F_p(w_p^{t,k})\|}$.
      $v_i^{t,k+1} = \alpha \nabla F_p(\widetilde{w}_p^{t,k}) + (1-\alpha)\triangle^t$.
      $w_i^{t,k+1} = w_i^{t,k} - \eta_l v_i^{t,k+1}$.
    End for.
    $\triangle_p^t = w_p^{t,K} - w_p^{t,0}$
  End for
  Aggregate $\triangle^{t+1} = \frac{\sum_{p \in S_t} \|\triangle_p^t\|}{|S_t| \|\sum_{p \in S_t} \triangle_p^t\|} \sum_{i \in S_t} \triangle_p^t$.
  Update global parameter $w^{t+1} = w^t - \eta_g \triangle^{t+1}$.
End for.

and normalizes the sum of the local updates as the global update direction. Utilizing the normalized aggregation scheme, the server can update the model with a larger step and improve the convergence.

Hyperbolic Graph Fusion. In FL, the Euclidean distances between parameters[11, 20] (or gradients[16, 42]) between the client and the server is utilized to correct the local training for alleviating the model inconsistency. However, the Euclidean distance between the model parameters can't correctly reflect the variation in functionality due to the complex structure of the modern machine learning model. The model inconsistency across the clients is still large, which impairs the performance of the FL framework. Since the most prevail machine learning model, neural network has a graph structure and the hyperbolic space exhibits minimal distortion in describing data with graph structure, the client maps the representations of the local and global model into the hyperbolic shallow space[29] and uses the squared Lorentzian distance[19] between the representations to measure the model inconsistency.

To eliminate the model inconsistency effectively, we adopt the hyperbolic graph fusion technique, adding the distance of representations in the hyperbolic space as a regularization term to the loss function. Then the original problem (1) can be reformulated as:

$$\min_{w_0} F(w_0) = \frac{1}{P} \sum_p [f_p(w_p) + \gamma * R(w_p, w_g)], \quad s.t.\ w_g = \frac{1}{P} \sum_p w_p \tag{2}$$

where $R(w_p, w_g)$ is the hyperbolic graph fusion regularization term, defined as:

$$R(w_p, w_g) = \exp\left(\|L_p - L_g\|_{\mathcal{L}}^2 / \sigma\right), \quad \|L_p - L_g\|_{\mathcal{L}}^2 = -2\beta - 2\langle L_p, L_g \rangle_{\mathcal{L}}. \tag{3}$$

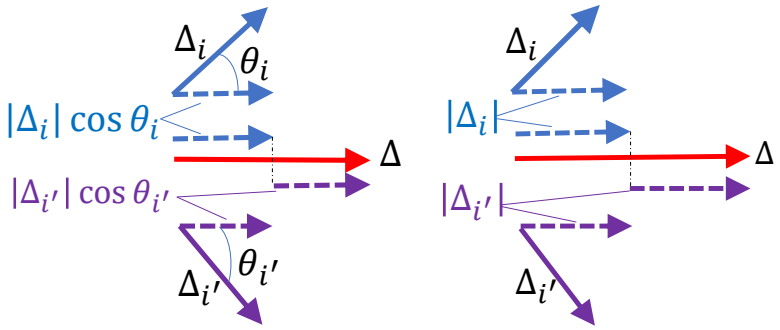
In (2) and (3), L_p and L_g are the mapped Lorentzian vectors corresponding to Z_p and Z_g, the representations from local model w_p and global model w_g. γ and σ are parameters to tune the impact of the model divergence on training process. β is the parameter of the Lorentz model and $\langle x, y \rangle_{\mathcal{L}}$ denotes the Lorentzian scalar product defined as:

$$\langle x, y \rangle_{\mathcal{L}} = -x_0 \cdot y_0 + \sum_{i=1}^{d} x_i \cdot y_i, \tag{4}$$

where x and y are $d+1$ dimensional mapped Lorentzian vectors. The new problem (2) can be divided into each client and client p uses its local optimizer to solve the following sub-problem:

$$\min_{w_p} F_p(w_p) = f_p(w_p) + \gamma * R(w_p, w_g). \tag{5}$$

The regularization term $R(w_p, w_g)$ has two benefits for local training: (1) It mitigates the local over-fitting by constraining the local representation to be closer to the global representation in the hyperbolic space (Lorentzian model); (2) It adopts representation distances in a low-dimensional hyperbolic space to measure model deviation, which can be more precisely and save computation.



(a) Vanilla Aggregation. (b) Normalized Aggregation.

Figure 2: A toy schematic to compare the naive aggregation and normalized aggregation of the local updates, where the number of clients is 2 and the local intervals are set as 1. The solid line indicates the client's local update $\triangle_i$, θ_i is the angle between the local update and the global update, and the dotted line represents the clients' contribution on the global update. The red lines are the aggregated global update $\triangle$. The main difference is $\|\triangle\|$, the norm of the global update. When adopting the naive aggregation method, the global norm $\|\triangle\| = \sum_i \|\triangle_i\| \cos \theta_i$. When adopting the normalized aggregation method, the global norm $\|\triangle\| = \sum_i \|\triangle_i\|$. We can see that the norm of global update in the direct aggregation is less than the norm in the normalized aggregation, due to the fact that $\cos \theta \leq 1$.

Normalized Aggregation of Local Updates. According to [3], after a number of communication rounds, the cosine similarities of the local updates across clients are almost zero, which comes from the model(gradient) inconsistency between the server and client sides. In the vanilla aggregation way, the server takes the mean of local updates from participated clients as the global gradient. As shown in Figure 2(a), client i makes $\|\triangle_i\| \cos \theta_i$ contribution on the aggregation result. When the data heterogeneous is significant, the cosine similarities across the local updates are small. Correspondingly, the cosine similarities $\cos \theta_i$ between the clients and the server narrows down, and the global update norm shrinks which slows down the convergence. To alleviate the negative impact of near-orthogonal local updates, we propose a new normalized aggregation method to compute the global update. The direction of the global update can be acquired by normalizing the sum of the local updates and the result is the same as the one obtained by the vanilla way. For the norm, the server computes it by taking the average of norms of the received local updates. As shown in Figure 2(b), with the proposed normalized aggregation method, the client i's contribution on the global update increases from $\|\triangle_i\| \cos \theta_i$ to $\|\triangle_i\|$. Accordingly, the norm of the global update $\|\triangle\|$ grows and accelerates the convergence.

Our proposed FedMRUR is characterized by Figure 1 and the detailed training process is summarized in Algorithm 1. Concretely, firstly, the server broadcasts the global parameter and updates it to the selected clients S_t. At the initialization stage of local training, client i utilizes the manifold regularization scheme to construct its own local loss function F_i with the received global parameter w^t. Then, client i adopts the Sharpness Aware Minimization (SAM) [7] optimizer to compute the gradient $\widetilde{g}_i$ with data sampled randomly. The local updater v_i consists of the stochastic gradient $\alpha \widetilde{g}_i$ and the momentum term $(1-\alpha)\triangle^t$, the received global update from the last round. Client i applies v_i to perform multiple SGD and uploads the accumulated local update $\triangle_i^t$ to the server. The server takes two steps to construct the global update: 1) aggregating and normalizing the accumulated local updates from the participated clients S_t as the direction of the global update; 2) averaging the norms of accumulated local updates as the norm of the global update. Finally, the server utilizes the constructed global update to perform one step SGD and get a new global parameter.

Remark 1. *FedMRUR is on the top of MoFedSAM [31] due to its excellent performance and our method can also be integrated with other federated learning methods, including FedExp, FedCM, SCAFFOLD, FedDYN, etc., to improve the performance.*

4 Convergence Analysis

In this section, we provide the theoretical analysis of our proposed FedMRUR for general non-convex FL setting. Due to space limitations, the detailed proofs are placed in **Appendix**. Before introducing the convergence results, we first state some commonly used assumptions as follows.

Assumption 1. *$f_p(x)$ is L-smooth and $R(x, x_0)$ is r-smooth with fixed x_0 for all client p, i.e.,*

$$\|\nabla f_p(a) - \nabla f_p(b)\| \leq L \|a - b\|, \quad \|\nabla R(a, x_0) - \nabla R(b, x_0)\| \leq r \|a - b\|.$$

Assumption 2. *The stochastic gradient $g_p^{t,k}$ with the randomly sampled data on the local client p is an unbiased estimator of $\nabla F_p(x_p^{t,k})$ with bounded variance, i.e.,*

$$E[g_p^{t,k}] = \nabla F_p(x_p^{t,k}),\ E\left\|g_p^{t,k} - \nabla F_p(x_p^{t,k})\right\|^2 \leq \sigma_l^2$$

Assumption 3. *The dissimilarity of the dataset among the local clients is bounded by the local and global gradients, i.e.,*

$$E\left\|\nabla F_p(x) - \nabla F(x)\right\|^2 \leq \sigma_g^2 \tag{6}$$

Assumption 1 guarantees the gradient Lipschitz continuity for the objective function and regularizer term. Assumption 2 guarantees the stochastic gradient is bounded by zero mean and constant variance. Assumption 3 gives the heterogeneity bound for the non-iid dataset across clients. All the above assumptions are widely used in many classical studies [1, 43, 33, 42, 14, 16], and our convergence analysis depends on them to study the properties of the proposed method.

Proof sketch. To explore the essential insights of the proposed FedMRUR, we first bound the client drift over all clients within the t-th communication round. Next, we characterize the global parameter moving within a communication round, which is similar to the one in centralized machine learning algorithms with momentum acceleration technology. Then, the upper bound for the global update $\triangle_t$ is provided. Lastly, we use $\|\nabla F(x_t)\|$ the global gradient norm as the metric of the convergence analysis of FedMRUR. The next theorem characterizes the convergence rate for FedMRUR.

Theorem 1. *Let all the assumptions hold and with partial client participation. If $\eta_l \leq \frac{1}{\sqrt{30\alpha KL}}$, $\eta_g \leq \frac{S}{2\alpha L(S-1)}$ satisfying $\frac{3}{4} - \frac{2(1-\alpha L)}{KN} - 70(1-\alpha)K^2(L+r)^2\eta_l^2 - \frac{90\alpha(L+r)^3\eta_g\eta_l^2}{S} - \frac{3\alpha(L+r)\eta_g}{2S}$, then for all $K \geq 0$ and $T \geq 1$, we have:*

$$\frac{1}{\sum_{t=1}^T d_t}\sum_{t=1}^T \mathbb{E}\left\|\nabla F(w^t)\right\|^2 d_t \leq \frac{F^0 - F^*}{C\alpha\eta_g \sum_{t=1}^T d_t} + \Phi, \tag{7}$$

where

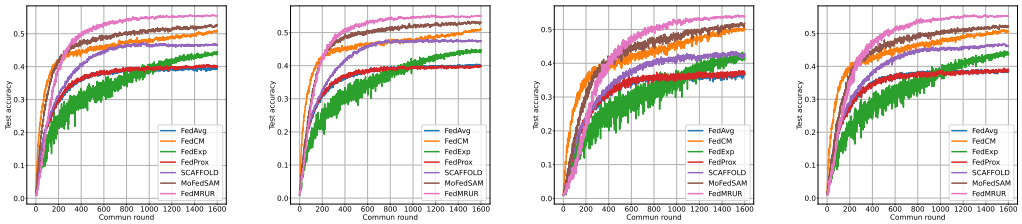
$$\begin{aligned}\Phi =& \frac{1}{C}\left[10\alpha^2(L+r)^4\eta_l^2\rho^2\sigma_l^2 + 35\alpha^2K(L+r)^2\eta_l^2 3(\sigma_g^2 + 6(L+r)^2\rho^2) + 28\alpha^2K^3(L+r)^6\eta_l^4\rho^2\right.\\ &+2K^2L^4\eta_l^2\rho^2 + \frac{\alpha(L+r)^3\eta_g^2\rho^2}{2KS}\sigma_l^2 + \frac{\alpha(L+r)\eta_g}{K^2SN}(30NK^2(L+r)^4\eta_l^2\rho^2\sigma_l^2\\ &+270NK^3(L+r)^2\eta_l^2\sigma_g^2 + 540NK^2(L+r)^4\eta_l^2\rho^2 + 72K^4(L+r)^6\eta_l^4\rho^2\\ &\left.+6NK^4(L+r)^2\eta_l^2\rho^2 + 4NK^2\sigma_g^2 + 3NK^2(L+r)^2\rho^2)\right].\end{aligned}$$

and $d_t = \frac{\sum_i \|\triangle_i^t\|}{\|\sum_i \triangle_i^t\|} \geq 1$. Specifically, we set $\eta_g = \Theta(\frac{\sqrt{SK}}{\sqrt{T}})$ and $\eta_l = \Theta(\frac{1}{\sqrt{ST}K(L+r)})$, the convergence rate of the FedMRUR under partial client participation can be bounded as:

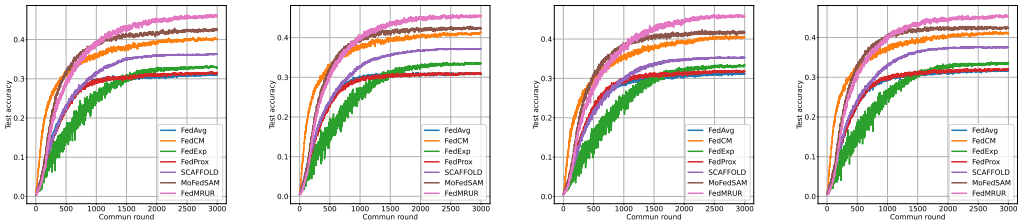
$$\sum_{t=1}^T E\left\|\nabla F(x_t)\right\|^2 = O\left(\frac{1}{\sqrt{SKT}}\right) + O\left(\frac{\sqrt{K}}{ST}\right) + O\left(\frac{1}{\sqrt{KT}}\right). \tag{8}$$

Remark 2. *Compared with the inequality $\frac{F^0 - F^*}{C\alpha\eta_g T} + \Phi$ of Theorem D.7 in MoFedSAM paper[31], the second constant term in (7) is same and the first term is less than the first term in MoFedSAM paper, which validates FedMRUR achieves faster convergence than MoFedSAM.*

Remark 3. *From (8), we can find that when T is large enough, the dominant term $O(\frac{1}{\sqrt{SKT}})$ in the bound achieves a linear speedup property with respect to the number of clients. It means that to achieve $\epsilon-$precision, there are $O(\frac{1}{SK\epsilon^2})$ communication rounds required at least for non-convex and L-smooth objective functions.*



(a) Test accuracy on CIFAR-100 in the non-iid ($\mu = 0.3$, $\mu = 0.6$, $n = 10$ and $n = 20$) settings.



(b) Test accuracy on TinyImageNet in the non-iid ($\mu = 0.3$, $\mu = 0.6$, $n = 40$ and $n = 80$) settings.

Figure 3: Test accuracy w.r.t. communication rounds of our proposed method and other approaches. Each method performs in 1600 communication rounds. To compare them fairly, the basic optimizers are trained with the same hyperparameters.

5 Experiments

In this section, we validate the effectiveness of the proposed FedMRUR algorithm using the experimental results on CIFAR-10/100 [18] and TinyImageNet [18]. We demonstrate that FedMRUR outperforms the vanilla FL baselines under heterogeneous settings. We also present that both manifold regularization and the proposed normalized update aggregation can improve the performance of SGD in FL. The experiments of CIFAR-10 are placed in the **Appendix.**

5.1 Experimental Setup

Datasets. We compare the performance of FL algorithms on CIFAR-10/100 and TinyImageNet datasets with 100 clients. The CIFAR-10 dataset consists of $50K$ training images and $10K$ testing images. All the images are with 32×32 resolution belonging to 10 categories. In the CIFAR-100 dataset, there are 100 categories of images with the same format as CIFAR-10. TinyImageNet includes 200 categories of $100K$ training images and $10K$ testing images, whose resolutions are 64×64. For non-iid dataset partitioning over clients, we use Pathological-n (abbreviated as Path(n)) and Dirichlet-μ (abbreviated as Dir(μ)) sampling as [13], where the coefficient n is the number of data categories on each client and μ measures the heterogeneity. In the experiments, we select the Dirichlet coefficient μ from $\{0.3, 0.6\}$ for all datasets and set the number of categories coefficient n from $\{3, 6\}$ on CIFAR-10, $\{10, 20\}$ on CIFAR-100 and $\{40, 80\}$ on TinyImageNet.

Implementation Details. For all algorithms on all datasets, following [1, 42], the local and global learning rates are set as 0.1 and 1.0, the learning rate decay is set as 0.998 per communication round and the weight decay is set as 5×10^{-4}. ResNet-18 together with group normalization is adopted as the backbone to train the model. The clients' settings for different tasks are summarized in Table 1. Other optimizer hyperparameters are as follow: $\rho = 0.5$ for SAM, $\alpha = 0.1$ for client momentum, $\gamma = 0.005$, $\sigma = 10000.0$ and $\beta = 1$ for manifold regularization.

Table 1: The experiments settings for different tasks.

Task	num of clients	participated ratio	batch size	local epoch
CIFAR	200	0.05	50	3
Tiny	500	0.02	20	2

Baselines. To compare the performances fairly, the random seeds are fixed. We compare the proposed FedMRUR with several competitive benchmarks: FedAvg [43], the most widely used baseline,

Table 2: Test accuracy (%) on CIFAR-100& TinyImagenet datasets in both Dir(μ) and Path(n)) distributions.

Algorithm	CIFAR-100				TinyImagenet			
	Dir(μ))		Path(n)		Dir(μ))		Path(n)	
	μ = 0.6	μ = 0.3	n = 20	n = 10	μ = 0.6	μ = 0.3	n = 80	n = 40
FedAvg	39.87	39.50	38.47	36.67	30.78	30.64	31.62	31.18
FedExp	44.51	44.26	43.58	41.00	33.49	32.68	33.65	33.39
FedProx	39.89	39.86	38.82	37.15	30.93	31.05	32.09	31.77
SCAFFOLD	47.51	46.47	46.23	42.45	37.14	36.22	37.48	35.32
FedCM	51.01	50.93	50.58	50.03	41.37	40.21	40.93	40.46
MoFedSAM	52.96	52.81	52.32	51.87	42.36	42.29	42.52	41.58
FedMRUR	55.81	55.49	55.21	53.69	45.54	45.41	45.42	45.71

Table 3: Convergence speed comparison on CIFAR100& TinyImageNet datasets. "Acc." represents the target test accuracy on the dataset. "∞" means that the algorithm is unable to achieve the target accuracy on the dataset.

Datasets	CIFAR-100					TinyImageNet				
Algorithms	Acc.	Dir(μ)		Path(n)		Acc.	Dir(μ)		Path(n)	
		0.6	0.3	20	10		0.6	0.3	80	40
FedAvg		513	494	655	∞		972	1078	1002	1176
FedExp		715	782	795	1076		1255	1362	1327	1439
FedProx		480	488	638	∞		1043	1030	1163	1615
SCAFFOLD	38%	301	322	389	585	30%	785	850	766	967
FedCM		120	126	157	255		342	401	366	474
MoFedSAM		154	146	211	300		436	447	415	460
Our		157	179	223	341		473	517	470	570
FedAvg		∞	∞	∞	∞		∞	∞	∞	∞
FedExp		985	1144	1132	1382		∞	∞	∞	∞
FedProx		∞	∞	∞	∞		∞	∞	∞	∞
SCAFFOLD	42%	406	449	558	998	35%	1289	1444	1206	2064
FedCM		173	193	260	527		599	735	674	879
MoFedSAM		197	192	260	392		598	624	583	685
Our		192	230	266	424		671	707	653	788
FedAvg		∞	∞	∞	∞		∞	∞	∞	∞
FedExp		∞	∞	∞	∞		∞	∞	∞	∞
FedProx		∞	∞	∞	∞		∞	∞	∞	∞
SCAFFOLD	45%	521	616	784	∞	40%	∞	∞	∞	∞
FedCM		276	353	470	842		1451	2173	1587	2186
MoFedSAM		243	278	400	575		950	1106	959	1162
Our		241	263	338	484		948	1050	953	1069

firstly applies local multiple training and partial participation for FL framework; SCAFFOLD [16] utilizes the SVRG method to mitigate the client drift issue; FedProx [20] uses a proximal operator to tackle data heterogeneity; FedCM [42] incorporates the client-momentum term in local training to maintain the model consistency among clients; Based on FedCM, MoFedSAM [31] improves the generalization performance with local SAM [7] optimizer; FedExp [14] determines the server step size adaptively based on the local updates to achieve faster convergence.

5.2 Evaluation Results

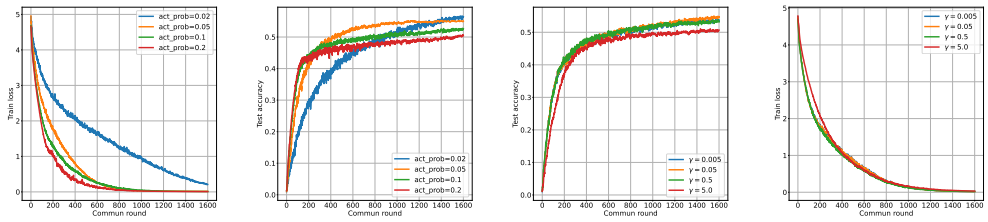
Figure 3 and Table 2 demonstrate the performance of ResNet-18 trained using multiple algorithms on CIFAR-100 and TinyImageNet datasets under four heterogeneous settings. We plot the test accuracy of the algorithms for a simple image classification task in the figure. We can observe that: our proposed FedMRUR performs well with good stability and effectively alleviates the negative impact of the model inconsistency. Specifically, on the CIFAR100 dataset, FedMRUR achieves 55.49% on the Dirichlet-0.3 setups, which is 5.07% higher than the second-best test performance. FedMRUR effectively reduces the model inconsistency and the enlarged global update improves the speed of convergence.

Table 3 depicts the convergence speed of multiple algorithms. From [13], a larger μ indicates less data heterogeneity across clients. We can observe that: 1) our proposed FedMRUR achieves the fastest convergence speed at most of the time, especially when the data heterogeneity is large. This

validates that FedMRUR can speed up iteration; 2) when the statistical heterogeneity is large, the proposed FedMRUR accelerates the convergence more effectively.

5.3 Ablation Study

Impact of partial participation. Figures 4(a) and 4(b) depict the optimization performance of the proposed FedMRUR with different client participation rates on CIFAR-100, where the dataset splitting method is Dirichlet sampling with coefficient $\mu = 0.3$ and the client participation ratios are chosen from 0.02 to 0.2. From this figure, we can observe that the client participation rate (PR) has a positive impact on the convergence speed, but the impact on test accuracy is little. Therefore, our method can work well under low PR settings especially when the communication resource is limited.



(a) (b) (c) (d)

Figure 4: (a). Training loss w.r.t different client participation rations; (b). Test accuracy w.r.t different client participation ratios. (c). Test accuracy with different γ. (d). Train loss with different γ. The performance of FedMRUR with different parameters on the CIFAR-100 dataset.

Hyperparameters Sensitivity. In Figures 4(c) and 4(d), we compare the performance of the proposed FedMRUR with different hyper-parameters on the CIFAR-100 dataset. From the results, we can see that our algorithm achieves similar test accuracy and training loss under different γ within a certain range ($\gamma \in [0.005, 0.5]$) and this indicates the proposed FedMRUR is insensitive to the hyperparameter γ. The hyperparameter γ represents the method the punishment on the model bias.

Impact of Each Component. Table 4 demonstrates the impact of each component of FedMRUR on the test accuracy for CIFAR-100 dataset on the Dirichlet-0.3 setups. For convenience, we abbreviate normalized aggregation as "normalized(N)" and hyperbolic graph fusion as "hyperbolic(H)", respectively. From the results, we can find that both the normalized local update aggregation scheme and the hyperbolic graph fusion can improve performance. This table validates that our algorithm design and theoretical analysis are correct and effective.

Table 4: Test accuracy % on CIFAR-100 datasets about without each ingredients of FedMRUR.

Algorithm	normalized(N)	hyperbolic(H)	Acc.
MoFedSAM	–	–	52.81
FedMRUR-N	✓	–	54.27
FedMRUR-H	–	✓	53.57
FedMRUR	✓	✓	55.49

6 Conclusion

In this work, we propose a novel and practical federated method, dubbed FedMRUR which applies the hyperbolic graph fusion technique to alleviate the model inconsistency in the local training stage and utilizes normalized updates aggregation scheme to compensate for the global norm reduction due to the *near-orthogonality* of the local updates. We provide the theoretical analysis to guarantee its convergence and prove that FedMRUR achieves a linear-speedup property of $O(\frac{1}{\sqrt{SKT}})$. We also conduct extensive experiments to validate the significant improvement and efficiency of our proposed FedMRUR, which is consistent with the properties of our analysis. This work inspires the FL framework design to focus on exploiting the manifold structure of the learning models.

Limitations&Broader Impacts. Our work focuses on the theory of federated optimization and proposes a novel FL algorithm. During the local training, the representations of the global model must be stored locally, which may bring extra pressure on the client. This will help us in inspiration for new algorithms. Since FL has wide applications in machine learning, Internet of Things, and UAV networks, our work may be useful in these areas.

Acknowledgements. This work is supported by National Key Research and Development Program of China under SQ2021YFC3300128, and National Natural Science Foundation of China under Grant 61971457. Thanks for the support from CENI-HEFEI and Laboratory for Future Networks in University of Science and Technology of China.

References

[1] Durmus Alp Emre Acar, Yue Zhao, Ramon Matas Navarro, Matthew Mattina, Paul N. Whatmough, and Venkatesh Saligrama. Federated learning based on dynamic regularization. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021.

[2] Keith Bonawitz, Hubert Eichner, Wolfgang Grieskamp, Dzmitry Huba, Alex Ingerman, Vladimir Ivanov, Chloe Kiddon, Jakub Konečnỳ, Stefano Mazzocchi, Brendan McMahan, et al. Towards federated learning at scale: System design. *Proceedings of machine learning and systems*, 1:374–388, 2019.

[3] Zachary Charles, Zachary Garrett, Zhouyuan Huo, Sergei Shmulyian, and Virginia Smith. On large-cohort training for federated learning. In *34th Advances in Neural Information Processing Systems, NeurIPS 2021, virtual, December 6-14, 2021*, pages 20461–20475, 2021.

[4] Zachary Charles and Jakub Konečnỳ. Convergence and accuracy trade-offs in federated learning and meta-learning. In *The 24th International Conference on Artificial Intelligence and Statistics, AISTATS 2021, April 13-15, 2021, Virtual Event*, pages 2575–2583. PMLR, 2021.

[5] Ashok Cutkosky and Róbert Busa-Fekete. Distributed stochastic optimization via adaptive sgd. *Advances in Neural Information Processing Systems*, 31, 2018.

[6] Shanshan Feng, Lisi Chen, Kaiqi Zhao, Wei Wei, Xuemeng Song, Shuo Shang, Panos Kalnis, and Ling Shao. Role: Rotated lorentzian graph embedding model for asymmetric proximity. *IEEE Transactions on Knowledge and Data Engineering*, 2022.

[7] Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur. Sharpness-aware minimization for efficiently improving generalization. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*, 2021.

[8] Mikhael Gromov. Hyperbolic groups. In *Essays in group theory*, pages 75–263. Springer, 1987.

[9] Farzin Haddadpour, Mohammad Mahdi Kamani, Aryan Mokhtari, and Mehrdad Mahdavi. Federated learning with compression: Unified analysis and sharp guarantees. In *The 24th International Conference on Artificial Intelligence and Statistics, AISTATS 2021, April 13-15, 2021, Virtual Event*, Proceedings of Machine Learning Research, pages 2350–2358. PMLR, 2021.

[10] William L Hamilton, Rex Ying, and Jure Leskovec. Representation learning on graphs: Methods and applications. *arXiv preprint arXiv:1709.05584*, 2017.

[11] Filip Hanzely and Peter Richtárik. Federated learning of a mixture of global and local models. *arXiv preprint arXiv:2002.05516*, 2020.

[12] Clare Elizabeth Heinbaugh, Emilio Luz-Ricca, and Huajie Shao. Data-free one-shot federated learning under very high statistical heterogeneity. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023.

[13] Tzu-Ming Harry Hsu, Hang Qi, and Matthew Brown. Measuring the effects of non-identical data distribution for federated visual classification. *arXiv preprint arXiv:1909.06335*, 2019.

[14] Divyansh Jhunjhunwala, Shiqiang Wang, and Gauri Joshi. Fedexp: Speeding up federated averaging via extrapolation. In *11-th International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*, 2023.

[15] Peter Kairouz, H Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kallista Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, et al. Advances and open problems in federated learning. *Foundations and Trends® in Machine Learning*, 14(1–2):1–210, 2021.

[16] Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank J. Reddi, Sebastian U. Stich, and Ananda Theertha Suresh. SCAFFOLD: stochastic controlled averaging for federated learning. In *Proc. 37th Int. Conf. Mach. Learn.*, pages 5132–5143, 2020.

[17] Prashant Khanduri, Pranay Sharma, Haibo Yang, Mingyi Hong, Jia Liu, Ketan Rajawat, and Pramod K. Varshney. STEM: A stochastic two-sided momentum algorithm achieving near-optimal sample and communication complexities for federated learning. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 6050–6061, 2021.

[18] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. *Technical report*, 2009.

[19] Marc Teva Law, Renjie Liao, Jake Snell, and Richard S. Zemel. Lorentzian distance learning for hyperbolic representations. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, Proceedings of Machine Learning Research, pages 3672–3681, 2019.

[20] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. In *Proc. Mach. Learn. Sys.*, 2020.

[21] Weishi Li, Yong Peng, Miao Zhang, Liang Ding, Han Hu, and Li Shen. Deep model fusion: A survey. 2023.

[22] Xin-Chun Li, Yi-Chu Xu, Shaoming Song, Bingshuai Li, Yinchuan Li, Yunfeng Shao, and De-Chuan Zhan. Federated learning with position-aware neurons. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pages 10072–10081. IEEE, 2022.

[23] Wei Yang Bryan Lim, Nguyen Cong Luong, Dinh Thai Hoang, Yutao Jiao, Ying-Chang Liang, Qiang Yang, Dusit Niyato, and Chunyan Miao. Federated learning in mobile edge networks: A comprehensive survey. *IEEE Communications Surveys & Tutorials*, 22(3):2031–2063, 2020.

[24] Chang Liu, Chenfei Lou, Runzhong Wang, Alan Yuhan Xi, Li Shen, and Junchi Yan. Deep neural network fusion via graph matching with applications to model ensemble and federated learning. In *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, pages 13857–13869. PMLR, 2022.

[25] Weibo Liu, Zidong Wang, Xiaohui Liu, Nianyin Zeng, Yurong Liu, and Fuad E Alsaadi. A survey of deep neural network architectures and their applications. *Neurocomputing*, 234:11–26, 2017.

[26] Yixing Liu, Yan Sun, Zhengtao Ding, Li Shen, Bo Liu, and Dacheng Tao. Enhance local consistency in federated learning: A multi-step inertial momentum approach. 2023.

[27] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, AISTATS 2017, 20-22 April 2017, Fort Lauderdale, FL USA*, Proceedings of Machine Learning Research, pages 1273–1282. PMLR, 2017.

[28] Dinh C Nguyen, Ming Ding, Pubudu N Pathirana, Aruna Seneviratne, Jun Li, and H Vincent Poor. Federated learning for internet of things: A comprehensive survey. *IEEE Communications Surveys & Tutorials*, 23(3):1622–1658, 2021.

[29] Maximilian Nickel and Douwe Kiela. Learning continuous hierarchies in the lorentz model of hyperbolic geometry. In Jennifer G. Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80, pages 3776–3785. PMLR, 2018.

[30] Wei Peng, Tuomas Varanka, Abdelrahman Mostafa, Henglin Shi, and Guoying Zhao. Hyperbolic deep neural networks: A survey. *IEEE Trans. Pattern Anal. Mach. Intell.*, 44(12):10023–10044, 2022.

[31] Zhe Qu, Xingyu Li, Rui Duan, Yao Liu, Bo Tang, and Zhuo Lu. Generalized federated learning via sharpness aware minimization. In *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, pages 18250–18280, 2022.

[32] Sashank Reddi, Zachary Charles, Manzil Zaheer, Zachary Garrett, Keith Rush, Jakub Konečnỳ, Sanjiv Kumar, and H Brendan McMahan. Adaptive federated optimization. *arXiv preprint arXiv:2003.00295*, 2020.

[33] Sashank J. Reddi, Zachary Charles, Manzil Zaheer, Zachary Garrett, Keith Rush, Jakub Konečný, Sanjiv Kumar, and Hugh Brendan McMahan. Adaptive federated optimization. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021.

[34] Sidak Pal Singh and Martin Jaggi. Model fusion via optimal transport. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

[35] Tao Sun, Dongsheng Li, and Bao Wang. Decentralized federated averaging. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

[36] Yan Sun, Li Shen, Shixiang Chen, Liang Ding, and Dacheng Tao. Dynamic regularized sharpness aware minimization in federated learning: Approaching global consistency and smooth landscape. In *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202, pages 32991–33013. PMLR, 2023.

[37] Yan Sun, Li Shen, Hao Sun, Liang Ding, and Dacheng Tao. Efficient federated learning via local adaptive amended optimizer with linear speedup. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, in press:1–12, 2023.

[38] Yue Tan, Guodong Long, Lu Liu, Tianyi Zhou, Qinghua Lu, Jing Jiang, and Chengqi Zhang. Fedproto: Federated prototype learning across heterogeneous clients. In *Proceedings of the 36th AAAI Conference on Artificial Intelligence*, pages 8432–8440, 2022.

[39] Hongyi Wang, Mikhail Yurochkin, Yuekai Sun, Dimitris S. Papailiopoulos, and Yasaman Khazaeni. Federated learning with matched averaging. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020.

[40] Jianyu Wang, Zachary Charles, Zheng Xu, Gauri Joshi, H Brendan McMahan, Maruan Al-Shedivat, Galen Andrew, Salman Avestimehr, Katharine Daly, Deepesh Data, et al. A field guide to federated optimization. *arXiv preprint arXiv:2107.06917*, 2021.

[41] Jianyu Wang, Qinghua Liu, Hao Liang, Gauri Joshi, and H. Vincent Poor. Tackling the objective inconsistency problem in heterogeneous federated optimization. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

[42] Jing Xu, Sen Wang, Liwei Wang, and Andrew Chi-Chih Yao. Fedcm: Federated learning with client-level momentum. *arXiv preprint arXiv:2106.10874*, 2021.

[43] Haibo Yang, Minghong Fang, and Jia Liu. Achieving linear speedup with partial worker participation in non-iid federated learning. In *9th Int. Conf. Learn. Representations*, 2021.

[44] Fuxun Yu, Weishan Zhang, Zhuwei Qin, Zirui Xu, Di Wang, Chenchen Liu, Zhi Tian, and Xiang Chen. Fed2: Feature-aligned federated learning. In *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining*, pages 2066–2074, 2021.

[45] Hao Yu, Rong Jin, and Sen Yang. On the linear speedup analysis of communication efficient momentum sgd for distributed non-convex optimization. In *International Conference on Machine Learning*, pages 7184–7193. PMLR, 2019.

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

[47] Jie Zhang, Chen Chen, Bo Li, Lingjuan Lyu, Shuang Wu, Shouhong Ding, Chunhua Shen, and Chao Wu. DENSE: data-free one-shot federated learning. In *NeurIPS*, 2022.

[48] Lin Zhang, Li Shen, Liang Ding, Dacheng Tao, and Ling-Yu Duan. Fine-tuning global model via data-free knowledge distillation for non-iid federated learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pages 10164–10173. IEEE, 2022.

In this part, we will introduce the proofs of the major theorems and some extra experiments. In Section A, we provide the full proofs of the major theorems. In section B, we provide some extra experiments on CIFAR-10 task.

A Proof for Convergence Analysis

In this section, we provide the convergence analysis for the proposed FedMRUR algorithm. Firstly, we state some preliminary lemmas as follows:

Lemma 2. *For random variables $x_1, ..., x_n$, we have*

$$\mathbb{E}\left[\|x_1 + ... + x_n\|^2\right] \leq n\mathbb{E}[\|x_1\|^2 + ... + \|x_n\|^2].$$

Lemma 3. *For independent, mean 0 random variables $x_1, ..., x_n$, we have*

$$\mathbb{E}\left[\|x_1 + ... + x_n\|^2\right] = \mathbb{E}[\|x_1\|^2 + ... + \|x_n\|^2]$$

Lemma 4. *The stochastic gradient $\nabla F_i(w, \xi_i)$ computed by the i-th client at model parameter w using minibatch ξ_i is an unbiased estimator of$\nabla F_i(w)$ with variance bounded by σ^2. The gradient of SAM is formulated by*

$$\mathbb{E}\left[\left\|\sum_{k=0}^{K-1} g_i^{t,k}\right\|^2\right] \leq K\sum_{k=0}^{K-1}\mathbb{E}\left[\left\|\nabla F_i(w_i^{t,k})\right\|^2\right] + \frac{K(L+r)^2\rho^2}{N}\sigma_l^2$$

Proof. we can bound the inequality as follows:

$$\begin{aligned}
&\mathbb{E}\left[\left\|\sum_{k=0}^{K-1} g_i^{t,k}\right\|^2\right] = \mathbb{E}\left[\left\|\sum_{k=0}^{K-1} \nabla F_i(w_i^{t,k})\right\|^2\right] + \mathbb{E}\left[\left\|\sum_{k=0}^{K-1} g_i^{t,k} - \nabla F_i(w_i^{t,k})\right\|^2\right] \\
\leq& K\sum_{k=0}^{K-1}\mathbb{E}\left[\left\|\nabla F_i(w_i^{t,k})\right\|^2\right] \qquad (9a) \\
&+ (L+r)^2\sum_{k=0}^{K-1}\mathbb{E}\left[\frac{1}{N}\sum_{i=1}^{N}\left(w_i^{t,k} + \delta_i^{t,k}(\tilde{w}_i^{t,k}; \xi_i^{t,k}) - w_i^{t,k} + \delta_i^{t,k}(\tilde{w}_i^{t,k})\right)\right] \\
\leq& K\sum_{k=0}^{K-1}\mathbb{E}\left[\left\|\nabla F_i(w_i^{t,k})\right\|^2\right] + \frac{K\rho^2\sigma_l^2}{N}(L+r)^2 \qquad (9b)
\end{aligned}$$

where (9a) is from Assumption 1 and (9b) is from Assumption 3 and Lemma 3. □

Lemma 5. *The variance of local and global gradients with perturbation can be bounded as follows:*

$$\|\nabla F_i(w + \delta_i) - \nabla F(w + \delta)\|^2 \leq 3\sigma_g^2 + 6(L+r)^2\rho^2.$$

Proof.

$$\begin{aligned}
&\|\nabla F_i(\tilde{w}) - \nabla F(\tilde{w})\|^2 = \|\nabla F_i(w + \delta_i) - \nabla F(w + \delta)\|^2 \\
=& \|\nabla F_i(w + \delta_i) - \nabla F_i(w) + \nabla F_i(w) - \nabla F(w) + \nabla F(w) - \nabla F(w + \delta)\|^2 \\
\leq& 3\|\nabla F_i(w + \delta_i) - \nabla F_i(w)\|^2 + 3\|\nabla F_i(w) - \nabla F(w)\|^2 + 3\|\nabla F(w) - \nabla F(w + \delta)\|^2 \qquad (10a) \\
\leq& 3\sigma_g^2 + 6(L+r)^2\rho^2, \qquad (10b)
\end{aligned}$$

where (10a) is from Lemma 2 and (10b) is from Assumption 1,2, the perturbation is limited by ρ. □

Below, we bound the average client drift over all clients within the $t-$ communication round. The average client drift is bounded by

Lemma 6. *Given* $\eta_l \leq \frac{1}{\sqrt{30}\alpha K(L+r)}$ *and* $\alpha \leq \frac{1}{2}$*, there is*

$$\begin{aligned}\epsilon_{t,k} =& \frac{1}{|S_t|}\sum_{i\in S_t}\mathbb{E}\left\|w_i^{t,k}-w^t\right\|^2 \\ \leq& 5K\eta_l^2\left[2\alpha^2(L+r)^2\eta_l^2\rho^2\sigma_l^2+7K\alpha^2\eta_l^2(3*\sigma^2+6(L+r)^2\rho^2)\right. \\ &\left.+14K\left(1-\alpha\right)^2\eta_l^2\left\|\nabla F(w_t)\right\|\right]+28K^3\alpha^2(L+r)^4\eta_l^4\rho^2.\end{aligned}$$

Proof. The term $\mathbb{E}\left\|w_i^{t,k}-w^t\right\|^2$ can be rewriteen as

$$\begin{aligned}&\mathbb{E}\left\|w_i^{t,k}-w^t\right\|^2=\mathbb{E}\left\|w_i^{t,k-1}-\eta_l\left[\alpha\tilde{g}_i^{t,k-1}+(1-\alpha\triangle^t)\right]-w^t\right\|^2 \\ \leq&\mathbb{E}\left\|w_i^{t,k-1}-w^t-\alpha\eta_l\left(\tilde{g}_i^{t,k-1}-\nabla F_i(\tilde{w}_i^{t,k-1})+\nabla F_i(\tilde{w}_i^{t,k-1})-\nabla F_i(\tilde{w}^t)+\nabla F_i(\tilde{w}^t)\right.\right. \quad (11a)\\ &\left.\left.-\nabla F(\tilde{w}^t)+\nabla F(\tilde{w}^t)\right)+(1-\alpha)\eta_l\delta^t\right\|^2 \\ \leq&(1+\frac{1}{2K-1}+2\alpha^2(L+r)^2\eta_l^2)\mathbb{E}\left\|w_i^{t,k-1}-w^t\right\|^2+2\alpha^2(L+r)^2\eta_l^2\rho^2\sigma_l^2 \quad (11b)\\ &+7K^2\alpha\eta_l^2\mathbb{E}\left\|\nabla F_i(\tilde{w}_i^{t,k-1})-\nabla F_i(\tilde{w})\right\|^2+7K\alpha^2\eta_l^2(3\sigma_g^2+6(L+r)^2\rho^2) \\ &+7K\alpha^2\eta_l^2\left\|\nabla F(\tilde{w}^t)\right\|^2+7K\eta_l^2(1-\alpha)^2\left\|\triangle^t\right\|^2 \\ \leq&(1+\frac{1}{2K-1}+2\alpha^2(L+r)^2\eta_l^2+14K\alpha(L+r)^2\eta_l^2)\mathbb{E}\left\|w_i^{t,k-1}-w^t\right\|^2+2\alpha^2(L+r)^2\eta_l^2\rho^2\sigma_l^2 \\ &\quad (11c)\\ &+7K(1-\alpha)^2\eta_l^2\mathbb{E}\left\|\triangle^t\right\|^2+14K\alpha^2(L+r)^2\eta_l^2\mathbb{E}\left\|\delta_i^{t,k}-\delta^t\right\|^2 \\ &+7K\alpha^2\eta_l^2(3\sigma_g^2+6(L+r)^2\rho^2)+7\alpha^2K\mathbb{E}\left\|\nabla F(\tilde{w^t})\right\|^2 \\ \leq&(1+\frac{1}{2K-1}+2\alpha^2(L+r)^2\eta_l^2+14K\alpha(L+r)^2\eta_l^2)\mathbb{E}\left\|w_i^{t,k-1}-w^t\right\|^2+2\alpha^2(L+r)^2\eta_l^2\rho^2\sigma_l^2 \\ &\quad (11d)\\ &+14K\alpha^2(L+r)^2\eta_l^2\mathbb{E}\left\|\delta_i^{t,k}-\delta^t\right\|^2+7K\alpha^2\eta_l^2(3\sigma_g^2+6(L+r)^2\rho^2) \\ &+14K(1-\alpha)^2\eta_l^2\left\|\nabla F(\tilde{w}^t)\right\|^2, \\ &\quad (11e)\end{aligned}$$

where (11a) follows from the fact that $\tilde{g}_i^{t,k-1}$ is an unbiased estimator of $\nabla F_i(\tilde{w}_i^{t,k-1})$ and Lemma 3; (11b) is from Lemma 2 and 5; (11c) is from Assumption 3 and Lemma 2; (11d) is from Assumption 2 and due to the fact that $\triangle \approx \nabla F(\tilde{w}^t)$ and $\alpha < \frac{1}{2}$.

Averaging over the clients i and learning rate satisfies $\eta_l \leq \frac{1}{\sqrt{30}\alpha K(L+r)}$, we have:

$$\begin{aligned}
\epsilon_{t,k} \leq & (1+\frac{1}{2K-1}+2\alpha^2(L+r)^2\eta_l^2+14K\alpha(L+r)^2\eta_l^2)\mathbb{E}\left\|w_i^{t,k-1}-w^t\right\|^2 \\
& +2\alpha^2(L+r)^2\eta_l^2\rho^2\sigma_l^2+14K\alpha^2(L+r)^2\eta_l^2\mathbb{E}\left\|\delta_i^{t,k}-\delta^t\right\|^2 \\
& +7K\alpha^2\eta_l^2(3\sigma_g^2+6(L+r)^2\rho^2)+14K(1-\alpha)^2\eta_l^2\left\|\nabla F(\tilde{w}^t)\right\|^2, \\
\leq & (1+\frac{1}{K-1})\frac{1}{N}\sum_{i=1}^{N}\mathbb{E}\left\|w_i^{t,k-1}-w^t\right\|^2 \qquad (12a) \\
& +2\alpha^2(L+r)^2\eta_l^2\rho^2\sigma_l^2+14K\alpha^2(L+r)^2\eta_l^2\mathbb{E}\left\|\delta_i^{t,k}-\delta^t\right\|^2 \\
& +7K\alpha^2\eta_l^2(3\sigma_g^2+6(L+r)^2\rho^2)+14K(1-\alpha)^2\eta_l^2\left\|\nabla F(\tilde{w}^t)\right\|^2, \\
\leq & \sum_{\tau=0}^{k-1}(1+\frac{1}{K-1})^\tau\left[2\alpha^2(L+r)^2\eta_l^2\rho^2\sigma_l^2+7K\alpha^2\eta_l^2(3\sigma_g^2+6(L+r)^2\rho^2)\right. \\
& \left.+14K\alpha^2(L+r)^2\eta_l^2\mathbb{E}\left\|\delta_i^{t,k}-\delta^t\right\|^2\right]+14K(1-\alpha)^2\eta_l^2\left\|\nabla F(\tilde{w}^t)\right\|^2, \\
\leq & 5K\left(2\alpha^2(L+r)^2\eta_l^2\rho^2\sigma_l^2+7K\alpha^2\eta_l^2(3\sigma_g^2+6(L+r)^2\rho^2)+14K(1-\alpha)^2\eta_l^2\left\|\nabla F(\tilde{w}^t)\right\|^2\right) \qquad (12b) \\
& +28\alpha^2K^3(L+r)^4\eta_l^4\rho^2,
\end{aligned}$$

where (12a) is due to the fact that $\eta_l \leq \frac{1}{\sqrt{30}\alpha K(L+r)}$ and $\alpha \leq \frac{1}{2}$; (12b) is from Lemma B.1 in [31]. □

The global update can be bounded by

Lemma 7. *For the partial client participation, we can bound* $\mathbb{E}_t\left[\left\|\triangle^{t+1}\right\|^2\right]$ *as follows:*

$$\mathbb{E}_t\left[\left\|\triangle^{t+1}\right\|^2\right] \leq \frac{K\eta_l^2\rho^2\sigma_l^2}{S}(L+r)^2+\frac{\eta_l^2}{S^2}\left[\left\|\sum_{i=1}^{N}\mathbb{P}\left\{i\in S^t\right\}\sum_{k=0}^{K-1}\nabla F_i(\tilde{w}_i^{t,k})\right\|^2\right]$$

Proof.

$$\begin{aligned}
& \mathbb{E}_t\left[\left\|\triangle^{t+1}\right\|^2\right]=\frac{1}{K^2S^2\eta_l^2}\mathbb{E}_t\left[\sum_{i\in S_t}\left\|\sum_k\left(\alpha\eta_l\tilde{g}_i^{t,k}+\eta_l(1-\alpha)\triangle^t\right)\right\|^2\right] \\
= & \frac{\alpha^2}{K^2S^2}\mathbb{E}_t\left[\sum_{i\in S_t}\left\|\sum_{k=0}^{K-1}\tilde{g}_i^{t,k}-\nabla F_i(\tilde{x}_i^{t,k})\right\|^2\right]+\frac{1}{K^2S^2}\mathbb{E}_t\left[\sum_{i\in S_t}\left\|\sum_k\left(\alpha\nabla F_i(\tilde{x}_i^{t,k})+(1-\alpha)\triangle^t\right)\right\|^2\right] \qquad (13a) \\
\leq & \frac{\alpha^2(L+r)^2\rho^2}{KS}\sigma_l^2+\frac{2(1-\alpha^2)}{KS}\left\|\nabla F(\tilde{w}^t)\right\|^2+\frac{2\alpha^2}{K^2S^2}\left[\sum_i\mathbb{P}\left\{i\in S_t\right\}\left\|\sum_{k=0}^{K-1}\nabla F_i(w_i^{t,k})\right\|^2\right] \qquad (13b) \\
= & \frac{\alpha^2(L+r)^2\rho^2}{KS}\sigma_l^2+\frac{2(1-\alpha^2)}{KS}\left\|\nabla F(\tilde{w}^t)\right\|^2+\frac{2\alpha^2}{K^2SN}\sum_{i=1}^{N}\mathbb{E}_t\left\|\sum_{k=0}^{K-1}\nabla F_i(\tilde{w}_i^{t,k})\right\|^2 \\
& +\frac{2\alpha^2(S-1)}{K^2SN^2}\mathbb{E}_t\left\|\sum_{i=1}^{N}\sum_{k=0}^{K-1}\nabla F_i(\tilde{w}_i^{t,k})\right\|^2, \qquad (13c)
\end{aligned}$$

where (13a) is from Lemma 5 and (13b) is from Lemma 4. □

Next, we provide the following lemma to demonstrate the descent behavior of FedMRUR under partial client participation setting.

Lemma 8. *For all $t \in [T-1]$ and $i \in S_t$, with the choice of learning rate, the iterates generated by FedMRUR under partial client participation satisfy:*

$$\begin{aligned}\mathbb{E}_t\left[F(w^{t+1})\right] \leq & F(\tilde{w}^t) - K\eta_g\eta_l d_t(\frac{1}{2} - 20K^2L^2\eta_l^2)\left\|\nabla F(\tilde{w}^t)\right\|^2 + K\eta_g\eta_l\left(6K^2\eta_l^2\alpha^4\rho^2\right. \\ & \left. +5K^2\eta_l\alpha^4\rho^2\sigma^2 + 20K^3\eta_l^3\alpha^2\sigma_g^2 + 16K^3\eta_l^4\alpha^6\rho^2 + \frac{\eta_g\eta_l\alpha^3\rho^2}{N}\sigma_l^2\right).\end{aligned}$$

Proof. Let's define $\epsilon_\delta = \frac{1}{N}\sum_i \mathbb{E}\left[\delta_{i,k} - \delta\right]^2$, where $\delta = \underset{\delta}{\operatorname{argmax}}\, F(w+\delta)$.

$$\begin{aligned}&\mathbb{E}_t\left[F(w^{t+1})\right] \leq F(w^t) + E_t\left\langle\nabla F(\tilde{w}^t), \tilde{w}^{t+1} - \tilde{w}^t\right\rangle + \frac{L+r}{2}\mathbb{E}_t\left[\left\|\tilde{w}^{t+1} - \tilde{w}^t\right\|^2\right] \\ =&F(w^t) - \alpha\eta_g\left\|\nabla F(\tilde{w}^t)\right\|^2 + \eta_g\left\langle\nabla F(\tilde{w}^t), \mathbb{E}\left[-\triangle^{t+1} + \alpha\nabla F(\tilde{w}^t)\right]\right\rangle + \frac{L+r}{2}\eta_g^2\mathbb{E}_t\left[\left\|\triangle^{r+1}\right\|^2\right]\end{aligned} \tag{14}$$

Let's denote the $\frac{\sum_{i\in S_t}\|\triangle_i^t\|}{\|\sum_{i\in S_t}\triangle_i^t\|}$ as d_t and bound the third term in (14) as follows:

$$\begin{aligned}\left\langle\nabla F(\tilde{w}^t), \mathbb{E}\left[-\triangle^{t+1} + \alpha\nabla F(\tilde{w}^t)\right]\right\rangle \leq & \left(\frac{3\alpha}{2} - 1\right)d_t\left\|\nabla F(\tilde{w}^t)\right\|^2 + \alpha(L+r)^2 d_t(\epsilon_{t,k} + \epsilon_\delta) \\ & - \frac{\alpha d_t}{2K^2N^2}\mathbb{E}_t\left\|\sum_{i,k}\nabla F_i(\tilde{w}_i^{t,k})\right\|^2\end{aligned} \tag{15}$$

Plugging (15) into (14), we have:

$$
\begin{aligned}
&\mathbb{E}_t\left[F(\tilde{w}^{t+1})\right] \\
\leq & F(\tilde{w}^t) - \left(\eta_g - \frac{\alpha\eta_g}{2}\right) d_t \left\|\nabla F(\tilde{w}^t)\right\|^2 + \alpha(L+r)^2\eta_g d_t(\epsilon_{t,k}+\epsilon_\delta) \\
& - \frac{\alpha\eta_g d_t}{2K^2N^2}\mathbb{E}_t\left\|\sum_{i,k}\nabla F_i(\tilde{w}_i^{t,k})\right\|^2 + \frac{(L+r)\eta_g^2}{2}\mathbb{E}_t\left[\left\|\triangle^{t+1}\right\|^2\right] \\
\leq & F(\tilde{w}^t) - \left(\frac{3\alpha\eta_g d_t}{4} - \frac{2(1-\alpha)^2(L+r)\eta_g d_t}{KS}\right)\left\|\nabla F(\tilde{w}^t)\right\|^2 + \alpha(L+r)^2\eta_g d_t(\epsilon_{t,k}+\epsilon_\delta) \qquad \text{(16a)} \\
& + \frac{\alpha^2(L+r)^3\rho^2\eta_g^2}{2KS}\sigma_l^2 - \frac{\alpha\eta_g d_t}{2K^2N^2}\mathbb{E}_t\left\|\sum_{i,k}\alpha\nabla F_i(\tilde{w}_i^{t,k})\right\|^2 \\
& + \frac{(L+r)\alpha^2\eta_g^2}{2K^2SN}\sum_i\mathbb{E}_t\left\|\sum_k\nabla F_i(\tilde{w}_i^{t,k})\right\|^2 + \frac{(L+r)\alpha^2(S-1)\eta_g^2}{K^2SN^2}\mathbb{E}_t\left\|\sum_k\nabla F_i(\tilde{w}_i^{t,k})\right\|^2 \\
\leq & F(\tilde{w}^t) - \alpha\eta_g d_t\left(\frac{3}{4} - \frac{2(1-\alpha)(L+r)}{KN} - 70(1-\alpha)K^2(L+r)^2\eta_l^2 - \frac{90\alpha(L+r)^3\eta_g\eta_l^2}{Sd_t}\right. \qquad \text{(16b)} \\
& \left. - \frac{3\alpha(L+r)\eta_g}{2S}\right)\left\|\nabla F(\tilde{w}^t)\right\|^2 + \beta\eta_g\left(10\alpha^2(L+r)^4\eta_l^2\rho^2\sigma_l^2 + 28\alpha^2K^3(L+r)^6\eta_l^4\rho^2\right. \\
& +35\alpha^2K(L+r)^2\eta_l^2(3\sigma_g^2 + 6(L+r)^2\rho^2) + 2K^2(L+r)^4\eta_l^2\rho^2 + \frac{\alpha(L+r)^3\eta_g^2 d_t^2\rho^2}{2KS}\sigma_l^2 \\
& +\frac{\alpha L\eta_g d_t}{K^2SN}(30NK^2L^4\eta_l^2\rho^2\sigma_l^2 + 270NK^3(L+r)^2\eta_l^2\sigma_g^2 + 540NK^2(L+r)^4\eta_l^2\rho^2 \\
& \left.+72K^4(L+r)^6\eta_l^4\rho^2 + 6NK^4L^2\eta_l^2\rho^2 + 4NK^2\sigma_g^2 + 3NK^2(L+r)^2\rho^2)\right) \\
\leq & F(\tilde{w}^t) - C\alpha\eta_g d_t\left\|\nabla F(\tilde{w}^t)\right\|^2 + \beta\eta_g\left(10\alpha^2(L+r)^4\eta_l^2\rho^2\sigma_l^2 + 28\alpha^2K^3(L+r)^6\eta_l^4\rho^2\right. \qquad \text{(16c)} \\
& +35\alpha^2K(L+r)^2\eta_l^2(3\sigma_g^2 + 6(L+r)^2\rho^2) + 2K^2(L+r)^4\eta_l^2\rho^2 + \frac{\alpha(L+r)^3\eta_g^2 d_t^2\rho^2}{2KS}\sigma_l^2 \\
& +\frac{\alpha L\eta_g d_t}{K^2SN}(30NK^2L^4\eta_l^2\rho^2\sigma_l^2 + 270NK^3(L+r)^2\eta_l^2\sigma_g^2 + 540NK^2(L+r)^4\eta_l^2\rho^2 \\
& \left.+72K^4(L+r)^6\eta_l^4\rho^2 + 6NK^4L^2\eta_l^2\rho^2 + 4NK^2\sigma_g^2 + 3NK^2(L+r)^2\rho^2)\right) \qquad \text{(16d)}
\end{aligned}
$$

where (16a) is from Lemma 7; (16b) is from Lemmas 6, Lemma B.1 in [31] and due to the fact that $\eta_g \leq \frac{S}{2\alpha L(S-1)}$ and (16c) is from the condition that $\frac{3}{4} - \frac{2(1-\alpha)L}{KN} - 70(1-\alpha)K^2(L+r)^2\eta_l^2 - \frac{90\alpha(L+r)^3\eta_g\eta_l^2}{S} - \frac{3\alpha(L+r)\eta_g}{2S} > C > 0$ and $\alpha \leq \frac{1}{2}$ hold. □

Finally, we provide following two theorems to charcterize the convergence rate of FedMRUR:

Theorem 9 (Extension of Theorem 1)**.** *Let all the assumptions hold and with partial client participation. If we choose learning rate* $\eta_l \leq \frac{1}{\sqrt{30}\alpha KL}$, $\eta_g \leq \frac{S}{2\alpha L(S-1)}$ *satisfying* $\frac{3}{4} - \frac{2(1-\alpha L)}{KN} - 70(1-\alpha)K^2(L+r)^2\eta_l^2 - \frac{90\alpha(L+r)^3\eta_g\eta_l^2}{S} - \frac{3\alpha(L+r)\eta_g}{2S}$, *then for all* $K \geq 0$ *and* $T \geq 1$, *we have:*

$$
\frac{1}{\sum_{t=1}^T d_t}\sum_{t=1}^T \mathbb{E}\left\|\nabla F(w^t)\right\|^2 d_t \leq \frac{F^0 - F^*}{C\alpha\eta_g\sum_{t=1}^T d_t} + \Phi,
$$

where

$$\Phi = \frac{1}{C}\left[10\alpha^2(L+r)^4\eta_l^2\rho^2\sigma_l^2 + 35\alpha^2K(L+r)^2\eta_l^2 3(\sigma_g^2 + 6(L+r)^2\rho^2) + 28\alpha^2K^3(L+r)^6\eta_l^4\rho^2\right.$$
$$+2K^2L^4\eta_l^2\rho^2 + \frac{\alpha(L+r)^3\eta_g^2\rho^2}{2KS}\sigma_l^2 + \frac{\alpha(L+r)\eta_g}{K^2SN}(30NK^2(L+r)^4\eta_l^2\rho^2\sigma_l^2$$
$$+270NK^3(L+r)^2\eta_l^2\sigma_g^2 + 540NK^2(L+r)^4\eta_l^2\rho^2 + 72K^4(L+r)^6\eta_l^4\rho^2$$
$$\left.+6NK^4(L+r)^2\eta_l^2\rho^2 + 4NK^2\sigma_g^2 + 3NK^2(L+r)^2\rho^2)\right].$$

If we set $\eta_g = \Theta(\frac{\sqrt{SK}}{\sqrt{T}})$ *and* $\eta_l = \Theta\left(\frac{1}{\sqrt{ST}K(L+r)}\right)$*, the convergence rate of the FedMRUR under partial client participation is:*

$$\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\left\|\nabla F(w^t)\right\|^2 = O(\frac{1}{\sqrt{SKT}}) + O\left(\frac{\sqrt{K}}{ST}\right) + O\left(\frac{1}{\sqrt{K}T}\right).$$

Proof. Summing (16c) in Lemma 8 over $t = \{1, ..., T\}$ and multiplying both sides by $\frac{1}{C\alpha\eta_g\sum_{t=1}^{T} d_t}$, we have

$$\begin{aligned}\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\left\|\nabla F(w^t)\right\|^2 \leq& \frac{F(\tilde{w}^t - F(\tilde{w}^{t+1}))}{C\alpha\eta_g\sum_{t=1}^{T} d_t} + \Phi \\ \leq& \frac{F^0 - F^*}{C\alpha\eta_g\sum_{t=1}^{T} d_t} + \Phi,\end{aligned}$$

where the second inequality comes from the fact that $F^0 - F^* \leq F(\tilde{w}^t) - F(\tilde{w}^{t+1})$. According to the definition of d_t in Lemma 8 and triangle inequality, we have $1 \leq \frac{\sum_{i\in S_t}\|\triangle_i^t\|}{\|\sum_{i\in S_t}\triangle_i^t\|} \leq S$ and $\sum_{t=1}^{T} d_t \geq T$. If we choose $\eta_g = \Theta(\frac{\sqrt{SK}}{\sqrt{T}})$, $\eta_l = \Theta\left(\frac{1}{\sqrt{ST}K(L+r)}\right)$ and $\rho = \Theta(\frac{1}{\sqrt{T}})$, the above ineqaulity can be rewriteen as

$$\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\left\|\nabla F(w^t)\right\|^2 = O(\frac{1}{\sqrt{SKT}}) + O\left(\frac{\sqrt{K}}{ST}\right) + O\left(\frac{1}{\sqrt{K}T}\right).$$

□

B Experiments

B.1 Results for CIFAR-10

Table 5 characterizes the convergence speed of multiple algorithms on CIFAR-10. For most of the time, our proposed method, FedMRUR outperforms the baselines. Therefore, we can conclude that: 1) our method achieves the fastest convergence speed, especially when the data heterogeneity is large, which validates the normalized update aggregation scheme accelerate the iteration; 2) when the statistical heterogeneity is large, the proposed FedMRUR accelerates the convergence more effectively, which validates that utilizing the hyperbolic graph fusion is able to alleviate the issue of the model inconsistency across clients.

Table 6 presents the final test accuracy of ResNet-18 trained using multiple algorithms on CIFAR-10 dataset under four heterogeneous settings. We plot the test accuracy of the algorithms for the image classification task in Figure 5. We can oberserve that the proposed FedMRUR performs well with good stability and efficently mitigates the negative effect of the model inconsistency. Specifically, on the Dirichlet-0.3 setups, FedMRUR achieves a test accuracy of 84.53%, which is 0.51% higher than the second-best algorithm, MoFedSAM. Based on these, we can conclude that FedMRUR reduces the model inconsistency and improves the convergence speed effectively.

B.2 Verification of Normalized Aggregation

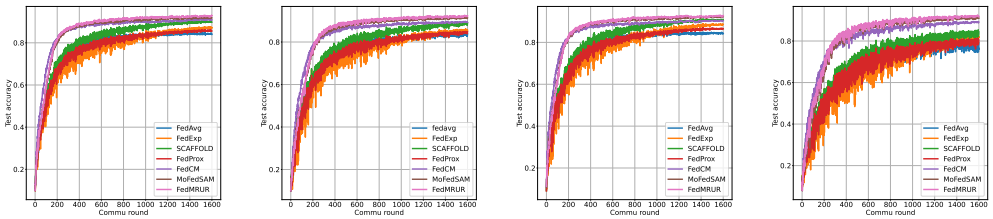
From the theoretical view, we can conclude that the "Normalized Aggregation of Local Updates" can accelerate the convergence in Theorem 9. In fact, using this operator in other baselines can also

Table 5: Convergence speed on CIFAR-10 dataset in both Dir(μ) and Path(n) distributions. "Acc." represents the target test accuracy on the dataset. "∞" means that the algorithm is unable to achieve the target accuracy on CIFAR-10 dataset.

Algorithms		FedAvg	FedExp	FedProx	SCAFFOLD	FedCM	MoFedSAM	Our
ACC.		70%						
Dir(μ)	0.6	392	538	354	263	95	119	125
	0.3	513	518	452	349	131	134	142
Path(n)	6	353	459	328	242	110	112	115
	3	∞	770	∞	466	177	178	192
ACC.		75%						
Dir(μ)	0.6	∞	788	∞	441	185	178	180
	0.3	∞	905	∞	588	229	205	221
Path(n)	6	∞	866	∞	426	171	166	166
	3	∞	1225	∞	1552	278	307	305
ACC.		80%						
Dir(μ)	0.6	∞	∞	∞	∞	471	384	393
	0.3	∞	∞	∞	∞	573	450	449
Path(n)	6	∞	∞	∞	1181	443	356	353
	3	∞	∞	∞	∞	810	636	630

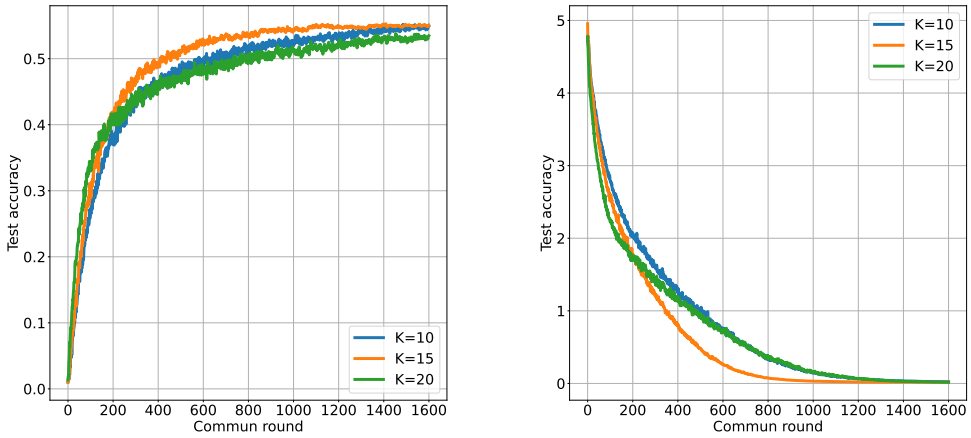
Table 6: Test accuracy (%) on CIFAR-10 dataset in both Dir(μ) and Path(n)) distributions.

Algorithm	CIFAR-10			
	Dir(μ)		Path(n)	
	μ = 0.6	μ = 0.3	n = 6	n = 3
FedAvg	72.96	71.44	73.22	67.78
FedExp	79.26	76.73	78.70	74.65
FedProx	73.69	72.15	73.95	68.46
SCAFFOLD	79.69	78.49	79.77	72.72
FedCM	84.48	82.95	84.15	83.10
MoFedSAM	84.99	84.03	85.10	84.13
FedMRUR	85.70	84.53	85.61	84.89



(a) (b) (c) (d)

Figure 5: Test accuracy w.r.t. communication rounds of our proposed method and other approaches. Each method performs in 1600 communication rounds. To compare them fairly, the basic optimizers are trained with the same hyperparameters on CIFAR-10 dataset.



(a) (b)

Figure 6: Test accuracy and train loss w.r.t communication rounds of FedMRUR with different local intervals K.

improve the performance. Here, we present the effect of the normalized aggregation method applied to FedCM and FedAvg in Table 7. From the results, we can find that the "Normalized Aggregation" can improve the convergence a lot (For example, when $\mu = 0.3$, it can improve the final acc 4% over FedCM).

Table 7: Test accuracy (%) on CIFAR-10 dataset in both Dir(μ) and Path(n)) distributions.

Algorithm	CIFAR-100			
	Dir(μ)		Path(n)	
	μ = 0.6	μ = 0.3	n = 20	n = 10
$FedAvg$	39.87	39.50	38.47	36.67
$FedAvg^+$	42.09	41.71	42.22	40.10
$FedCM$	51.01	50.93	50.58	50.03
$FedCM^+$	52.53	52.32	52.59	52.50

B.3 Validation for the linear speedup propoerty

In this part, we present the experiment results which veritfies the linear speedup propoerty of the proposed FedMRUR. Because the whole dataset is fixed, increasing the number of clients changes the amount of data in the local data, which changes the entire optimization problem, we conduct the experiment under various settings of local intervals K fixing the number of clients to verify the linear speedup property. From Figure 6, when K increase to 15, the algorithm achieves $1.5\times$ than $K = 10$. From (8), when local interval K is increased to $\mathcal{O}\left(ST\right)^{\frac{1}{2}}$, the impact of the second term in Theorem 1 becomes greater and the first term becomes less. Therefore, when K increase from 15 to 20, the speedup of convergence is not obvious.

B.4 Impact of Hyperbolic space

Since hyperbolic geometry is a Riemann manifold with a constant negative curvature, its typical geometric property is that the volume grows exponentially with its radius, whereas the Euclidean space grows polynomially. Such a geometric trait has 2 advantages:

- The hyperbolic space exhibits minimal distortion and it fits the hierarchies particularly well since the space closely matches the growth rate of graph-like data while the Euclidean space doesn't.
- Even with a low-embedding dimensional space, hyperbolic models are surprisingly able to produce a high quality representation, which makes them to be particularly advantageous in low-memory and low-storage scenarios.

In realistic scenarios, there exists many graph-like data structure, such as the hypernym structure in NLP, the subordinate structure of entities in the knowledge graph and the power-law distribution in recommender systems. In FL, the machine learning models have a graph-like structure, so adopting the Lorentz metric of hyperbolic space makes use of the hierarchical information in neural networks, which are helpful to fuse the model further bring prediction gains. Using Euclidean metric, or some Riemann metric defined by a positive definite matrix is an interesting idea. Here, we show the results of experiments using different geometric spaces as follow (Table 8). From these results, we can find that Lortenz metric of hyperbolic space can help the algorithm achieving the highest test accuracy.

Table 8: Test accuracy (%) on CIFAR-100 dataset using different manifolds.

Space	Euclidean	Hyperbolic
Test Acc.	54.01(0.36)	55.64(0.41)

The representations generated by the model have fewer dimensions than the data. Mapping the representations to the hyperbolic space introduces less computation overhead than mapping the data. We also conduct experiments mapping the original data to the hyperbolic space over 8 seeds. The results are as presented in Table 9.

Table 9: Test accuracy (%) on CIFAR-100 dataset using hyperbolic model in different ways.

Orignial data	Representation
56.03(0.56)	55.64(0.41)

From the table, we can see that both methods achieve similar performance. Thus, considering the computation overhead and performance, we only map representations to hyperbolic space and do not treat the entire learning process in hyperbolic space.

To study the impact of β for hyperbolic graph manifold regularization on the performance, we conduct the experiment on CIFAR100 task with different β settings and present the final test accuracy in Table 10. From this table, we can find that has a limited impact on the final performance of the algorithm.

Table 10: Test accuracy (%) on CIFAR-100 dataset using different β.

β	0.1	0.5	1.0	5.0	10.0
Test Acc.	54.67	54.69	55.04	54.79	54.91

B.5 Training time

Test Experiments: Nvidia GTX-3090 GPU, CUDA Driver 11.4, Driver Version 470.10.3.01, Pytorch-1.11.1

Table 11 shows the wall-clock time costs on the CIFAR-100 of Dirichlet-0.3 dataset split. Due to the double computation of the gradients via SAM optimizer, MoFedSAM and FedMRUR will take more time in a single communication round, about $1.46\times$ over the FedCM method. However, the communication rounds required is less than FedCM. Considering the total wall-clock time costs, the acceleration ratio of FedMRUR achieves $2.46\times$ compared with MoFedSAM ($3.67\times$ compared with FedCM) at the final. Therefore, we can conclude that the FedMRUR is more efficient with respect to the communication round and wall-clock time when high-performance models are required.

Table 11: Test accuracy (%) on CIFAR-100 dataset to achieve 50% test accuracy.

Algorithm	Times(s/round)	Rounds	Total(s)	Cost Ratio
FedAvg	9.23	∞	∞	∞
FedExp	14.82	∞	∞	∞
SCAFFOLD	15.52	∞	∞	∞
FedProx	12.89	∞	∞	∞
FedCM	11.53	1407	16222.71	$3.67\times$
MoFedSAM	15.53	701	10886.53	$2.46\times$
FedMRUR	16.82	263	4423.66	$1\times$