Communication-Efficient Federated Self-Distillation Method for Medical Image Segmentation

Chen Qing[1, 2], Li Xinwei[1], Xia Jinhong[1], Lin Yifeng[1], Zheng Shenhai[1, 2, *]

[1]College of Computer Science and Technology, Chongqing University of Posts and Telecommunications, Chongqing 400065, China

[2]Chongqing Key Laboratory of Image Cognition, Chongqing University of Posts and Telecommunications, Chongqing 400065, China

Abstract: Federated learning is a distributed learning framework designed to facilitate collaborative training of deep learning models at the network edge, ensuring the privacy and security of users while effectively responding to the challenges posed by data silos within the healthcare domain. However, deploying intricate medical segmentation network models in realistic settings often leads to substantial communication overhead for federated learning clients, thereby severely undermining the operational efficiency of federated systems. In light of these issues, we propose an innovative federated learning approach known as FedSKD, which is intended to alleviate the significant communication overhead incurred during medical image segmentation tasks in federated learning. This method strikes a delicate balance between communication costs and model training accuracy. A key feature of the FedSKD framework is incorporating a self-knowledge distillation mechanism, which encompasses both prediction map distillation loss and feature map distillation loss. This integration plays a key role in enhancing model accuracy without the need for training an additional resource-intensive mentor model, thereby optimizing computational efficiency. Additionally, we introduce parameter difference dynamic compression technology to compress communications, resulting in a reduction of communication costs by more than 30% within the federated system. Furthermore, extensive experiments utilizing FedSKD were conducted on widely recognized public medical segmentation datasets such as KiTS19 and COVID-19. The experimental findings unequivocally demonstrate that our FedSKD not only significantly diminishes the expenses associated with federated communication but also enhances overall model accuracy.

Keywords: Federated Learning; Medical Image Segmentation; Self-Knowledge Distillation; Dynamic Parameter Difference Compression

DOI: 10.57237/j.cst.2024.03.001

1 Introduction

Deep Neural Networks have significantly improved the performance of various computer vision-related tasks in the medical domain, greatly enhancing the efficiency and accuracy of medical diagnosis [1]. These tasks include medical image segmentation [2, 3], boundary detection [4], image fusion [5], and more. As we all know, access to

Funding: National Natural Science Foundation of China (No. 61902046); Science and Technology Research Program of Chongqing Municipal Education Commission (No. KJZD-K202200606); Natural Science Foundation of Chongqing (No. CSTB2022NSCQ-MSX0277).

*Corresponding author: Zheng Shenhai, zhengsh@cqupt.edu.cn

Received: 29 May 2024; Accepted: 12 July 2024; Published Online: 18 July 2024 http://www.computscitech.com

rich and diverse data is crucial for training high-quality deep learning models. However, in reality, many types of data cannot be centrally stored and shared due to ethical and privacy concerns, such as patients' medical health records held by hospitals [6]. Data privacy and security pose significant challenges for traditional centralized deep learning methods.

FL enables secure collaboration among multiple institutions for training deep neural network models while maintaining data privacy [7]. Key FL algorithms such as FedAvg [8], FedProx [9], and SCAFFOLD [10] significantly improve model efficiency and privacy protection, ensuring safer advancements in medical data analysis and utilization. However, the frequent transmission of model data between clients and servers results in significant communication overhead. To reduce the data volume on the server, it is effective to involve fewer clients in aggregation. However, this approach may lead to imbalanced client model training [11]. Compressing and quantizing federated learning model parameters or gradients can reduce data transmission [12-14], but it may lead to some loss of accuracy, potentially impacting the FL model's performance and generalization.

In real-world scenarios, the imbalanced distribution of data among healthcare institutions can cause significant variations in the performance of models trained by each institution. For example, when dealing with tasks such as abdominal multi-organ segmentation [15], institutions with different medical images may encounter issues of data imbalance and domain shift, ultimately resulting in decreased overall performance. The selection of suitable clients can help partially mitigate the impact of data heterogeneity in the system to some extent [16, 17]. KD [18] is a model compression technique that involves treating the target model as a mentee, learning additional knowledge from the powerful representational capabilities of a mentor model [19–21]. However, traditional KD often requires training a more complex teacher model, which can lead to a significant consumption of computational resources. SKD can enhance model accuracy and reduce computational costs by distilling its own knowledge [22–24], but it is limited by inherent knowledge constraints.

In response to the aforementioned issues, this paper presents a FL framework based on SKD. We have designed a DPDC, which dynamically adjusts the decomposition threshold when transmitting parameters between clients and the server. This can compress communication parameter size between clients and the server by over 30% while simultaneously improving model accuracy. Unlike traditional KD, which necessitates additional training of a cumbersome Mentor model, we propose a self-knowledge distillation framework for local models. This framework guides the model network to optimize for more precise locations. Experimental analyses on the public medical segmentation datasets KiTS19 and COVID-19 validate the effectiveness of our framework. Comparative experiments with various methods also demonstrate the efficiency of our proposed approach.

2 Proposed Methods

The communication process of FedSKD is illustrated in Figure 1. At the onset of the *t*-th federated communication round, each client C_i computes the parameter difference $d_{i,t}$ between the local model before and after training on a per-layer basis. Based on the current state, it adaptively adjusts its decomposition threshold T_i and applies dynamic difference compression to $d_{i,t}$ before uploading it to server S. After collecting the compressed parameters uploaded by all clients, S reconstructs the parameter differences and aggregates them to obtain the total parameter difference $d_{s,t}$ according to client weights θ_i, then updates the server model $w_{s,t}$. S computes the parameter difference $d_{s,t} - d_{i,t}$ for each client C_i and performs dynamic difference compression on each one, broadcasting the compressed data back to C_i.

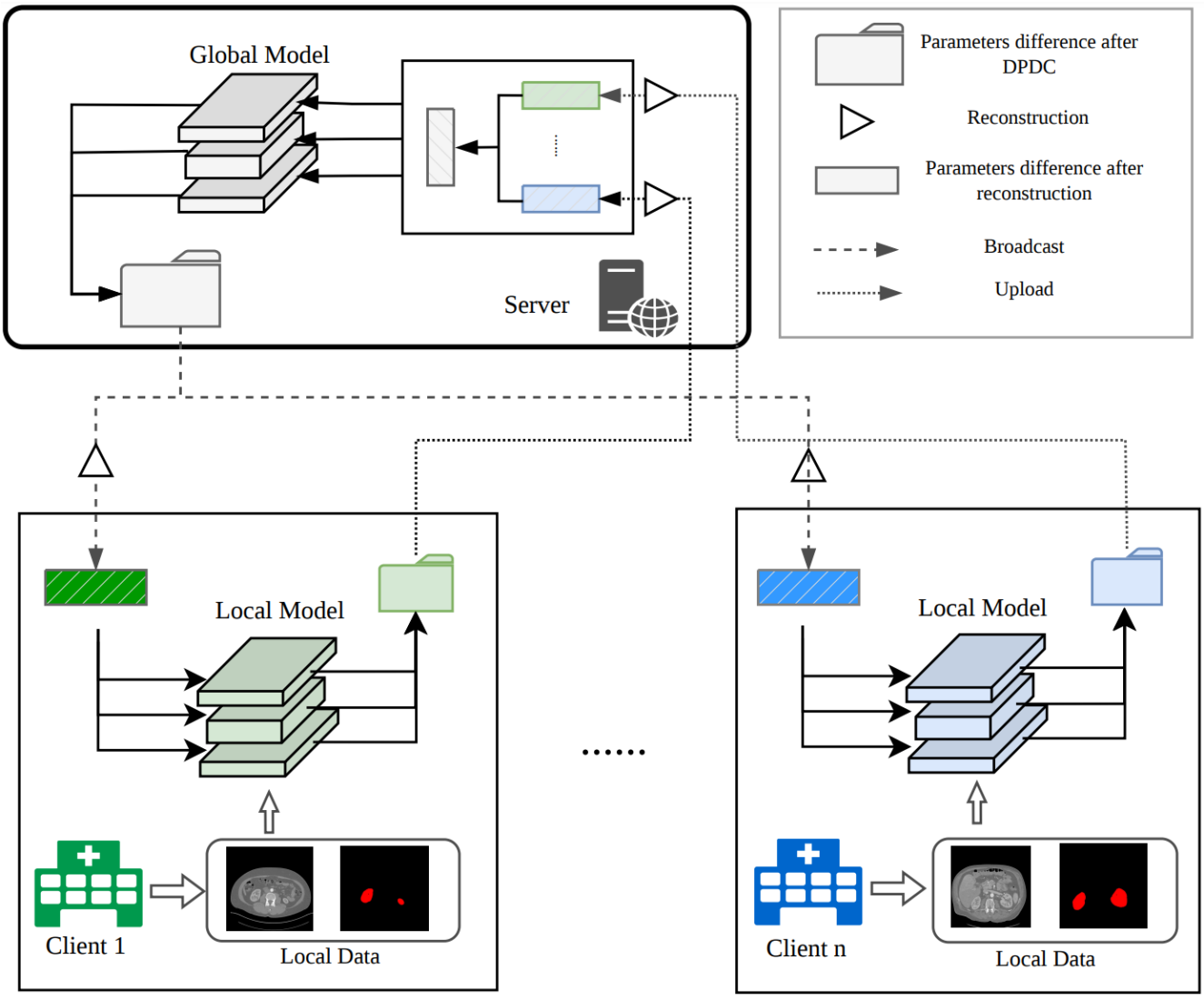


Figure 1 In the FedSKD process, clients use their local datasets for multiple SKD training rounds. Following local training, each client compresses data for each model layer using DPDC and uploads it to the server. The server reconstructs and combines parameter differences from all clients, applies a customized DPDC to each client's data, and broadcasts it back to them. Clients then integrate the parameter difference updates to rebuild their local models, initiating the next communication round

2.1 Dynamic Parameter Difference Compression (DPDC)

In our approach, DPDC is the core component that relies on truncated singular value decomposition (SVD). SVD can reduce the dimensionality of data and approximate the original data. Let's consider a matrix $A = [a_{i,j}](A \in R^{n\times m})$ with a rank of r. By performing truncated SVD, we obtain $A = U\sum V^T$, where U is an m-order orthogonal matrix, V is an n-order orthogonal matrix, and $\sum$ is an $m \times n$diagonal matrix with its diagonal elements being the non-negative singular values σ_i of A, sorted in descending order.

Let's consider a collection of matrices $M \in R^{n\times m}$, where all matrices have ranks not exceeding k. Within this collection, there exists a matrix $B \in M$ with a rank of k represents the optimal approximation of matrix A in terms of the Frobenius norm. In other words, finding the optimal approximation target B can be transformed into identifying the top k singular values that satisfy an approximation of the squared Frobenius norm of A, as Eq. (1):

$$||B||_F^2 = ||A||_F^2 - ||A - B||_F^2 = \sum_{i=1}^{k} \sigma_i^2 \approx ||A||_F^2 \quad (1)$$

In medical image segmentation tasks, models are often highly complex. In FL, a large number of parameter uploads and downloads can lead to significant communica-

tion overhead. Therefore, we are considering utilizing dynamic parameter difference compression to reduce the parameter size exchanged between clients and the server, thereby alleviating communication burden. This motivation is based on the scenario where, after multiple rounds of local training during the *t*-th round of communication, the local testing loss remains almost unchanged to meet the approximation requirements for the original parameter information.

Specifically, we set a decomposition threshold T_i to dynamically adjust the number of singular values that client C_i needs to retain to approximate the original parameter information A. The value of T_i is determined by the test error of the local model and can be adjusted as follows:

$$T_i = T_i + \frac{t+1}{t_{total}} \times \gamma \tag{2}$$

Where t_{total} represents the total communication rounds and γ is a hyperparameter for balancing the adjustment scale. The adjustment of the size of T_i occurs within the limit of the set maximum decomposition threshold value, T_{max}. This approach allows the model to adapt the magnitude of the SVD threshold based on its own training progress.

$$\min_k \sum_{i=1}^{k} \sigma_i^2 >= T_i \times ||A||_F^2 = T_i \times \sum \sigma^2 \tag{3}$$

The appropriate decomposition threshold can determine the number of singular values retained, effectively reducing the parameter size when clients upload parameter differences, and the server broadcasts parameter differences. This simple approach greatly reduces communication expenses in FL.

2.2 Federated Self-Knowledge Distillation (SKD)

During the model training process, the model's prediction results often contain rich information. Distilling knowledge from the previous round's model predictions can assist the current model. The deepest-layer feature maps of the model contain advanced semantic information learned from the original data. Learning from past feature information also helps the model comprehend and differentiate complex patterns. In FedSKD, the goal is adaptively learn from the knowledge of the previous round's model predictions and the deepest-layer feature knowledge model. Since the precision of model training is a gradually improving process, we dynamically adjust the degree of distillation for past predictions and features from shallower to deeper levels by setting α_T and β_T accordingly as we progress from shallower to deeper levels. The supervisory loss for training consists of three components in total, as illustrated in Figure 2. The first component is the MaskLoss:

$$l_m = BCEWithLogitsLoss(p_i^t, y_i) \tag{4}$$

Where p_i^t represents the segmentation prediction of client C_i in the *t*-th communication round, and y_i is the ground truth of the CT image. The second component is the Prediction Distillation Loss (PDLoss) l_p, which involves the Kullback-Leibler (KL) divergence loss between the predictions from the previous round and the current round:

$$l_p = KLDivLoss(p_i^t, p_i^{t-1}) \tag{5}$$

Since the model predictions become more accurate as the rounds progress, the parameter α_t for dynamically adjusting the weight of l_p is defined as $\alpha_t = \alpha_T \times \frac{t+1}{t_{total}}$.

The third component of the loss is the Feature Distillation Loss (FDLoss) l_f, which is composed of the L2 loss between the deepest layer features from the previous round and the current round:

$$l_f = L2Loss(f_i^t, f_i^{t-1}) \tag{6}$$

Where f_i^t represents the deepest-layer feature map of the model from client C_i in the *t*-th communication round. For l_f, similarly, the weight parameter needs to be dynamically adjusted as well: $\beta_t = \beta_T \times \frac{t+1}{t_{total}}$. This adjustment considers the increasing importance of feature knowledge as training advances throughout the iterations.

Through the L2 loss, the implicit knowledge stored in the deepest layer features of previous models is integrated into the current model. This process encourages the feature mappings of the current model's network layers to align with those of past models.

To sum up, the complete SKD framework's loss function is composed of the three aforementioned losses, which are written as:

$$\text{TotalLoss} = l_m + \alpha_t \times l_p + \beta_t \times l_f \tag{7}$$

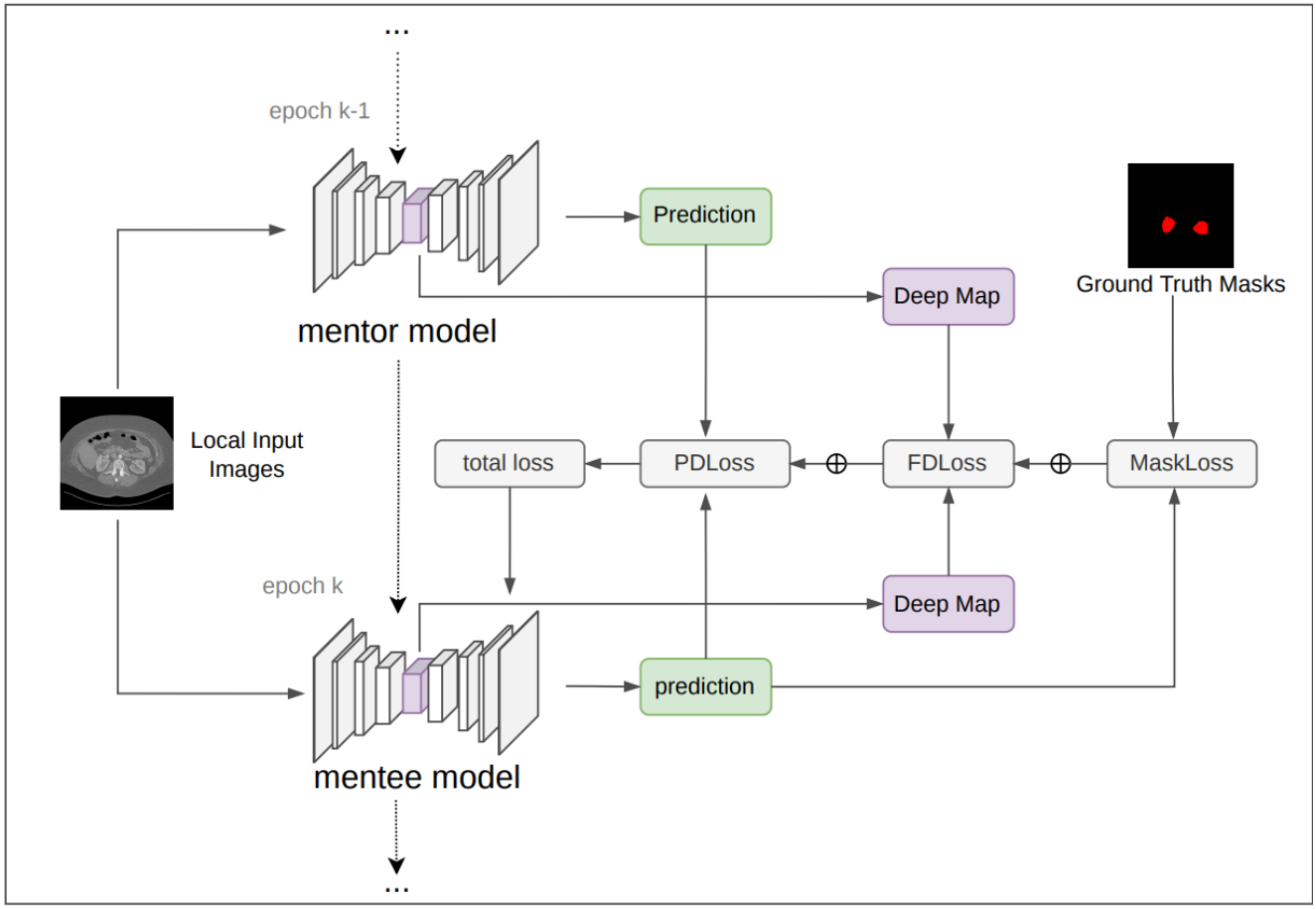


Figure 2 SKD framework. The local model distills knowledge from the previous epoch's predictions and the deepest layer network features, resulting in PDLoss and FDLoss

3 Experiments

We conducted a series of compelling experiments using the widely adopted U-Net segmentation architecture. In our proposed method, we utilized the Stochastic Gradient Descent (SGD) optimizer with a momentum of 0.9. The initial local learning rate for Unet was set to 0.01, and we employed the *ReduceLROnPlateau* strategy to adjust the learning rate. The batch size was set to 16, and the number of epochs for local model training was set to 3, with a total of 50 communication rounds. The dynamic adjustment γ for SVD threshold was set to 0.01, and the initial α and β for l_p and l_f were both set to 0.5. The training and testing were conducted on an A40 GPU.

3.1 Datasets and Evaluation Metrics

KiTS19 is a challenging dataset from MICCAI. We processed 210 original 3D CT volumes into 2D slices. It obtained a total of 4180 2D 512×512 cases. For simplicity, we treated kidney tumors as part of the kidneys and performed segmentation for both the background and kidneys. COVID-19 is available on Kaggle. It consists of 3616 frontal chest X-ray images. We resize the sample images to match the dimensions of the 256×256 ground truth images using nearest-neighbor interpolation. Then, we divided the two datasets into a training and test set with a 9:1 ratio, distributing the training set data to clients. Each client used a split of 8.5:1.5 for training and validation. The test data was evenly distributed to each client to validate the performance of the local models.

We quantitatively assess the proposed FedSKD method using the Dice coefficient (*Dice*), Jaccard similarity coefficient (*Jac*), and Sensitivity (*Se*). In practical performance evaluation, higher mean values of *Dice*, *Jac*, and *Se* reflect a more accurate segmentation performance of the model.

3.2 Federated Performance Evaluation

In our federated framework, we have five clients with varying dataset sizes, mimicking real-world imbalances, where differences in training data size exceed 1000 samples. We allocated aggregation weights to clients based on the proportions of their datasets. We compared the global model performance of FedSKD with the following approaches: (1) FedAvg, where the server aggregates the average of all client parameters. (2) FedProx is controlled by a hyperparameter $\mu = 0.01$, which regulates the weight of the proximal term. (3) SCAFFOLD: This method introduces a server control variable c and client control variables c_i to manage the direction of model updates in the presence of gradient differences. Additionally, we compared these methods to models trained individually for each client. Each client was trained for 150 epochs, and we monitored the model's performance every 3 epochs.

Table 1 Comparison of FedSKD with other methods and clients individual training on KiTS19 and COVID-19

Model	KiTS19				COVID-19			
	Dice	**Jac**	**Se**	**comm.cost**	**Dice**	**Jac**	**Se**	**comm.cost**
Client1	0.920 ±0.012	0.834 ±0.025	0.910 ±0.017	-	0.976 ±0.012	0.955 ±0.017	0.972 ±0.014	-
Client2	0.702 ±0.013	0.582 ±0.028	0.709 ±0.019	-	0.933 ±0.020	0.878 ±0.029	0.939 ±0.022	-
Client3	0.921 ±0.016	0.855 ±0.017	0.917 ±0.014	-	0.967 ±0.017	0.939 ±0.019	0.966 ±0.013	-
Client4	0.891 ±0.016	0.840 ±0.021	0.901 ±0.018	-	0.973 ±0.010	0.952 ±0.018	0.978 ±0.015	-
Client5	0.856 ±0.012	0.781 ±0.018	0.859 ±0.021	-	0.968 ±0.013	0.941 ±0.023	0.965 ±0.021	-
FedAvg	0.927 ±0.013	0.868 ±0.018	0.928 ±0.017	65.88MB	0.979 ±0.010	0.960 ±0.017	0.981 ±0.010	65.88MB
FedProx	0.849 ±0.022	0.786 ±0.027	0.910 ±0.019	65.88MB	0.969 ±0.011	0.954 ±0.019	0.973 ±0.016	65.88MB
SCAFFOLD	0.837 ±0.011	0.761 ±0.028	0.883 ±0.015	65.88MB	0.941 ±0.014	0.895 ±0.024	0.919 ±0.027	65.88MB
FedSKD	0.930 ±0.017	0.860 ±0.017	0.930 ±0.012	38.87MB	0.980 ±0.009	0.963 ±0.018	0.976 ±0.021	43.48MB

Table 1 displays the tested results (*mean* ±*std*), and it is evident that our FedSKD method demonstrates the best.

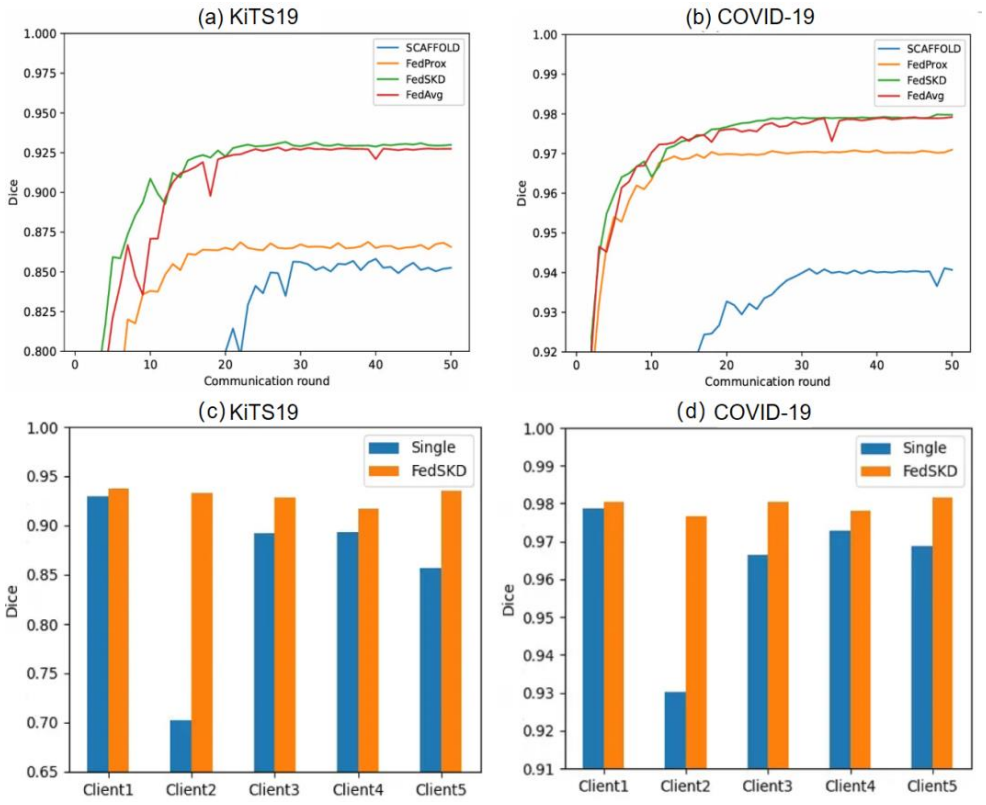


Figure 3 (a), (b) present the model performance comparison of FedSKD against other FL methods on KiTS19 and COVID-19. (c), (d) show the Dice score comparison between FedSKD clients and clients trained individually (Single) on KiTS19 and COVID-19

Dice and *Se* values (in bold font) on the KiTS19 dataset compared to other methods. For COVID-19, it also achieves the highest *Dice* and *Jac* values. Among them, "comm.cost" represents the average communication cost per round during the federated communication process. It can be observed that with a slight improvement in computational performance, there is a significant reduction in federated communication costs on both datasets. Figure 3 (a) and (b) further illustrate that FedSKD outperforms FedAvg, FedProx, and SCAFFOLD in training the optimal model performance. Figure 3(c) and (d) demonstrate that, by upholding data privacy and security, FedSKD allows all clients to attain better model performance compared to individual training (Single). Notably, clients with limited data, such as Client 2, experience particularly significant performance improvements. Figure 4 presents visual examples of these methods using the COVID-19 and KiTS19 datasets. It can be observed that FedSKD can produce segmentation results that are closer to the ground truth compared to other methods.

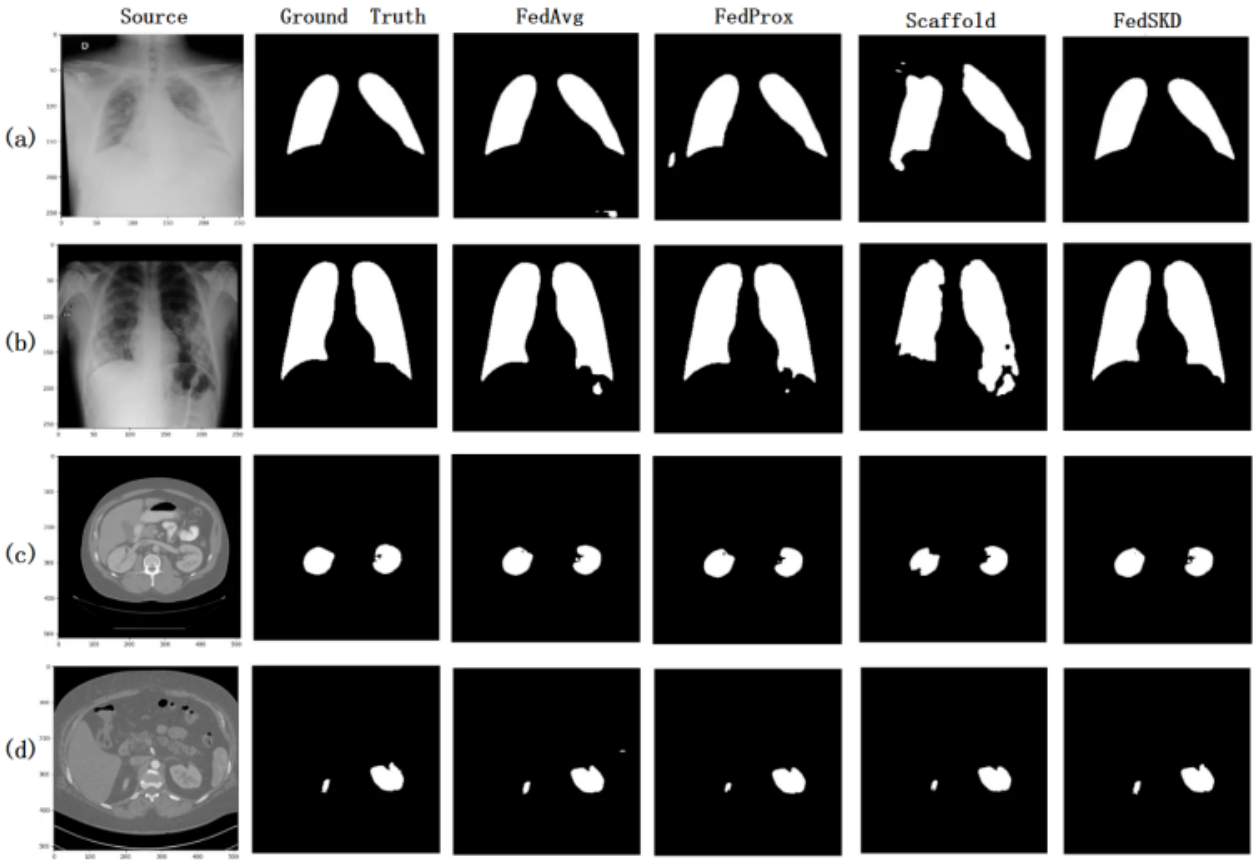
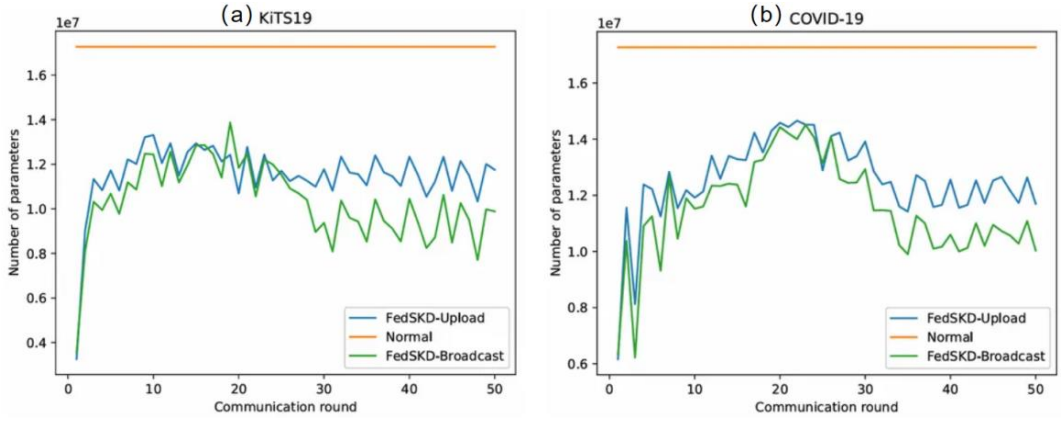


Figure 4 Segmentation results of FedSKD on the COVID-19 (a and b) and KiTS19 (c and d)



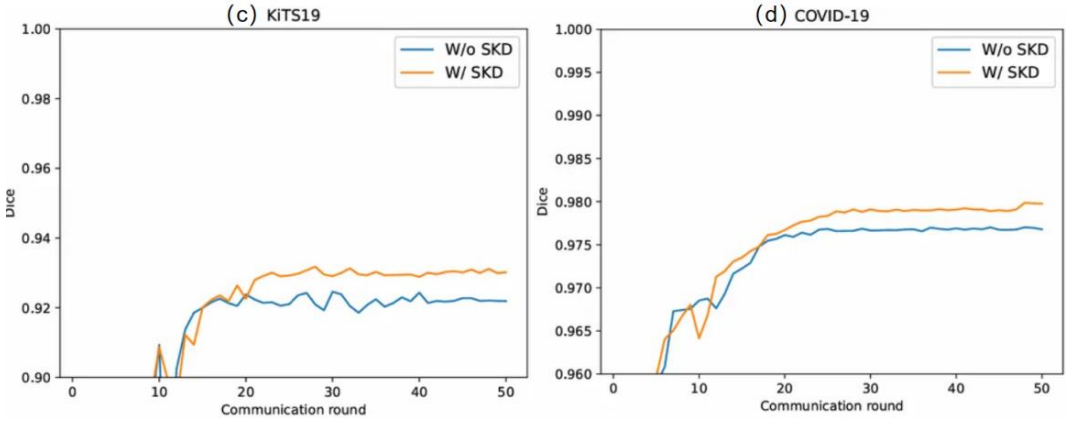


Figure 5 (a), (b) The effect of DPDC on KiTS19 and COVID-19. (c), (d) The effect of SKD on KiTS19 and COVID-19

3.3 Ablation Study

Effects of DPDC. We compared the data transmission sizes on the KiTS and COVID-19 datasets throughout the entire federated communication process when utilizing DPDC and when not using it (Normal). As shown in Figure 5 (a) and (b), when DPDC is not employed, clients and the server need to communicate all model parameters during each communication round, resulting in a fixed parameter volume of approximately 17.27 million for each round. However, with DPDC, the size of the communication data experiences a relatively rapid increase from the beginning of training to the convergence phase. After model convergence, the size of the communication data gradually decreases and stabilizes. FedSKD achieves an average reduction in communication size of approximately 34% and 41% during the upload and broadcast stages for the KiTS19 dataset, and 28% and 34% for the COVID-19 dataset. In summary, FedSKD can significantly reduce communication costs in the upload and broadcast stages, achieving over 30% savings on both datasets. This outcome holds crucial practical significance, especially in large-scale distributed systems, where it can significantly enhance model training efficiency and speed, particularly in cases with limited communication bandwidth.

Effects of SKD. Effects of SKD. Under constant conditions, we compared the changes in model accuracy on the KiTS19 and COVID-19 datasets when utilizing the SKD module versus not using it, as depicted in Figure 5 (c) and (d). It's clear that the federated model with the SKD module performs better than the one without SKD. This confirms the model's effectiveness in improving accuracy by learning from its previous states.

4 Conclusion

Our work presents FedSKD, an efficient federated learning framework for medical image segmentation. Prioritizing client privacy with DPDC significantly reduces communication costs. Distilling past model knowledge effectively enhances the accuracy of local models. Evaluation of the KiTS19 and COVID-19 datasets demonstrates the model's effectiveness. Future efforts aim to enhance client model accuracy and further decrease communication overhead in complex scenarios.

References

[1] Singh A, Sengupta S, Lakshminarayanan V. Explainable deep learning models in medical image analysis [J]. Journal of imaging, 2020, 6(6): 52.

[2] Liu S, Liu C, Ji Y, et al. Regional consistency for semi-supervised segmentation of 3D medical images [J]. IEEE Signal Processing Letters, 2023.

[3] Ronneberger O, Fischer P, Brox T. U-net: Convolutional networks for biomedical image segmentation [C] // Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18. Springer International Publishing, 2015: 234-241.

[4] Zhang Y, Xi R, Zeng L, et al. Structural Priors Guided Network for the Corneal Endothelial Cell Segmentation [J]. IEEE Transactions on Medical Imaging, 2023.

[5] Zhang X, Liu A, Jiang P, et al. MSAIF-Net: A Multi-Stage Spatial Attention based Invertible Fusion Network for MR Images [J]. IEEE Transactions on Instrumentation and Measurement, 2023.

[6] Langer S G. Cyber-security issues in healthcare information technology [J]. Journal of digital imaging, 2017, 30: 117-125.

[7] Xu J, Glicksberg B S, Su C, et al. Federated learning for healthcare informatics [J]. Journal of healthcare informatics research, 2021, 5: 1-19.

[8] McMahan B, Moore E, Ramage D, et al. Communication-efficient learning of deep networks from decentralized data [C] // Artificial intelligence and statistics. PMLR, 2017: 1273-1282.

[9] Li T, Sahu A K, Zaheer M, et al. Federated optimization in heterogeneous networks [J]. Proceedings of Machine learning and systems, 2020, 2: 429-450.

[10] Karimireddy S P, Kale S, Mohri M, et al. Scaffold: Stochastic controlled averaging for federated learning [C] // International conference on machine learning. PMLR, 2020: 5132-5143.

[11] Ribero M, Vikalo H. Communication-efficient federated learning via optimal client sampling [J]. arxiv preprint arxiv:2007.15197, 2020.

[12] Rothchild D, Panda A, Ullah E, et al. Fetchsgd: Communication-efficient federated learning with sketching [C] // International Conference on Machine Learning. PMLR, 2020: 8253-8265.

[13] Bernstein J, Wang Y X, Azizzadenesheli K, et al. signSGD: Compressed optimisation for non-convex problems [C] // International Conference on Machine Learning. PMLR, 2018: 560-569.

[14] Konecnýj, McMahan H B, Yu F X, et al. Federated learning: Strategies for improving communication efficiency [J]. arxiv preprint arxiv: 1610.05492, 2016, 8.

[15] Liu P, Sun M, Zhou S K. Multi-site organ segmentation with federated partial supervision and site adaptation [J]. arxiv preprint arxiv: 2302.03911, 2023.

[16] Abouei J, Seyedmohammadi S J, Sheikholeslami S M, et al. MoFLeuR: Motion-based Federated Learning Gesture Recognition [J]. Authorea Preprints, 2023.

[17] Fraboni Y, Vidal R, Kameni L, et al. Clustered sampling: Low-variance and improved representativity for clients selection in federated learning [C] // International Conference on Machine Learning. PMLR, 2021: 3407-3416.

[18] Hinton G, Vinyals O, Dean J. Distilling the knowledge in a neural network [J]. arxiv preprint arxiv:1503.02531, 2015.

[19] Wu C, Wu F, Lyu L, et al. Communication-efficient federated learning via knowledge distillation [J]. Nature communications, 2022, 13(1): 2032.

[20] Li K, Yu L, Wang S, et al. Towards cross-modality medical image segmentation with online mutual knowledge distillation [C] // Proceedings of the AAAI conference on artificial intelligence. 2020, 34(01): 775-783.

[21] Zhu Z, Hong J, Zhou J. Data-free knowledge distillation for heterogeneous federated learning [C] // International conference on machine learning. PMLR, 2021: 12878-12889.

[22] Kim K, Ji B M, Yoon D, et al. Self-knowledge distillation with progressive refinement of targets [C] // Proceedings of the IEEE/CVF international conference on computer vision. 2021: 6567-6576.

[23] Ji M, Shin S, Hwang S, et al. Refine myself by teaching myself: Feature refinement via self-knowledge distillation [C] // Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2021: 10664-10673.

[24] Yang S, Zheng X, Xu Z, et al. A Lightweight Approach for Network Intrusion Detection based on Self-Knowledge Distillation [C] // ICC 2023-IEEE International Conference on Communications. IEEE, 2023: 3000-3005.