Multi-Level Branched Regularization for Federated Learning

Jinkyu Kim [*1] **Geeho Kim** [*1] **Bohyung Han** [1 2]

Abstract

A critical challenge of federated learning is data heterogeneity and imbalance across clients, which leads to inconsistency between local networks and unstable convergence of global models. To alleviate the limitations, we propose a novel architectural regularization technique that constructs multiple auxiliary branches in each local model by grafting local and global subnetworks at several different levels and that learns the representations of the main pathway in the local model congruent to the auxiliary hybrid pathways via online knowledge distillation. The proposed technique is effective to robustify the global model even in the non-iid setting and is applicable to various federated learning frameworks conveniently without incurring extra communication costs. We perform comprehensive empirical studies and demonstrate remarkable performance gains in terms of accuracy and efficiency compared to existing methods. The source code is available in our project page[1].

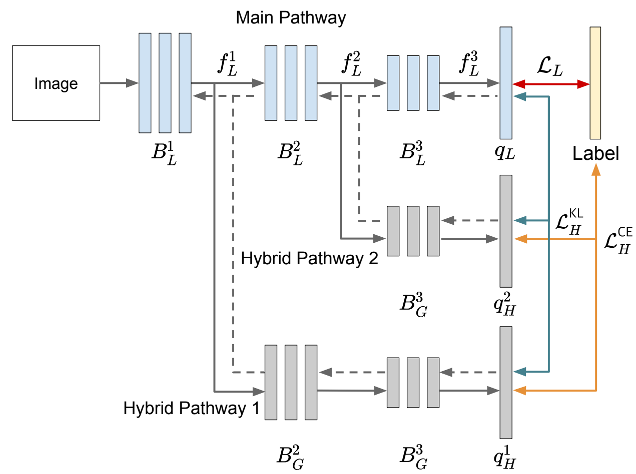


Figure 1. Illustration of the local network for the proposed multi-level branched regularization framework. B_L and B_G denote network blocks extracted from local and global networks, respectively, and their superscripts indicate block indices. q_L and q_H indicate softmax outputs of the main pathway and hybrid pathways, respectively. Solid and dashed gray lines represent forward and backpropagation flows, respectively. For training, the standard cross-entropy losses are applied to all branches and the KL-divergence losses between the main pathway and the rest of the pathways are employed for regularization. Note that we update the parameters in the main pathway only, which are illustrated in sky blue color in this figure.

1. Introduction

Training deep neural networks typically relies on centralized algorithms, where computing resources and training data are located within a single server. Recently, to deal with large-scale models and/or distributed data, the learning frameworks based on multiple remote machines become widespread in machine learning research and development. Federated learning (McMahan et al., 2017) is a unique distributed learning framework that takes advantage of computing resources and training data in clients, typically edge devices, which is helpful to secure data privacy but often suffers from insufficient computing power, low communication bandwidth, and extra battery consumption. As practical solutions to handle the challenges, each edge device often runs a small number of iterations and minimizes the communication rounds with the central server.

FedAvg (McMahan et al., 2017), the standard optimization method of federated learning, maintains the global model in the server by aggregating the local models trained independently in multiple clients. Each client utilizes its own dataset to update the local model instead of sharing the data with the server or other clients, sends the locally optimized model to the server for aggregation, and downloads the updated global model for the next stage. FedAvg works well when the individual datasets of the clients are iid and the participation rate of the clients is high, but it struggles with low convergence speed otherwise. This is because, when the data distributions of individual clients are different from

[*]Equal contribution [1]Computer Vision Laboratory, Department of Electrical and Computer Engineering & ASRI, Seoul National University, Korea [2]Interdisciplinary Program of Artificial Intelligence, Seoul National University, Korea. Correspondence to: Bohyung Han <bhhan@snu.ac.kr>.

Proceedings of the 39th International Conference on Machine Learning, Baltimore, Maryland, USA, PMLR 162, 2022.

[1]http://cvlab.snu.ac.kr/research/FedMLB

the global distribution, local updates are prone to drift and increase divergence with respect to the global model (Zhao et al., 2018; Karimireddy et al., 2020).

Our goal is to make each client preserve the latest global representations and prevent model drift caused by independent local updates. Meanwhile, we make the server learn client-specific knowledge by a simple aggregation of local networks and achieve a high-performance model well-suited for all clients, eventually. To mitigate the discrepancy between the local models updated by heterogeneous datasets, we propose a novel architectural regularization technique via knowledge distillation that grafts the local and global subnetworks and constructs multi-level branches. The auxiliary branches reduce the deviation of the representations in the local models from the feature space of the global model. To this end, a participating client first modularizes the downloaded global network into multiple blocks. The client trains the main pathway of the local model with the parameters corresponding to the global subnetworks fixed, while the output representations of the hybrid pathways are made similar to those in the main pathway using additional regularization terms based on knowledge distillation. Our regularization approach is unique compared to the methods based on the standard knowledge distillation because it constrains the representations of the main pathway in the local model using the on-the-fly outputs of the hybrid pathways. This idea is motivated by a recent knowledge distillation approach that aims to learn a student-friendly teacher network (Park et al., 2021). Figure 1 illustrates our main idea.

The main contributions of this work are as follows:

- We propose a simple but effective regularization technique to reduce the drift problem of a local model, via online knowledge distillation between the main pathway and the multiple hybrid pathways reflecting the global representations partially.
- Our approach is robust to typical challenges of federated learning including data heterogeneity and low client participation rate, and is applicable to various federated learning frameworks.
- The proposed method requires no additional communication cost and spends no extra memory to store the history of local or global states and the auxiliary information.
- We demonstrate through extensive experiments that our architectural regularization technique improves accuracy and convergence speed consistently compared to the state-of-the-art federated learning algorithms.

The organization of this paper is as follows. We first discuss existing work related to the optimization in federated learning and review the basic concept with a simple solution in Section 2 and 3, respectively. Section 4 describes the proposed federated learning framework, and Section 5 demonstrates the effectiveness of the proposed approach via extensive experiments. Finally, we conclude this paper in Section 6.

2. Related Work

Federated learning (McMahan et al., 2017) is a distributed learning framework with the characteristics such as non-iid client data, data privacy requirement, massive distribution, and partial participation. Although FedAvg provides a practical solution for the issues, it still suffers from the heterogeneity of data across clients (Zhao et al., 2018). On the theoretical side, there exist several works that derive convergence rates with respect to the data heterogeneity (Li et al., 2020a; Wang et al., 2019; Khaled et al., 2019; Li et al., 2020b; Hsieh et al., 2020; Wang et al., 2020).

To alleviate the limitations of FedAvg, the local model updates are often regularized to prevent a large deviation from the global model. FedProx (Li et al., 2020a) imposes a quadratic penalty over the distance between the server and client parameters while SCAFFOLD (Karimireddy et al., 2020) and FedDANE (Li et al., 2019) employ a form of variance reduction techniques such as control variates. On the other hand, FedPD (Zhang et al., 2020) and FedDyn (Acar et al., 2021) penalize each client's risk objective dynamically based on its local gradient. Some approaches adopt a trick motivated by data augmentation (Yoon et al., 2021), or contrastive learning (Li et al., 2021) to ensure the similarity of the representations between the downloaded global model and local networks. However, these methods typically rely on unrealistic or impractical assumptions such as high participation rates, additional communication costs, or extra memory requirements in clients.

Meanwhile, the server-side optimization techniques have been discussed actively for the acceleration of the convergence. For example, FedAvgM (Hsu et al., 2019) adds a momentum term to speed up training, and FedADAM (Reddi et al., 2021) adopts an adaptive gradient-descent method. Our approach and these aggregation-based methods are orthogonal, so combining them together can yield additional performance gains.

There is another line of research that utilizes knowledge distillation to tackle data heterogeneity issue in federated learning. The algorithms in this category perform knowledge distillation either in the server or clients. As client distillation methods, FD (Seo et al., 2020) shares the representations between clients for knowledge distillation with their ensemble features while FedLS-NTD (Lee et al., 2021) transfers the knowledge of the global model to local networks except the activations for ground-truth labels. FedGKD (Yao et al., 2021) employs representations from the ensemble of the historical global models to refine local models, and FedGen (Zhu et al., 2021) learns a global generator to aggregate the local information and distill global knowledge to clients. For the distillation in the server, FedDF (Lin et al., 2020) utilizes the averaged representations of local models on proxy data for aggregation. However, these methods require additional communication overhead (Yao et al., 2021; Zhu et al., 2021) or auxiliary data (Seo et al., 2020; Lin et al., 2020). In particular, as discussed in (Wang et al., 2021), federated learning algorithms are sensitive to the communication cost and the use of globally-shared auxiliary data should be cautiously performed. To the contrary, our approach incurs no additional communication cost and has no requirement of auxiliary data. The proposed algorithm belongs to the method for local optimization based on knowledge distillation.

3. Preliminaries

This section briefly discusses the concept and procedure of the basic federated learning algorithm.

3.1. Problem setting and notations

Given N clients, the goal of the federated learning is to learn a global model θ that minimizes the average losses of all clients as follows:

$$\underset{\theta}{\text{argmin}} \left[\mathcal{L}(\theta) = \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}_i(\theta) \right], \tag{1}$$

where $\mathcal{L}_i(\theta) = \mathbb{E}_{(x,y)\sim D_i}[\mathcal{L}_i(x, y; \theta)]$ is the loss in the i^{th} client given by the expected loss over all instances in the client, denoted by D_i. Note that clients may have heterogeneous data distributions and exchanges of training data are strictly prohibited due to privacy issues.

3.2. FedAvg algorithm

FedAvg (McMahan et al., 2017) is a standard solution of federated learning, where the server simply aggregates all the participating client models to obtain the global model. Specifically, in the t^{th} communication round, a central server first sends a global model θ^{t-1} to each of the clients. Each client sets its initial parameter $\theta_{i,0}^t$ to θ^{t-1}, *i.e.*, $\theta_{i,0} = \theta^{t-1}$, performs K steps of the gradient descent optimization to minimize its local loss, and then returns the resultant model parametrized by $\theta_{i,K}^t$ to the server. An updated global model for the next round training is obtained by averaging all the participating local models in the current communication round. The local loss of FedAvg at the k^{th} local iteration ($k = 1, \ldots, K$) is defined by

$$\mathcal{L}_i(\theta_{i,k}^t) = \mathbb{E}_{(x,y)\sim D_i}[\mathcal{L}_i(x, y; \theta_{i,k}^t)], \tag{2}$$

where the cross-entropy loss is typically used for $\mathcal{L}_i$.

Multiple local updates in FedAvg before the aggregation step in the server decreases the communication cost for training apparently. However, in practice, it typically leads to the so-called client drift issue (Karimireddy et al., 2020), where the individual client updates are prone to be inconsistent due to overfitting on local client data. This phenomenon inhibits FedAvg from converging to the optimum of the average loss over all clients.

4. Proposed Algorithm: FedMLB

This section describes the details of our approach for federated learning with multi-level branched regularization, referred to as FedMLB, which exploits the representations of multiple hybrid pathways.

4.1. Overview

The main objective of FedMLB is to prevent the representations of the local model from being deviated too much by local updates while accommodating new knowledge from each client with heterogeneous datasets through independent local updates. We achieve this goal via indirect layer-wise online knowledge distillation using the architecture illustrated in Figure 1.

Although there exist several regularization approaches based on knowledge distillation for federated learning (Lee et al., 2021; Yao et al., 2021; Zhu et al., 2021), FedMLB is unique in the sense that it constructs multiple hybrid pathways combining the subnetworks of local and global models at various levels and learns the representations of the local model similar to those of the hybrid pathways. Note that the proposed approach performs knowledge distillation using on-the-fly targets given by the hybrid pathways although the subnetworks of the global model remain fixed. The regularization using the multiple auxiliary branches plays a critical role to make the individual blocks of the local network aligned well to the matching subnetworks of the global model. Consequently, local updates in each client allow the local model to be less deviated from the global counterpart, and such a regularization also reduces the variations of the models collected from multiple clients. We describe the details of the proposed algorithm next.

4.2. Multi-level hybrid branching

To perform the proposed regularization, we first divide the network into M exclusive blocks, which are based on the depths and the feature map sizes of the architecture. Let $\{B_L^m\}_{m=1}^M$ and $\{B_G^m\}_{m=1}^M$ be the sets of blocks in a local model and the global network, respectively. The main pathway consists of local blocks $B_L^{1:M}$ and we create multiple hybrid pathways by augmenting a subnetwork in the global

Algorithm 1 FedMLB

Input: # of clients N, # of communication rounds T, # of local iterations K, initial server model θ^0
for each round $t = 1, \dots, T$ **do**
  Sample a subset of clients $S_t \subseteq \{1, \dots, N\}$.
  Server sends θ^{t-1} to each of all clients $i \in S_t$.
  for each $i \in S_t$, **in parallel do**
    $\theta^t_{i,0} \leftarrow \theta^{t-1}$
    for $k = 1, \dots, K$ **do**
      for each (x, y) in a batch **do**
        $q_L(x; \tau) \leftarrow \text{softmax}\left(\frac{f_L(x;\theta^t_{i,k-1})}{\tau}\right)$
        $q^m_H(x; \tau) \leftarrow \text{softmax}\left(\frac{f^m_H(x;\theta^t_{i,m,k-1})}{\tau}\right),$
          $m = 1, \dots, M-1$
      end for
      $\mathcal{L}(\theta^t_{i,k-1}) \leftarrow \mathcal{L}_L + \lambda_1 \cdot \mathcal{L}^{\text{CE}}_H + \lambda_2 \cdot \mathcal{L}^{\text{KL}}_H$
      $\theta^t_{i,k} \leftarrow \theta^t_{i,k-1} - \eta \nabla \mathcal{L}$
    end for
    Client sends $\theta^t_{i,K}$ back to the server
  end for
  In server:
    $\theta^t = \frac{1}{|S_t|} \sum_{i \in S_t} \theta^t_{i,K}$ FedAvg
end for

Client KD，提取两种，基于客戶端的本地数据集

model $B^{m+1:M}_G$ to a local subnetwork $B^{1:m}_L$. Depending on branching locations, from 1 to $M-1$, several different hybrid pathways, denoted by $\{B^{1:m}_L, B^{m+1:M}_G\}_{m \in \{1,\dots,M-1\}}$, are constructed in parallel as illustrated in Figure 1.

The constructed network by multi-level hybrid branching has M pathways altogether for predictions, which includes one main pathway and $M-1$ hybrid pathways. The softmax output of the main pathway, $q_L(x; \tau)$, is given by

$$q_L(x; \theta, \tau) = \text{softmax}\left(\frac{f_L(x; \theta)}{\tau}\right), \tag{3}$$

where x is the input of the network, θ is the model parameters, τ is the temperature of the softmax function, and $f_L(\cdot; \cdot)$ denotes the logit of the main pathway. Similarly, the softmax output of the hybrid pathway stemming from B^m_L is given by

Softmax

$$q^m_H(x; \theta_m, \tau) = \text{softmax}\left(\frac{f^m_H(x; \theta_m)}{\tau}\right), \tag{4}$$

where $f^m_H(\cdot; \cdot)$ and θ_m denote the logit and the model parameter of the m^{th} hybrid pathway, respectively.

4.3. Knowledge distillation

Our goal is to learn the representation of the main pathway similar to those of the hybrid pathways by using knowledge distillation. To this end, we employ two different kinds of loss terms; one is the cross-entropy loss and the other is the knowledge distillation loss.

The cross-entropy loss of the main pathway is given by

client train with private data

$$\mathcal{L}_L = \text{CrossEntropy}(q_L, y), \tag{5}$$

while the overall cross-entropy loss of the hybrid pathways is defined as

$$\mathcal{L}^{\text{CE}}_H = \frac{1}{M-1} \sum_{m=1}^{M-1} \text{CrossEntropy}(q^m_H, y). \tag{6}$$

从global model各层提取出的softmax与真实y的CE loss

On the other hand, we encourage the representations of individual hybrid pathways to be similar to the main branch of the local network and employ the following knowledge distillation loss additionally:

各层提取出的sclaed softmax与local model softmax的 KL loss

$$\mathcal{L}^{\text{KL}}_H = \frac{1}{M-1} \sum_{m=1}^{M-1} \text{KL}(\tilde{q}^m_H, \tilde{q}_L), \tag{7}$$

where $\text{KL}(\cdot, \cdot)$ denotes the Kullback-Leibler (KL) divergence between two normalized vectors, and $\tilde{q}$ is the temperature-scaled softmax output using the hyperparameter $\tilde{\tau}$.

The total loss function of the proposed method is given by

$$\mathcal{L} = \mathcal{L}_L + \lambda_1 \cdot \mathcal{L}^{\text{CE}}_H + \lambda_2 \cdot \mathcal{L}^{\text{KL}}_H, \tag{8}$$

where λ_1 and λ_2 are hyperparameters that determine the weights of individual terms.

The local model is optimized by backpropagation based on the loss function in (8). Note that we update the model parameters in the main pathway while the blocks from the global network in the hybrid pathways remain unchanged during the local updates.

4.4. Learning procedure

Our federated learning follows the standard protocol and starts from local updates in clients. Each client first builds a model with either pretrained or randomized parameters and updates the parameters for a small number of iterations using local data. The updated models are sent to the server, and the server aggregates the models by a simple model averaging. Since the communication between the server and the clients is not stable, only a small fraction of clients typically participate in each round of the training procedure. Finally, the server broadcasts the new model given by model averaging and initiates a new round. Note that FedMLB is a client-side optimization approach and there is no special operation in the server. Algorithm 1 describes the detailed learning procedure of FedMLB.

Table 1. Comparisons between FedMLB and the baselines on CIFAR-100 and Tiny-ImageNet for two different federated learning settings. For (a) moderate-scale experiments, the number of clients and the participation rate are set to 100 and 5%, respectively, while (b) large-scale experiments have 500 clients with 2% participation rate. The accuracy at the target round and the number of communication rounds to reach the target test accuracy are based on the exponential moving average with the momentum parameter 0.9. The arrows indicate whether the higher (↑) or the lower (↓) is better.

(a) Moderate-scale with Dir(0.3): 100 clients, 5% participation

Method	CIFAR-100				Tiny-ImageNet			
	Accuracy (%, ↑)		Rounds (#, ↓)		Accuracy (%, ↑)		Rounds (#, ↓)	
	500R	1000R	47%	53%	500R	1000R	38%	42%
FedAvg (McMahan et al., 2017)	41.88	47.83	924	1000+	33.94	35.42	1000+	1000+
FedMLB	**47.39**	**54.58**	**488**	**783**	**37.20**	**40.16**	**539**	1000+
FedAvgM (Hsu et al., 2019)	46.98	53.24	515	936	36.10	38.36	794	1000+
FedAvgM + FedMLB	**53.02**	**58.97**	**349**	**499**	**40.93**	**43.52**	**380**	**642**
FedADAM (Reddi et al., 2021)	47.07	54.19	499	947	**36.98**	40.60	647	1000+
FedADAM + FedMLB	**48.59**	**58.23**	**472**	**645**	35.81	**42.90**	**552**	**873**
FedDyn (Acar et al., 2021)	48.38	55.78	425	735	37.35	41.17	573	1000+
FedDyn + FedMLB	**57.33**	**61.81**	**299**	**377**	**43.05**	**46.55**	**324**	**446**

(b) Large-scale with Dir(0.3): 500 clients, 2% participation

Method	CIFAR-100				Tiny-ImageNet			
	Accuracy (%, ↑)		Rounds (#, ↓)		Accuracy (%, ↑)		Rounds (#, ↓)	
	500R	1000R	36%	40%	500R	1000R	26%	32%
FedAvg (McMahan et al., 2017)	29.87	37.48	858	1000+	23.63	29.48	645	1000+
FedMLB	**32.03**	**42.61**	**642**	**800**	**28.39**	**33.67**	**429**	**710**
FedAvgM (Hsu et al., 2019)	31.80	40.54	724	955	26.75	33.26	457	836
FedAvgM + FedMLB	**36.21**	**47.75**	**496**	**636**	**32.00**	**37.53**	**307**	**500**
FedADAM (Reddi et al., 2021)	36.07	47.04	480	653	29.65	35.91	345	642
FedADAM + FedMLB	**38.28**	**52.68**	**442**	**527**	**32.14**	**39.54**	**311**	**524**
FedDyn (Acar et al., 2021)	31.58	41.02	691	927	24.35	29.54	595	1000+
FedDyn + FedMLB	**36.50**	**49.65**	**478**	**611**	**30.77**	**37.68**	**384**	**541**

4.5. Discussion

FedMLB maintains the knowledge in the global model while accommodating new information from the client datasets in each communication round. This objective is similar to that of continual learning, and knowledge distillation (KD) is widely used to this end. However, vanilla KD-based methods (Lee et al., 2021; Yao et al., 2021) matches the representations at the logit level, which derives model updates primarily at deeper layers while the parameters in the lower layers are less affected. This issue can be alleviated by using the layer-wise KD techniques (Romero et al., 2015), but the independent supervisions at multiple layers may lead to inconsistent and restrictive updates of model parameters. Contrary to these two options, the proposed approach constructs separate pathways using the static network blocks of the global model and updates the representations in each block of the local model to induce the proper outputs of the network. At the same time, FedMLB effectively distributes the workload of the network across multiple blocks for adapting to the local data, which is helpful for maintaining the representations of the global model during local iterations.

Compared to the federated learning techniques that handle client heterogeneity by employing global gradient information for the local update, the proposed algorithm has the following major advantages. First, FedMLB does not require any additional communication overhead such as global gradient information (Karimireddy et al., 2020; Xu et al., 2021). Note that the increase in communication cost challenges many realistic federated learning applications involving clients with limited network bandwidths. Also, unlike (Karimireddy et al., 2020; Acar et al., 2021; Li et al., 2021; Yao et al., 2021), the clients are not supposed to store their local states or historical information of the model, which is particularly desirable for the low-rate participation situations in federated learning.

Meanwhile, FedMLB incurs a moderate increase of computational cost due to the backpropagation through the additional branches. However, it achieves impressive accuracy with fewer communication rounds compared to the baselines. The performance of FedMLB is particularly good with a relatively large number of local iterations, which is helpful for reducing the number of communication rounds even further to achieve the target accuracy.

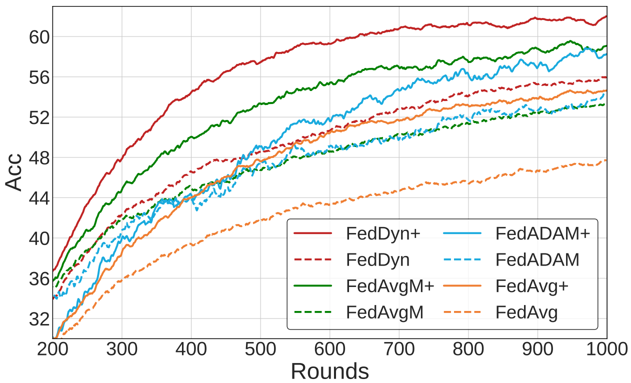


Figure 2. The convergence plots of FedMLB and other federated learning baselines on CIFAR-100. The + symbol indicates the incorporation of FedMLB. The number of clients, the participation rate, and the symmetric Dirichlet parameter are set to 100, 5%, and 0.3, respectively.

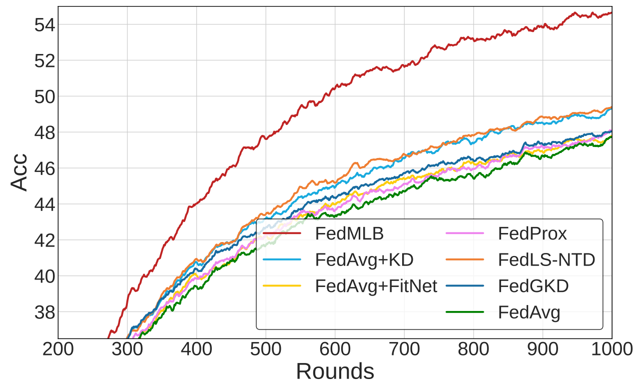


Figure 3. The convergence plots of FedMLB and other local optimization techniques on CIFAR-100. The number of clients, the participation rate, and the symmetric Dirichlet parameter are set to 100, 5%, and 0.3, respectively.

5. Experiments

This section demonstrates the effectiveness of FedMLB by incorporating it into various baseline algorithms for federated learning.

5.1. Experimental setup

Datasets and baselines We conduct a set of experiments on the CIFAR-100 and Tiny-ImageNet (Le & Yang, 2015) datasets. To simulate non-iid data, we sample examples with heterogeneous label ratios using symmetric Dirichlet distributions parametrized by two different concentration parameters $\{0.3, 0.6\}$, following (Hsu et al., 2019). We maintain the training dataset sizes balanced, so each client holds the same number of examples. For comprehensive evaluation, we employ several state-of-the-art federated learning techniques including FedAvg (McMahan et al., 2017), FedAvgM (Hsu et al., 2019), FedADAM (Reddi et al., 2021), and FedDyn (Acar et al., 2021). We also compare with existing regularization-based approaches such as FedProx (Li et al., 2020a), FedLS-NTD (Lee et al., 2021), and FedGKD (Yao et al., 2021). We choose a ResNet-18 (He et al., 2016) as the backbone network for all benchmarks, but replace the batch normalization with group normalization as suggested in (Hsieh et al., 2020).

Evaluation metrics To evaluate generalization performance of each algorithm, we use the whole test set in the CIFAR-100 (Krizhevsky et al., 2009) and Tiny-ImageNet datasets. Since both convergence speed as well as final performance are important metrics for federated learning as discussed in (Al-Shedivat et al., 2021), we measure the performance attained at two specific rounds and the number of required rounds to achieve the desired levels of the target accuracy. For the methods that fail to accomplish the target accuracy within the maximum communication rounds, we append a + sign to the number indicating the communication round.

Implementation details We use PyTorch (Paszke et al., 2019) to implement the proposed method and other baselines. FedMLB divides its backbone network, ResNet-18, into six blocks based on the depths of its layers and feature map sizes; each of conv1, conv2_x, conv3_x, conv4_x, conv5_x, and fc layers constitutes a single block. Following (Acar et al., 2021; Xu et al., 2021), we adopt the SGD optimizer for local updates with the learning rate of 0.1 for all benchmarks, except for the FedADAM whose learning rate is set to 0.01. We apply the exponential decay to the local learning rate with the parameter of 0.998. There is no momentum in the local SGD but the weight decay with a factor of 0.001 is employed to prevent overfitting. We also perform gradient clipping to stabilize the algorithms. The number of local training epochs is set to 5, and the batch size is determined to make the total number of iterations for local updates 50 for all experiments unless specified otherwise. The global learning rate is 1 for all methods except for FedADAM with 0.01. We list the details of the hyperparameters specific to FedMLB and the baseline algorithms in Appendix A.

5.2. Main results

FedMLB with server-side optimization techniques We first present the performance of the proposed approach, FedMLB, on CIFAR-100 and Tiny-ImageNet based on four federated learning baselines that perform server-side optimizations. Our experiments have been performed on two different settings; one is with moderate-scale, which involves 100 devices with 5% participation rate per round, and the other is with a large number of clients, 500 with 2% participation rate. Note that the number of clients in

Table 2. Comparison between FedMLB and the baselines based on other local objectives on CIFAR-100 with two different federated learning settings. The accuracy at the target round and the number of communication rounds to reach the target test accuracy are based on the exponential moving average with the momentum parameter 0.9.

Method	Dir(0.3), 100 clients, 5%				Dir(0.3), 500 clients , 2%			
	Accuracy (%, ↑)		Rounds (#, ↓)		Accuracy (%, ↑)		Rounds (#, ↓)	
	500R	1000R	40%	48%	500R	1000R	30%	36%
FedAvg (McMahan et al., 2017)	41.88	47.83	428	1000+	29.87	37.48	504	858
FedAvg + KD (Hinton et al., 2014)	42.99	49.17	389	842	29.83	37.65	505	859
FedAvg + FitNet (Romero et al., 2015)	42.04	47.67	419	1000+	29.92	37.63	503	860
FedProx (Li et al., 2020a)	42.03	47.93	419	1000+	29.28	36.16	533	966
FedLS-NTD (Lee et al., 2021)	43.22	49.29	386	825	28.66	35.99	546	1000+
FedGKD (Yao et al., 2021)	42.28	47.96	397	1000+	29.27	37.25	530	896
FedMLB (ours)	**47.39**	**54.58**	**339**	**523**	**32.03**	**42.61**	**446**	**642**

Table 3. Effect of more local iterations, $K = 100$ and 200, for FedMLB and the baselines with other local objectives on CIFAR-100 with 100 clients and 5% participation rate. The accuracy at the target round and the number of communication rounds to reach the target test accuracy are based on the exponential moving average with the momentum parameter 0.9.

Method	$K = 100$ (Dir(0.3), 100 clients, 5%)				$K = 200$ (Dir(0.3), 100 clients, 5%)			
	Accuracy (%, ↑)		Rounds (#, ↓)		Accuracy (%, ↑)		Rounds (#, ↓)	
	500R	1000R	40%	48%	500R	1000R	40%	48%
FedAvg (McMahan et al., 2017)	41.92	48.15	398	987	41.45	49.25	433	897
FedAvg + KD (Hinton et al., 2014)	42.58	49.15	385	905	42.84	51.48	381	751
FedAvg + FitNet (Romero et al., 2015)	39.48	45.80	531	1000+	37.17	43.97	651	1000+
FedProx (Li et al., 2020a)	42.01	48.17	391	992	41.30	48.67	458	959
FedLS-NTD (Lee et al., 2021)	41.34	47.09	423	1000+	39.82	46.72	508	1000+
FedGKD (Yao et al., 2021)	42.04	46.79	387	1000+	42.26	48.85	387	889
FedMLB (ours)	**52.53**	**58.52**	**223**	**359**	**53.12**	**58.91**	**183**	**325**

the large-scale setting is 5 times more than the moderate-scale experiment, which reduces the number of examples per client by 80%. Table 1 demonstrates that FedMLB improves accuracy and convergence speed by significant margins consistently on all the four baselines for most cases. Figure 2 also illustrates the effectiveness of FedMLB when it is combined with the four baseline methods. Note that the overall performance in the large-scale setting is lower than the case with a moderate number of clients. This is because, as the number of training data per client decreases, each client has even more distinct properties and is prone to drift. Nevertheless, we observe that FedMLB outperforms the baseline methods consistently on all benchmarks.

Comparisons with other local objectives To understand the effectiveness of FedMLB compared to other local optimization techniques, we compare our objective with the following two baselines: 1) employing the vanilla knowledge distillation for regularization (FedAvg + KD) (Hinton et al., 2014)), and 2) adopting knowledge distillation on block-wise features between the local and global model (FedAvg + FitNet (Romero et al., 2015)).

Table 2 illustrates the outstanding performance of our multi-level branched regularization using online knowledge distillation on CIFAR-100, and Figure 3 visualizes the convergence curves of all compared algorithms with different local objectives. One noticeable result is that the baseline methods with knowledge distillation only achieve marginal gains or sometimes degrade accuracy. This is partly because the predictions of the downloaded global model are not fully-trustworthy during training, especially in early communication rounds. Therefore, merely simulating the outputs of the global model is suboptimal and hampers the learning process at the local model. In this respect, FedMLB is more robust to heterogeneous characteristics of clients and more flexible to learn the new knowledge in local models. Note that, FedGKD requires 1.5 times communication costs compared to other methods since the server transmits the historical global model along with the latest model for server-to-client communication.

5.3. Effect of more local iterations

The increase of local iterations under a heterogeneous environment is beneficial because we can reduce the number of communication rounds between the server and clients. Table 3 presents the results of FedMLB and the baselines with more local iterations, *i.e.*, $K \in \{100, 200\}$, on CIFAR-100. The results demonstrate that FedMLB outperforms the compared methods by significant margins. An interesting observation is that the baseline methods fail to benefit from additional iterations. This is because the increase of local iterations is prone to result in more divergence across mul-

Table 4. Ablation study results from two different compositions of the hybrid pathways on CIFAR-100 in two federated learning settings. The accuracy at the target round and the number of communication rounds to reach the target test accuracy are based on the exponential moving average with the momentum parameter 0.9.

Method	Dir(0.3), 100 clients, 5%				Dir(0.3), 500 clients, 2%			
	Accuracy (%, ↑)		Rounds (#, ↓)		Accuracy (%, ↑)		Rounds (#, ↓)	
	500R	1000R	40%	48%	500R	1000R	30%	36%
FedAvg (McMahan et al., 2017)	41.88	47.83	428	1000+	29.87	37.48	504	858
FedMLB$_{\mathrm{G}\rightarrow\mathrm{L}}$	42.41	47.40	386	1000+	28.53	35.46	571	1000+
FedMLB$_{\mathrm{L}\rightarrow\mathrm{G}}$ (ours)	**47.39**	**54.58**	**339**	**523**	**32.03**	**42.61**	**446**	**642**

Table 5. Accuracy after $1K$ rounds of FedMLB with various configurations of the hybrid pathways on CIFAR-100 in the moderate-scale setting. The symmetric Dirichlet parameter is set to 0.3.

Hybrid pathway index 1	2	3	4	5	Acc.
✓	✓	✓	✓	✓	54.58
✓	✓	✓	✓		55.18
✓	✓	✓			55.03
✓	✓				54.16
✓					50.79
				✓	48.45
			✓	✓	46.78
		✓	✓	✓	51.73
	✓	✓	✓	✓	52.65

Table 6. Effect of the number of hybrid pathways employed for FedMLB on CIFAR-100 in the moderate-scale setting. The average accuracy and the standard deviation are computed over all possible combinations with the same number of hybrid pathways. We measure the accuracy after $1K$ rounds, where the symmetric Dirichlet parameter is set to 0.3 for non-iid sampling.

Number of pathways	1	2	3	4	5
Average accuracy	47.89	51.86	52.89	53.69	**54.58**
Standard deviation	3.99	2.45	1.78	1.43	-

Table 7. Sensitivity of FedMLB to the weights of the two regularization loss terms with respect to the accuracy after $1K$ round on CIFAR-100 in the moderate-scale setting. The symmetric Dirichlet parameter is set to 0.3.

λ_1 \ λ_2	0	1	2	3
0	47.83	53.27	54.51	54.21
1	52.10	54.58	55.52	**56.32**
2	52.37	53.05	54.28	54.16
3	49.54	53.00	54.34	53.57

Table 8. Sensitivity of FedMLB to τ' with respect to the accuracy after $1K$ rounds on CIFAR-100 in the moderate-scale setting with two different values of the symmetric Dirichlet parameter.

τ'	0.75	1	1.5	2	3	5
Dir(0.3)	53.90	54.58	53.87	53.98	53.60	53.68
Dir(0.6)	56.10	56.70	56.72	56.87	54.98	54.02

tiple client models and eventually leads to degradation of performance. In contrast, FedMLB consistently improves its accuracy and convergence speed substantially compared to the results with 50 iterations shown in Table 2. Although the accuracies with 100 and 200 iterations are similar, the numbers of required iterations to achieve 40% and 48% are noticeably smaller with 200 iterations. These results imply that FedMLB handles the client drift issue effectively.

5.4. Analysis of auxiliary branches

Effectiveness of local-to-global pathways Each hybrid pathway in FedMLB is composed of a set of local blocks followed by a global subnetwork. To show the effectiveness of the current design of the hybrid pathways, we evaluate the performance of the opposite architecture design, the local model with multi-level global-to-local pathways. As in the original version of FedMLB, the newly considered model denoted by FedMLB$_{\mathrm{G}\rightarrow\mathrm{L}}$ also updates the parameters in the local blocks only. Table 4 presents that the knowledge distillation with the hybrid pathways stemming from local blocks to global ones outperforms the opposite composition method. The reason is that FedMLB$_{\mathrm{G}\rightarrow\mathrm{L}}$ constrains the output of each local blocks excessively and reduces the flexibility of the main pathway in the local network significantly. Although the new strategy is helpful for preserving the knowledge in the global model, it interferes learning new knowledge.

Effect of multi-level branches FedMLB employs the features from multi-level auxiliary branches to compute the cross-entropy loss $\mathcal{L}_H^{\mathrm{CE}}$ and KL-divergence loss $\mathcal{L}_H^{\mathrm{KL}}$. Table 5 illustrates that the pathways grafted from the lower layers are generally more helpful for accuracy gains. Also, according to Table 6, the accuracy of FedMLB generally improves as we increase the number of pathways.

5.5. Effect of hyperparameters in FedMLB

Ablation study for loss function To show the effectiveness of the cross-entropy loss of the hybrid pathways $\mathcal{L}_H^{\mathrm{CE}}$ and the KL-divergence loss $\mathcal{L}_H^{\mathrm{KL}}$, we conduct the comprehensive experiments by varying λ_1 and λ_2 in (8), which controls the weight of each of the two loss terms. Table 7 shows that both of the loss terms contribute to performance gains while the KL-divergence loss is more critical than the

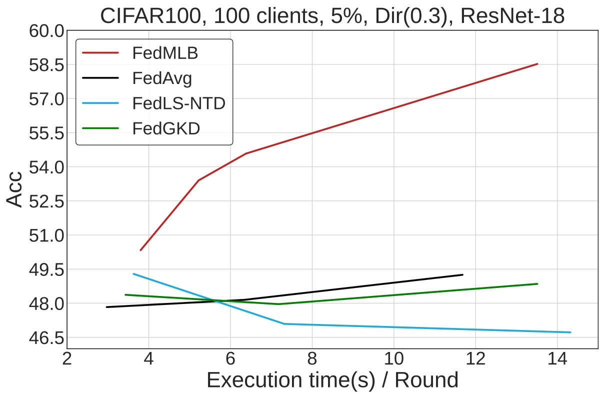


Figure 4. Performance of algorithms by varying their local computational costs controlled by the number of local iterations while maintaining the total communication costs. Accuracies are measured after $1K$ rounds on CIFAR-100 using ResNet18.

cross-entropy loss in the hybrid pathways. Note that we set $\lambda_1 = \lambda_2 = 1$ in our experiment.

Softmax function temperature in knowledge distillation The temperature parameter τ' in (7) controls the smoothness of the softmax function output for KL divergence loss, $\mathcal{L}_H^{\text{KL}}$. Table 8 presents that the performance of FedMLB is consistent with respect to the variations of τ' in the two different values of the symmetric Dirichlet parameter.

5.6. Local computation overheads

While FedMLB requires additional computation compared to the baselines under the same number of local epochs, its benefit outweighs its cost as illustrated in Figure 4. FedMLB outperforms other methods at the same computational cost and the gap gets more significant as they have more local iterations. An important observation from Figure 4 is that FedLS-NTD and FedGKD are inconsistent with the number of local iterations compared to FedMLB and FedAVG. Also, note that the performance of FedMLB improves substantially with more iterations, which is effective for reducing the communication cost, which is critical in federated learning. Also, Table 5 presents that removing a couple of pathways from deeper layers sometimes improves accuracy, which means there still exists room for optimization in terms of computational cost.

5.7. Experiments with other backbone models

To confirm the generality of FedMLB with respect to backbone networks, we conduct experiments with additional architectures, which include VGG-9 (Simonyan & Zisserman, 2015), MobileNet (Sandler et al., 2018), ShuffleNet (Zhang et al., 2018), and SqueezeNet (Iandola et al., 2016). VGG

Table 9. Results with different CNN backbone architectures on CIFAR-100.

Architecture	Dir(0.3), 100 clients, 5%			
	FedAvg	FedLS-NTD	FedGKD	FedMLB
VGG-9	47.04	51.37	48.62	**54.54**
MobileNet	38.52	47.66	47.72	**48.34**
ShuffleNet	37.04	39.27	38.47	**42.29**
SqueezeNet	39.86	39.56	42.28	**42.71**
ResNet-18	47.83	49.29	47.96	**54.58**

is widely used network without skip connections while MobileNet, ShuffleNet and SqueezeNet are lightweight networks suitable for edge devices. As for implementation, since these modern deep neural networks are typically modularized, we typically branch a pathway after a module, *e.g.*, ResBlock (no branches from the layers enclosed by a skip connection). Table 9 shows that FedMLB clearly outperforms other algorithms regardless of the backbone architectures.

6. Conclusion

We presented a practical solution to improve the performance of federated learning, where a large number of clients with heterogeneous data distributions and limited participation rates are involved in the learning process. To address the critical limitations, we proposed a novel regularization technique via online knowledge distillation. Our approach employs multi-level hybrid branched networks, which reduces the drift of the representations in the local models from the feature space of the global model. The proposed federated learning framework has the following two desirable properties; it requires no additional communication cost and spends no extra memory to store the history of local states. We demonstrated that the proposed approach, referred to as FedMLB, achieves outstanding performance in terms of accuracy and efficiency, through a comprehensive evaluation on multiple standard benchmarks under various environments.

Acknowledgments This work was partly supported by Samsung Electronics Co., Ltd., and by the NRF Korea grant [No. 2022R1A2C3012210, Knowledge Composition via Task-Distributed Federated Learning] and the IITP grants [2021-0-02068, Artificial Intelligence Innovation Hub; 2021-0-01343, Artificial Intelligence Graduate School Program (Seoul National University)] funded by the Korea government (MSIT).

References

Acar, D. A. E., Zhao, Y., Matas, R., Mattina, M., Whatmough, P., and Saligrama, V. Federated learning based on dynamic regularization. In *ICLR*, 2021.

Al-Shedivat, M., Gillenwater, J., Xing, E., and Rostamizadeh, A. Federated learning via posterior averaging: A new perspective and practical algorithms. In *ICLR*, 2021.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *CVPR*, 2016.

Hinton, G., Vinyals, O., and Dean, J. Distilling the knowledge in a neural network. In *NIPSW*, 2014.

Hsieh, K., Phanishayee, A., Mutlu, O., and Gibbons, P. The non-iid data quagmire of decentralized machine learning. In *ICML*, 2020.

Hsu, T.-M. H., Qi, H., and Brown, M. Measuring the effects of non-identical data distribution for federated visual classification. *arXiv preprint arXiv:1909.06335*, 2019.

Iandola, F. N., Han, S., Moskewicz, M. W., Ashraf, K., Dally, W. J., and Keutzer, K. Squeezenet: Alexnet-level accuracy with 50x fewer parameters and $<$ 0.5 MB model size. In *arXiv preprint arXiv:1602.07360*, 2016.

Karimireddy, S. P., Kale, S., Mohri, M., Reddi, S. J., Stich, S. U., and Suresh, A. T. SCAFFOLD: stochastic controlled averaging for on-device federated learning. In *ICML*, 2020.

Khaled, A., Mishchenko, K., and Richtárik, P. First analysis of local GD on heterogeneous data. In *NeurIPSW*, 2019.

Krizhevsky, A., Hinton, G., et al. Learning multiple layers of features from tiny images. 2009.

Le, Y. and Yang, X. Tiny imagenet visual recognition challenge. *CS 231N*, 7(7):3, 2015.

Lee, G., Shin, Y., Jeong, M., and Yun, S.-Y. Preservation of the global knowledge by not-true self knowledge distillation in federated learning. *arXiv preprint arXiv:2106.03097*, 2021.

Li, Q., He, B., and Song, D. Model-contrastive federated learning. In *CVPR*, 2021.

Li, T., Sahu, A. K., Zaheer, M., Sanjabi, M., Talwalkar, A., and Smithy, V. FedDANE: A federated newton-type method. In *ACSCC*, 2019.

Li, T., Sahu, A. K., Zaheer, M., Sanjabi, M., Talwalkar, A., and Smith, V. Federated optimization in heterogeneous networks. In *MLSys*, 2020a.

Li, X., Huang, K., Yang, W., Wang, S., and Zhang, Z. On the convergence of fedavg on non-iid data. In *ICLR*, 2020b.

Lin, T., Kong, L., Stich, S. U., and Jaggi, M. Ensemble distillation for robust model fusion in federated learning. In *NeurIPS*, 2020.

McMahan, B., Moore, E., Ramage, D., Hampson, S., and y Arcas, B. A. Communication-efficient learning of deep networks from decentralized data. In *AISTATS*, 2017.

Park, D. Y., Cha, M.-H., Jeong, C., Kim, D. S., and Han, B. Learning student-friendly teacher networks for knowledge distillation. In *NeurIPS*, 2021.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al. Pytorch: An imperative style, high-performance deep learning library. In *NeurIPS*, 2019.

Reddi, S. J., Charles, Z., Zaheer, M., Garrett, Z., Rush, K., Konečnỳ, J., Kumar, S., and McMahan, H. B. Adaptive federated optimization. In *ICLR*, 2021.

Romero, A., Ballas, N., Kahou, S. E., Chassang, A., Gatta, C., and Bengio, Y. Fitnets: Hints for thin deep nets. In *ICLR*, 2015.

Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., and Chen, L.-C. Mobilenetv2: Inverted residuals and linear bottlenecks. In *CVPR*, 2018.

Seo, H., Park, J., Oh, S., Bennis, M., and Kim, S.-L. Federated knowledge distillation. *arXiv preprint arXiv:2011.02367*, 2020.

Simonyan, K. and Zisserman, A. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015.

Wang, J., Liu, Q., Liang, H., Joshi, G., and Poor, H. V. Tackling the objective inconsistency problem in heterogeneous federated optimization. In *NeurIPS*, 2020.

Wang, J., Charles, Z., Xu, Z., Joshi, G., McMahan, H. B., Al-Shedivat, M., Andrew, G., Avestimehr, S., Daly, K., Data, D., et al. A field guide to federated optimization. *arXiv preprint arXiv:2107.06917*, 2021.

Wang, S., Tuor, T., Salonidis, T., Leung, K. K., Makaya, C., He, T., and Chan, K. Adaptive federated learning in resource constrained edge computing systems. *IEEE Journal on Selected Areas in Communications*, 37(6): 1205–1221, 2019.

Xu, J., Wang, S., Wang, L., and Yao, A. C.-C. Fedcm: Federated learning with client-level momentum. *arXiv preprint arXiv:2106.10874*, 2021.

Yao, D., Pan, W., Dai, Y., Wan, Y., Ding, X., Jin, H., Xu, Z., and Sun, L. Local-global knowledge distillation in heterogeneous federated learning with non-iid data. *arXiv preprint arXiv:2107.00051*, 2021.

Yoon, T., Shin, S., Hwang, S. J., and Yang, E. Fedmix: Approximation of mixup under mean augmented federated learning. In *ICLR*, 2021.

Zhang, X., Zhou, X., Lin, M., and Sun, J. Shufflenet: An extremely efficient convolutional neural network for mobile devices. In *CVPR*, 2018.

Zhang, X., Hong, M., Dhople, S., Yin, W., and Liu, Y. FedPD: a federated learning framework with optimal rates and adaptivity to non-iid data. In *arXiv preprint arXiv:2005.11418*, 2020.

Zhao, Y., Li, M., Lai, L., Suda, N., Civin, D., and Chandra, V. Federated learning with non-iid data. *arXiv preprint arXiv:1806.00582*, 2018.

Zhu, Z., Hong, J., and Zhou, J. Data-free knowledge distillation for heterogeneous federated learning. In *ICML*, 2021.

A. Implementation Details

For the experiments on CIFAR-100, the number of local training epochs is 5, and the local learning rate is 0.1 except for 0.01 in FedADAM. We set the batch sizes of local updates to 50 and 10 for the experiments with 100 and 500 clients, respectively. The parameter for learning rate decay in each algorithm is set as 0.998. The global learning rate is 1 except for FedAdam, which adopts 0.01. For the Tiny-ImageNet experiments, we match the total number of local iterations with other benchmarks by setting the batch sizes of local update as 100 for 100 clients and 20 for 500 clients.

There are several hyperparameters specific to each of the existing algorithms, which are typically determined to achieve the best performance by referring to the setting in the original paper. For example, α in FedDyn is 0.1, τ in FedADAM is 0.001, and γ in FedGKD is 0.2. We select β in FedAvgM from $\{0.4, 0.6, 0.8\}$, and β in FedProx is from $\{0.1, 0.01, 0.001\}$. When incorporating FedMLB into the baselines, we inherit all their hyperparameter settings. For all experiments for FedMLB, both λ_1 and λ_2 are set to 1 while τ' is 1.

B. Additional Analysis

B.1. Auxiliary branches

FedMLB employs the features from multi-level auxiliary branches to compute the cross-entropy loss $\mathcal{L}_H^{\text{CE}}$ and the KL-divergence loss $\mathcal{L}_H^{\text{KL}}$. Figure 5 illustrates that the use of all auxiliary branches leads to the best performance and the proposed multi-level regularization contributes to additional performance gains. It also implies that the branches stemming from shallower local blocks are generally more helpful, which is consistent with the results in Table 6.

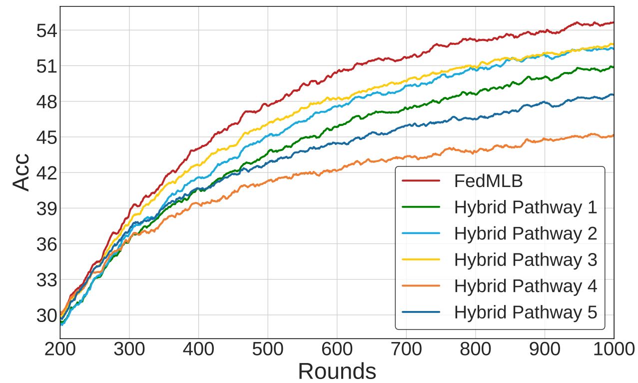


Figure 5. The benefit of each hybrid pathway to accuracy during training. The number of clients, the participation rate, and the symmetric Dirichlet parameter are set to 100, 5% and 0.3, respectively. The hybrid pathway index follows the notation in Figure 1.

B.2. Level of data heterogeneity

To demonstrate the generality of the proposed approach on the level of data heterogeneity, we test FedMLB with two different data partitioning strategies, which include the method based on a symmetric Dirichlet distribution with a higher concentration parameter (0.6) and the iid setting. Tables 10 and 11 verify that incorporating FedMLB into four different baseline methods improves accuracy and convergence speed with large margins for most cases.

B.3. Convergence

To further analyze the effectiveness of the proposed method, we investigate the convergence characteristics of several algorithms including FedMLB in diverse settings, which are configured by varying the number of clients, the level of data heterogeneity, and participation rate. The accuracies at each round on CIFAR-100 and Tiny-ImageNet are demonstrated in Figure 6 and 7, respectively. The results show that FedMLB is indeed helpful for improving accuracy throughout the training procedure, facilitating convergence.

Figure 8 also illustrates the convergence of FedMLB in comparison to other regularization-based methods. We observe the consistent and non-trivial improvements of FedMLB over FedAvg during training while other methods only achieve marginal gains compared to FedAvg or are even worse, especially in a more challenging condition with a less participation rate. Furthermore, in Figure 9 we notice that FedMLB continuously outperforms the baselines large margins even when we increase the number of local iterations, which is effective for reducing the communication cost.

Table 10. Comparisons of FedMLB and the baselines on CIFAR-100 and Tiny-ImageNet for two different federated learning settings, where the symmetric Dirichlet parameter is 0.6. The accuracy at the target round and the number of communication rounds to reach the target test accuracy are based on the exponential moving average with the momentum parameter 0.9. The arrows indicate whether higher (↑) or lower (↓) is better.

(a) Moderate-scale with Dir(0.6): 100 clients, 5% participation

Method	CIFAR-100				Tiny-ImageNet			
	Accuracy (%, ↑)		Rounds (#, ↓)		Accuracy (%, ↑)		Rounds (#, ↓)	
	500R	1000R	48%	52%	500R	1000R	38%	42%
FedAvg (McMahan et al., 2017)	43.43	48.71	926	1000+	36.15	37.72	1000+	1000+
FedMLB	**49.36**	**56.70**	**465**	**602**	**39.34**	**42.15**	**441**	**917**
FedAvgM (Hsu et al., 2019)	46.66	52.49	572	937	38.41	40.75	475	1000+
FedAvgM + FedMLB	**54.62**	**60.91**	**331**	**417**	**43.96**	**46.82**	**294**	**401**
FedADAM (Reddi et al., 2021)	50.81	56.95	427	569	40.13	42.75	415	813
FedADAM + FedMLB	**53.34**	**61.49**	**364**	**467**	**40.67**	**44.96**	415	**610**
FedDyn (Acar et al., 2021)	50.51	56.79	427	580	39.39	42.97	423	875
FedDyn + FedMLB	**57.51**	**62.43**	**298**	**355**	**43.35**	**47.43**	**294**	**412**

(b) Large-scale with Dir(0.6): 500 clients, 2% participation

Method	CIFAR-100				Tiny-ImageNet			
	Accuracy (%, ↑)		Rounds (#, ↓)		Accuracy (%, ↑)		Rounds (#, ↓)	
	500R	1000R	36%	44%	500R	1000R	26%	32%
FedAvg (McMahan et al., 2017)	29.36	36.36	966	1000+	24.48	29.94	600	1000+
FedMLB	**33.74**	**43.53**	**571**	1000+	**28.97**	**35.08**	**415**	**650**
FedAvgM (Hsu et al., 2019)	32.44	41.40	680	1000+	24.65	31.54	575	1000+
FedAvgM + FedMLB	**38.35**	**49.65**	**421**	**690**	**32.33**	**37.60**	**308**	**483**
FedADAM (Reddi et al., 2021)	37.33	47.73	463	756	31.40	37.03	**286**	533
FedADAM + FedMLB	**39.57**	**53.53**	**402**	**621**	**33.71**	**41.15**	292	**444**
FedDyn (Acar et al., 2021)	31.63	41.58	677	1000+	26.42	31.80	485	1000+
FedDyn + FedMLB	**38.90**	**52.72**	**440**	**639**	**32.52**	**38.28**	**350**	**712**

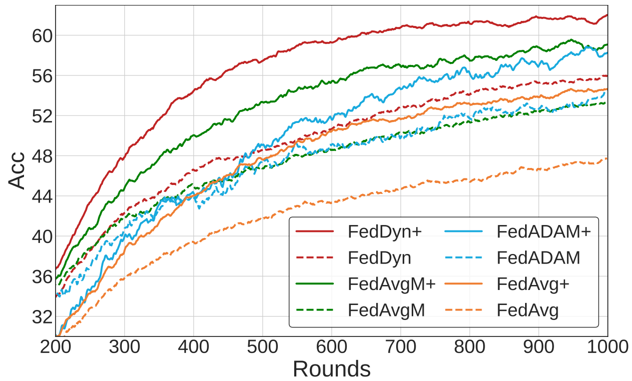
Table 11. Comparisons of FedMLB and the baselines on CIFAR-100 and Tiny-ImageNet for two different iid federated learning settings. The accuracy at the target round and the number of communication rounds to reach the target test accuracy are based on the exponential moving average with the momentum parameter 0.9. The arrows indicate whether higher (↑) or lower (↓) is better.

(a) Moderate-scale with IID: 100 clients, 5% participation

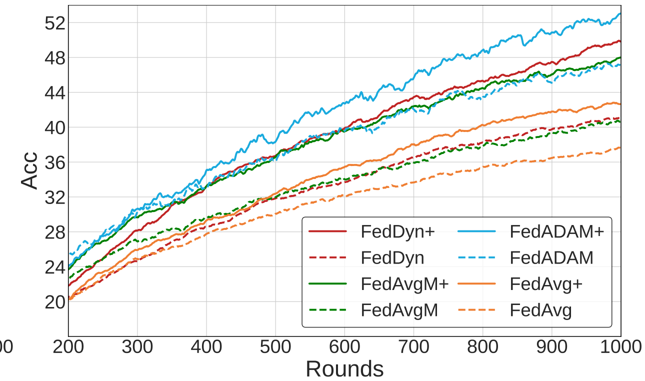
Method	CIFAR-100				Tiny-ImageNet			
	Accuracy (%, ↑)		Rounds (#, ↓)		Accuracy (%, ↑)		Rounds (#, ↓)	
	500R	1000R	48%	52%	500R	1000R	38%	42%
FedAvg (McMahan et al., 2017)	43.60	48.01	997	1000+	37.96	38.92	504	1000+
FedAvg + FedMLB	**50.12**	**56.40**	**426**	**584**	**40.69**	**42.98**	**376**	**640**
FedAvgM (Hsu et al., 2019)	47.43	52.83	532	880	39.79	41.34	346	1000+
FedAvgM + FedMLB	**55.29**	**61.16**	**294**	**377**	**44.02**	**47.03**	**271**	**382**
FedADAM (Reddi et al., 2021)	54.35	60.35	303	416	43.54	46.12	**257**	377
FedADAM + FedMLB	**57.13**	**62.58**	**285**	**363**	**44.27**	**47.36**	276	**372**
FedDyn (Acar et al., 2021)	50.37	56.88	397	592	39.49	42.42	350	848
FedDyn + FedMLB	**56.97**	**61.41**	**298**	**366**	**44.28**	**46.62**	**277**	**361**

(b) Large-scale with IID: 500 clients, 2% participation

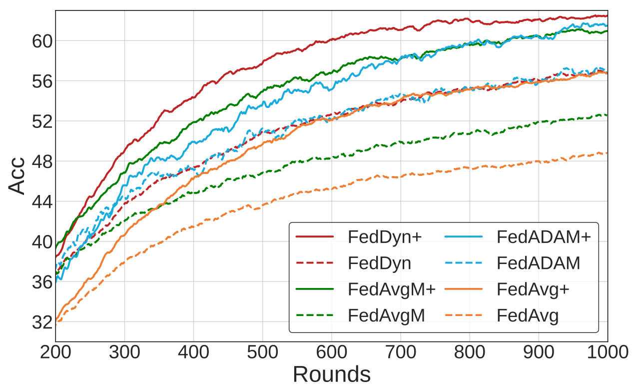
Method	CIFAR-100				Tiny-ImageNet			
	Accuracy (%, ↑)		Rounds (#, ↓)		Accuracy (%, ↑)		Rounds (#, ↓)	
	500R	1000R	36%	44%	500R	1000R	26%	32%
FedAvg (McMahan et al., 2017)	29.96	36.93	903	1000+	23.25	28.92	701	1000+
FedAvg + FedMLB	**34.60**	**44.95**	**549**	**935**	**28.27**	**35.51**	**435**	**689**
FedAvgM (Hsu et al., 2019)	32.47	41.04	679	1000+	27.52	34.08	445	800
FedAvgM + FedMLB	**38.60**	**50.52**	**418**	**676**	**33.51**	**39.37**	**281**	**444**
FedADAM (Reddi et al., 2021)	38.32	48.70	430	712	33.30	37.55	252	441
FedADAM + FedMLB	**42.48**	**55.24**	**338**	**539**	**35.34**	**41.75**	**250**	**384**
FedDyn (Acar et al., 2021)	35.77	47.34	509	806	24.79	31.75	565	1000+
FedDyn + FedMLB	**39.37**	**53.11**	**429**	**619**	**30.46**	**37.89**	**384**	**540**



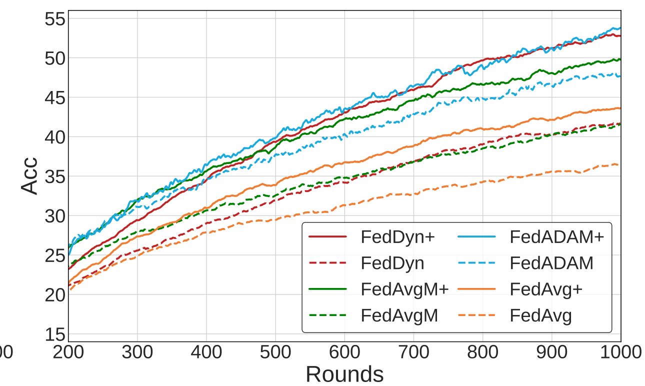
(a) Dir(0.3), 100 clients, 5% participation



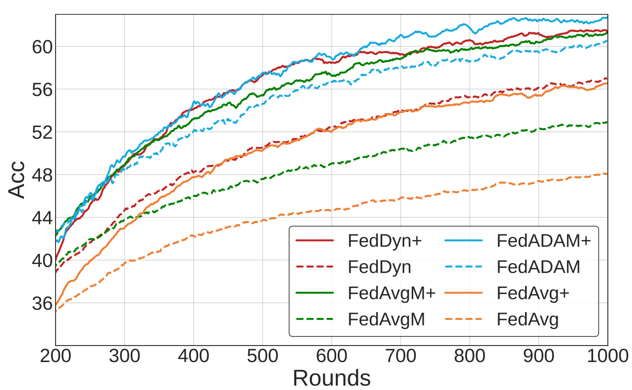
(b) Dir(0.3), 500 clients, 2% participation



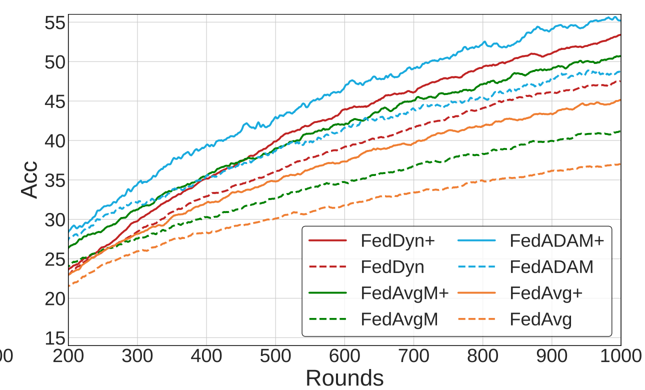
(c) Dir(0.6), 100 clients, 5% participation



(d) Dir(0.6), 500 clients 2% participation

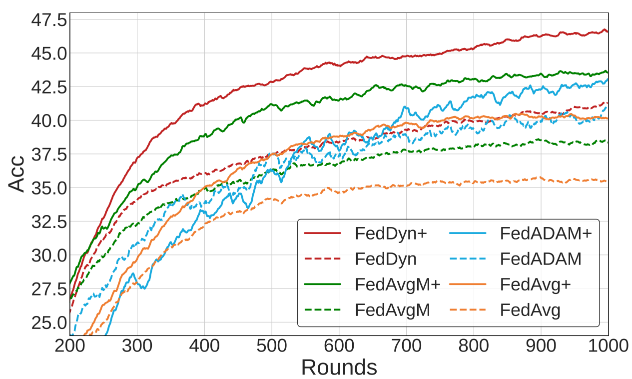


(e) IID, 100 clients, 5% participation

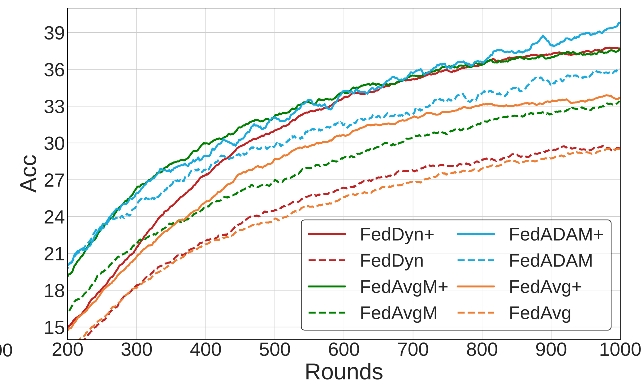


(f) IID, 500 clients, 2% participation

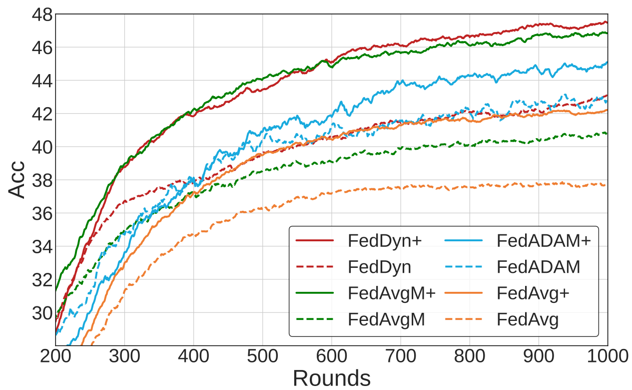
Figure 6. The convergence of several federated learning algorithms on CIFAR-100 in various settings. Note that + symbol indicates the incorporation of FedMLB.



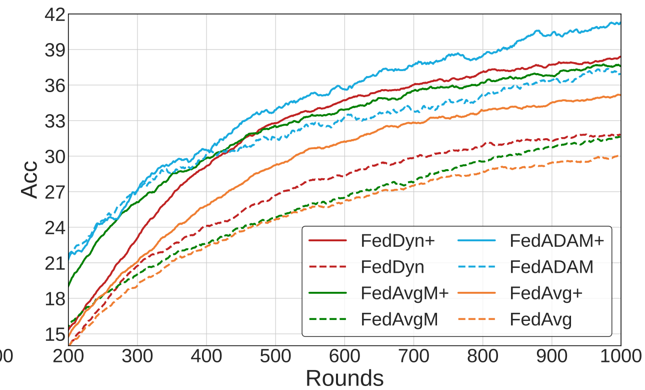
(a) Dir(0.3), 100 clients, 5% participation



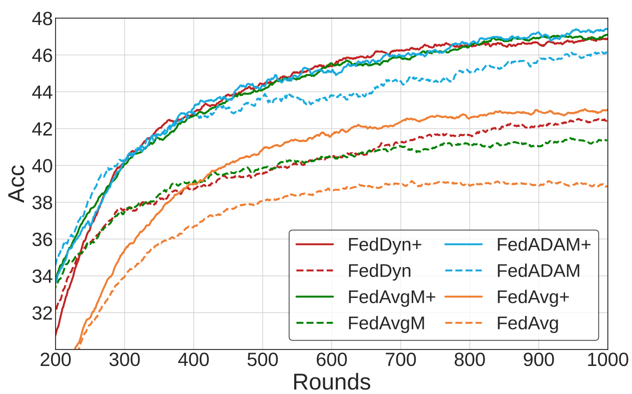
(b) Dir(0.3), 500 clients, 2% participation



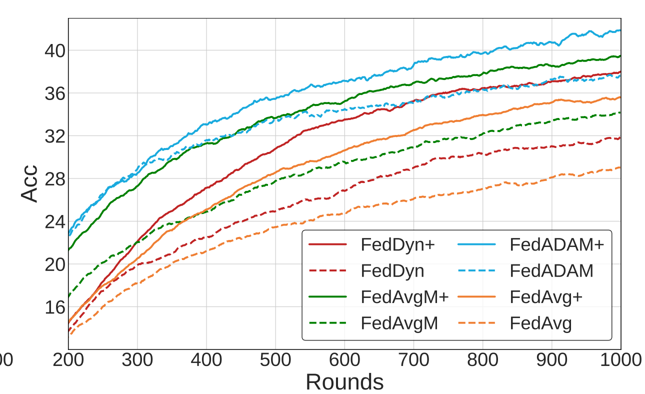
(c) Dir(0.6), 100 clients, 5% participation



(d) Dir(0.6), 500 clients, 2% participation

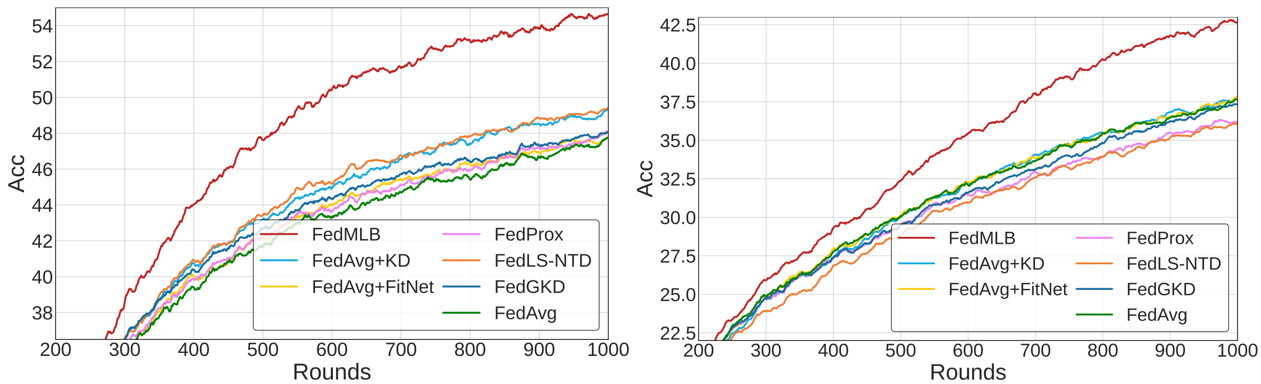


(e) IID, 100 clients, 5% participation



(f) IID, 500 clients, 2% participation

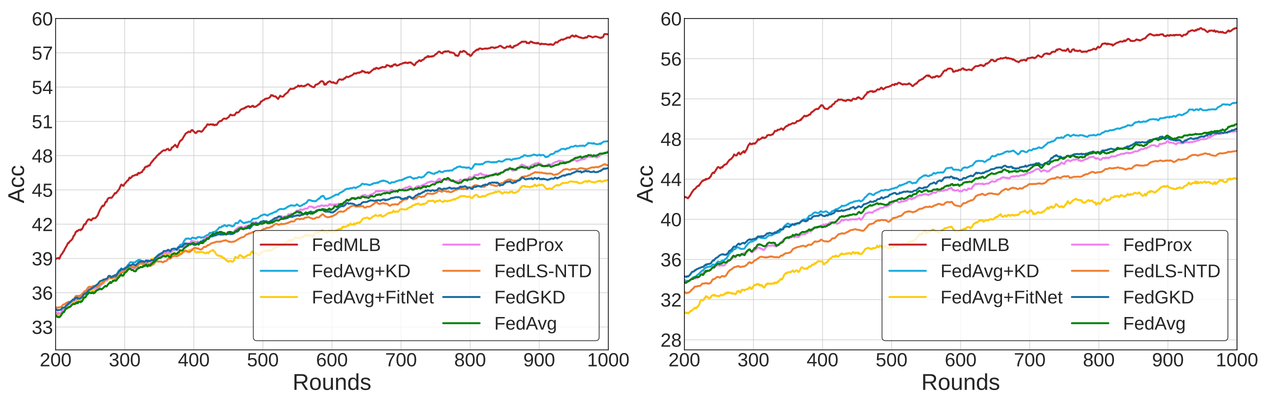
Figure 7. The convergence of several federated learning algorithms on Tiny-ImageNet in various settings. Note that + symbol indicates the incorporation of FedMLB.



(a) Dir(0.3), 100 clients, 5% participation

(b) Dir(0.3), 500 clients, 2% participation

Figure 8. The convergence of FedMLB and other local optimization approaches on CIFAR-100.



(a) $K = 100$ (Dir(0.3), 100 clients, 5%)

(b) $K = 200$ (Dir(0.3), 100 clients, 5%)

Figure 9. The convergence of FedMLB and other local optimization approaches on CIFAR-100 with two different numbers of local iterations, K.