Tackling Resource-Constrained and Data-Heterogeneity in Federated Learning with Double-Weight Sparse Pack

Qiantao Yang[1], Liquan Chen[1, 2*], Mingfu Xue[3], Songze Li[1]

[1] School of Cyber Science and Engineering, Southeast University, China
[2] Purple Mountain Laboratories, China
[3] School of Communication and Electronic Engineering, East China Normal University, China
{yangqiantao, lqchen, songzeli}@seu.edu.cn, mfxue@cee.ecnu.edu.cn

Abstract

Federated learning has drawn widespread interest from researchers, yet the data heterogeneity across edge clients remains a key challenge, often degrading model performance. Existing methods enhance model compatibility with data heterogeneity by splitting models and knowledge distillation. However, they neglect the insufficient communication bandwidth and computing power on the client, failing to strike an effective balance between addressing data heterogeneity and accommodating limited client resources. To tackle this limitation, we propose a personalized federated learning method based on cosine sparsification parameter packing and dual-weighted aggregation (FedCSPACK), which effectively leverages the limited client resources and reduces the impact of data heterogeneity on model performance. In FedCSPACK, the client packages model parameters and selects the most contributing parameter packages for sharing based on cosine similarity, effectively reducing bandwidth requirements. The client then generates a mask matrix anchored to the shared parameter package to improve the alignment and aggregation efficiency of sparse updates on the server. Furthermore, directional and distribution distance weights are embedded in the mask to implement a weighted-guided aggregation mechanism, enhancing the robustness and generalization performance of the global model. Extensive experiments across four datasets using ten state-of-the-art methods demonstrate that FedCSPACK effectively improves communication and computational efficiency while maintaining high model accuracy.

Code — https://github.com/NigeloYang/FedCSPACK

Introduction

Federated learning (FL) has attracted attention for its ability to leverage the massive data of clients while protecting privacy by sharing model parameters between clients without exchanging raw data for collaborative training of models (McMahan et al. 2017; Alam et al. 2022). However, the data generated and collected by edge client devices across different geographical locations exhibit significant heterogeneity, resulting in Non-Independent and Identically Distributed (Non-IID) characteristics in their data. Its computing, storage, bandwidth, and other resources also vary, which causes heterogeneity in system resources. In this context, a single global model struggles to encompass all client-side local datasets, and traditional methods using a single global model often result in significant deviations in the distribution of local data across clients, leading to slow convergence, poor inference performance, and even the inability to deploy in resource-constrained environments (Li et al. 2021, 2022; Oh, Kim, and Yun 2022). To solve the problem of heterogeneity in FL, some works propose personalized federated learning (PFL) (Huang et al. 2021; Tan et al. 2022a), which tailors a local model for each client according to their data distribution and system resources to enhance the model's ability to solve the heterogeneity problem.

Existing works demonstrate that the essence of model performance degradation is client drift caused by data heterogeneity. When significant aggregation discrepancies exist between the local model and global model, the aggregated global model fails to adapt to the heterogeneous data distributions across clients adequately. To reduce client drift, (Karimireddy et al. 2020) uses the difference between the old and global rounds of local models to compensate for the current round of local updates. (Li et al. 2020; Acar et al. 2021; Zhang et al. 2022) add regularization to the local loss function to facilitate the local model to be close to the global model. There are also some model aggregation strategies (Luo and Wu 2022; Zhang et al. 2023b) and knowledge distillation (Tan et al. 2022b; Xu, Tong, and Huang 2023). While these works improve the performance of models in the face of data heterogeneity, they overlook the problem of limited client resources.

In FL, clients are typically composed of physical devices with varying processors, memory, and bandwidth, which creates resource differences between devices (system resource heterogeneity). Since FL relies on frequent model interaction, limited resources clients often struggle to keep pace with the collaborative training of a complex global model. This leads to communication bottlenecks, computational delays, and participation imbalance (Caldas et al. 2018a; Diao, Ding, and Tarokh 2021). Some works reduce resource constraints and improve the interaction efficiency and overall performance of the model at both ends by sharing sparse model parameters, optimizing model architecture, and model splitting (Jiang et al. 2024; Horváth et al. 2021; Isik et al. 2023; Zhang et al. 2023a; Wu et al. 2023). How-

*Corresponding author

ever, these works only consider the inconsistency of limited system resources, lacking consideration of the data heterogeneity of clients. In the real world, data heterogeneity and client resource constraints are not isolated but rather intertwined, synergistic core challenges. Data heterogeneity impacts model convergence speed and performance, while system resources constrain communication and computational efficiency. Achieving an effective balance between these two factors can reduce the burden on limited-resource clients while improving model performance, thereby promoting the viable deployment of FL in the real world.

Given the limitations of existing works, we proposed a cosine sparsification parameter packing and dual-weighted aggregation method for PFL, which effectively balances data heterogeneity and limited system resources, aiming to reduce communication bottlenecks while maintaining model performance. Specifically, the client divides the model parameters into multiple flat parameter packages and adaptively selects the K parameter packages with the highest similarity through cosine similarity for sharing. This reduces communication overhead during transmission and effectively helps resource-constrained clients overcome communication bottlenecks. When the client computes the shared parameter packet, it generates a mask matrix that anchors the parameter packet and assigns a dual weight value to the mask, consisting of a direction weight and a distribution distance weight, helping the server to effectively align the parameter packet. During the aggregation phase, the server uses the dual weight values of the mask to complete the weighted aggregation, which accelerates the global model absorption of new knowledge and mitigates the negative impact of client data heterogeneity on the training process. To verify the effectiveness of FedCSPACK, we conduct extensive experiments across various heterogeneous scenarios. The results demonstrate that FedCSPACK can effectively balance data heterogeneity and resource constraints while ensuring model performance and client communication efficiency. The main contributions of our work are as follows:

- To the best of our knowledge, FedCSPACK is the first personalized federated learning method implemented at the package level, which achieves a balance between limited client resources and data heterogeneity through parameter packaging and dual-weight aggregation.
- Packaging local model parameters effectively reduces communication overhead. A personalized mask is generated based on the parameter package, and a dual weight is calculated for the parameter package using KL divergence and cosine similarity, which improves the server's aggregation efficiency and knowledge absorption capacity for scattered parameter packages, mitigating the impact of data heterogeneity on model performance.
- We compare FedCSPACK with 10 state-of-the-art (SOTA) methods and conduct extensive evaluations on real-world datasets with different data heterogeneity. The results show that FedCSPACK improves training speed by 2-5× and model accuracy by 3.34%, while maintaining computational efficiency.

Relate Works

FL may not be as good as the individual training performed locally by each client in a variety of heterogeneous scenarios (Tan et al. 2022c; Lee et al. 2022). To address these challenges, personalized federated learning was developed (Huang et al. 2021; Yi et al. 2023; Zhang et al. 2023a), which enables each client to train a PFL model that suits the clients.

In PFL, FedProx (Li et al. 2020) constrains the offset between the local model and the global model by introducing L2 into the local objective function, but the size of the L2 will affect the balance between the degree of the model and the convergence speed. FedNova (Wang et al. 2020) normalizes the global model based on the number of local steps for each party. FedGH (Yi et al. 2023) uploads the calculated local average representation and category labels to the server, which trains the global prediction header and broadcasts it to the client, which replaces the local prediction header. FedDBE (Zhang et al. 2023a) reduces the domain differences between the two in the representation space by facilitating bidirectional knowledge transfer between the server and the client. FedAS (Yang, Huang, and Ye 2024) enhances the localization of global parameters by combining them with local insights. However, these schemes do not deeply consider the division between model parameters, which can easily lead to bottlenecks in model performance. FedNTD (Lee et al. 2022) uses global models and historical models as teachers to promote comprehensive knowledge distillation and complete the transfer of global generalized knowledge and historical personalized knowledge to local models, thereby mitigating catastrophic forgetting. FedPAC (Xu, Tong, and Huang 2023) improves the accuracy and robustness of the model as a whole through feature alignment and classifier collaboration, it can be challenging to implement and optimize the model for resource-constrained clients. Li et al. (Li, He, and Song 2021) corrected the user model optimization direction by comparing the losses of the global model in two rounds of training. FedALA (Zhang et al. 2023b) fuses the global model of each round with the local model of the previous round to preserve the local personalized information.

Additionally, there are the Top-k-based model-sparse methods, e.g., FedSPU (Niu, Dong, and Qin 2025). Although these methods effectively solve heterogeneous problems or resource constraints in FL, they only consider one of two factors separately, ignoring the challenges of coordination between data heterogeneity and limited system resources. FedCSPACK aims to improve the computational efficiency of FL on resource-constrained devices and alleviate communication bottlenecks. Through a dual-weight aggregation mechanism, FedCSPACK reduces the impact of data heterogeneity on client and server model performance, achieving efficient and robust personalized model training.

Problem Formulation

In FL, suppose N clients are participating in training, their local dataset is $D_1, D_2, ..., D_N$ and the local model is $w_1, w_2, ..., w_N$, and the training goal of the collaboration is

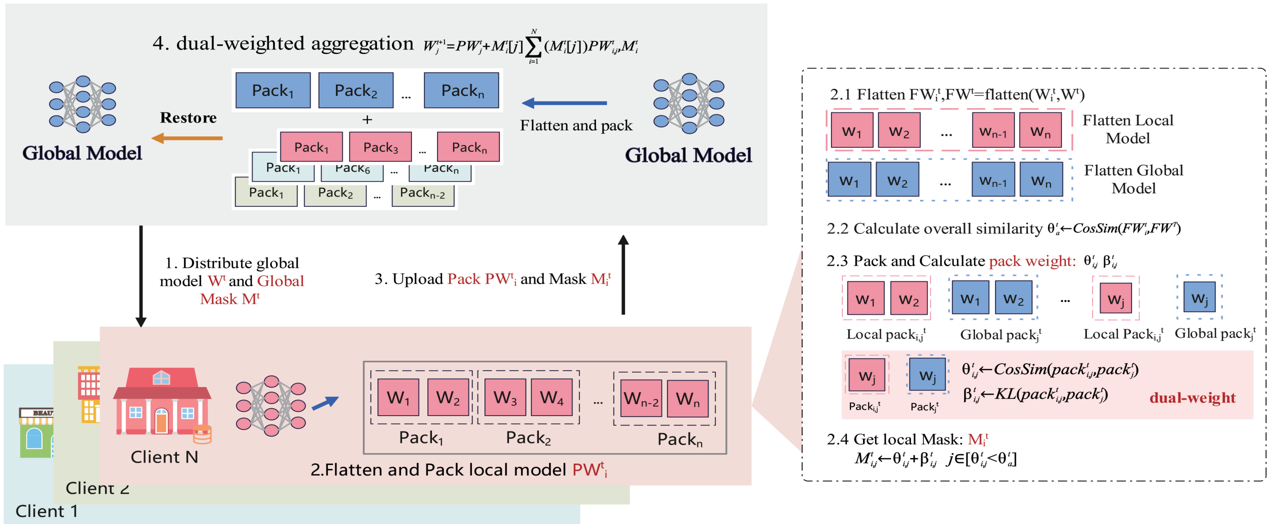


Figure 1: **FedCSPACK Overview.** The client shows the detailed process of the generation and selection of the local parameters package and mask. The server side displays the details of the dual-weighted aggregation.

to minimize the average loss of the global model $F(w)$ on all client data:

$$minF(w) = \sum_{i=1}^{N} \frac{|D_i|}{\sum |D_i|} L_i(F(w), D_i) \quad (1)$$

where the parameter w of the global model is a d-dimensional vector. $L_i(F(w), D_k)$ is the loss function of the global model $F(w)$ on the local dataset D_i on client i. $|D_i|$ is the dataset size for client i, $sum\,|D_i|$ is the sum of all client dataset sizes.

Because client data is the Non-IID distribution, the best global model obtained through training does not guarantee the best generalization performance across all clients. The goal of the PFL is to determine the optimal set of local models $w_1^a st, w_2^a st, ..., w_N^a st$, allowing multiple models to coexist, ensuring that each client i learns an optimal personalized model based on its local goals:

$$w_1^*, ..., w_N^* = \underset{w_1,...,w_N}{argmin} \sum_{i=1}^{N} \frac{|D_i|}{\sum |D_i|} L_i(F_i(w_i), D_i) \quad (2)$$

where w_i^* is the optimal model for client $(i \in N)$, and $L_i(F_i(w_i), D_i)$ is the loss function for client i. Considering that data heterogeneity and limited system resources coexist in the real scenario, FedCSPACK can effectively obtain the best local model and minimize the overall experience loss through parameter packaging and double-weighted aggregation, thereby achieving a balance between limited client resources and data heterogeneity.

FedCSPACK Design

Overview

To address the above challenges, we propose FedCSPACK, which efficiently utilizes limited resources by packaging model parameters and selectively sharing partial parameter packages, while enhancing the model's compatibility with heterogeneous data via a double-weight mask aggregation mechanism. The framework of the FedCSPACK is illustrated in fig. 1. In the t-round communication, FedCSPACK's steps are as follows:

- **Step 1.** The server completes the aggregation process of the global model W^t and the global mask M^t, and then broadcasts them to all clients.
- **Step 2-3.** Client i updates the local model W_i^t on the local data D_i under the guidance of M^t and W^t. Before sharing W_i^t, client i will convert W_i^t, W^t into one-dimensional vectors FW_i^t, FW^t and use cosine similarity to calculate the overall similarity θ_a^t between the two models. Then, according to the size of $PACK$, FW_i^t and FW^t are packaged into parameter packages $PW_{i,j}^t$ and PW_j^t, and the similarity threshold $\theta_{i,j}^t$ and the discrete value $\beta_{i,j}^t$ of each $PW_{i,j}^t$ and PW_j^t are calculated. Meanwhile, use Top-k to select K $PW_{i,k}^t, k < j$ by $\theta_{i,j}^t < \theta_a^t$ as the client i shared parameter pack $PW_{i,k}^t$. Finally, to effectively align the $PW_{i,k}^t$ of client i, let the valid weight value of the mask $M_{i,j}^t = \theta_{i,j}^t + \beta_{i,j}^t$.
- **Step 4.** The server collects local model W_i^t and local mask M_i^t shared by client i, integrates M_i^t, and completes the alignment-weighted aggregation task of the new round of global model W^{t+1} according to its effective weight value.

Perform steps 1-4 above until the specified epoch or model W_i converges for client $i \in [N]$. Details of the FedCSPACK implementation can be found in algorithm 1.

Top-k Based Cosine Parameter Packing

Frequent model interactions between clients and the server pose a significant challenge to limited resource clients, as

Algorithm 1: FedCSPACK

Input: Local datasets D_i, Number of client N, Global epoch T, Local epoch E, Learning rate η

Output: Global Model W^T

Server Executes:
Initialize Global Model w^t
for *global epoch* $t \in [T]$ **do**
  Random Sample Clients Subset S^t
  for *each client* $i \in S_t$ *parallel* **do**
    $PW_{i,k}^t, M_i^t \leftarrow$ **Client**(i, W^t, M^t)
  Update $W^{t+1}, M^{t+1} \leftarrow$ Eq. 9
  Send W^{t+1}, M^{t+1} to all clients
Return W^T
Local Executes: **Client**(i, W^t, M^t)
Receive Global Model: W^t
Update: $W_i^t \leftarrow update(W^t, M^t)$
for *local epoch* $le \in [E]$ **do**
  $W_i^t \leftarrow W_i^t - \eta \nabla f_i(W_i^t)$
$\Delta W_i^t \leftarrow W^t - W_i^t$
Flattened: $FW_i^t, FW^t \leftarrow flattened(W_i^t, W^t)$
Get Mask and Parameters pack: $M_i^t, PW_{i,k}^t \leftarrow$ Eqs. 5 and 8
Return $PW_{i,k}^t, M_i$ to the server

the frequent interaction of massive model parameters reduces their interaction efficiency and imposes a heavy communication burden on limited resource clients. Top-k (Aji and Heafield 2017) is a commonly used communication optimization method, model parameters change dynamically during iterative training, and traditional Top-K sparsity methods find it difficult to select an appropriate parameter subset in each iteration. To alleviate the communication bottleneck and accurately select parameter subsets, we propose a packet-level Top-K parameters packaging and sharing mechanism guided by cosine similarity, because lower cosine similarities mean higher inference loss, and improving these similarities may lead to better model performance.

After client i receives global model W^t and global mask M^t, it flattens the local model W_i^t and W^t to obtain FW_i^t, FW^t, then packages FW_i^t, FW^t into PW_i^t, PW^t according to $PACK$, and completes the local model update $PW_i^t[j] \leftarrow PW^t[j], if M^t[j] = Valid$. Then, $PW_i^t[j]$ is restored and starts t-round local training. Before sharing a local model, client i flattens W_i^t and W^t to obtain FW_i^t, FW^t, using cosine similarity (eq. 3) to calculate the overall similarity threshold θ_a^t between FW_i^t and FW^t.

$$\theta_a^t = CosSim(FW_i^t, FW^t) = \frac{FW_i^t \cdot FW^t}{\| FW_i^t \| \| FW^t \|} \tag{3}$$

Then, client i will package FW_i^t and FW^t according to the size of $PACK$, ensuring that each package $PW_{i,j}^t \leftarrow \frac{FW_i^t}{PACK}, j \in \left|\frac{FW_i^t}{PACK}\right|$ contains $PACK$ parameters (global model $PW_j^t \leftarrow \frac{FW^t}{PACK}$), calculating the package similarity threshold $\theta_{i,j}^t$ according to eq. 4.

$$\theta_{i,j}^t = \frac{PW_{i,j}^t \cdot PW_j^t}{\| PW_{i,j}^t \| \| PW_j^t \|},\ j \in \left|PW_i^t\right| \tag{4}$$

Next, client i uses eq.5 to select K packages ($PW_{i,k}^t, k \in j$) that is $\theta_{i,j}^t < \theta_a^t$ as the shared parameter packages of client i, and the rest of the parameter packets $PW_{i,j\setminus k}^t$ will become unique features of the client, which avoids the interference of the heterogeneity of the rest of the clients and maximizes the utility of each communication round.

$$PW_{i,k}^t = TopK(PW_{i,j}^t, \theta_{i,j}^t, \theta_a^t),\ j \in \left|PW_i^t\right| \tag{5}$$

Although $PW_{i,k}^t$ improves communication efficiency, misaligned or erroneous aggregation of $PW_{i,k}^t$ can lead to biased global model updates, degrading overall model performance. To ensure that server efficiently indexes the position of $PW_{i,k}^t$ and effectively completes global model aggregation, we design a client mask $M_{i,j}^t$ to locate the position of $PW_{i,j}^t$ and considers $\theta_{i,k}^t \leftarrow \theta_{i,j}^t < \theta_a^t$ as the valid index position of $PW_{i,k}^t$ and the proportion of the aggregate weight occupied according to eq.6.

$$M_{i,j}^t = \begin{cases} \theta_{i,k}^t & \theta_{i,k}^t \in \theta_{i,j}^t \\ 0 & \theta_{i,j} \setminus \theta_{i,k}^t \end{cases} \tag{6}$$

Through this packet-level model parameter selection and sharing mechanism, our method ensures that personalized models maintain model performance while communicating efficiently on resource-constrained clients.

Mask Double Weight Aggregation

Due to data heterogeneity, the global model aggregated by servers often struggles to adapt to the varying data distribution across clients. Although cosine similarity can maintain consistency between local updates and global directions in parameter space, it only reflects directional alignment. It cannot measure distance differences and magnitude shifts between parameter values. This insensitivity to magnitude can lead to overweighting the corresponding mask $M_{i,j}^t$, excessively influencing the aggregation results, and compromising model stability. To address this, we propose a dual-weight parameter pack aggregation mechanism. Based on directional weights based on cosine similarity, we further introduce the Kullback-Leibler (KL) divergence to quantify the distance shift between local and global model parameter updates. Specifically, When calculating the packet similarity threshold $\theta_{i,j}^t$ of $PW_{i,j}^t$ and PW_j^t, client i will use KL divergence (eq. 7) to calculate the distance difference threshold $\beta_{i,j}^t$ between $PW_{i,j}^t$ and PW_j^t, and add β_j as a supplement weight of $M_{i,j}^t$.

$$\beta_{i,j}^t = \sum PW_{i,j}^t \log \frac{PW_{i,j}^t}{PW_j^t},\ j \in \left|PW_i^t\right| \tag{7}$$

Then, we perform a new allocation process for the mask weights $M_{i,j}^t$ in eq. 8 according to eq. 6 to ensure that customers extract the weight ratio of each model parameter package more comprehensively. This strategy preserves the benefits of directional alignment while effectively mitigating the impact of distributional discrepancies, improving the common-sense ability of the global model to absorb each parameter package.

$$M_{i,j}^t = \begin{cases} \theta_{i,k}^t + \beta_{i,k}^t & \theta_{i,k}^t \in \theta_{i,j} \\ 0 & \theta_{i,j} \setminus \theta_{i,k}^t \end{cases} \tag{8}$$

After the server receives shared parameter packages $PW_{i,k}^t, i \in S^t$ and mask $M_i^t, i \in S^t$, it first updates the global mask $M^{t+1} = sum(M_i^t)$ to obtain the total weight of each parameter packet position. Then, flatten the global model W^t and complete the parameter packing ($PW_{i,k}^t$) according to the PACK size. Under the guidance of eq. 9, the server performs weighted aggregation on the client's parameter pack and fuses the new result $PW_{i,j}^t$ with PW^t to obtain a new global model $PW_j^{t+1} \leftarrow PW_j^t + PW_{i,j}^t$.

$$PW_{i,j}^t = \begin{cases} \sum_{i=1}^{S_t} \frac{M_{i,j}^t}{M_j^{t+1}} PW_{i,k}^t & M_{i,j}^t \neq 0 \\ 0 & else \end{cases} \tag{9}$$

As training progresses, the global model can efficiently and quickly absorb client model knowledge through mask double-weighted aggregation at the packet level, preserving its generalization capability. When clients merge a new global model, the stability and generalization performance of the global model under cross-client heterogeneous data are enhanced because the local parameter package retained by the client contains the local personalized knowledge.

Experiments

Experiment Setting

Dataset Setup: We use four publicly available image classification datasets as our experimental data: Fashion-MNIST (FMNIST), EMNIST, CIFAR-10, and CIFAR-100. To verify the effectiveness of **FedCSPACK**, we use CNN and ResNet-18 as the test model. To simulate this heterogeneity, we employ Dirichlet (Dir(α)) and pathological sampling, which are widely used in FL (Caldas et al. 2018b; Acar et al. 2021; Li et al. 2022). A smaller α value corresponds to greater Non-IID characteristics. Dirichlet sampling is shown in fig. 2. Pathological Sampling and experimental setup information are in Appendix A. All experiments are repeated three times, and the average value is taken as the experimental result.

Baseline: We compare FedCSPACK with ten SOTA methods. FedAvg (McMahan et al. 2017), FedProx (Li et al. 2020), and FedNova (Wang et al. 2020) are traditional methods; FedALA (Zhang et al. 2023b) in the aggregation scheme helps weak clients to choose the appropriate ratio of integration between the global and local models. Model Spliting (MOON (Li, He, and Song 2021), FedDBE (Zhang et al. 2023a), and FedAS (Yang, Huang, and Ye 2024)) and Knowledge Distillation (FedNTD (Lee et al. 2022) and FedPAC (Xu, Tong, and Huang 2023)) can effectively improve accuracy and communication efficiency. FedSPU (Niu, Dong, and Qin 2025) improves communication efficiency by sparsifying model parameters.

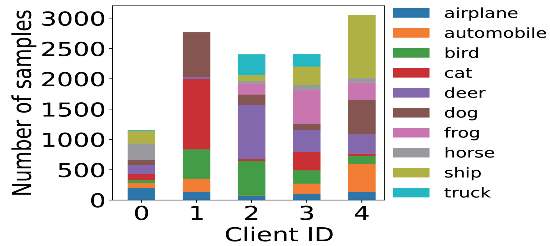


Figure 2: Data heterogeneity of 5 clients is simulated using dirichlet sampling ($Dir(0.6)$) on CIFAR-10.

Performance Comparison

The Overall Performance of the Global Model. Table 1 lists the results of FedCSPACK and other FL SOTA methods on the original dataset for basic models from shallow to deep. On FMNIST, as the data distribution shifts from $Dir(0.3)$ to $Dir(1.0)$, the model's accuracy generally improves. FedCSPACK performs well under all data distributions, especially at $Dir(1.0)$. It achieved an accuracy rate of 90.73%, significantly higher than that of other methods. On complex CIFAR-10 and CIFAR-100, FedCSPACK reached 78.71% and 43.20% on $Dir(1.0)$, respectively. On EMNIST, the model's accuracy is generally high. This might be because the increase in data volume helps the model learn features better. As the data distribution shifted from $Dir(1.0)$ to $Dir(0.3)$, the model's accuracy declined. This might be due to the homogenization of data distribution leading to a reduction in samples of certain categories, thereby affecting the generalization ability of other scheme models. However, FedCSPACK can still ensure the high accuracy of the model through the dual-weight package aggregation method, especially reaching 86.26% at $Dir(0.6)$ on EMNIST. The pathological result is in Appendix B. Overall, FedCSPACK can maintain a high model accuracy rate through dual-weight aggregation, effectively adapting to datasets ranging from simple to complex and from small to massive.

The Generalization Ability of the Global Model on the Client. As shown in fig. 3, the global model trained by FedCSPACK shows effective model performance in adapting to the heterogeneous data of the client. On Client 5 with the best generalization performance, FedCSPACK achieves a model accuracy of 0.66, while the highest accuracy of other methods was 0.62, representing a 6.5% improvement. On client 3 with the worst generalization performance, FedCSPACK's generalization performance remains stable, 20% higher than the best SOTA (FedDBE and FedSPU). Although FedCSPACK's generalization performance is lower than FedPAC and FedAS on Client 2 and Client 7, it is not as good as FedCSPACK on other clients. This shows that FedCSPACK's use of Top-k cosine parameter packs and dual-weighted aggregation scheme can effectively ensure the gen-

DataSet	FMNIST			EMNIST			CIFAR-10			CIFAR-100		
Dirichlet(α)	0.3	0.6	1.0	0.3	0.6	1.0	0.3	0.6	1.0	0.3	0.6	1.0
FedAvg	84.39	86.96	87.65	83.12	84.03	83.98	69.71	74.71	75.01	39.15	40.44	40.34
FedProx	84.39	86.96	87.65	84.55	84.12	83.99	69.58	**74.59**	74.54	38.48	39.58	39.68
FedNova	84.40	**87.50**	**88.68**	**84.97**	84.18	84.03	**70.93**	74.37	75.53	38.65	40.30	39.96
MOON	85.44	86.40	87.90	83.17	84.08	83.94	70.03	74.63	**76.00**	38.43	39.85	39.90
FedDBE	85.06	86.28	88.24	83.14	84.08	83.83	69.66	74.76	74.80	38.33	40.03	39.92
FedAS	42.47	57.27	71.96	70.46	77.32	77.63	43.97	70.00	69.46	20.51	24.12	31.42
FedNTD	84.35	86.67	88.01	83.56	**84.49**	**84.32**	70.32	74.68	75.52	**39.44**	**40.67**	**40.88**
FedPAC	41.53	65.73	69.34	58.11	59.02	66.98	57.23	68.78	58.60	35.39	38.63	39.54
FedALA	84.39	86.96	87.65	83.29	83.98	83.82	69.49	74.64	74.96	38.93	40.03	40.24
FedSPU	**85.29**	88.76	88.25	83.56	84.22	83.91	67.38	74.52	74.62	37.81	38.25	39.35
FedCSPACK(ours)	**88.13**	**89.50**	**90.73**	**85.55**	**86.26**	**86.19**	**73.23**	**77.15**	**78.71**	**41.60**	**42.96**	**43.20**

Table 1: The test Top-1 accuracy (%) across three Dirichlet sampling with $\alpha \in [0.3, 0.6, 1.0]$ on four datasets. The best result is FedCSPACK, while the bold values denote the best methods in SOTA.

DataSet / Model	Metric	FedAvg	Tradition	Split	Distillation	FedALA	FedSPU	**FedCSPACK**
FMNIST/CNN	Traffic	0.30	0.30	0.30	0.32	0.30	0.07	**0.02**
	Time	1.13	1.45	2.34	1.80	1.18	1.41	**1.21**
EMNIST/CNN	Traffic	18.18	18.18	18.18	18.29	18.18	4.29	**0.73**
	Time	11.16	14.37	23.11	17.92	11.67	13.92	**11.32**
CIFAR-10/CNN	Traffic	2.59	2.59	2.59	2.61	2.59	0.61	**0.20**
	Time	1.40	1.83	2.89	2.23	1.46	1.75	**1.42**
CIFAR-100/ResNet18	Traffic	251.00	251.00	251.00	251.80	251.00	49.22	**9.24**
	Time	0.81	1.05	1.67	1.30	0.84	1.01	**0.88**

Table 2: On four datasets, with training rounds T=100 and Dir(0.3), the total training time (hours) and the number of parameters transmitted (GB) of FedCSPACK and SOTA are compared. Tradition: FedProx and FedNova. Split: MOON, FedDBE, and FedAS. Distillation: FedNTD and FedPAC. Parameter packing improves the computational and communication performance.

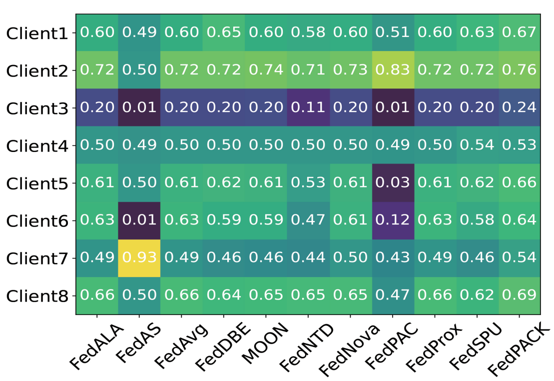


Figure 3: The generalization of the global model on the clients in CIFAR-10 and Dir(0.3).

eralization performance of the global model and make it more stable. More results are in Appendix C.

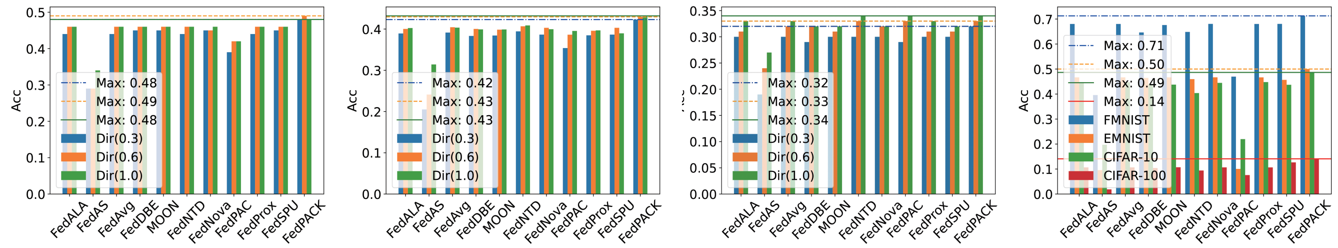
Resource Consumption

Table 2 presents comparative results demonstrating improved computational performance and optimal resource usage. **Communication Traffic**. FedCSPACK significantly reduces transmission overhead across various datasets. For example, on EMNIST, the communication traffic dropped from 18.18GB to 0.73GB, achieving a compression ratio of 96.0%, the lowest among all compared methods. On CIFAR-100, FedCSPACK compressed the communication traffic of the ResNet18 from 251.00GB to 9.24GB, representing a 27$\times$ reduction; it is 5$\times$ lower than FedSPU. This result demonstrates that FedCSPACK's packet-level sparse communication mechanism effectively reduces the number of uploaded parameters, demonstrating exceptional compression capabilities, particularly for a complex model, significantly outperforming SOTA methods.

Training Time. On EMNIST and CIFAR-10, FedCSPACK's training time barely increases compared to the baseline, remaining highly stable. On CIFAR-100, although training time increases slightly from 0.81H to 0.88H, FedCSPACK is still 89.8%, 47.7%, and 38.11% lower than Split, Distillation, and FedSPU, respectively, demonstrating superior computational efficiency. These results demonstrate that while the Top-K parameter pack selection mechanism introduces a small amount of additional computation, the resulting increase in training time is minimal, and even improves it in some scenarios.

Numbers of Limited Resources Clients

Fig. 4 shows the impact of CPR on the model's performance and stability. In figs 4a to 4c, model performance fluctu-



(a) CPR=1.0,Dir(0.5) (b) CPR=0.6,Dir(0.5) (c) CPR=0.3,Dir(0.5) (d) CPR=0.3,Pathological

Figure 4: The influence of the limited resources client participation ratio (CPR) on CIFAR-100, where CPR represents the ratio of the number of clients per round to the total number of clients. The horizontal line indicates the accuracy of the FedCSPACK.

ates significantly as client participation declines. In low-participation scenarios (fig. 4c), the impact of data heterogeneity on existing SOTA methods is further exacerbated due to the incomplete distribution of available data, leading to a significant performance drop. However, FedCSPACK maintains good stability and robustness, achieving an accuracy of 0.32, 0.33, and 0.34. In fig. 4d, FedCSPACK demonstrates superior model performance in four different pathological data partitioning scenarios. This result further demonstrates FedCSPACK's effectiveness and generalization capabilities under diverse data distributions. More participation rate results are in Appendix D.

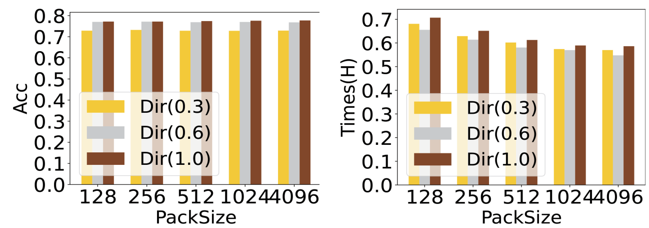
Ablation Studies

Influence of Weight. We assessed the training impact of key weights: cosine similarity (CS) weight and KL divergence weight on model performance in FedCSPACK to demonstrate their individual effectiveness. As shown in table 3, compared to using CS alone, FedCSPACK using dual-weighted aggregation improved the average accuracy by 0.01, 0.06, 0.04, and 0.05, respectively; and compared to using KL alone, it improved the accuracy by 0.03, 0.06, 0.03, 0.07, and 0.1, respectively. The result demonstrates that the dual-weighted aggregation combining parameter direction and parameter distance can more comprehensively evaluate the contribution of the parameter package and effectively improve the training efficiency of the model.

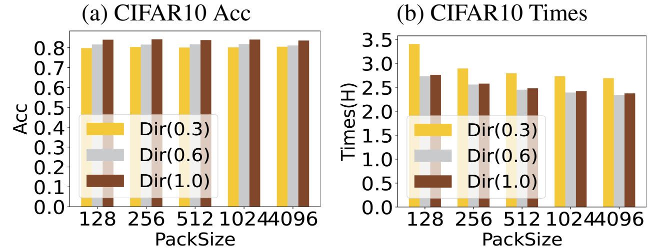
Weight	Round				
	10	30	50	70	100
CS	0.33	0.62	0.65	0.70	0.74
KL	0.30	0.57	0.68	0.67	0.69
Ours	0.33	0.63	0.71	0.74	0.79

Table 3: Impact of key weights on model accuracy at different training stages on CIFAR-10 and Dir(0.5).

Influence of PACK. Fig. 5 further verifies the effect of $PACK$ on model performance. In figs 5a and 5c, as the $PACK$ increases, the overall model performance remains largely unchanged under varying data heterogeneity conditions. This demonstrates that the parameter pack selection technique selects the K most contributing local model parameter packs. The dual-weight aggregation mechanism further enhances the global model's ability to integrate local model knowledge, thereby ensuring model stability. In figs 5b and 5d, the overall time consumption of the model gradually decreases as the size of the $PACK$ increases, and the improvement of FedCSPACK is more obvious in heterogeneous scenarios with a large amount of data. Appendix E has more details.



(a) CIFAR10 Acc (b) CIFAR10 Times



(c) EMNIST Acc (d) EMNIST Times

Figure 5: The impact of the $PACK$ size on model performance and computation time.

Conclusion

In this work, we propose a novel parameter package-level PFL method (FedCSPACK) to address the coexistence of resource constraints and data heterogeneity. The key insight is to package the client model parameters, using cosine similarity and Top-k to calculate the shared parameter package, reducing the client's demand for limited resources. Through mask matrix and dual weights, FedCSPACK can ensure that the server efficiently aggregates the sparse parameter packages of client models, guaranteeing the robustness and generalization ability of the global model. Extensive experiments on various datasets have demonstrated that FedCSPACK achieves SOTA performance.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (U22B2026, 62572121, 62372231), Jiangsu Province Frontier Technology Research and Development Program (BF2025067), Aeronautical Science Foundation (2022Z071052008), Fundamental Research Funds for the Central Universities (2242025K30025), and the Big Data Computing Center of Southeast University.

References

Acar, D. A. E.; Zhao, Y.; Navarro, R. M.; Mattina, M.; Whatmough, P. N.; and Saligrama, V. 2021. Federated Learning Based on Dynamic Regularization. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Aji, A. F.; and Heafield, K. 2017. Sparse Communication for Distributed Gradient Descent. In Palmer, M.; Hwa, R.; and Riedel, S., eds., *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*, 440–445. Association for Computational Linguistics.

Alam, S.; Liu, L.; Yan, M.; and Zhang, M. 2022. FedRolex: Model-Heterogeneous Federated Learning with Rolling Sub-Model Extraction. In Koyejo, S.; Mohamed, S.; Agarwal, A.; Belgrave, D.; Cho, K.; and Oh, A., eds., *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

Caldas, S.; Konečný, J.; McMahan, H. B.; and Talwalkar, A. 2018a. Expanding the Reach of Federated Learning by Reducing Client Resource Requirements. *CoRR*, abs/1812.07210.

Caldas, S.; Wu, P.; Li, T.; Konečný, J.; McMahan, H. B.; Smith, V.; and Talwalkar, A. 2018b. LEAF: A Benchmark for Federated Settings. *CoRR*, abs/1812.01097.

Diao, E.; Ding, J.; and Tarokh, V. 2021. HeteroFL: Computation and Communication Efficient Federated Learning for Heterogeneous Clients. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Horváth, S.; Laskaridis, S.; Almeida, M.; Leontiadis, I.; Venieris, S. I.; and Lane, N. D. 2021. FjORD: Fair and Accurate Federated Learning under heterogeneous targets with Ordered Dropout. In Ranzato, M.; Beygelzimer, A.; Dauphin, Y. N.; Liang, P.; and Vaughan, J. W., eds., *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, 12876–12889.

Huang, Y.; Chu, L.; Zhou, Z.; Wang, L.; Liu, J.; Pei, J.; and Zhang, Y. 2021. Personalized Cross-Silo Federated Learning on Non-IID Data. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, 7865–7873. AAAI Press.

Isik, B.; Pase, F.; Gündüz, D.; Weissman, T.; and Zorzi, M. 2023. Sparse Random Networks for Communication-Efficient Federated Learning. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Jiang, Z.; Xu, Y.; Xu, H.; Wang, Z.; Liu, J.; Qian, C.; and Qiao, C. 2024. Computation and Communication Efficient Federated Learning With Adaptive Model Pruning. *IEEE Trans. Mob. Comput.*, 23(3): 2003–2021.

Karimireddy, S. P.; Kale, S.; Mohri, M.; Reddi, S. J.; Stich, S. U.; and Suresh, A. T. 2020. SCAFFOLD: Stochastic Controlled Averaging for Federated Learning. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, 5132–5143. PMLR.

Lee, G.; Jeong, M.; Shin, Y.; Bae, S.; and Yun, S. 2022. Preservation of the Global Knowledge by Not-True Distillation in Federated Learning. In Koyejo, S.; Mohamed, S.; Agarwal, A.; Belgrave, D.; Cho, K.; and Oh, A., eds., *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

Li, Q.; Diao, Y.; Chen, Q.; and He, B. 2022. Federated Learning on Non-IID Data Silos: An Experimental Study. In *38th IEEE International Conference on Data Engineering, ICDE 2022, Kuala Lumpur, Malaysia, May 9-12, 2022*, 965–978. IEEE.

Li, Q.; He, B.; and Song, D. 2021. Model-Contrastive Federated Learning. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, 10713–10722. Computer Vision Foundation / IEEE.

Li, T.; Hu, S.; Beirami, A.; and Smith, V. 2021. Ditto: Fair and Robust Federated Learning Through Personalization. In Meila, M.; and Zhang, T., eds., *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, 6357–6368. PMLR.

Li, T.; Sahu, A. K.; Zaheer, M.; Sanjabi, M.; Talwalkar, A.; and Smith, V. 2020. Federated Optimization in Heterogeneous Networks. In Dhillon, I. S.; Papailiopoulos, D. S.; and Sze, V., eds., *Proceedings of Machine Learning and Systems 2020, MLSys 2020, Austin,TX, USA, March 2-4, 2020*. mlsys.org.

Luo, J.; and Wu, S. 2022. Adapt to Adaptation: Learning Personalization for Cross-Silo Federated Learning. In Raedt, L. D., ed., *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI 2022, Vienna, Austria, 23-29 July 2022*, 2166–2173. ijcai.org.

McMahan, B.; Moore, E.; Ramage, D.; Hampson, S.; and y Arcas, B. A. 2017. Communication-Efficient Learning of Deep Networks from Decentralized Data. In Singh, A.; and Zhu, X. J., eds., *Proceedings of the 20th International*

Conference on Artificial Intelligence and Statistics, AISTATS 2017, 20-22 April 2017, Fort Lauderdale, FL, USA, volume 54 of *Proceedings of Machine Learning Research*, 1273–1282. PMLR.

Niu, Z.; Dong, H.; and Qin, A. K. 2025. FedSPU: Personalized Federated Learning for Resource-Constrained Devices with Stochastic Parameter Update. In Walsh, T.; Shah, J.; and Kolter, Z., eds., *AAAI-25, Sponsored by the Association for the Advancement of Artificial Intelligence, February 25 - March 4, 2025, Philadelphia, PA, USA*, 19721–19729. AAAI Press.

Oh, J.; Kim, S.; and Yun, S. 2022. FedBABU: Toward Enhanced Representation for Federated Image Classification. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Tan, A. Z.; Yu, H.; Cui, L.; and Yang, Q. 2022a. Towards Personalized Federated Learning. *IEEE Transactions on Neural Networks and Learning Systems*, 1–17.

Tan, Y.; Long, G.; Liu, L.; Zhou, T.; Lu, Q.; Jiang, J.; and Zhang, C. 2022b. FedProto: Federated Prototype Learning across Heterogeneous Clients. In *Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022, Thirty-Fourth Conference on Innovative Applications of Artificial Intelligence, IAAI 2022, The Twelveth Symposium on Educational Advances in Artificial Intelligence, EAAI 2022 Virtual Event, February 22 - March 1, 2022*, 8432–8440. AAAI Press.

Tan, Y.; Long, G.; Ma, J.; Liu, L.; Zhou, T.; and Jiang, J. 2022c. Federated Learning from Pre-Trained Models: A Contrastive Learning Approach. In Koyejo, S.; Mohamed, S.; Agarwal, A.; Belgrave, D.; Cho, K.; and Oh, A., eds., *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

Wang, J.; Liu, Q.; Liang, H.; Joshi, G.; and Poor, H. V. 2020. Tackling the Objective Inconsistency Problem in Heterogeneous Federated Optimization. In Larochelle, H.; Ranzato, M.; Hadsell, R.; Balcan, M.; and Lin, H., eds., *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Wu, X.; Liu, X.; Niu, J.; Zhu, G.; and Tang, S. 2023. Bold but Cautious: Unlocking the Potential of Personalized Federated Learning through Cautiously Aggressive Collaboration. In *IEEE/CVF International Conference on Computer Vision, ICCV 2023, Paris, France, October 1-6, 2023*, 19318–19327. IEEE.

Xu, J.; Tong, X.; and Huang, S. 2023. Personalized Federated Learning with Feature Alignment and Classifier Collaboration. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Yang, X.; Huang, W.; and Ye, M. 2024. FedAS: Bridging Inconsistency in Personalized Federated Learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2024, Seattle, WA, USA, June 16-22, 2024*, 11986–11995. IEEE.

Yi, L.; Wang, G.; Liu, X.; Shi, Z.; and Yu, H. 2023. FedGH: Heterogeneous Federated Learning with Generalized Global Header. In El-Saddik, A.; Mei, T.; Cucchiara, R.; Bertini, M.; Vallejo, D. P. T.; Atrey, P. K.; and Hossain, M. S., eds., *Proceedings of the 31st ACM International Conference on Multimedia, MM 2023, Ottawa, ON, Canada, 29 October 2023- 3 November 2023*, 8686–8696. ACM.

Zhang, J.; Hua, Y.; Cao, J.; Wang, H.; Song, T.; Xue, Z.; Ma, R.; and Guan, H. 2023a. Eliminating Domain Bias for Federated Learning in Representation Space. In Oh, A.; Naumann, T.; Globerson, A.; Saenko, K.; Hardt, M.; and Levine, S., eds., *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

Zhang, J.; Hua, Y.; Wang, H.; Song, T.; Xue, Z.; Ma, R.; and Guan, H. 2023b. FedALA: Adaptive Local Aggregation for Personalized Federated Learning. In Williams, B.; Chen, Y.; and Neville, J., eds., *Thirty-Seventh AAAI Conference on Artificial Intelligence, AAAI 2023, Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence, IAAI 2023, Thirteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2023, Washington, DC, USA, February 7-14, 2023*, 11237–11244. AAAI Press.

Zhang, J.; Li, Z.; Li, B.; Xu, J.; Wu, S.; Ding, S.; and Wu, C. 2022. Federated Learning with Label Distribution Skew via Logits Calibration. In Chaudhuri, K.; Jegelka, S.; Song, L.; Szepesvári, C.; Niu, G.; and Sabato, S., eds., *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, 26311–26329. PMLR.