Dynamic Corrected Split Federated Learning With Homomorphic Encryption for U-Shaped Medical Image Networks

Ziyuan Yang, Yingyu Chen, Huijie Huangfu, Maosong Ran, Hui Wang, Xiaoxiao Li, *Member, IEEE*, and Yi Zhang, *Senior Member, IEEE*

***Abstract*—U-shaped networks have become prevalent in various medical image tasks such as segmentation, and restoration. However, most existing U-shaped networks rely on centralized learning which raises privacy concerns. To address these issues, federated learning (FL) and split learning (SL) have been proposed. However, achieving a balance between the local computational cost, model privacy, and parallel training remains a challenge. In this articler, we propose a novel hybrid learning paradigm called Dynamic Corrected Split Federated Learning (DC-SFL) for U-shaped medical image networks. To preserve data privacy, including the input, model parameters, label and output simultaneously, we propose to split the network into three parts hosted by different parties. We propose a Dynamic Weight Correction Strategy (DWCS) to stabilize the training process and avoid the model drift problem due to data heterogeneity. To further enhance privacy protection and establish a trustworthy distributed learning paradigm, we propose to introduce additively homomorphic encryption into the aggregation process of client-side model, which helps prevent potential collusion between parties and provides a better privacy guarantee for our proposed method. The proposed DC-SFL is evaluated on various medical image tasks, and the experimental results demonstrate its effectiveness. In comparison with state-of-the-art distributed learning methods, our method achieves competitive performance.**

***Index Terms*—Split federated learning, U-shaped medical image network, privacy-preserving, homomorphic encryption.**

Manuscript received 23 May 2023; revised 6 August 2023; accepted 16 September 2023. Date of publication 20 September 2023; date of current version 6 December 2023. This work was supported in part by the National Natural Science Foundation of China under Grant 62271335, in part by the Sichuan Science and Technology Program under Grant 2021JDJQ0024, and in part by the Sichuan University "From 0 to 1" Innovative Research Program under Grant 2022SCUH0016. *(Corresponding author: Yi Zhang.)*

This work involved human subjects or animals in its research. Approval of all ethical and experimental procedures and protocols was granted by the Institutional Review Board (IRB) of Mayo Clinic.

Ziyuan Yang is with the College of Computer Science, Sichuan University, Chengdu 610065, China, and also with the Key Laboratory of Data Protection and Intelligent Management, Ministry of Education, Sichuan University, Chengdu 610207, China (e-mail: cziyuanyang@gmail.com).

Yingyu Chen, Huijie Huangfu, Maosong Ran, and Hui Wang are with the College of Computer Science, Sichuan University, Chengdu 610065, China (e-mail: 2021323040005@stu.scu.edu.cn; huangfu.huijie@qq.com; maosongran@gmail.com; whmzfc@gmail.com).

Xiaoxiao Li is with the Department of Electrical and Computer Engineering, The University of British Columbia, Vancouver, BC V6T1Z4, Canada (e-mail: xiaoxiao.li@ece.ubc.ca).

Yi Zhang is with the School of Cyber Science and Engineering, Sichuan University, Chengdu 610207, China, and also with the Key Laboratory of Data Protection and Intelligent Management, Ministry of Education, Sichuan University, Chengdu 610207, China (e-mail: yzhang@scu.edu.cn).

Related codes will be released at https://github.com/Zi-YuanYang/DC-SFL.

Digital Object Identifier 10.1109/JBHI.2023.3317632

I. Introduction

U-SHAPED medical image networks have gained significant success in various medical image tasks over the past decade, including segmentation [1], and restoration [2]. These deep learning methods are always data-hungry, but gathering data from multiple sources is hindered by privacy protection regulations. To address this challenge, researchers have increasingly turned to distributed collaborative machine learning (DCML) [3], [4], [5], which allows for decentralized model training without requiring access to raw data from a single institution [6].

Federated Learning (FL) [7], [8] is a prominent DCML paradigm that facilitates parallel training and has achieved satisfactory training time overhead. However, FL faces two significant challenges. First, each client needs to train a full model, which requires high computational resources and may not be suitable for resource-constrained environments. Second, model privacy is a concern, as all participants, including clients and the server, have full access to the entire model, increasing potential privacy and security risks [9].

Split Learning (SL) was recently proposed as a solution to FL's challenges by dividing the full model into server-side and client-side models [10]. This approach offers two significant benefits. First, clients only need to train a part of the model, while the most resource-intensive aspect is trained on the server. Second, each party, including clients and the server, only has partial access to the full model. Hence, SL enhances model privacy and requires fewer client-side computational resources than FL. Despite the above merits, SL's primary concern is training cost. The training process of SL is sequential, which only allows one client to engage with the server at one instance [11].

In addition to the challenges mentioned above, data heterogeneity remains a significant issue for all DCML paradigms [12]. Generally, the heterogeneity problem is caused by the non-independent and identically distributed (non-iid) data [13]. For

instance, the medical images from various hospitals may exhibit significant differences in noise level, contrast, and resolution. These data are typically non-iid. The heterogeneity problem leads to a substantial gap between the optimization directions of local and global models. On the one hand, local optimization aims to find an optimal model for local data with little regard for global performance. On the other hand, global optimization seeks to achieve a globally optimal solution, which can tolerate sub-optimal local performance. These different optimization objectives can lead to a drift between local and global models.

To relieve the above problems in a unified framework, we propose **D**ynamic **C**orrected **S**plit **F**ederated **L**earning (**DC-SFL**) for U-shaped medical image networks. DC-SFL is a hybrid learning paradigm that combines SL and FL to achieve the best of both worlds. We observed that, unlike in classification tasks, the labels and outputs in segmentation and restoration tasks often contain privacy-sensitive information. To address this issue and reduce the computational costs on the client-side, we split the U-shaped network into three parts: the head, body, and tail models. There are three parties in the proposed framework: a computation server, an aggregation server, and clients. The head and tail models are hosted by clients, while the body, which requires significant computational resources, is hosted by the computation server. Each client corresponds to one body model in the computation server to enable parallel training. All clients perform forward propagation of their client-side models in parallel and upload the outputs of head models to the computation server. The body models are executed separately in parallel and transmit their outputs to clients as the inputs of tail models. The training process are independently performed in each client, and client-side and server-side models are aggregated in the aggregation and computation servers after a communication round, respectively. It is noteworthy that the input, label, and output are not transferred to other parties, and each party only partially accesses the full model throughout the learning process.

Moreover, to address the data heterogeneity problem, we propose a dynamic weight correction strategy (DWCS) to mitigate model drift and correct the global model. Specifically, we design a weight correction loss to quantify the drift between models from two consecutive communication rounds, and optimize the correction model by minimizing this loss. We then treat the weighted sum of the correction and last round models as the final corrected model. At the beginning of the training process, we assign a large weight to the current round model as it is underfitting. As the training proceeds, we increase the weight for the correction model to address model drift caused by local training and approach the global optimal solution. In summary, we expect to stabilize the training process by dynamically adjusting the weights of the two models to prevent model drift.

The aforementioned learning process operates under the assumption that all parties are benign. However, this leaves the model privacy vulnerable and the complete model susceptible to unauthorized access, especially in cases where the aggregation server and the computation server collude. To mitigate such risks, we have incorporated the Homomorphic Encryption (HE) methodology to enable secure multi-party computation. Specifically, we have chosen to utilize the Paillier cryptosystem [14], for its exceptional capacity to uphold stringent privacy measures without compromising the accuracy of results. The main contributions of this articler are summarized as follows:

- To simultaneously protect the privacy of input, model parameters, label and output, we propose a novel distributed learning framework DC-SFL for U-shaped networks, which can further facilitate parallel training and reduce local computation cost.
- To relieve the problems of data heterogeneity and model drift, we propose DWCS to dynamically correct the global model.
- We have integrated HE into SFL to facilitate reliable distributed learning, which offers robust privacy protection. To the best of our knowledge, this is the first attempt at merging HE and SFL.

II. Related Workds

A. U-Shaped Medical Image Networks

Since U-Net was proposed [15], most segmentation methods chose U-shaped architecture as the backbone and achieved encouraging performances [16]. For example, Fakhry et al. [17] combined the residual connection with U-Net. Similarly, some researchers used DenseNet block to replace regular convolutional layer [18]. Oktay et al. [19] proposed a novel attention gate segmentation model, which is dubbed Attention U-Net. Isensee et al. [20] proposed nnUNet, which trains the vanilla U-Net with multiple preprocessing steps and surpassed most existing approaches. The success of these methods demonstrates the effectiveness of U-shaped structure in medical segmentation [21].

Besides, U-shaped architecture is also widely used in medical restoration tasks [22]. For example, Chen et al. [23] introduced the residual block into the U-shaped autoencoder for low-dose CT restoration, which is dubbed residual encoder-decoder convolutional neural network (RED-CNN). Wang et al. [24] proposed generative adversarial networks with dual-domain U-Net-based discriminators for low-dose CT restoration. Wang et al. [25] combined the transformer block and U-Net, and achieved impressive performance. On the other hand, many recent proposed works chose U-shaped networks as the benchmark for medical image restoration [26], [27], [28]. Although the above segmentation and imaging methods achieved competitive performance, they need to collect numerous samples from multiple different data sources and usually ignore data privacy.

B. Distributed Collaborative Machine Learning

FL is one of the most popular DCML paradigms, which trains a full network at each client in parallel, and then the local gradients are transferred to the server for aggregation [29]. Typically, McMahan et al. [30] proposed FedAvg, which learns the global model by aggregating local models. Furthermore, Li et al. [31] proposed FedProx, which can be considered a re-parametrization of FedAvg. Li et al. [32] represented a personalized federated learning method FedBN, which alleviates the feature shift using personalized batch normalization (BN) in clients. However, current FL methods usually suffer from two limitations: 1) Clients need to train the full model, which may be challenging for resource-constrained clients; 2) Each participant can access the full model, which risks privacy leaks.

Recently, SL was proposed to alleviate the above problems by splitting a full model into multiple parts [10] and each client

only needs to train a part of the model. SL seems to be a better choice than FL in computational resource-constrained environment [11]. In [33], graph neural network is combined with SL to protect model privacy. Jeon et al. [34] proposed parallel SL (PSL) learning to reduce the training time overhead. In this method, each client preserves its local model and doesn't upload it to other clients.

Recently, Researchers have been enthusiastic about introducing DCML to the healthcare field to protect data privacy. Feng et al. [35] proposed a personalized magnetic resonance imaging method FedMRI, which consists of a globally shared encoder and client-specific decoders. Yang et al. [36] leveraged the prior information of scanning parameters to modulate different local models for CT imaging. In [37], federated domain generalization (FedDG) utilized the frequency information from different clients to handle the data heterogeneity. Roth et al. [38] combined split learning and U-Net, but labels and inputs are hosted in different parties, which is against the privacy settings.

However, FL and SL-based methods are hampered in some challenges [39]. To enjoy the best of both worlds, Thapa et al. [9] proposed split federated learning (SFL), the hybrid of FL and SL, which achieved satisfactory training overhead and prediction accuracy simultaneously. Zhang et al. [40] evaluated the performance of SFL in several medical tasks, but this method transfers outputs or labels between different parties, violating the privacy setting, it is unsuitable for the widely used U-shaped networks for medical image segmentation and restoration.

C. Privacy-Preserving Methods

Privacy concerns continue to be a significant hurdle for DCML methods. For example, research conducted by Geiping et al. [41] has shown the possibility of recovering private data from gradients. DP [42] and HE [43] are two popular strategies to mitigate this issue. DP is a non-cryptographic approach wherein each user perturbs gradients locally. However, this method can potentially compromise performance, as the perturbed gradients are then incorporated into the aggregation operation [44].

Unlike DP, HE-based methods allow specific mathematical operations to be performed directly on ciphertexts without requiring decryption, thus maintaining privacy [14]. The Paillier cryptosystem, which upholds additive homomorphism, has gained wide attention in DCML methods in recent years. For instance, Fang and Qian [45] incorporated the Paillier cryptosystem into a federated medical diagnosis framework, achieving satisfactory results.

In an effort to lessen the encryption and communication overhead imposed by HE, Ma et al. [46] made a concerted effort. Tang et al. [47] introduced a robust, privacy-preserving federated learning framework that leverages the Paillier cryptosystem for gradient verification to ensure data privacy. Furthermore, Liang and Shin [48] proposed an auditable Federated Learning (FL) scheme endowed with Byzantine robustness, applying homomorphic encryption to enable clients to securely aggregate gradients as a defense mechanism against attacks.

III. Proposed Method

A. Problem Formulation

Suppose that there are N clients, denoted as C_1,..., C_N. Each client C_i has a local specific dataset $\mathcal{D}_i$. The goal of FL is to learn a full model $\mathcal{M}$ from $\mathcal{D} = \bigcup_{i=1}^{N} \mathcal{D}_i$. Then, the optimization of FL can be formulated as follows:

$$\arg\min_{\theta} \mathcal{L} = \sum_{i=1}^{N} \frac{|\mathcal{D}_i|}{|\mathcal{D}|} \mathbb{E}_{(x,y)\sim\mathcal{D}_i} \left[\mathcal{L}_i(\mathcal{M}(x,\theta), y)\right], \quad (1)$$

where $\mathcal{L}_i$ denotes the loss function for C_i, and $\mathcal{L}$ means the overall loss function across all clients. x and y represent the input and label, respectively. $|\mathcal{D}_i|$ and $|\mathcal{D}|$ are the number of samples in $\mathcal{D}_i$ and $\mathcal{D}$. θ is the parameter set of $\mathcal{M}$.

Different from FL, whose optimal objective is the full network, the optimal objective of SL is composed of multiple partitions. Assume $\mathcal{M}$ is split into two parts, the client-side model $\mathcal{M}_c$ and the server-side model $\mathcal{M}_s$. The optimization of SL is formulated as:

$$\arg\min_{\theta_s,\theta_c} \mathcal{L} = \sum_{i=1}^{N} \mathbb{E}_{(x,y)\sim\mathcal{D}_i} \left[\mathcal{L}_i(\mathcal{M}_s(\mathcal{M}_c(x,\theta_c),\theta_s), y)\right], \quad (2)$$

where θ_c and θ_s are the parameter sets of $\mathcal{M}_c$ and $\mathcal{M}_s$, respectively.

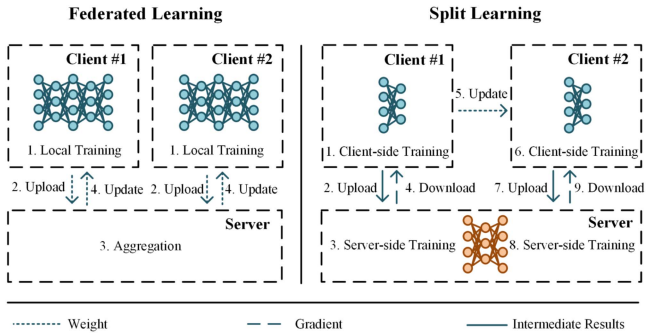

Fig. 1. Learning process of FL and SL. The numbers represent the order of processing, and different lines denote different kinds of dataflow.

B. Architecture of DC-SFL

The learning processes of FL and SL are illustrated in Fig. 1. It can be observed that the optimization objectives of these two methods are different. FL expects an optimal full network, but SL tries to optimize the different partition networks. SL is effective at protecting model privacy, and its local computational costs are low. However, its learning process is sequential, leading to significant training time overhead and under-utilization of client computational resources. In contrast, FL can be implemented in a parallel way, but its clients need to train full models locally, which requires high client-side computational resources.

Actually, SL and FL are complementary and their drawbacks can be addressed by the other. The split operation of SL can reduce the client-side computational costs and protect model privacy. Meanwhile, the aggregation operation in FL can leverage local computational resources and significantly lower training time overhead. To enjoy the benefits of both worlds, we propose a hybrid learning paradigm for U-shaped medical image networks. A similar idea was also proposed in [40], but its splitting method is not suitable for segmentation and restoration. The main reasons lie in that:

1) *Violating privacy setting:* This method transfers the label or output to other parties and these data contain patients' privacy information in segmentation, restoration and denoising tasks.

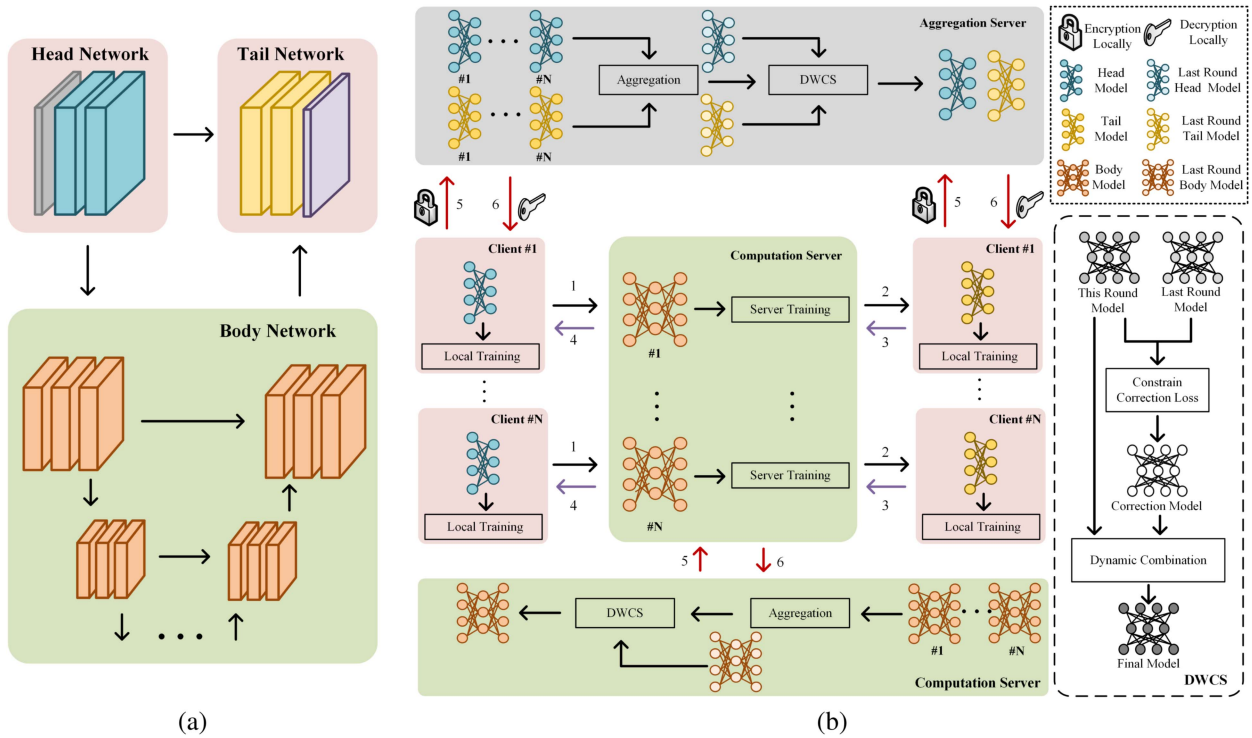

Fig. 2. Overall framework of the proposed DC-SFL. (a) Represents the proposed split strategy for U-shaped networks. (b) Denotes the whole training process of DC-SFL. The numbers represent the order of processing. The black arrow, purple arrow, and red arrow denote the intermediate result, gradient, and weight dataflows, respectively.

2) *Extra bandwidth:* This method requires extra bandwidth to transfer intermediate features of shortcut connections between different parties. However, shortcut connections are a key contribution of U-shaped medical image networks, and it may be inappropriate to remove them to improve communication efficiency.

These challenges motivate us to propose a split method that does not require the sharing of input, model parameters, output, and labels between different parties. As shown in Fig. 2(a), the full network is split into three parts: the head, body, and tail networks. The lightweight head and tail networks are hosted on clients to reduce local computational costs, while the computational resource-intensive body network is hosted on a server with high-performance computational resources. We can then formulate the forward process as follows:

$$\mathcal{F}(x) = \mathcal{M}_t(\mathcal{M}_b(\mathcal{M}_h(x, \theta_h), \theta_b), \theta_t), \tag{3}$$

where $\mathcal{M}_h$, $\mathcal{M}_b$ and $\mathcal{M}_t$ denote the head, body and tail networks with parameter sets θ_h, θ_b and θ_t, respectively.

As mentioned above, the output of each encoder layer is the input of the corresponding decoder layer. In situations where the connected encoder and decoder layers are hosted on different parties, the output of the encoder layers must be uploaded to another party, leading to increased communication costs. In our method, the corresponding encoder and decoder layers are hosted in the same party, eliminating the need for additional communication bandwidth for feature transfer. Notably, the input, label, and output are preserved locally without any sharing requirements in our method, so data privacy is well protected.

C. DC-SFL Implementation

Based on the proposed splitting strategy, we introduce a collaborative distributed training framework, DC-SFL, that enables parallel training while protecting the privacy of input, output, model and label. The flowchart of DC-SFL is illustrated in Fig. 2(b). As previously mentioned, clients and servers all have full access to the model in FL, which can violate the principle of model privacy protection. To address this issue, we propose a three-party training approach where clients, computation servers, and aggregation servers each only have access to a apart of the model. Specifically, clients and aggregation servers can access the head and tail models, while the computation server accesses the body model. To enable parallel training, we build N body models on the computation server corresponding to N clients, and each client's body model is executed separately to reduce training time overhead.

At the beginning of training, client-side and server-side models are initialized in the aggregation and computation servers, respectively. All clients perform forward propagation of head models locally: $\hat{y}_h = \mathcal{M}_h(x, \theta_h)$, and then deliver the encoded results to the computation server. Benefiting from the setting of multiple body models, the forward propagation of body models can be executed in parallel: $\hat{y}_b = \mathcal{M}_b(\hat{y}_h, \theta_b)$. At the end of the forward path, $\hat{y}_b$ is delivered to the clients to generate the final prediction of tail networks as $\hat{y} = \mathcal{M}_t(\hat{y}_b, \theta_t)$.

After forward propagation, each client calculates the loss and starts backpropagation. Concretely, the gradients about $\mathcal{M}_t$ and $\hat{y}_b$ are calculated at first. Then, the gradients of $\hat{y}_b$ are transmitted to the computation server, and the server executes the backpropagation on $\mathcal{M}_b$ and deliver the gradients of $\hat{y}_h$ to clients.

Finally, with the received gradients of $\hat{y}_h$, the client executes the backpropagation of $\mathcal{M}_h$. So far, one backpropagation pass between clients and the server is completed. To make full use of all local data and get optimal global models, we aggregate the client-side and server-side models in the aggregation and computation servers, respectively. Until now, one complete global training round has been finished. It can be observed that the training processes of different clients are executed in parallel, which greatly reduce the training time overhead of SL. Meanwhile, there is no party in our framework that can access the full model, which effectively protects model privacy. Then, the servers provide aggregated models by averaging local models as:

$$\theta^k = \sum_{i=1}^{N} \frac{|\mathcal{D}_i|}{|\mathcal{D}|} \theta_i^k, \tag{4}$$

where k and i represent the current training round and client index, respectively. $\theta^k = \{\theta_h^k, \theta_b^k, \theta_t^k\}$ represents the parameter sets of $\mathcal{M}^k = \{\mathcal{M}_h^k, \mathcal{M}_b^k, \mathcal{M}_t^k\}$. Then our optimization problem is formulated as:

$$\arg\min_{\theta_h, \theta_b, \theta_t} \mathcal{L} = \sum_{i=1}^{N} \frac{|\mathcal{D}_i|}{|\mathcal{D}|} \mathbb{E}_{(x,y)\sim\mathcal{D}_i} \left[\mathcal{L}_i(\mathcal{F}(x), y)\right]. \tag{5}$$

The proposed framework comprises two parties engaged in parallel training: the clients and the computation server. Similar to other FL methods, each client has independent computation resources, enabling it to perform the training process in parallel with other clients. As mentioned before, the computation server is equipped with substantial computational resources, such as GPU or CPU cluster. In the practical implementation, when the computation server receives intermediate features from clients, it allocates idle computation resources to each client to facilitate the corresponding training process. As a result, basically, if the total number of computation resources (such as the number of GPUs) is greater than or equal to the number of clients participating in the training, the computation server can achieve parallel training.

D. Dynamic Weight Correction Strategy

The original DCML suffers from the data heterogeneity problem, which leads to a huge gap between the optimization directions of local and global models. Since there is always a distribution gap between D_i and D in practice, local training will lead the local model to work badly in other data domains. As a result, it may generate a poor optimization solution to the global model and cause model collapse after aggregation. This problem happens more commonly in healthcare tasks, in which the collected data inevitably suffer from serious data heterogeneity caused by several factors, such as different hardware, scanning protocols and patients.

To recover from this situation, we propose DWCS to avoid the model drift problem. Specifically, we treat the model of the last communication round as the anchor model and propose a weight correction loss to quantify the drift between the anchor model and its adjacent communication round model. Then we get the correction model by minimizing the weight correction loss, and the weighted sum of the correction and last round models is treated as the final result. The weight correction loss function is

Algorithm 1: Main Steps of DC-SFL Without Encryption.

Function Main: ▷ Computation Server Executes
  Initialize θ_b^0
  for *round* $k = 1, 2, ..., R$ **do**
    AggSer$(k-1)$
    for *client* $n = 1, 2, ..., N$ **in parallel do**
      $\theta_{b,n}^k \leftarrow \theta_b^{k-1}$
    for *epoch* $i = 1, 2, ..., E$ **do**
      for *client* $n = 1, 2, ..., N$ **in parallel do**
        $\hat{y}_h \leftarrow$**HeadForward**(n, k)
        $\hat{y}_b \leftarrow \mathcal{M}_b(\hat{y}_h, \theta_{b,n}^k)$
        $\frac{\partial \mathcal{L}_n}{\partial \hat{y}_b} \leftarrow$ **TailMain**$(n, \hat{y}_b)$ & Backprop
        HeadBack$(n, \frac{\partial \mathcal{L}_n}{\partial \hat{y}_b})$
        $\theta_{b,n}^k \leftarrow \theta_{b,n}^k - \eta \frac{\partial \mathcal{L}_n}{\partial \theta_{b,n}^k}$
    $\theta_b^k \leftarrow \sum_{n=1}^{N} \frac{|\mathcal{D}_n|}{|\mathcal{D}|} \theta_{b,n}^k$
    $\theta_b^k \leftarrow$**Correct**$(\theta_b^k, \theta_b^{k-1})$
Function HeadForward(n, k): ▷ Client C_n Executes
  $x_n \leftarrow$ Sampled input batch from D_i
  return $\mathcal{M}_h(x_n, \theta_{h,n}^k)$
Function TailMain$(n, \hat{y}_b)$: ▷ Client C_n Exceutes
  $y_n \leftarrow$ Sampled label batch from D_i
  $\hat{y}_t \leftarrow \mathcal{M}_t(\hat{y}_b, \theta_{t,n}^k)$
  $\mathcal{L}_n \leftarrow$TaskLoss$(\hat{y}_t, y_n)$
  Backprop & $\theta_{t,n}^k \leftarrow \theta_{t,n}^k - \eta \frac{\partial \mathcal{L}_n}{\partial \theta_{t,n}^k}$
  return $\frac{\partial \mathcal{L}_n}{\partial \hat{y}_{b,n}^k}$
Function HeadBack$(n, \frac{\partial \mathcal{L}_n}{\partial \hat{y}_b})$: ▷ Client C_n Executes
  Backprop & $\theta_{h,n}^k \leftarrow \theta_{h,n}^k - \eta \frac{\partial \mathcal{L}_n}{\partial \theta_{h,n}^k}$
Function AggSer(k): ▷ Aggregation Server Exceutes
  if $k = 0$ **then**
    Initlialize θ_h^k, θ_t^k
  else
    $(\theta_h^k, \theta_t^k) \leftarrow (\sum_{n=1}^{N} \frac{|\mathcal{D}_i|}{|\mathcal{D}|} \theta_{h,n}^k, \sum_{n=1}^{N} \frac{|\mathcal{D}_i|}{|\mathcal{D}|} \theta_{t,n}^k)$
    $(\theta_h^k, \theta_t^k) \leftarrow$(**Correct**$(\theta_h^k, \theta_h^{k-1})$,**Correct**$(\theta_t^k, \theta_t^{k-1})$)
  Deliver θ_h^k, θ_t^k to clients
Function Correct(θ^k, θ^{k-1}):
  $\mathcal{L}_{con} \leftarrow$WeightCorrectionLoss$(\theta^k, \theta^{k-1})$
  $\theta_c^k \leftarrow \theta^k - \eta \frac{\partial \mathcal{L}_{con}}{\partial \theta^{k-1}}$
  $\theta_r^k \leftarrow (1-\alpha)\theta^k + \alpha \theta_c^k$
  return θ^k

defined as:

$$\arg\min_{\theta^k} \mathcal{L}_{con}(\theta^k, \theta^{k-1}) = \frac{\mu}{2} ||\theta^k - \theta^{k-1}||_2^2, \tag{6}$$

where $\mathcal{L}_{con}$ represents the weight correction loss, and μ is the hyperparameter constraining the optimization step factor. Then, our correction model can be formulated as:

$$\theta_c^k = \theta^k + \eta \bigtriangledown \mathcal{L}_{con}(\theta^k, \theta^{k-1}), \tag{7}$$

where η is the learning rate, and our correction model is $\theta_c^k = \{\theta_{c,h}^k, \theta_{c,b}^k, \theta_{c,t}^k\}$.

In the early stage of training, a small weight should be assigned to the correction model to accelerate convergence. With the training continuing, the model almost converges, but the local

training may cause severe model drift, which leads to global model collapse by aggregating. To alleviate the above issue, inspired by [49], we propose a dynamic adjustment strategy to stabilize the training process and minish the model drift. Then, a robust model θ_r^k can be obtained with the weighted summation of θ_c^k and θ^k, which can be defined as:

$$\theta_r^k = (1-\alpha)\theta^k + \alpha\theta_c^k. \tag{8}$$

where $\alpha = \min(1 - \frac{1}{k+1}, \beta)$ is the balancing factor, and β denotes the maximum constraint value, which is set to 0.99 in this articler.

To help readers to follow our method, the main steps of DC-SFL without encryption are introduced in Algorithm 1 in pseudocode style. It can be noticed that the proposed DC-SFL is a parallel DCML method without sharing the input, model parameters, output and label. As a result, the model privacy can be well protected in the whole training phase. Additionally, we only need to perform DWCS once during one communication round, so our method is lighter than regularization-based methods.

E. Encryption

To avoid collusion among different parties and provide a strong privacy guarantee, we incorporate HE into the proposed DC-SFL framework. More specifically, we utilize the Paillier cryptosystem, a public-key encryption scheme that capitalizes on the complexity of calculating discrete logarithms as well as the decisional composite residuosity assumption as per [14]. Paillier encryption is chosen due to its additive homomorphism, making it suitable for the aggregation process in our proposed method.

Initially, each client independently generates public and private keys as follows:

$$n = pq, \tag{9}$$

where p and q are two different large prime numbers that satisfy $\gcd(pq, (p-1)(q-1)) = 1$, where $\gcd(a, b)$ denotes the greatest common divisor of a and b. Subsequently, we can arbitrarily choose a random number $g \in \mathbb{Z}_{n^2}^*$ such that $\gcd(\mathcal{P}(g^\lambda \bmod n^2), n) = 1$, where $\mathcal{P}(u) = \frac{u-1}{n}$, and $\lambda = \mathrm{lcm}(p-1, q-1)$ is the private key. $\mathrm{lcm}(a, b)$ denotes the least common multiple of a and b.

Once each local training session has concluded, clients must encrypt their models before transmitting the ciphertext message to the aggregation server. More precisely, we first select a random number r that satisfies $0 < r < n$ and $\gcd(r, n) = 1$, before computing the ciphertext as:

$$c = g^m \cdot r^n \bmod n^2, \tag{10}$$

where $c \in \mathbb{Z}_{n^2}^*$ is the ciphertext, and m denotes the parameters of the client-side models.

After encryption, the ciphertext c^i of the i-th client will be sent to the aggregation server. The server then aggregates the set of ciphertexts into $c_{agg} = \prod_{i=1}^{k} c^i$. After that, the aggregated parameters are returned to the clients, who decrypt the ciphertext and compute the plaintext as:

$$m = L(c_{agg}^\lambda \bmod n^2) \cdot \mu \bmod n, \tag{11}$$

where μ is the modular inverse of $\mathcal{P}(g^\lambda \bmod n^2)$ modulo n, and m refers to the decrypted ciphertext.

With that, the entire encryption process is presented. Both the encryption and decryption procedures are performed locally, which effectively bolsters data privacy. Consequently, the potential for collusion among different parties is notably mitigated.

IV. Experiments

A. Implementation Details

To validate the effectiveness of our method for U-shaped medical image networks in different tasks, we compare our method with several state-of-the-art DCML methods, such as SL [10], PSL [34], FedAvg [30], FedProx [31], FedBN [32], FedDG [37], FedMRI [35] and centralized learning (CL) method, which is treated as the benchmark. Notely, the network splitting strategy used in SL and PSL is what we proposed in this articler. Adam [50] is adopted to optimize the models. The learning rate is set to 1×10^{-4}, and the weight decay is 1×10^{-8}. All codes are implemented in PyTorch and the experiments are performed on an NVIDIA GTX 3090 GPU. We first evaluate the performance of DC-SFL in different tasks without encryption in this section, and then investigate the privacy-enhanced protection experiments in Section IV-E.

B. Segmentation Experiments

We conduct segmentation experiments on the public dataset Automated Cardiac Diagnosis Challenge (ACDC) [51], which contains 200 annotated short-axis cardiac MR-cine images from 100 patients. All short-axis slices within 3D scans are resized to 256×256 as 2D images. In our segmentation experimental setup, each patient's data only appears at one client. We randomly divide 80 patients on average into 4 clients as the training set, and the remaining 20 patients are treated as the testing set. There exists data heterogeneity between different patients, due to the variations in patient body size, organ size, lesion area, organ diastolic and systolic periods, patient age and gender, etc. In summary, since each patient only appears in one client and the training and testing data are from different patients, there exists a non-iid problem between clients in our segmentation settings. Furthermore, we visualize the heterogeneity degree of non-iid among different clients using the kernel density estimation (KDE) plot in Fig. 3. In Fig. 3, it can be observed that, although the scanned regions are similar in different clients, the data distributions differ due to the heterogeneity of patients.

Dice Similarity Coefficient (DSC), 95% Hausdorff Distance (HD95), Average Surface Distance (ASD), and Jaccard Index (JC) are chosen as the quantitative metrics in this articler. These metrics are commonly used to quantitatively validate the segmentation performance. DSC denotes the spatial overlap index between the predicted mask and the ground truth mask. HD95 is calculated by measuring the 95th percentile of the Hausdorff Distance, which represents the maximum distance between the predicted mask and the ground truth mask. ASD metric calculates the average distance from the segmentation result to ground truth to measure the shape fidelity, which is less sensitive to outliers than HD. JC computes the ratio of the intersection to the union of the predicted mask and the ground truth. These metrics can comprehensively evaluate the segmentation performance from various perspectives [52].

Segmentation networks are all optimized with cross-entropy loss and dice loss. The numbers of communication rounds and local training epochs are set to 300 and 1, respectively. To

TABLE I
AVERAGE QUANTITATIVE RESULTS FOR THE SEGMENTATION TASK

Method	DSC↑	HD95↓	ASD↓	JC↑
CL	0.8858	2.663	0.7291	0.8010
SL [10]	0.8760	3.002	0.9198	0.7893
PSL [34]	0.8542	3.267	0.9858	0.7557
FedAvg [30]	0.8600	3.426	1.1176	0.7642
FedProx [31]	0.8718	3.228	1.0322	0.7806
FedBN [32]	0.8629	3.422	1.0504	0.7672
FedMRI [35]	0.4805	23.183	7.6595	0.3489
FedDG [37]	0.8614	3.857	1.0877	0.7668
DC-SFL w/o DWCS	0.8733	2.897	0.8473	0.7839
DC-SFL w/ DWCS	**0.8807**	**2.594**	**0.7844**	**0.7938**

The best results are shown in boldface.

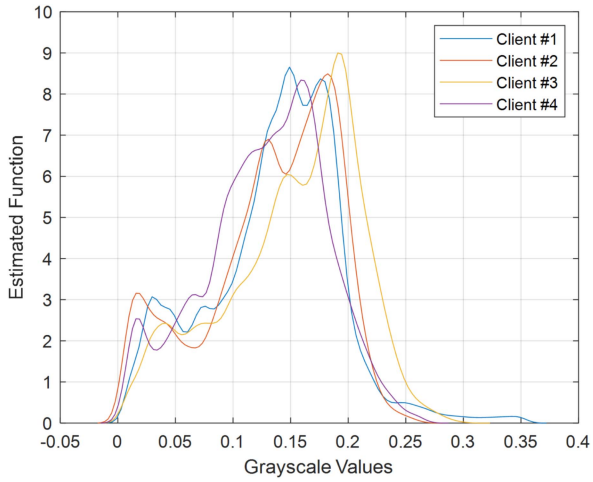

Fig. 3. KDE results of the data from different clients for segmentation task.

validate the robustness of the performances, we show the average results with three seeds.

The quantitative results are listed in Table I . It is observed that our method can achieve competitive performance to CL and outperforms other DCML methods. SL achieves satisfactory performance, but its training process is sequential, which leads the training time overhead is N times (the number of clients) longer than our DC-SFL and other FL methods. SL beats most FL-based methods. Our method inherits the advantages of SL and performs much better than FL-based methods. Three representative visual results are shown in Fig. 4 . Our method accurately identifies the boundaries of organ regions, and significantly outperforms other methods. In the first case, our result is even better than CL. We must mention that FedMRI is proposed for MRI restoration, which is unsuitable for segmentation task. In the ablation experiment, we find that DWCS can effectively improve the segmentation performance due to its good trade-off between the convergence acceleration and model drift recovery.

C. Restoration Experiments

The well-known NIH-AAPM-Mayo Low-Dose CT dataset [53], which contains 5936 full-dose CT images from 10 patients, is used to verify the performance of our method. Eight patients are randomly divided into four clients on average as the training dataset, and two patients are treated as testing dataset. To simulate real environments, we generate multi-source non-iid low-dose CT (LDCT) data following [54], [55]. Poisson noise and electronic noise were added to the measured projection data to simulate the low-dose case as follows:

$$p = \ln \frac{I_0}{\text{Poisson}\left(I_0 \exp(-\hat{p})\right) + \text{Normal}\left(0, \sigma_e^2\right)}, \tag{12}$$

where $\hat{p}$ represents the clean projection, and σ_e denotes the variance of electronic noise. I_0 represents the number of photons. In this articler, we fixed the electronic noise variance at $\sigma_e^2 = 10$ and treat $I_0 = 1 \times 10^6$ following [54], [55].

In this articler, four cases with different sparse-view and low-dose data are simulated and the corresponding geometric parameters and dose levels are listed in Table II. We must note that the levels of data heterogeneity in our experiments for segmentation and restoration are different. Since the corrupted data in the restoration task are simulated using various sparse and/or low-dose sampling strategies, the data heterogeneity level for the restoration task is more severe.

In this articler, Peak Signal to Noise Ratio (PSNR) and Structural Similarity (SSIM) are employed as the quantitative metrics. These metrics are widely used to quantitatively evaluate image restoration performance [56]. PSNR assesses the quality of images by measuring the ratio between the maximum possible pixel value and the mean squared error between the restored image and the ground truth. SSIM can effectively measure the structural similarity between the restored image and the ground truth.

Restoration networks are optimized with mean-squared error (MSE) loss. The numbers of communication rounds and local training epochs are set to 500 and 1, respectively. Similar to the segmentation experiments, the results demonstrate the average results with three seeds.

The quantitative results are shown in Table III. It can be noticed that our method achieves the best performance in comparison with other DCML methods, and even works better than CL in some clients. Similar to the segmentation task, FL-based methods are inferior to SL-based methods in most cases. The possible reason lies in that the transferred weights are only with limited information and they cannot make full use of the information from local data. However, as mentioned above, the training time overheads of FL-based methods are much less than those of SL-based methods. Our method combines the merits of both learning paradigms and achieves the best performance. In

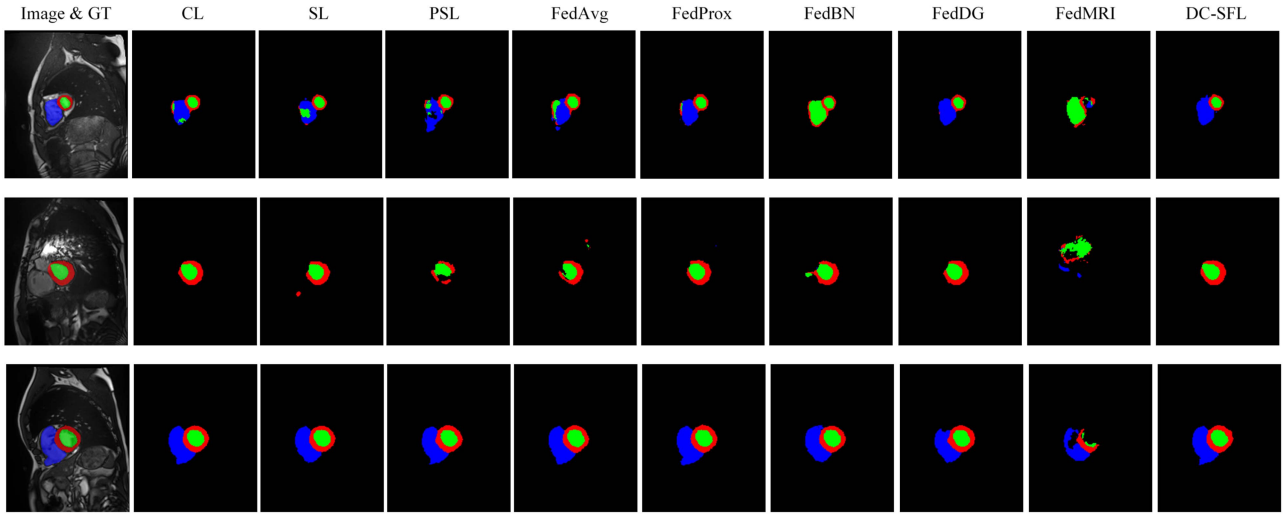

Fig. 4. Qualitative comparison on some typical segmentation results of different methods.

TABLE II
THE GEOMETRY PARAMETERS AND DOSE LEVELS IN DIFFERENT CLIENTS

	Client #1	Client #2	Client #3	Client #4
Number of views	1024	128	512	384
Number of detector bins	512	768	768	600
Pixel length (mm)	0.66	0.78	1.0	1.4
Detector bin length (mm)	0.72	0.58	1.23	1.64
Distance between the source and rotation center (mm)	250	350	500	350
Distance between the detector and rotation center (mm)	250	300	400	300
Intensity of X-rays	1e5	1e6	5e4	1.25e5

TABLE III
THE AVERAGE QUANTITATIVE RESULTS FOR THE IMAGING TASK

	Client #1		Client #2		Client #3		Client #4		Average	
Method	PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑
CL	40.17	0.9689	42.81	0.9677	45.69	0.9873	43.37	0.9715	43.01	0.9739
SL [10]	39.81	0.9622	**41.82**	**0.9606**	45.46	0.9865	43.13	0.9693	42.55	0.9696
PSL [34]	38.98	0.9609	41.80	0.9597	45.47	0.9861	42.23	0.9646	42.12	0.9678
FedAvg [30]	38.12	0.9475	39.84	0.9372	44.10	0.9799	42.31	0.9648	41.09	0.9574
FedProx [31]	37.35	0.9309	38.10	0.9049	43.23	0.9750	40.34	0.9471	39.75	0.9395
FedBN [32]	38.09	0.9477	40.01	0.9321	44.73	0.9843	42.33	0.9628	41.29	0.9567
FedDG [37]	38.95	0.9528	39.34	0.9231	44.06	0.9788	42.10	0.9623	41.11	0.9542
FedMRI [35]	37.48	0.9384	40.91	0.9451	44.57	0.9842	39.95	0.9430	40.73	0.9524
DC-SFL w/o DWCS	39.58	0.9668	40.82	0.9527	45.17	0.9851	43.00	0.9694	42.14	0.9685
DC-SFL w/ DWCS	**42.15**	**0.9749**	40.92	0.9513	**45.48**	**0.9870**	**43.71**	**0.9750**	**42.82**	**0.9721**

The best results are shown in boldface.

the ablation experiment about DWCS, we can easily find that it improves the overall performance. We can notice that the results of other methods in Client #1 have a significant performance gap with those in other clients, which is probably caused by the model drift problem. Benefiting from DWCS, which effectively alleviates the model drift problem by correcting the optimal solution, our method has no noticeable performance gap between different nodes. Fig. 5 shows several typical slices denoised using different methods. It can be observed that results denoised by other methods still contain noise or artifacts to varying degrees, but our method can effectively remove them. In some results of other methods, edges are blurry and some tiny structures are wrongly restored. Compared with them, our method correctly restores those structural details and edges, and they are clearer in our results.

D. Ablation Experiments

In this subsection, we evaluate the impact of the hyperparameter μ on the performance, which is used to control the optimization step of calculating the correction model. The results are shown in Fig. 6, where the left and right vertical axes denote

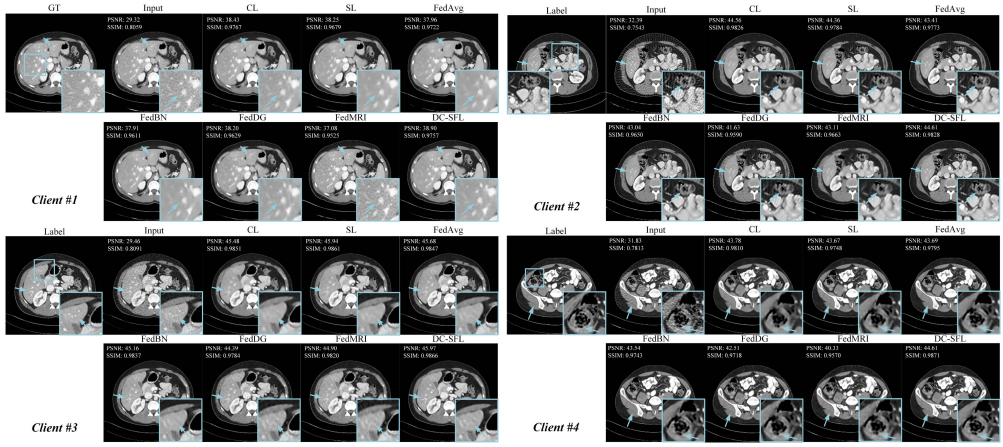

Fig. 5. Visual comparisons with state-of-the-art methods on different geometries and dose levels. The display window is [−160,240]HU.

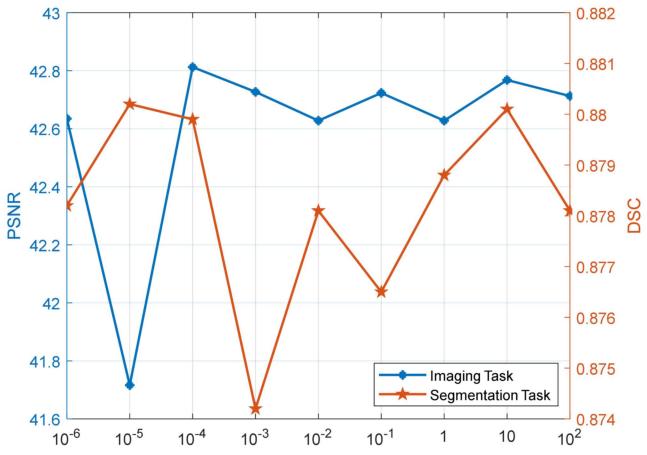

Fig. 6. Ablation study about the selection of hyperparameter μ in (6).

TABLE IV
ANALYSIS OF DIFFERENT NUMBERS OF LOCAL EPOCHS IN RESTORATION TASK. (COMMUNICATION ROUND/LOCAL EPOCH)

	1/500	2/250	4/125	5/100
PSNR	42.82	42.38	41.89	41.90
SSIM	0.9721	0.9699	0.9650	0.9649

the PSNR and DSC values for imaging and segmentation tasks, respectively. We conduct experiments with μ from 1×10^{-6} to 100. It can be observed that our method is not very sensitive to μ, and our performance is better than other DCML methods even under the worst case. As suggested in Fig. 6, 1×10^{-4} is chosen as the default selection in this articler.

The domain shift problem between clients may cause the global model to deviate from the global optimal solution after aggregation. We conduct experiments to sense the impact of the number of local training epochs and communication rounds on the performance. For all the cases, we set the numbers of training iterations equal, and the results of restoration and segmentation tasks are shown in Tables IV and V, respectively. We notice that increasing the number of local training epochs would decrease

TABLE V
ANALYSIS OF DIFFERENT NUMBERS OF LOCAL EPOCHS IN SEGMENTATION TASK. (COMMUNICATION ROUND/LOCAL EPOCH)

	1/300	2/150	4/75	5/60
Dice	88.02	88.21	87.52	88.00
HD95	2.4560	2.3756	2.5918	2.5810

TABLE VI
PRIVACY-RELATED EXPERIMENTS IN THE SEGMENTATION TASK

	baseline	DP	HE	DP and HE
Dice	88.02	87.67	87.93	87.63
HD95	2.4560	3.7000	2.9477	3.9415

TABLE VII
PRIVACY-ENHANCED EXPERIMENT

	baseline	DP	HE	DP and HE
PSNR	42.82	42.61	42.71	42.46
SSIM	0.9721	0.9721	0.9718	0.9691

the imaging performance, but the segmentation performance is not significantly affected. The domain gap in imaging task is greater than that in segmentation task since the scanner and scanning parameters may be different, which lead to a more serious model drift problem in imaging. The model drift is more serious as the number of local training epochs increases, which will lead the global model to deviate from the global optimal solution. As a result, we empirically decrease the number of local training epochs to avoid the above issues.

E. Privacy-Enhanced Protection Experiments

As previously mentioned, the aforementioned experiments operate under the assumption of a benign environment. However, concerns arise in instances where the aggregation server may collude with the computation server. To address this, we have incorporated HE into our DC-SFL approach.

For the privacy-enhanced protection experiments, we have used DC-SFL without encryption as the baseline. In addition to the previously mentioned HE methodology, we have also

TABLE VIII
COMMUNICATION COSTS AND TRAINING TIME OF DIFFERENT LEARNING PARADIGMS

	Communication Cost			Training Time
	Feature	Label / Output	Model	
FL	0	0	$2N\theta$	$T + T_{FL} + \frac{2N\theta}{\mathcal{S}}$
SL	$2\|\|\mathcal{D}\|\|(\mathcal{Q}_{int} + \sum(\mathcal{Q}_s^1 + ... + \mathcal{Q}_s^I))$	$2\|\|\mathcal{D}\|\|\mathcal{O}$	$2N\theta_{cli}$	$T + T_{ser} + T_{cli} + \frac{2\|\|\mathcal{D}\|\|(\mathcal{Q}_{int}+\sum(\mathcal{Q}_s^1+...+\mathcal{Q}_s^I)+\mathcal{O}+\theta_{cli})}{\mathcal{S}}$
Splitfed [9]	$2\|\|\mathcal{D}\|\|(\mathcal{Q}_{int} + \sum(\mathcal{Q}_s^1 + ... + \mathcal{Q}_s^I))$	$2\|\|\mathcal{D}\|\|\mathcal{O}$	$2N\theta_{cli}$	$T + T_{ser} + T_{cli} + \frac{2\|\|\mathcal{D}\|\|(\mathcal{Q}_{int}+\sum(\mathcal{Q}_s^1+...+\mathcal{Q}_s^I+\mathcal{O}+\theta_{cli})}{\mathcal{S}}$
Ours	$2\|\|\mathcal{D}\|\|(\mathcal{Q}_{int}^1 + \mathcal{Q}_{int}^2)$	0	$2N(\theta_h + \theta_t)$	$T + T_{ser} + T_{cli} + T_{DWCS} + \frac{2\|\|\mathcal{D}\|\|(\mathcal{Q}_{int}^1+\mathcal{Q}_{int}^2+\theta_h+\theta_t)}{\mathcal{S}}$

introduced Laplace noise, with a mean of 0 and a variance of 0.00001, into the gradient as a DP mechanism to further enhance privacy. The results of segmentation and restoration are illustrated in Tables VI and VII, respectively.

The results indicate that the performance of our proposed method, facilitated by HE, is on par with the baseline. This suggests that the application of HE does not significantly degrade performance. Furthermore, incorporating DP into the encrypted learning framework serves as a contingency solution, should parties need to bolster privacy protections in light of potential risks such as hackers obtaining private keys. However, as the results demonstrate, the addition of DP may negatively impact performance. Therefore, for applications demanding high performance, utilizing HE may present a potential solution for privacy preservation.

F. Discussion

In this subsection, we discuss the communication costs of different distributed learning paradigms. Because the learning paradigms can be composed of different backbones, we need to give some definitions. $||\mathcal{D}||$ denotes the number of training samples in the whole training dataset. I is the number of shortcut ways in θ, $\mathcal{Q}_{int}$ and $\mathcal{Q}_s$ represent the intermediate feature and shortcut feature, respectively. θ_{cli} denotes the client-side model for SL and [9]. $\mathcal{O}$ denotes the output of θ. $\mathcal{S}$ is the communication rate. T stands for the time cost for one forward and backward propagation on the full model. T_{FL} denotes the aggregation time of full model (T_{ser} and T_{cli} represent the aggregation time of server-side and client-side models, respectively). The time cost for DWCS is represented as T_{DWCS}.

The communication costs and training times of different methods are listed in Table VIII. The costs of all transferred contents need to be doubled for themselves and their corresponding gradients. Since the communication costs of FL-based methods are almost equal, we discuss them together rather than comparing them individually.

As previously mentioned, all parties in FL have access to the full model parameters θ, which necessitates transferring the full model between different parties. However, no parameter transfer occurs during the training phase. On the other hand, while SL and Splitfed [9] can ensure model privacy, they may also raise privacy concerns regarding the sensitive information contained in the output or the label. Besides, they need to transfer shortcut features. In contrast, our method does not require transferring the output or the shortcut features, simultaneously mitigating privacy concerns and communication costs. Although two intermediate features are transferred between parties in our method, the communication cost is much smaller than that of the shortcut features in FL and Splitfed. On the other hand, since our model only contains two small client-side models (head and tail models), the communication cost of our method are similar to those of SL and Splitfed.

We have to admit that our communication cost is slightly higher than FL due to the transmission of intermediate features, but this cost allows us to effectively protect the privacy of model parameters. Despite the increased communication cost, our approach maintains a significant performance improvement, making it a competitive learning paradigm. In comparison to SL and Splitfed, our communication cost is smaller, and we can effectively protect the privacy of the output and label, ensuring a secure learning process. The computational cost of DWCS is low, and DWCS works only once during each training round, costing only a little training time. Considering the significant performance improvement, the additional time is acceptable. Moreover, T_{DWCS} is usually less than the time required for transferring shortcut features. In summary, our method exhibits less training time than SL and Splitfed in practice, and this advantage becomes more significant as the number of clients increases. It is an interesting topic to reduce the communication cost in our future work.

V. CONCLUSION

Current U-shaped medical image networks have achieved impressive success without considering privacy issues. To simultaneously protect the data privacy of input, model parameters, output and label, we propose dynamic corrected split federated learning (DC-SFL) for U-shaped medical image networks. Except for privacy protection, DC-SFL also has other important merits, such as low training time overhead and local computational resource. Meanwhile, we focus on the model drift problem in distributed learning, and propose dynamic weight correction strategy (DWCS) to correct the optimization solution and stabilize the training. Meanwhile, to provide a strong privacy guarantee, we propose to use HE to encrypt the gradients, which can effectively protect the data privacy and avoid potential risks caused by collusion among different parties. Extensive experiments on different tasks demonstrate the effectiveness of our method. On the other side, although DC-SFL can achieve satisfactory performance on different domains, it ignores the test data belonging to an unseen domain. As a result, how to improve the performance for an unseen domain seems an interesting research field in our future work.

V. COMPLIANCE WITH ETHICAL STANDARDS

This research study was conducted retrospectively using real clinical exams acquired at the University Hospital of Dijon and Mayo Clinic. Ethical approval was not required as confirmed by the license attached with the open access data.

REFERENCES

[1] L. Liu, J. Cheng, Q. Quan, F.-X. Wu, Y.-P. Wang, and J. Wang, "A survey on U-shaped networks in medical image segmentations," *Neurocomputing*, vol. 409, pp. 244–258, 2020.

[2] W. Xia, H. Shan, G. Wang, and Y. Zhang, "Physics-/model-based and data-driven methods for low-dose computed tomography: A survey," *IEEE Signal Process. Mag.*, vol. 40, no. 2, pp. 89–100, 2023.

[3] G. Xu et al., "Hercules: Boosting the performance of privacy-preserving federated learning," *IEEE Trans. Dependable Secure Comput.*, vol. 20, no. 5, pp. 4418–4433, Sep./Oct. 2023.

[4] Y. Liu, Z. Ma, Y. Yang, X. Liu, J. Ma, and K. Ren, "RevFRF: Enabling cross-domain random forest training with revocable federated learning," *IEEE Trans. Dependable Secure Comput.*, vol. 19, no. 6, pp. 3671–3685, Nov./Dec. 2022.

[5] G. Xu, H. Li, Y. Zhang, S. Xu, J. Ning, and R. H. Deng, "Privacy-preserving federated deep learning with irregular users," *IEEE Trans. Dependable Secure Comput.*, vol. 19, no. 2, pp. 1364–1381, Mar./Apr. 2022.

[6] C. Thapa, M. A. P. Chamikara, and S. A. Camtepe, "Advancements of federated learning towards privacy preservation: From federated learning to split learning," in *Federated Learning Systems*. Berlin, Germany: Springer, 2021, pp. 79–109.

[7] B. Huang, X. Li, Z. Song, and X. Yang, "FL-NTK: A neural tangent kernel-based framework for federated learning analysis," in *Proc. Int. Conf. Mach. Learn.*, 2021, pp. 4423–4434.

[8] M. Jiang, H. Yang, X. Li, Q. Liu, P.-A. Heng, and Q. Dou, "Dynamic bank learning for semi-supervised federated image diagnosis with class imbalance," in *Proc. Int. Conf. Med. Image Comput. Comput.- Assist. Interv.*, 2022, pp. 196–206.

[9] C. Thapa, P. C. M. Arachchige, S. Camtepe, and L. Sun, "SplitFed: When federated learning meets split learning," in *Proc. AAAI Conf. Artif. Intell.*, 2022, pp. 8485–8493.

[10] P. Vepakomma, O. Gupta, T. Swedish, and R. Raskar, "Split learning for health: Distributed deep learning without sharing raw patient data," 2018, *arXiv:1812.00564*.

[11] A. Singh, P. Vepakomma, O. Gupta, and R. Raskar, "Detailed comparison of communication efficiency of split learning and federated learning," 2019, *arXiv:1909.09145*.

[12] L. Qu et al., "Rethinking architecture design for tackling data heterogeneity in federated learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 10051–10061.

[13] X. Ma, J. Zhu, Z. Lin, S. Chen, and Y. Qin, "A state-of-the-art survey on solving non-IID data in federated learning," *Future Gener. Comput. Syst.*, vol. 135, pp. 244–258, 2022.

[14] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," in *Proc. Int. Conf. Theory Appl. Cryptographic Techn.*, 1999, pp. 223–238.

[15] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. Int. Conf. Med. Image Comput. Comput.- Assist. Interv.*, 2015, pp. 234–241.

[16] B. Lei et al., "Self-co-attention neural network for anatomy segmentation in whole breast ultrasound," *Med. Image Anal.*, vol. 64, 2020, Art. no. 101753.

[17] A. Fakhry, T. Zeng, and S. Ji, "Residual deconvolutional networks for brain electron microscopy image segmentation," *IEEE Trans. Med. Imag.*, vol. 36, no. 2, pp. 447–456, Feb. 2017.

[18] Z. Zhou, M. M. R. Siddiquee, N. Tajbakhsh, and J. Liang, "UNet++: Redesigning skip connections to exploit multiscale features in image segmentation," *IEEE Trans. Med. Imag.*, vol. 39, no. 6, pp. 1856–1867, Jun. 2020.

[19] O. Oktay et al., "Attention U-Net: Learning where to look for the pancreas," in *Proc. Med. Imag. Deep Learn.*, 2018.

[20] F. Isensee, P. F. Jaeger, S. A. Kohl, J. Petersen, and K. H. Maier-Hein, "nnU-Net: A self-configuring method for deep learning-based biomedical image segmentation," *Nature Methods*, vol. 18, no. 2, pp. 203–211, 2021.

[21] N. Siddique, S. Paheding, C. P. Elkin, and V. Devabhaktuni, "U-Net and its variants for medical image segmentation: A review of theory and applications," *IEEE Access*, vol. 9, pp. 82031–82057, 2021.

[22] K. Kulathilake, N. A. Abdullah, A. Q. M. Sabri, and K. W. Lai, "A review on deep learning approaches for low-dose computed tomography restoration," *Complex Intell. Syst.*, vol. 9, pp. 2713–2745, 2021.

[23] H. Chen et al., "Low-dose CT with a residual encoder-decoder convolutional neural network," *IEEE Trans. Med. Imag.*, vol. 36, no. 12, pp. 2524–2535, Dec. 2017.

[24] J. Wang et al., "Domain adaptive denoising network for low-dose CT via noise estimation and transfer learning," *Med. Phys.*, vol. 50, pp. 74–88, 2023.

[25] Z. Wang, X. Cun, J. Bao, W. Zhou, J. Liu, and H. Li, "Uformer: A general U-shaped transformer for image restoration," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 17662–17672.

[26] S. Lee, M. Negishi, H. Urakubo, H. Kasai, and S. Ishii, "Mu-net: Multi-scale U-net for two-photon microscopy image denoising and restoration," *Neural Netw.*, vol. 125, pp. 92–103, 2020.

[27] Z. Huang, J. Zhang, Y. Zhang, and H. Shan, "DU-GAN: Generative adversarial networks with dual-domain u-net-based discriminators for low-dose CT denoising," *IEEE Trans. Instrum. Meas.*, vol. 71, pp. 1–12, 2021.

[28] Y. Han and J. C. Ye, "Framing U-Net via deep convolutional framelets: Application to sparse-view CT," *IEEE Trans. Med. Imag.*, vol. 37, no. 6, pp. 1418–1429, Jun. 2018.

[29] J. Wicaksana, Z. Yan, X. Yang, Y. Liu, L. Fan, and K.-T. Cheng, "Customized federated learning for multi-source decentralized medical image classification," *IEEE J. Biomed. Health Inform.*, vol. 26, no. 11, pp. 5596–5607, Nov. 2022.

[30] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proc. 20th Int. Conf. Artif. Intell. Statist.*, 2017, pp. 1273–1282.

[31] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated optimization in heterogeneous networks," in *Proc. Mach. Learn. Syst.*, 2020, pp. 429–450.

[32] X. Li, M. Jiang, X. Zhang, M. Kamp, and Q. Dou, "FedBN: Federated learning on non-IID features via local batch normalization," in *Proc. Int. Conf. Learn. Representations*, 2020.

[33] C. Shan, H. Jiao, and J. Fu, "Towards representation identical privacy-preserving graph neural network via split learning," 2021, *arXiv:2107.05917*.

[34] J. Jeon and J. Kim, "Privacy-sensitive parallel split learning," in *Proc. IEEE Int. Conf. Inf. Netw.*, 2020, pp. 7–9.

[35] C.-M. Feng, Y. Yan, S. Wang, Y. Xu, L. Shao, and H. Fu, "Specificity-preserving federated learning for MR image reconstruction," *IEEE Trans. Med. Imag.*, vol. 42, no. 7, pp. 2010–2021, Jul. 2023.

[36] Z. Yang, W. Xia, Z. Lu, Y. Chen, X. Li, and Y. Zhang, "Hypernetwork-based personalized federated learning for multi-institutional CT imaging," 2022, *arXiv:2206.03709*.

[37] Q. Liu, C. Chen, J. Qin, Q. Dou, and P.-A. Heng, "FedDG: Federated domain generalization on medical image segmentation via episodic learning in continuous frequency space," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 1013–1023.

[38] H. R. Roth et al., "Split-U-Net: Preventing data leakage in split learning for collaborative multi-modal brain tumor segmentation," in *Proc. Int. Workshop Distrib. Collaborative Federated Learn., Workshop Affordable Healthcare AI Resour. Diverse Glob. Health*, 2022, pp. 47–57.

[39] V. Turina, Z. Zhang, F. Esposito, and I. Matta, "Federated or split? A performance and privacy analysis of hybrid split and federated learning architectures," in *Proc. IEEE Int. Conf. Cloud Comput.*, 2021, pp. 250–260.

[40] M. Zhang, L. Qu, P. Singh, J. Kalpathy-Cramer, and D. L. Rubin, "SplitAVG: A heterogeneity-aware federated deep learning method for medical imaging," *IEEE J. Biomed. Health Inform.*, vol. 26, no. 9, pp. 4635–4644, Sep. 2022.

[41] J. Geiping, H. Bauermeister, H. Dröge, and M. Moeller, "Inverting gradients-how easy is it to break privacy in federated learning?," in *Proc. Annu. Conf. Neural Inf. Process. Syst.*, 2020, pp. 16937–16947.

[42] M. Akter, N. Moustafa, T. Lynar, and I. Razzak, "Edge intelligence: Federated learning-based privacy protection framework for smart healthcare systems," *IEEE J. Biomed. Health Inform.*, vol. 26, no. 12, pp. 5805–5816, Dec. 2022.

[43] L. Zhang, J. Xu, P. Vijayakumar, P. K. Sharma, and U. Ghosh, "Homomorphic encryption-based privacy-preserving federated learning in IoT-enabled healthcare system," *IEEE Trans. Netw. Sci. Eng.*, vol. 10, no. 5, pp. 2864–2880, Sep./Oct. 2023.

[44] M. Abadi et al., "Deep learning with differential privacy," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, 2016, pp. 308–318.

[45] H. Fang and Q. Qian, "Privacy preserving machine learning with homomorphic encryption and federated learning," *Future Internet*, vol. 13, no. 4, 2021, Art. no. 94.

[46] C. Zhang, S. Li, J. Xia, W. Wang, F. Yan, and Y. Liu, "Batchcrypt: Efficient homomorphic encryption for cross-silo federated learning," in *Proc. USENIX Annu. Tech. Conf.*, 2020, pp. 493–506.

[47] X. Tang, M. Shen, Q. Li, L. Zhu, T. Xue, and Q. Qu, "PILE: Robust privacy-preserving federated learning via verifiable perturbations," *IEEE Trans. Dependable Secure Comput.*, early access, Jan. 23, 2023, doi: 10.1109/TDSC.2023.3239007.

[48] Y. Liang, Y. Li, and B.-S. Shin, "Auditable federated learning with byzantine robustness," *IEEE Trans. Comput. Soc. Syst.*, early access, Apr. 24, 2023, doi: 10.1109/TCSS.2023.3266019.

[49] A. Tarvainen and H. Valpola, "Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results," in *Proc. Annu. Conf. Neural Inf. Process. Syst.*, 2017, pp. 1195–1204.

[50] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Representations*, 2015.

[51] O. Bernard et al., "Deep learning techniques for automatic MRI cardiac multi-structures segmentation and diagnosis: Is the problem solved?," *IEEE Trans. Med. Imag.*, vol. 37, no. 11, pp. 2514–2525, Nov. 2018.

[52] K.-N. Wang et al., "AWSnet: An auto-weighted supervision attention network for myocardial scar and edema segmentation in multi-sequence cardiac magnetic resonance images," *Med. Image Anal.*, vol. 77, 2022, Art. no. 102362.

[53] C. McCollough, "TU-FG-207A-04: Overview low dose CT grand challenge," *Med. Phys.*, vol. 43, no. 6, pp. 3759–3760, 2016.

[54] W. Xia et al., "CT reconstruction with PDF: Parameter-dependent framework for data from multiple geometries and dose levels," *IEEE Trans. Med. Imag.*, vol. 40, no. 11, pp. 3065–3076, Nov. 2021.

[55] S. Niu et al., "Sparse-view x-ray CT reconstruction via total generalized variation regularization," *Phys. Med. Biol.*, vol. 59, no. 12, 2014, Art. no. 2997.

[56] A. Hore and D. Ziou, "Image quality metrics: PSNR vs. SSIM," in *Proc. IEEE Int. Conf. Pattern Recognit.*, 2010, pp. 2366–2369.