Medical Image Analysis 81 (2022) 102564

Contents lists available at ScienceDirect

Medical Image Analysis

journal homepage: www.elsevier.com/locate/media

Distributed contrastive learning for medical image segmentation☆

Yawen Wu[a,*], Dewen Zeng[b], Zhepeng Wang[c], Yiyu Shi[b], Jingtong Hu[a]

[a] *University of Pittsburgh, Pittsburgh PA 15260, USA*
[b] *University of Notre Dame, Notre Dame IN 46556, USA*
[c] *George Mason University, Fairfax VA 22030, USA*

ARTICLE INFO

MSC:
68T10
68T45
92C55
68W15

Keywords:
Federated learning
Contrastive learning
Self-supervised learning
Image segmentation

ABSTRACT

Supervised deep learning needs a large amount of labeled data to achieve high performance. However, in medical imaging analysis, each site may only have a limited amount of data and labels, which makes learning ineffective. Federated learning (FL) can learn a shared model from decentralized data. But traditional FL requires fully-labeled data for training, which is very expensive to obtain. Self-supervised contrastive learning (CL) can learn from unlabeled data for pre-training, followed by fine-tuning with limited annotations. However, when adopting CL in FL, the limited data diversity on each site makes federated contrastive learning (FCL) ineffective. In this work, we propose two federated self-supervised learning frameworks for volumetric medical image segmentation with limited annotations. The first one features high accuracy and fits high-performance servers with high-speed connections. The second one features lower communication costs, suitable for mobile devices. In the first framework, features are exchanged during FCL to provide diverse contrastive data to each site for effective local CL while keeping raw data private. Global structural matching aligns local and remote features for a unified feature space among different sites. In the second framework, to reduce the communication cost for feature exchanging, we propose an optimized method FCLOpt that does not rely on negative samples. To reduce the communications of model download, we propose the predictive target network update (PTNU) that predicts the parameters of the target network. Based on PTNU, we propose the distance prediction (DP) to remove most of the uploads of the target network. Experiments on a cardiac MRI dataset show the proposed two frameworks substantially improve the segmentation and generalization performance compared with state-of-the-art techniques.

1. Introduction

Deep learning (DL) provides state-of-the-art medical image segmentation performance by learning from large-scale labeled datasets (Ronneberger et al., 2015; Milletari et al., 2016; Xu et al., 2019; Dong et al., 2017), without which the performance of DL will significantly degrade (Kairouz et al., 2019). However, medical data exist in isolated medical centers and hospitals (Yang et al., 2019), and combining a large dataset consisting of very sensitive and private medical data in a single location is impractical and even illegal. It requires multiple medical institutions to share medical patient data such as medical images, which is constrained by the Health Insurance Portability and Accountability Act (HIPAA) (Kairouz et al., 2019) and EU General Data Protection Regulation (GDPR) (Truong et al., 2020). Federated learning (FL) is an effective machine learning approach in which distributed clients (i.e. individual medical institutions) collaboratively learn a shared model while keeping private raw data local (Rieke et al., 2020; Sheller et al., 2018, 2020; Dou et al., 2021). By applying FL to medical image segmentation, an accurate model can be collaboratively learned and data is kept local for privacy.

Conventional FL approaches usually use supervised learning on each client and require that all data are labeled. However, annotating all the medical images is usually unrealistic due to the high labeling cost and requirement of expertise. The deficiency of labels makes supervised FL impractical. Self-supervised learning can address this challenge by pre-training a neural network encoder with unlabeled data, followed by fine-tuning for a downstream task with limited labels. Contrastive learning (CL), a variant of the self-supervised learning approach, can effectively learn high-quality image representations. By integrating CL to FL as federated contrastive learning (FCL), clients can learn models by first collaboratively learning a shared image-level representation. Then the learned model will be fine-tuned by using limited annotations. Compared with local CL, FCL can learn a better encoder as the initialization for fine-tuning, and provide higher segmentation performance.

☆ This work was supported in part by NSF, United States CNS-2122320, CNS-212220, CNS-1822099, CNS-2007302.

* Corresponding author.
E-mail address: yawen.wu@pitt.edu (Y. Wu).

https://doi.org/10.1016/j.media.2022.102564
Received 22 February 2022; Received in revised form 6 July 2022; Accepted 1 August 2022
Available online 12 August 2022

In this way, a high-quality model can be learned by using limited annotations while data privacy is preserved.

Based on CL, we propose two frameworks to enable federated self-supervised learning for medical image segmentation. The first framework exchanges encoded features for a higher accuracy and has higher communication cost than the second one. It is suitable for distributed medical institutions with high-performance servers and high-speed connections. The second framework does not exchange encoded features and has less model synchronization. It has lower communication costs and is suitable for mobile devices with high communication costs.

Integrating FL with CL to achieve good performance is nontrivial. Simply applying CL to each client and then aggregating the models is not the optimal solution for the following two reasons: First, each client only has a small amount of unlabeled data with limited diversity. Since existing contrastive learning frameworks (Chen et al., 2020a; He et al., 2020) rely on datasets with diverse data to learn distinctive representations, directly applying CL on each client will result in an inaccurate learned model due to the lack of data diversity. Second, if each client only focuses on CL on its local data while not considering others' data, each client will have its own feature space based on its raw data and these feature spaces are inconsistent among different clients. When aggregating local models, the inconsistent feature space among local models will degrade the performance of the aggregated model.

To address these challenges, in our **first** FCL framework, we develop a two-stage FCL method to enable effective FCL for volumetric medical image segmentation with limited annotations. The first stage is feature exchange (FE), in which each client exchanges the features (i.e. low-dimensional vectors) of its local data with other clients. It provides more diverse data to compare with for better local contrastive learning while avoiding raw data sharing. In the learning process, the improved data diversity in feature space provides more accurate and complete contrastive information in the local learning process on each client and improves the learned representations.

The second stage is global structural matching (GSM), in which we leverage structural similarity of 3D medical images to align similar features among clients for better FCL. The intuition is that the same anatomical region for different subjects has similar content in volumetric medical images such as MRI. By leveraging the structural similarity across volumetric medical images, GSM aligns the features of local images to the shared features of the same anatomical region from other clients. In this way, the learned representations of local models are more unified among clients and they further improve the global model after model aggregation.

In the first framework, feature exchange requires additional communication. To reduce the communication cost, we further propose the **second** framework FCLOpt. It is an optimized method that does not rely on negative samples and reduces the communication costs of feature sharing. Based on FCLOpt, to further reduce the communications of model download, we propose the predictive target network update (PTNU) that predicts the target network by fast forward. Based on PTNU, we propose the distance prediction (DP) to remove the upload of the target network.

Experimental results show that the proposed FCL methods substantially improve the segmentation performance over state-of-the-art techniques, and the FCLOpt including the proposed PTNU and DP methods effectively reduces the communication cost while preserving the segmentation performance of FCL.

The conference version of this paper (Wu et al., 2021b) appeared at the MICCAI 2021 conference proceedings. The extensions to the original conference paper are described in Section 7.

The rest of this paper is organized as follows. The background and related work are described in Section 2. The FCL method with feature sharing is introduced in Section 3. The communication-optimized method FCLOpt is described in Section 4. The experimental settings and results are reported in Section 5, and this paper is concluded in Section 6.

2. Background and related work

Federated Learning. Federated learning (FL) learns a shared model by aggregating locally updated models on clients while keeping raw data accessible on local clients for privacy (McMahan et al., 2017; Li et al., 2020; Zhao et al., 2018; Li et al., 2018). In FL, the training data are distributed among clients. FL is performed round-by-round by repeating the local model learning and model aggregation process until convergence.

The main drawback of these works is that fully labeled data are needed to perform FL, which results in high labeling costs. To solve this problem, an FL approach using limited annotations while achieving good performance is needed.

Contrastive Learning. Contrastive learning (CL) is a self-supervised approach to learn useful visual representations by using unlabeled data (Hadsell et al., 2006; Misra and Maaten, 2020; Tian et al., 2019). The learned model provides good initialization for fine-tuning on the downstream task with few labels (He et al., 2020; Chen et al., 2020a,b; Zeng et al., 2021; Chaitanya et al., 2020; Wu et al., 2021a). CL performs a proxy task of instance discrimination (Wu et al., 2018), which maximizes the similarity of representations from similar pairs and minimizes the similarity of representations from dissimilar pairs (Wang and Isola, 2020).

The main drawback of existing CL approaches is that they are designed for centralized learning on large-scale datasets with sufficient data diversity. However, when applying CL to FL on each client, the limited data diversity will greatly degrade the performance of the learned model. Therefore, an approach to increase the local data diversity while avoiding raw data sharing for privacy is needed. Besides, while Chaitanya et al. (2020) leverages structural information in medical images for improving centralized CL, it requires accessing raw images of similar pairs for learning. Since sharing raw medical images is prohibitive due to privacy, Chaitanya et al. (2020) cannot be applied to FL. Therefore, an approach to effectively leverage similar images across clients without sharing raw images is needed.

Federated Self-supervised Pre-training. Some concurrent works employ federated pre-training on unlabeled data. Van Berlo et al. (2020) employs auto-encoders in FL for pre-training on time-series data, but the more effective contrastive learning for visual tasks is not explored in FL. Bercea et al. (2021) uses auto-encoders for federated self-supervised medical image segmentation. Different from these works, we employ self-supervised contrastive learning, which has demonstrated superior performance to auto-encoders in centralized training (Wu et al., 2018). FedCA (Zhang et al., 2020) combines contrastive learning with FL. However, it relies on a shared dataset available on each client, which is impractical for medical images due to privacy concerns. Different from this, we do not share raw data among clients to preserve privacy. Dong and Voiculescu (2021) uses CL method MoCo to perform self-supervised learning on each client. Metadata is shared among clients to improve local CL. Zhuang et al. (2021, 2022) updates local models of clients adaptively using the exponential moving average (EMA) of the global model. The proposed work differs from these works in the following ways. First, in our FCL method, we leverage the structural similarity of volumetric images across clients to improve the quality of representation learning. Second, these works only communicate one network between the server and clients even if they use two networks for local learning. Different from this, in our FCLOpt method, we predict the parameters of the second network to achieve higher performance while keeping the communication cost similar to communicating only one network.

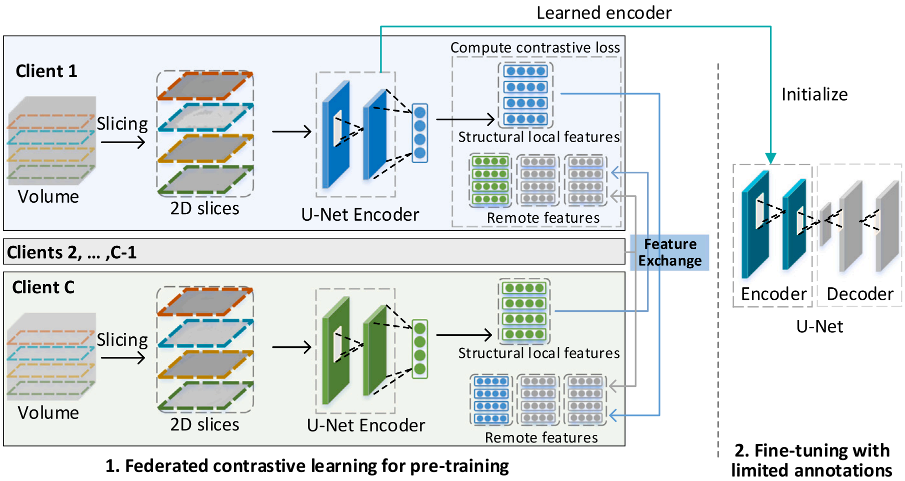


Fig. 1. Federated contrastive learning with structural feature exchange for learning the encoder with unlabeled data. Then the learned encoder initializes the encoder in U-Net for fine-tuning with limited annotations.

3. Federated Contrastive Learning (FCL)

3.1. Overview of Federated Contrastive Learning

The overview of the proposed FCL process is shown in Fig. 1. Distributed clients first collaboratively learn a shared encoder by FCL with unlabeled data. Then the learned encoder initializes the encoder in U-Net (Ronneberger et al., 2015) for fine-tuning with limited annotations, either independently on each client by supervised learning or collaboratively by supervised federated learning. Since the supervised fine-tuning can be trivially achieved by using available annotations, in the rest of the paper, we focus on FCL to learn a good encoder as the initialization for fine-tuning.

As shown in Fig. 1, in the FCL stage, given a volumetric 3D image on one client, multiple 2D slices are sampled from the volume while keeping structural order along the slicing axis. Then the ordered 2D images are fed into the 2D encoder to generate feature vectors, one vector for each 2D image.

To improve the data diversity in local contrastive learning, one natural way is to share raw images (Zhao et al., 2018). However, sharing raw medical images is prohibitive due to privacy concerns. To solve this problem, the proposed FCL framework exchanges the feature vectors instead of raw images among clients, which can improve the data diversity while preserving privacy. As shown in Fig. 1, client 1 generates structural local features denoted as blue vectors and shares them with other clients. Meanwhile, client 1 collects structural features from other clients, such as remote features shown in green and gray vectors. After that, the contrastive loss is computed based on both local and remote features.

3.2. Contrastive learning with feature exchange

With feature exchange, each client has both remote and local features and is ready to perform local CL in each round of FCL. The exchanged features from other clients provide more diverse features to compare with and improve the learned representations. As shown in Fig. 2, we use MoCo (He et al., 2020) architecture for local CL since it has a memory bank for negatives, which can leverage local and remote features. There are two encoders, including the main encoder and the momentum encoder. The main encoder will be learned and used as the initialization for fine-tuning, while the momentum encoder is the slowly-evolving version of the main encoder and generates features to contrast with and for sharing. Now the most important steps are to construct negatives and positives from both local and remote features.

Negatives from local and remote features. Local features are generated by the momentum encoder from local images and used as local negatives. Each client has a memory bank of local features and a memory bank of remote features. Let $Q_{l,c}$ be the size-K memory bank of local features on client c, which are used as local negatives. $Q_{l,c}$ is progressively updated by replacing the oldest features with the latest ones. In each round of FCL, the remote negatives from other clients will be shared with client c to form its aggregated memory bank including local and remote negatives as:

$$Q = Q_{l,c} \cup \{Q_{l,i} \mid 1 \leq i \leq |C|, i \neq c\}. \tag{1}$$

where C is the set of all clients and $Q_{l,i}$ is the local memory bank on client i.

Compared with using only local memory bank $Q_{l,c}$, the aggregated memory bank Q provides more data diversity to improve CL. However, Q is $|C|$ times the size of the local memory bank $Q_{l,c}$. More negatives make CL more challenging since for one local feature q, more negatives need to be simultaneously pushed away from it than when using $Q_{l,c}$, which can result in ineffective learning. To solve this problem, instead of using all negatives in Q, for each q we sample a size-K (i.e. the same size as $Q_{l,c}$) subset of Q as negatives, which is defined as:

$$Q' = \{Q_i \mid i \sim \mathcal{U}(|Q|, K)\}. \tag{2}$$

where $i \sim \mathcal{U}(|Q|, K)$ means i is a set of indices sampled uniformly from $[|Q|]$.

Local positives. We leverage the structural similarity in the volumetric medical images to define the local positives, in which the same anatomical region from different subjects has similar content (Chaitanya et al., 2020). Each volume is grouped into S partitions, and one image sampled from partition s of volume i is denoted as x_s^i. *Local positives* are features of images from the same partition in different volumes. Given an image x_s^i, its feature q_s^i and corresponding positives $P(q_s^i) = \{k_s^{i+}, k_s^{j+}\}$ are formed as follows. Two transformations (e.g. cropping) are applied to x_s^i to get $\tilde{x}_s^i$ and $\hat{x}_s^i$, which are then fed into the main encoder and momentum encoder to generate two representation vectors q_s^i and k_s^{i+}, respectively. Then another image x_s^j is sampled from partition s of volume j, and its features q_s^j and k_s^{j+} are generated accordingly. In this way, the local positives for both q_s^i and q_s^j are formed as $P(q_s^i) = P(q_s^j) = \{k_s^{i+}, k_s^{j+}\}$.

Loss function for local positives. By using the sampled memory bank Q' consisting of both *local* negatives and *remote* negatives, one local feature q is compared with its local positives $P(q)$ and each negative in Q'. The contrastive loss is defined as:

$$\begin{aligned}\mathcal{L}_{local} &= \ell_{q,P(q),Q'} \\ &= -\frac{1}{|P(q)|} \sum_{k^+ \in P(q)} \log \frac{\exp(q \cdot k^+/\tau)}{\exp(q \cdot k^+/\tau) + \sum_{n \in Q'} \exp(q \cdot n/\tau)}.\end{aligned} \tag{3}$$

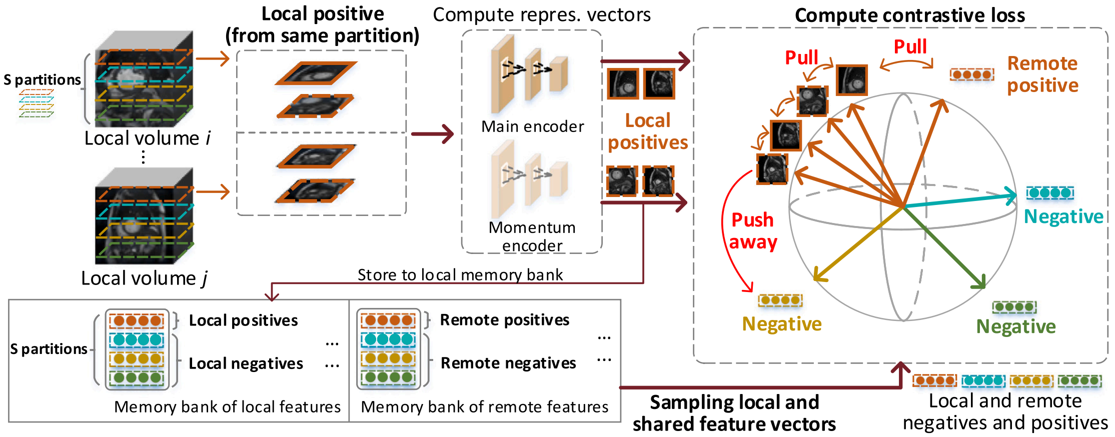


Fig. 2. Contrastive learning on one client with exchanged features. The exchanged features consist of remote negatives and remote positives, in which remote negatives improve the local data diversity and remote positives are used for global structural matching to learn a unified feature space among clients.

where τ is the temperature and the operator $\cdot$ is the dot product between two vectors. By minimizing the loss, the distance between q and each local positive is minimized, and the distance between q and each negative in Q' is maximized.

3.3. Global structural matching

Remote positives. We use the remote positives from the shared features to further improve the learned representations. Inspired by Chaitanya et al. (2020) that aligns the features of images in the same partition for centralized learning, on each client, we align the features of one image to the features of images in the same partition from other clients. In this way, the features of images in the same partition across clients will be aligned in the feature space and more unified representations can be learned among clients. To achieve this, for one local feature q, in addition to its local positives $P(q)$, we define remote positives $\Lambda(q)$ as features in the sampled memory bank Q' which are in the same partition as q.

$$\Lambda(q) = \{p \mid p \in Q', partition(p) = partition(q)\}. \tag{4}$$

$partition(\cdot)$ is the partition number of one feature and Q' is defined in Eq. (2).

Final loss function. By replacing local positives $P(q)$ in Eq. (3) with remote positives $\Lambda(q)$ as $\mathcal{L}_{remote}$, the final loss function for one feature q is defined as:

$$\mathcal{L}_q = \mathcal{L}_{remote} + \mathcal{L}_{local} = \ell_{q,\Lambda(q),Q'} + \ell_{q,P(q),Q'}. \tag{5}$$

With $\mathcal{L}_q$, the loss for one batch of images is defined as $\mathcal{L}_B = \frac{1}{|B|}\sum_{q\in B}\mathcal{L}_q$, where B is the set of features generated by the encoder from the batch of images.

3.4. Privacy of feature exchange

To protect the shared features against attacks, image encryption methods or representation perturbation methods can be employed. For the image encryption method, Huang et al. (2020) encrypt images before learning and keep the utility of encrypted images for learning. Images are encrypted before being fed into the encoder for generating representations. As a result, only features of encrypted images are shared, which effectively mitigates the potential vulnerability by feature exchange and maintains the utility of exchanged features for local learning. For the representation perturbation method, Sun et al. (2021) learns to perturb data representation such that the quality of the potentially leaked information is severely degraded, while FL performance is maintained.

4. FCLOpt for reducing communications

In the previous Sections, we have introduced the FCL method, aiming at improving the quality of learned representations and the segmentation performance of the downstream task. However, sharing features require additional communication. To solve this problem, we eliminate the need for feature sharing by proposing an optimized method *FCLOpt* that does not rely on shared features as negative samples. To further reduce the communications of model downloading, we propose the predictive target network update (PTNU) that predicts the target network by fast forward. Based on PTNU, we propose the distance prediction (DP) to remove most of the uploads of the target network and only use sporadic upload for calibration.

4.1. Revisiting self-supervised learning method BYOL

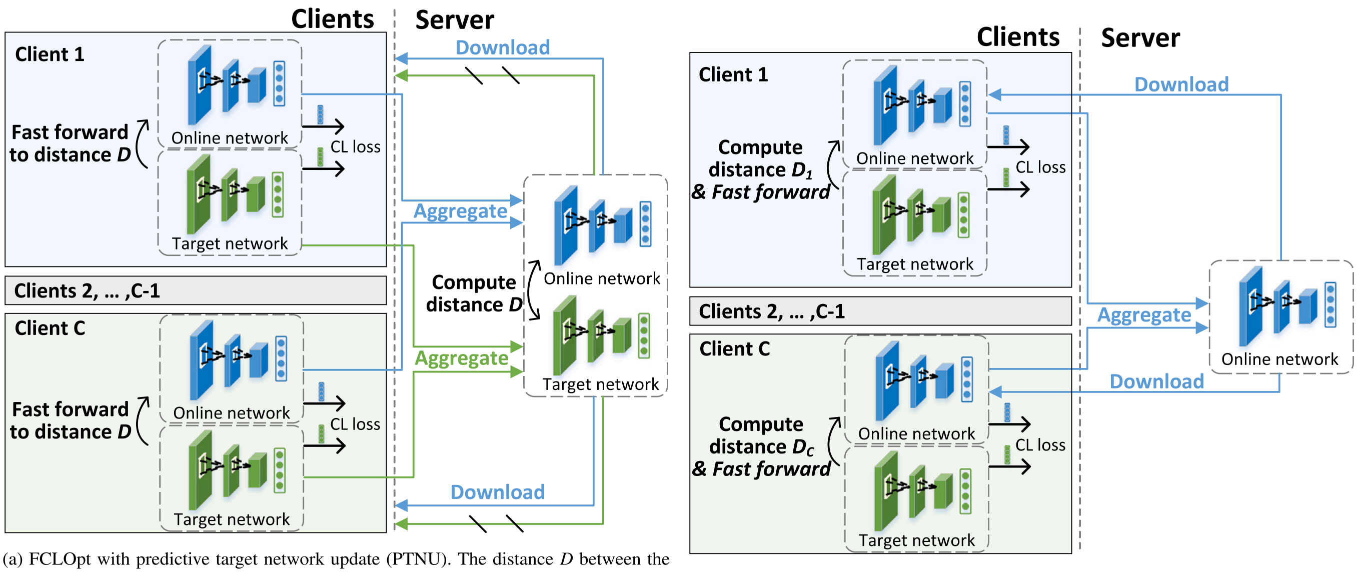
BYOL (Grill et al., 2020) is a self-supervised learning method without negative pairs. Conventional CL performs learning by attracting the positive sample pairs and repulsing the negative sample pairs. Different from this, BYOL directly predicts the output of one sample in a positive pair from the other one. Since no positive pairs are used in BYOL, it has the potential to eliminate the need for feature exchange and reduces communication cost.

BYOL has a Siamese network architecture, consisting of the online network and the target network. The online network consists of an encoder and a predictor. The target network has the same architecture as the encoder in the online network, but different parameters. The target network provides the learning targets to train the online network, and it is updated by an exponential moving average (EMA) of the parameters of the online network. Details of using BYOL for local learning on clients will be introduced in Section 4.3.

4.2. FCLOpt overview

The overview of the optimized method FCLOpt is shown in Fig. 3. Compared with the FCL method introduced in Section 3 that solely seeks for high model accuracy, FCLOpt reduces the communication cost, simplifies the system complexity, while keeping a comparable or even better segmentation performance.

In FCLOpt, there is a server that coordinates multiple clients to upload locally updated online networks and target networks, which are then aggregated on the server as the global online network and the global target network. The server also downloads the global online network and target network to clients. In each training round, synchronizing both the online network and target network needs extra communication, while synchronizing only one network will greatly



(a) FCLOpt with predictive target network update (PTNU). The distance D between the global online network and global target network is computed on the server. Each client uses D to predict the global target network instead of downloading it.

(b) FCLOpt-PTNU with distance prediction (DP) between the global online network and the global target network to remove the upload of the target network.

Fig. 3. FCLOpt for reducing communications. Based on the proposed FCL method, we develop FCLOpt to eliminate the need for feature sharing while keeping a high accuracy of the learned model. In FCLOpt, we reduce the download and upload of the target encoder. (a) Predictive target network update (PTNU) to eliminate the download of the global target network. (b) Distance prediction (DP) to remove most of the uploads of the target network.

degrade the learned model. To solve this problem, predictive target network update (PTNU) and distance prediction (DP) are proposed to reduce the synchronization of the target network. In this way, the communication cost is comparable to that of synchronizing only one network while the accuracy of the learned network is similar to that of synchronizing two networks. FedOpt is summarized in Algorithm 1.

4.3. FCLOpt: Local training and aggregation

In FCLOpt, each client has two networks, the online network and the target network, following the model architectures from BYOL (Grill et al., 2020). The online network consists of an online encoder f_θ and a predictor q_θ. The target network is a target encoder f_ξ, which has the same model architecture as the online encoder f_θ but different parameters.

During local training, two augmentations are applied to a 2D image x and generate two transformed images t and t'. t and t' are then fed into the online network and the target network, respectively. The contrastive loss to update parameters θ of the online network is defined as:

$$\mathcal{L}_{\theta,\xi} = \|z - z'\|_2^2 = 2 - 2 \cdot \frac{\langle z, z' \rangle}{\|z\|_2 \cdot \|z'\|_2}. \tag{6}$$

where $z = q_\theta(f_\theta(t))$ is the output of the online network, and $z' = f_\xi(t')$ is the output of the target network. No negatives samples are used in this contrastive loss and therefore no shared features are needed. The online network with parameters θ is updated by gradient descent to minimize $\mathcal{L}_{\theta,\xi}$, and the target network with parameters ξ is updated by exponential moving average (EMA) of the parameters θ of the online encoder f_θ:

$$\xi = m\xi + (1 - m)\theta. \tag{7}$$

where $m \in (0, 1]$ is the momentum parameter controlling the update speed of the target network. For conciseness, in the rest of this paper, we use f_θ to denote the whole online network consisting of the online encoder and the online predictor.

On each client, the training is performed for E epochs before uploading the updated local networks to the server for aggregation and downloading the aggregated models from the server to initiate the networks for training in the next round.

The aggregation is performed on the server as follows. In round r, denoting the online network and target network after local training on client c as $f^r_{\theta_c}$ and $f^r_{\xi_c}$, the global online network F^{r+1}_θ and the global target network F^{r+1}_ξ are aggregated as:

$$F^{r+1}_\theta = \sum_{c \in C} \frac{n_c}{n} f^r_{\theta_c}. \tag{8}$$

$$F^{r+1}_\xi = \sum_{c \in C} \frac{n_c}{n} f^r_{\xi_c}. \tag{9}$$

where n_c is the number of samples on client c, and n is the total number of samples on all clients. After network aggregation, F^{r+1}_θ and F^{r+1}_ξ are downloaded to clients to start the training of round $r + 1$.

4.4. Predictive target network update

Since there are two networks to synchronize between the server and clients, to reduce the communication cost of the target network, we propose a predictive target network update (PTNU) method to eliminate the need for target network download.

We first introduce how to eliminate the download of the global target network by predicting the parameters ξ of global target network F^{r+1}_ξ on clients.

At the beginning of round $r+1$, the server computes the average ℓ_1 distance between the parameters of the aggregated F^{r+1}_θ and F^{r+1}_ξ:

$$d(F^{r+1}_\theta, F^{r+1}_\xi) = \|F^{r+1}_\theta - F^{r+1}_\xi\|_1 = \frac{1}{N} \sum_{a\in\theta,\ b\in\xi} |a - b|. \tag{10}$$

where $N = \|\xi\|_0$ is the number of parameters in the global target network.

On client c, ideally, both F^{r+1}_θ and F^{r+1}_ξ are downloaded from the server to initiate its local $f^{r+1}_{\theta_c}$ and $f^{r+1}_{\xi_c}$. To reduce the communication cost, we only download F^{r+1}_θ and the scalar value $d(F^{r+1}_\theta, F^{r+1}_\xi)$, and predict F^{r+1}_ξ on the client instead of downloading it. Following Algorithm 2, given the latest global online network F^{r+1}_θ, the local target network $f^r_{\xi_c}$, and the distance $d(F^{r+1}_\theta, F^{r+1}_\xi)$, we predict the parameters of F^{r+1}_ξ by an iterative update on client c:

$$\xi_c = m_d \xi_c + (1 - m_d)\theta^{r+1} \tag{11}$$

Algorithm 1: Communication-optimized Federated Contrastive Learning (FCLOpt) with Predictive Target Network Update (PTNU) and Distance Prediction (DP)

```
Input: number of training rounds $R$, number of local epochs $E$,
       learning rate $\eta$, local batch size $B$, distance calibrator $\alpha$
Output: $F_\theta$
Server ():
    Initialize the global online network $F_\theta^0$ and global target
     network $F_\xi^0$;
    for each round r from 1 to R do
        $C_r \leftarrow$ (random set of $K$ clients);
        // DP: Predict distance between global online and
         target networks by Eq. (13) and Eq. (14);
        for client $c \in C_r$ in parallel do
            $dp_c$ = ClientDistance($F_\theta^r$, $r$);
        end
        $\tilde{d} = \alpha \cdot \sum_{c \in C_r} \frac{1}{|C_r|} dp_c$;
        for client $c \in C_r$ in parallel do
            //DP eliminates the upload of $f_\xi^c$;
            $f_\theta^c, f_\xi^c \leftarrow$ ClientTrain($F_\theta^r$, $\tilde{d}$, $r$);
        end
        // Model aggregation of online networks;
        $F_\theta^{r+1} \leftarrow \sum_{c \in C_r} \frac{n_c}{n} f_{\theta_c}^r$;
        // Model aggregation of target networks. DP eliminates
         the following two lines;
        $F_\xi^{r+1} \leftarrow \sum_{c \in C_r} \frac{n_c}{n} f_{\xi_c}^r$;
        $\tilde{d} = d(F_\theta^{r+1}, F_\xi^{r+1})$;
    end
    return $f_\phi^R$;
ClientDistance ($F_\theta^r$, $r$):
    $dist \leftarrow \|F_\theta^r - f_{\xi_c}^{r-1}\|_1$;
    return $dist$;
ClientTrain ($F_\theta^r$, $\tilde{d}$, $r$):
    $f_\theta \leftarrow F_\theta^r$ // Model download;
    // PTNU: Predict global target network by Algo. 2;
    $F_\xi^r \leftarrow$ PTNU($F_\theta^r$, $f_\xi^{r-1}$, $\tilde{d}$);
    $\mathcal{B} \leftarrow$ (form batches of size $B$);
    // Local training with Eq. (6) and Eq. (7);
    for each local epoch i from 1 to E do
        for batch $b \in \mathcal{B}$ do
            $\theta \leftarrow \theta - \eta \nabla_\theta \mathcal{L}_{\theta,\xi}(\theta; b)$;
            $\xi \leftarrow \tau\xi + (1 - \tau)\theta$;
        end
    end
    return $f_\theta$, $f_\xi$ // Model upload;
```

Algorithm 2: Predictive target network update (PTNU).

```
Input: Global online network $F_\theta$, local target network $f_{\xi_c}$, and
       their distance $d(F_\theta, F_\xi)$
Output: Predicted target model parameters $f_\xi$
PTNU ($F_\theta$, $f_\xi$, $d$):
    // Compute the initial model distance by Eq. (10);
    $dist \leftarrow d(F_\theta, f_\xi)$;
    while $dist > d$ do
        // Update model by exponential moving average (EMA)
         using Eq. (11);
        $\xi = m_d \xi + (1 - m_d)\theta$;
        // Compute updated model distance by Eq. (10);
        $dist \leftarrow d(F_\theta, f_\xi)$;
    end
    return $f_\xi$;
```

which is performed iteratively until $d(F_\theta^{r+1}, f_{\xi_c}^r) \leq d(F_\theta^{r+1}, F_\xi^{r+1})$ is satisfied. After this update, the parameters of $f_{\xi_c}^r$ are used to initialize F_ξ^{r+1} on client c to start the training of round $r+1$.

The intuition of PTNU is that we can approximate F_ξ^{r+1} by gradually approaching F_θ^{r+1} from a certain direction until a certain frontier is reached. Ideally, F_ξ^{r+1} is the weighted average of all $f_{\xi_c}^r, c \in C$ as defined in Eq. (8), and F_θ^{r+1} is the weighted average of all $f_{\theta_c}^r, c \in C$ as defined in Eq. (9). For each $c \in C$, since $f_{\xi_c}^r$ is updated by EMA in Eq. (7) to approach $f_{\theta_c}^r$ from the direction of $f_{\xi_c}^r$, F_ξ^{r+1} can be treated as approaching F_θ^{r+1} from the direction of each $f_{\xi_c}^r$ simultaneously.

On client c, since we only know $f_{\xi_c}^r$ and do not have access to $f_{\xi_i}^r, i \neq c$, we gradually approach F_θ^{r+1} from the direction of $f_{\xi_c}^r$, instead of from the direction of each $f_{\xi_c}^r$. By updating $f_{\xi_c}^r$ with Eq. (11), we draw $f_{\xi_c}^r$ near to F_θ^{r+1} until their distance is comparable to the distance between F_θ^{r+1} and F_ξ^{r+1}. In this way, the updated $f_{\xi_c}^r$ can approximate F_ξ^{r+1}.

The PTNU eliminates the need for downloading the aggregated global target network F_ξ^{r+1} to clients. Considering the communications of four components, including upload of the local online network $f_{\theta_c}^r$ and the local target network $f_{\xi_c}^r$, and the download of the global online network F_θ^{r+1} and the global target network F_ξ^{r+1}, PTNU reduces about 25% of the communication.

4.5. Distance prediction for predictive target network update

To further reduce the communication cost, we propose a distance prediction (DP) method that eliminates most of the uploads of the local target network $f_{\xi_c}^r$ and only uses sporadic upload for calibration. In this way, by combining DP with PTNU, most of the communications regarding the target networks are removed, which can reduce about 50% of the communications.

To perform PTNU by Eq. (11), the exact distance $d(F_\theta^{r+1}, F_\xi^{r+1})$ between the aggregated online network F_θ^{r+1} and the target network F_ξ^{r+1} is required, which is computed on the server by Eq. (10). Since F_ξ^{r+1} is aggregated by the uploaded local target models $f_{\xi_c}^r, c \in C$ with Eq. (9), to reduce the upload of $f_{\xi_c}^r$, instead of computing the exact $d(F_\theta^{r+1}, F_\xi^{r+1})$ on the server, we approximate it by a proxy distance $\tilde{d}(F_\theta^{r+1}, F_\xi^{r+1})$ computed on the clients.

At the beginning of round $r+1$, each client $c \in C$ downloads F_θ^{r+1} from the server, and computes dp_c as:

$$dp_c = d(F_\theta^{r+1}, f_{\xi_c}^r) = \|F_\theta^{r+1} - f_{\xi_c}^r\|_1 \tag{12}$$

which is then uploaded to the server to predict the distance $\tilde{d}(F_\theta^{r+1}, F_\xi^{r+1})$ as:

$$DP = \frac{1}{|C|} \sum_{c \in C} dp_c. \tag{13}$$

$$\tilde{d}(F_\theta^{r+1}, F_\xi^{r+1}) = \alpha DP. \tag{14}$$

Compared with the exact distance $d(F_\theta^{r+1}, F_\xi^{r+1}) = d(F_\theta^{r+1}, \sum_{c \in C} \frac{n_c}{n} f_{\xi_c}^r)$, which computes the weighted average of local target networks $f_{\xi_c}^r$ on the server before computing the ℓ_1 distance, the distance prediction first computes the distance between the global online network F_θ^{r+1} and the target network $f_{\xi_c}^r$ on the client, after which the distance averaging is taken on the server. Since the value of DP and the ground truth distance $d(F_\theta^{r+1}, F_\xi^{r+1})$ are slightly different, we use a calibration parameter α, which is slightly smaller than 1, to accommodate for the difference between DP and $d(F_\theta^{r+1}, F_\xi^{r+1})$. In this way, $\tilde{d}(F_\theta^{r+1}, F_\xi^{r+1})$ as the adjusted DP can accurately approximate the $d(F_\theta^{r+1}, F_\xi^{r+1})$.

dp_c is computed on each client directly from local target networks $f^r_{\xi_c}$ without the need for uploading $f^r_{\xi_c}$. After computing $d(F^{r+1}_\theta, f^r_{\xi_c})$ on each client c, this distance is uploaded to the server for the averaging in Eq. (13) and scaling in Eq. (14). Then the value of $\tilde{d}(F^{r+1}_\theta, F^{r+1}_\xi)$ is downloaded to clients and used as the target distance defined in Eq. (10) for performing PTNU.

In the training process, the scaling factor between $d(F^{r+1}_\theta, F^{r+1}_\xi)$ and DP can gradually shift and the calibration parameter α needs to be adjusted to make the prediction $\tilde{d}(F^{r+1}_\theta, F^{r+1}_\xi)$ closely follow the real distance $d(F^{r+1}_\theta, F^{r+1}_\xi)$. To achieve this, in every R rounds, we periodically upload the local target networks to the server for computing the ground-truth distance $d(F^{r+1}_\theta, F^{r+1}_\xi)$ by Eq. (10), and adjust the calibration parameter as:

$$\alpha = \frac{d(F^{r+1}_\theta, F^{r+1}_\xi)}{DP}. \tag{15}$$

where DP is the distance prediction by Eq. (13). By using the calibrated α, we can perform accurate distance prediction by Eq. (14) for the following training rounds, which further helps the PTNU process.

5. Experiments

Dataset and preprocessing. We evaluate the proposed approaches on the ACDC MICCAI 2017 challenge dataset (Bernard et al., 2018). It consists of 100 patients with 3D cardiac MRI images. Each patient has about 15 volumes covering a full cardiac cycle, and only volumes for the end-diastolic and end-systolic phases are annotated by experts for three structures, including the left ventricle, myocardium, and right ventricle. The HVSMR MICCAI 2016 challenge dataset (Pace et al., 2015) contains 10 3D cardiac MRI images captured in an axial view using a 1.5T scanner with expert annotations of the blood pool and ventricular myocardium. In the pre-processing, for both datasets, following Chaitanya et al. (2020) we first normalize the intensity of each 3D volume x using min–max normalization to $[x_1, x_{99}]$, where x_p is the pth intensity percentile in x. Then we resample the 2D images and corresponding annotations to a fixed pixel size $r_f = 1.25 \times 1.25mm^2$ and $r_f = 0.7 \times 0.7mm^2$ for ACDC and HVSMR, respectively.

Federated and training setting. Following Zhao et al. (2018), we use 10 clients. We randomly split 100 patients in ACDC dataset into 10 partitions, each with 10 patients. Then each client is assigned one partition with 10 patients. We use the proposed FCL approaches to pre-train the U-Net encoder on the assigned dataset partition on each client without labels. Then the pre-trained encoder (i.e. the final global encoder after pre-training) is used as the initialization for fine-tuning the U-Net segmentation model by using a small number of labeled samples. The U-Net model follows the standard 2D U-Net architecture (Ronneberger et al., 2015) with the initial number of channels set to 48. We evaluate with three settings for fine-tuning: *local fine-tuning*, *federated fine-tuning*, and *centralized fine-tuning*. In local fine-tuning, each client fine-tunes the model on its local annotated data. In federated fine-tuning, all clients collaboratively fine-tune the model by supervised FL with a small number of annotations. In centralized fine-tuning, all data are combined for sampling the annotated data, following a standard evaluation protocol (Caron et al., 2020; Chen et al., 2020a).

Evaluation. During fine-tuning, we use 5-fold cross-validation to evaluate the segmentation performance. In each fold, 10 patients on one client are split into a training set of 8 patients and a validation set of 2 patients. For each fold, we fine-tune with annotations from $N \in \{1, 2, 4, 8\}$ patients in the training set, and validate on the validation set of the same fold on all clients (i.e. 20 patients). Dice similarity coefficient (DSC) is used as the metric for evaluation.

Training details. The FCL is performed for 200 rounds. The percentage of active clients per round is 1.0 and the number of local epochs per communication round is 1. The size of the memory bank is 4096. The temperature τ for contrastive loss is 0.1 and the momentum is 0.99. The SGD optimizer is used with momentum 0.9 and weight decay 0.0001. The batch size is 32 and the learning rate is 0.05 with a cosine decay schedule. For FCLOpt, FCLOpt-PTNU, and FCLOpt-PTNU-DP, the learning rate is 0.5 with a cosine decay schedule. The momentum parameter m for the target network update is 0.99. m_d for PTNU is 0.995. The calibration parameter α is adjusted every 10 training rounds. As in Chaitanya et al. (2020), we group each volume into 4 partitions. In the fine-tuning stage, the model is trained for 200 epochs in local fine-tuning or 200 rounds in federated fine-tuning. Adam optimizer is used with a batch size of 10, a learning rate of 0.0005, and a cosine schedule. The training is performed on one Nvidia V100 GPU.

Baselines. We compare the proposed approaches with multiple baselines. *Random init* fine-tunes the model from random initialization. *Local CL* performs contrastive learning on each client by the SOTA approach (Chaitanya et al., 2020) with unlabeled data for pre-training the encoder before fine-tuning. *Rotation* (Gidaris et al., 2018) is a self-supervised pre-training approach by predicting the image rotations. *SimCLR* (Chen et al., 2020a), *SwAV* (Caron et al., 2020), *MoCo* (He et al., 2020), and *BYOL* (Grill et al., 2020) are the SOTA self-supervised learning approaches for pre-training. We combine these three self-supervised approaches with *FedAvg* (McMahan et al., 2017) as their federated variants *FedRotation*, *FedSimCLR*, *FedSwAV*, *FedMoCo*, and *FedBYOL* for pre-training the encoder. *FedGL* is the combination of the SOTA self-supervised learning approach for volumetric medical image segmentation (Chaitanya et al., 2020) with *FedAvg*. *FedCA* (Zhang et al., 2020) and *FedU* (Zhuang et al., 2021) are two federated self-supervised learning methods for pre-training. In the experimental results, we denote the method introduced in Section 3 as *FCL*, the method described in Section 4 without PTNU or DP as *FCLOpt*, and denote the methods with PTNU and DP enabled as *FCLOpt-PTNU* and *FCLOpt-PTNU-DP*, respectively.

5.1. Results of local fine-tuning

We evaluate the performance of the proposed approaches by fine-tuning locally on each client with limited annotations. As shown in Table 1, the proposed approaches substantially outperform the baselines. First, with 1, 2, 4, or 8 annotated patients, our FCL method outperforms the best-performing baseline by 0.065, 0.045, 0.042, and 0.029 dice score, respectively. Our communication-optimized methods FCLOpt, FCL-PTNU, and FCL-PTNU-DP achieve a similar or higher dice score than our method FCL. Second, the proposed approaches significantly improve the annotation efficiency. For example, with 2 or 4 annotated patients, our FCLOpt method performs on par with the best-performing baseline with 2× annotations (0.655 vs. 0.703 and 0.745 vs. 0.795), respectively.

5.2. Results of federated fine-tuning

We evaluate the performance of the proposed approaches by collaborative federated fine-tuning with limited annotations. Similar to local fine-tuning, the proposed approaches significantly outperform the SOTA techniques as shown in Table 2. First, with 1, 2, 4, or 8 annotated patients per client (i.e. 10, 20, 40, or 80 annotated patients in total), our FCLOpt method outperforms the best-performing baselines by 0.148, 0.063, 0.027, and 0.018 dice score, respectively. Second, the proposed approaches effectively reduce the annotations needed for fine-tuning. For example, with 2 or 4 annotated patients per client, our FCLOpt method achieves better performance than the best-performing baseline with 2× annotated patients per client (0.853 vs. 0.850 and 0.877 vs. 0.879, respectively), which achieve more than 2× labeling-efficiency. Third, compared with local fine-tuning in Table 1, all the approaches achieve a higher dice score. This is because federated fine-tuning with annotations on distributed clients leverages more annotations than local fine-tuning with only local annotations.

Table 1
Comparison of the proposed approaches and baselines on **local fine-tuning** with limited annotations on the ACDC dataset. N is the number of annotated patients for fine-tuning on each client. The average dice score and standard deviation across 10 clients are reported, in which on each client the dice score is averaged on 5-fold cross-validation. The proposed approaches substantially outperform all the baselines with different numbers of annotations.

Methods	N=1	N=2	N=4	N=8
Random init	0.280 ± 0.037	0.414 ± 0.070	0.618 ± 0.026	0.766 ± 0.027
Local CL (Chaitanya et al., 2020)	0.320 ± 0.106	0.456 ± 0.095	0.637 ± 0.043	0.770 ± 0.029
FedRotation (Gidaris et al., 2018)	0.357 ± 0.058	0.508 ± 0.054	0.660 ± 0.021	0.783 ± 0.029
FedSimCLR (Chen et al., 2020a)	0.288 ± 0.049	0.435 ± 0.046	0.619 ± 0.032	0.765 ± 0.033
FedSwAV (Caron et al., 2020)	0.323 ± 0.066	0.480 ± 0.067	0.659 ± 0.019	0.782 ± 0.030
FedCA (Zhang et al., 2020)	0.280 ± 0.047	0.417 ± 0.042	0.610 ± 0.030	0.766 ± 0.029
FedMoCo (He et al., 2020)	0.287 ± 0.056	0.442 ± 0.066	0.626 ± 0.034	0.767 ± 0.030
FedBYOL (Grill et al., 2020)	0.431 ± 0.057	0.554 ± 0.052	0.685 ± 0.021	0.781 ± 0.027
FedU (Zhuang et al., 2021)	0.441 ± 0.047	0.586 ± 0.043	0.703 ± 0.018	0.795 ± 0.022
FedGL (Chaitanya et al., 2020)	0.260 ± 0.036	0.404 ± 0.063	0.633 ± 0.028	0.765 ± 0.040
FCL (ours)	0.506 ± 0.056	0.631 ± 0.051	**0.745** ± 0.017	**0.824** ± 0.025
FCLOpt (ours)	**0.524** ± 0.052	**0.655** ± 0.039	**0.745** ± 0.020	0.821 ± 0.020
FCL-PTNU (ours)	0.517 ± 0.061	0.622 ± 0.045	0.730 ± 0.019	0.810 ± 0.022
FCL-PTNU-DP (ours)	0.512 ± 0.053	0.621 ± 0.050	0.729 ± 0.016	0.810 ± 0.027

Table 2
Comparison of the proposed approaches and baselines on **federated fine-tuning** with limited annotations on the ACDC dataset. N is the number of annotated patients for fine-tuning on each client. L is the total number of annotated patients from all clients. The proposed approaches significantly outperform all the baselines with different numbers of annotations.

Annotated patients per client	N=1	N=2	N=4	N=8
Annotated patients of all clients	L=1 × 10	L=2 × 10	L=4 × 10	L=8 × 10
Random init	0.445 ± 0.012	0.572 ± 0.061	0.764 ± 0.017	0.834 ± 0.011
Local CL (Chaitanya et al., 2020)	0.473 ± 0.013	0.717 ± 0.024	0.784 ± 0.015	0.847 ± 0.009
FedRotation (Gidaris et al., 2018)	0.516 ± 0.015	0.627 ± 0.074	0.821 ± 0.015	0.867 ± 0.010
FedSimCLR (Chen et al., 2020a)	0.395 ± 0.023	0.576 ± 0.046	0.788 ± 0.014	0.859 ± 0.011
FedSwAV (Caron et al., 2020)	0.500 ± 0.015	0.594 ± 0.058	0.815 ± 0.015	0.862 ± 0.010
FedCA (Zhang et al., 2020)	0.397 ± 0.020	0.561 ± 0.047	0.784 ± 0.015	0.858 ± 0.011
FedMoCo (He et al., 2020)	0.467 ± 0.016	0.675 ± 0.053	0.782 ± 0.018	0.846 ± 0.011
FedBYOL (Grill et al., 2020)	0.621 ± 0.065	0.790 ± 0.011	0.840 ± 0.006	0.871 ± 0.006
FedU (Zhuang et al., 2021)	0.576 ± 0.082	0.717 ± 0.105	0.850 ± 0.010	0.879 ± 0.007
FedGL (Chaitanya et al., 2020)	0.468 ± 0.019	0.687 ± 0.051	0.813 ± 0.015	0.865 ± 0.009
FCL (ours)	0.646 ± 0.052	0.824 ± 0.004	0.871 ± 0.007	0.894 ± 0.006
FCLOpt (ours)	**0.769** ± 0.025	**0.853** ± 0.006	**0.877** ± 0.006	**0.897** ± 0.004
FCL-PTNU (ours)	0.680 ± 0.086	0.840 ± 0.007	0.868 ± 0.009	0.889 ± 0.006
FCL-PTNU-DP (ours)	0.778 ± 0.016	0.840 ± 0.002	0.873 ± 0.006	0.894 ± 0.004

Table 3
Comparison of the proposed approaches and baselines on **centralized fine-tuning** with limited annotations on the ACDC dataset. N is the number of annotated patients for fine-tuning. The proposed approaches significantly outperform all the baselines with different numbers of annotations.

Methods	N=1	N=2	N=4	N=8
Random init	0.296 ± 0.091	0.528 ± 0.064	0.677 ± 0.056	0.797 ± 0.028
Local CL (Chaitanya et al., 2020)	0.314 ± 0.058	0.544 ± 0.065	0.691 ± 0.040	0.805 ± 0.014
FedRotation (Gidaris et al., 2018)	0.374 ± 0.072	0.583 ± 0.061	0.686 ± 0.056	0.815 ± 0.021
FedSimCLR (Chen et al., 2020a)	0.287 ± 0.030	0.524 ± 0.065	0.658 ± 0.037	0.802 ± 0.022
FedSwAV (Caron et al., 2020)	0.334 ± 0.096	0.575 ± 0.063	0.726 ± 0.044	0.805 ± 0.020
FedCA (Zhang et al., 2020)	0.320 ± 0.067	0.527 ± 0.067	0.653 ± 0.049	0.793 ± 0.024
FedMoCo (He et al., 2020)	0.310 ± 0.068	0.520 ± 0.055	0.695 ± 0.045	0.802 ± 0.017
FedBYOL (Grill et al., 2020)	0.472 ± 0.073	0.633 ± 0.042	0.729 ± 0.039	0.805 ± 0.021
FedU (Zhuang et al., 2021)	0.485 ± 0.133	0.633 ± 0.057	0.747 ± 0.047	0.820 ± 0.031
FedGL (Chaitanya et al., 2020)	0.331 ± 0.057	0.520 ± 0.066	0.686 ± 0.045	0.795 ± 0.020
FCL (ours)	0.575 ± 0.113	0.702 ± 0.041	**0.790** ± 0.026	**0.844** ± 0.024
FCLOpt (ours)	**0.587** ± 0.109	**0.708** ± 0.055	0.785 ± 0.038	0.837 ± 0.031
FCL-PTNU (ours)	0.547 ± 0.130	0.691 ± 0.052	0.771 ± 0.041	0.837 ± 0.020
FCL-PTNU-DP (ours)	0.556 ± 0.132	0.674 ± 0.069	0.763 ± 0.055	0.831 ± 0.031

5.3. Results of centralized fine-tuning

We evaluate the performance of the proposed approaches by centralized fine-tuning with limited annotations, which is a standard evaluation protocol for generic self-supervised learning. As shown in Table 3, the proposed approaches FCL and FCLOpt achieve significantly higher performance than the baselines. First, with 1, 2, 4, or 8 annotated patients, our FCL method outperforms the best-performing baseline by 0.090, 0.069, 0.043, and 0.024 dice score, respectively, while our communication-optimized FCLOpt, FCL-PTNU, and FCL-PTNU-DP perform on par with FCL. Second, all our methods significantly improve the annotation efficiency. For example, with 2 or 4 annotated patients, FCL performs on par with the best-performing baseline with 2× annotations (0.702 vs. 0.747 and 0.790 vs. 0.820), respectively, which roughly improves labeling-efficiency by 2×.

In addition to the default learning rate of 0.0005, we further explore more learning rates for the random init baseline in the centralized fine-tuning setting. As shown in Table 4, increasing the learning rate only results in marginal improvement and even degrades the performance of

Table 4
Impact of learning rate on the random init baseline in the centralized fine-tuning setting. N is the number of annotated patients for fine-tuning.

LR	N=1	N=2	N=4	N=8
0.0005 (default)	0.296 ± 0.091	0.528 ± 0.064	0.677 ± 0.056	0.797 ± 0.028
0.0010	0.311 ± 0.035	0.546 ± 0.062	0.686 ± 0.037	0.813 ± 0.028
0.0020	0.294 ± 0.035	0.519 ± 0.080	0.680 ± 0.044	0.798 ± 0.027
0.0050	0.280 ± 0.033	0.485 ± 0.055	0.666 ± 0.069	0.795 ± 0.029
0.0100	0.290 ± 0.069	0.480 ± 0.043	0.668 ± 0.055	0.806 ± 0.017

Table 5
Comparison of the proposed methods and baselines on **transfer learning** from ACDC to HVSMR dataset. M is the number of annotated patients for fine-tuning. The average dice score and standard deviation on 5-fold cross-validation are reported. The proposed approaches outperform all the baselines, which shows the proposed approaches can learn useful and transferable representations to be used on the downstream task.

Methods	M=1	M=2	M=4	M=8
Random init	0.792 ± 0.051	0.814 ± 0.049	0.827 ± 0.049	0.859 ± 0.039
Local CL (Chaitanya et al., 2020)	0.798 ± 0.053	0.811 ± 0.045	0.825 ± 0.052	0.855 ± 0.044
FedRotation (Gidaris et al., 2018)	0.800 ± 0.054	0.816 ± 0.055	0.834 ± 0.052	0.864 ± 0.037
FedSimCLR (Chen et al., 2020a)	0.797 ± 0.048	0.799 ± 0.048	0.815 ± 0.053	0.854 ± 0.040
FedSwAV (Caron et al., 2020)	0.802 ± 0.044	0.814 ± 0.054	0.842 ± 0.039	0.862 ± 0.040
FedCA (Zhang et al., 2020)	0.790 ± 0.043	0.802 ± 0.050	0.817 ± 0.056	0.861 ± 0.037
FedMoCo (He et al., 2020)	0.794 ± 0.049	0.815 ± 0.043	0.828 ± 0.045	0.857 ± 0.039
FedBYOL (Grill et al., 2020)	0.797 ± 0.047	0.802 ± 0.042	0.834 ± 0.045	0.865 ± 0.031
FedU (Zhuang et al., 2021)	0.806 ± 0.039	0.819 ± 0.042	0.843 ± 0.040	0.862 ± 0.047
FedGL (Chaitanya et al., 2020)	0.791 ± 0.054	0.813 ± 0.049	0.827 ± 0.054	0.860 ± 0.032
FCL (ours)	**0.814** ± 0.046	0.823 ± 0.048	**0.849** ± 0.038	**0.872** ± 0.033
FCLOpt (ours)	**0.814** ± 0.045	**0.828** ± 0.039	0.844 ± 0.042	**0.872** ± 0.028
FCL-PTNU (ours)	0.812 ± 0.040	0.829 ± 0.039	0.843 ± 0.044	0.868 ± 0.032
FCL-PTNU-DP (ours)	0.813 ± 0.040	0.832 ± 0.048	0.847 ± 0.040	0.868 ± 0.032

Table 6
Ablation study of FCL on the ACDC dataset. The average dice score and standard deviation across 10 clients by local fine-tuning are reported. N is the number of annotated patients for fine-tuning on each client.

Methods	N=1	N=2	N=4	N=8
Without FE	0.287 ± 0.056	0.442 ± 0.066	0.626 ± 0.034	0.767 ± 0.030
FE	0.296 ± 0.048	0.445 ± 0.069	0.634 ± 0.035	0.768 ± 0.028
FE+NS	0.373 ± 0.057	0.524 ± 0.064	0.678 ± 0.021	0.787 ± 0.027
FCL (FE+NS+GSM)	**0.506** ± 0.056	**0.631** ± 0.051	**0.745** ± 0.017	**0.824** ± 0.025

the random init baseline when the learning rate is too large. Therefore, the default learning rate we used is a good one. Besides, we use the same learning rate for all the baselines for fine-tuning, which is a fair comparison.

5.4. Results of transfer learning

We evaluate the generalization performance of the learned encoder by transferring to a new downstream task. We pre-train the encoder on ACDC by different methods and use the pre-trained encoder as the initialization for fine-tuning on the HVSMR dataset with limited annotations. The results are shown in Table 5. Under different numbers of annotations for fine-tuning, the proposed approaches consistently outperform the baselines. While the acquisition view and resolutions are different on the source ACDC dataset and target HVSMR dataset, the proposed approaches can still learn useful and transferable representations to be used on the downstream task.

5.5. Ablation studies of FCL

We perform ablation studies to evaluate the effectiveness of each component in the FCL with feature sharing introduced in Section 3. The influences of feature exchange (FE) by Eq. (1), negative sampling (NS) by Eq. (2), and global structural matching (GSM) by Eq. (5) on federated contrastive learning are evaluated. By progressively adding the proposed FE, NS, and GSM, the average dice score increases, which shows the effectiveness of each of the proposed approaches. As shown in Table 6, by using a given number of annotations for fine-tuning, enabling the proposed components FE, NS, and GSM one by one effectively improves the dice score after fine-tuning. For example, with 4 annotated patients, adding FE+NS improves the dice score from 0.626 to 0.678, and adding GSM further improves the dice score to 0.745. These results show the effectiveness of each component of FCL.

5.6. Results of reduced communication cost

We evaluate the effectiveness of *FCLOpt*, *PTNU*, and *DP* for reducing the communication cost while keeping a high segmentation performance, which are introduced in Section 4. The results are shown in Table 7, where the results of fine-tuning locally on each client are reported. We report the amount of data communication in each round of the federated pre-training, and the segmentation performance after fine-tuning. The communication cost is normalized such that the method FCL has a cost of 1.0 and standards for 124.2 MB communication per round. First, all of our four methods FCL (without communication optimization), FCLOpt, FCL-PTNU, and FCL-PTNU-DP achieve a high segmentation performance compared with the baselines FedMoCo and FedBYOL. We compare with the baselines FedMoCo and FedBYOL because our FCL employs the same base CL method MoCo (He et al., 2020) as FedMoCo, and FCLOpt, FCL-PTNU, FCL-PTNU-DP use the same base CL method BYOL (Grill et al., 2020) as FedBYOL. Second, compared with FCL, our communication-optimized FCLOpt effectively reduces the communication cost from $1.000\times$ to $0.645\times$. Adding $PTNU$ to FCLOpt as FCL-PTNU further reduces the communication cost to $0.509\times$ and adding DP reduces the communication cost to $0.386\times$. Compared with FedBYOL which only synchronizes the online network, our FCL-PTNU-DP has a comparable communication

Table 7
Comparison of the proposed approaches and baselines on **local fine-tuning** with limited annotations on the ACDC dataset. *Communication* is the amount of data communications in the FCL pretraining process. N is the number of annotated patients for fine-tuning on each client.

Methods	Comm.	N=1	N=2	N=4	N=8
Baselines					
FedMoCo	0.544 ×	0.287 ± 0.056	0.442 ± 0.066	0.626 ± 0.034	0.767 ± 0.030
FedBYOL	0.373 ×	0.431 ± 0.057	0.554 ± 0.052	0.685 ± 0.021	0.781 ± 0.027
Our methods: high accuracy with feature sharing					
FCL (ours)	1.000 ×	**0.506** ± 0.056	**0.631** ± 0.051	**0.745** ± 0.017	**0.824** ± 0.025
Our methods: optimizing communication					
FCLOpt (ours)	0.645 ×	**0.524** ± 0.052	**0.655** ± 0.039	**0.745** ± 0.020	**0.821** ± 0.020
FCL-PTNU (ours)	0.509 ×	0.517 ± 0.061	0.622 ± 0.045	0.730 ± 0.019	0.810 ± 0.022
FCL-PTNU-DP (ours)	0.386 ×	0.512 ± 0.053	0.621 ± 0.050	0.729 ± 0.016	0.810 ± 0.027

Table 8
Communicated model components in different methods.

Methods	Online network	Online predictor	Target network	Encoded representations
MoCo-based methods				
FedMoCo	✓		✓	
FCL (ours)	✓		✓	✓
BYOL-based methods				
FedBYOL	✓	✓		
FCLOpt (ours)	✓	✓	✓	
FCLOpt-PTNU (ours)	✓	✓	Upload only	
FCLOpt-PTNU-DP (ours)	✓	✓	~ 0	

cost but significantly better segmentation performance. These results show all of our methods achieve a high segmentation performance, and enabling the communication optimizations can effectively reduce the communication cost.

5.7. Comparison of communicated model components

To better understand different methods, we show the communicated model components in Table 8, in which the top half compares MoCo-based methods, while the bottom half compares BYOL-based methods. First, both MoCo-based baseline FedMoCo and our FCL communicate the online network and the target network (the online predictor does not exist in MoCo). The difference is that our FCL also communicates the encoded features. As shown in Table 7, our FCL greatly outperforms FedMoCo in terms of model performance at the cost of increased communication cost. This is desirable when model performance is the main goal while communication has a marginal cost for medical institutions with high-speed connections. Second, for BYOL-based methods, the goal of our FCLOpt with PTNU and DP is to achieve a similar communication cost as FedBYOL while having substantially higher model performance. More specifically, in FedBYOL, only the online network and the online predictor are communicated, while the target network is not. Based on FedBYOL, we propose FCLOpt which further communicates the target network for higher model performance. Then, we propose PTNU which eliminates the upload of the target network. After that. we propose DP to eliminate most of the downloads of the target network. As shown in Table 7, our FCLOpt-PTNU-DP achieves much higher model performance than FedBYOL and has almost the same communication cost as FedBYOL.

5.8. Visualization

We visualize the segmentation results of the ACDC dataset in Fig. 4. The input image and the ground truth annotations are shown in the first two images, followed by segmentation results of the baseline methods, and the results of our methods are shown in the third row. Our methods generate better visual segmentation results than the baselines and are more similar to the ground-truth annotations, which are consistent with the quantitative results.

6. Conclusion

This work aims to enable federated contrastive learning (FCL) for volumetric medical image segmentation with limited annotations. Clients first learn a shared encoder on distributed unlabeled data and then a model is fine-tuned on annotated data. Feature exchange is proposed to improve data diversity for contrastive learning while avoiding sharing raw data. Global structural matching is developed to learn an encoder with unified representations among clients. To reduce the communication cost of FCL, an optimized method FCLOpt that does not rely on negative samples is proposed. Based on FCLOpt, predictive target network update (PTNU) is developed by predicting the target network by fast forward to further reduce the communications of model downloading. Distance prediction (DP) is proposed to remove the uploading of the target network. The experimental results show significantly improved segmentation performance and labeling-efficiency compared with state-of-the-art techniques.

7. Description of the extensions

The original version of this paper was published on MICCAI 2021 proceedings (Wu et al., 2021b). Compared with the original version, we made the following extensions in this manuscript.

1. We added Section 4 to describe the communication-optimized method FCLOpt. The FCL method introduced in the original MICCAI paper (described in Section 3 in this manuscript) requires additional communication for feature sharing, aiming at improving the segmentation performance. We extend the original method by proposing an optimized method FCLOpt (Sections 4.2 and 4.3) that does not rely on negative samples to eliminate the communication costs of feature sharing.
2. Based on FCLOpt, to further reduce the communications of model download, we propose the predictive target network update (PTNU) that predicts the target network by fast forward (Section 4.4).
3. Based on PTNU, we propose the distance prediction (DP) to remove the uploading of the target network (Section 4.5).
4. We added experiments for the extended methods. More specifically, we added experiments for the FCLOpt, FCL-PTNU, and FCL-PTNU-DP in Section 5. The results of segmentation performance by local fine-tuning, federated fine-tuning, centralized fine-tuning, and transfer learning are added to Section 5.1, Section 5.2, Section 5.3, and Section 5.4, respectively. The results of communication cost are added to Section 5.6 to show the reduced communication by the added methods compared with the FCL method in the original paper. The visualization of segmentation results is added to Section 5.8. These results show the extended FCLOpt including the PTNU and DP methods effectively reduces the communication cost while preserving the segmentation performance of the FCL method in the original paper.

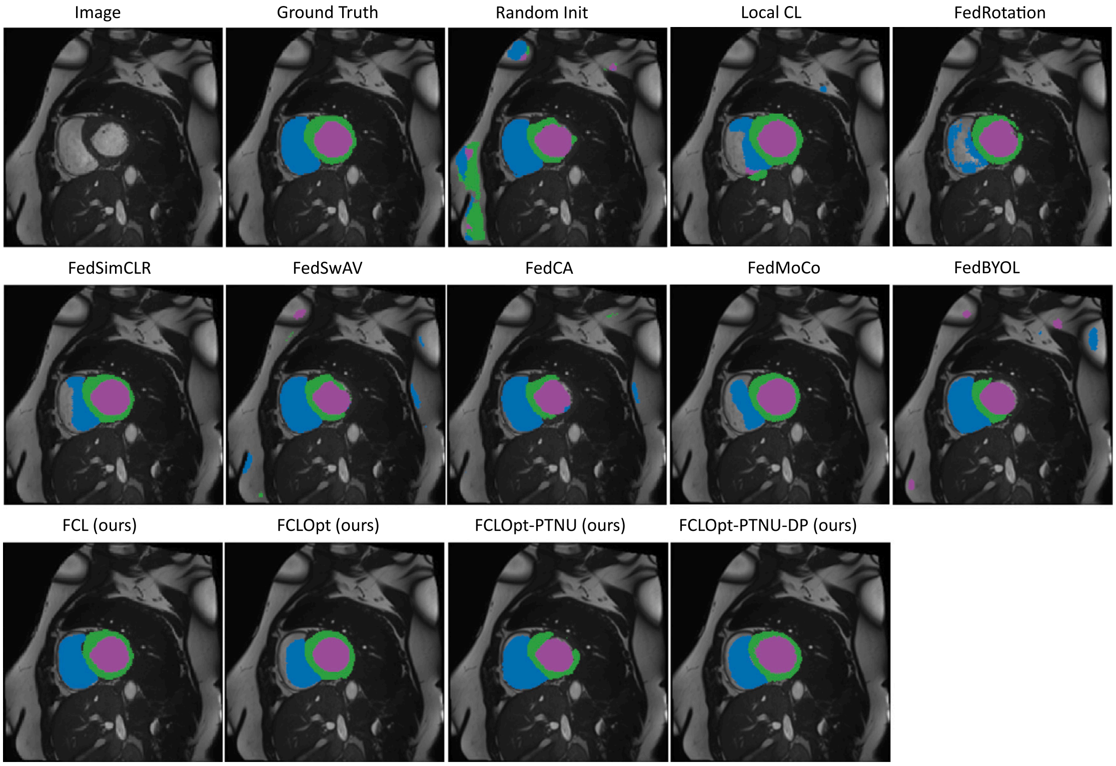


Fig. 4. Visualization of segmentation results on ACDC dataset. The results are generated from the fine-tuned model when 2 annotated patients are used for fine-tuning ($N = 2$). The proposed approaches achieve significantly better segmentation performance than the baselines.

5. In addition to the extended methods and corresponding experiments, in the experimental results of Section 5, we added two baseline methods FedMoCo and FedBYOL to the experimental results for comparison. We also added the evaluation protocol centralized fine-tuning with limited annotations, which is a standard evaluation protocol for generic self-supervised learning.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Jingtong Hu reports financial support was provided by National Science Foundation. Yiyu Shi reports financial support was provided by National Science Foundation.

Acknowledgments

This work used the Extreme Science and Engineering Discovery Environment (XSEDE), which is supported by National Science Foundation, United States grant number ACI-1548562. Specifically, it used the Bridges system, which is supported by NSF, United States award number ACI-1445606, at the Pittsburgh Supercomputing Center (PSC).

References

Bercea, C.I., Wiestler, B., Rueckert, D., Albarqouni, S., 2021. Feddis: Disentangled federated learning for unsupervised brain pathology segmentation. arXiv preprint arXiv:2103.03705.

Bernard, O., Lalande, A., Zotti, C., Cervenansky, F., Yang, X., Heng, P.-A., Cetin, I., Lekadir, K., Camara, O., Ballester, M.A.G., et al., 2018. Deep learning techniques for automatic MRI cardiac multi-structures segmentation and diagnosis: is the problem solved? IEEE Trans. Med. Imaging 37 (11), 2514–2525.

Caron, M., Misra, I., Mairal, J., Goyal, P., Bojanowski, P., Joulin, A., 2020. Unsupervised learning of visual features by contrasting cluster assignments. In: Proceedings of Advances in Neural Information Processing Systems (NeurIPS).

Chaitanya, K., Erdil, E., Karani, N., Konukoglu, E., 2020. Contrastive learning of global and local features for medical image segmentation with limited annotations. Adv. Neural Inf. Process. Syst. 33, 12546–12558.

Chen, T., Kornblith, S., Norouzi, M., Hinton, G., 2020a. A simple framework for contrastive learning of visual representations. In: III, H.D., Singh, A. (Eds.), Proceedings of the 37th International Conference on Machine Learning. In: Proceedings of Machine Learning Research, vol. 119, PMLR, pp. 1597–1607.

Chen, T., Kornblith, S., Swersky, K., Norouzi, M., Hinton, G., 2020b. Big self-supervised models are strong semi-supervised learners. arXiv preprint arXiv:2006.10029.

Dong, N., Voiculescu, I., 2021. Federated contrastive learning for decentralized unlabeled medical images. In: International Conference on Medical Image Computing and Computer-Assisted Intervention. Springer, pp. 378–387.

Dong, H., Yang, G., Liu, F., Mo, Y., Guo, Y., 2017. Automatic brain tumor detection and segmentation using u-net based fully convolutional networks. In: Annual Conference on Medical Image Understanding and Analysis. Springer, pp. 506–517.

Dou, Q., So, T.Y., Jiang, M., Liu, Q., Vardhanabhuti, V., Kaissis, G., Li, Z., Si, W., Lee, H.H., Yu, K., et al., 2021. Federated deep learning for detecting COVID-19 lung abnormalities in CT: a privacy-preserving multinational validation study. NPJ Digit.l Med. 4 (1), 1–11.

Gidaris, S., Singh, P., Komodakis, N., 2018. Unsupervised representation learning by predicting image rotations. arXiv preprint arXiv:1803.07728.

Grill, J.-B., Strub, F., Altché, F., Tallec, C., Richemond, P., Buchatskaya, E., Doersch, C., Avila Pires, B., Guo, Z., Gheshlaghi Azar, M., et al., 2020. Bootstrap your own latent-a new approach to self-supervised learning. Adv. Neural Inf. Process. Syst. 33, 21271–21284.

Hadsell, R., Chopra, S., LeCun, Y., 2006. Dimensionality reduction by learning an invariant mapping. In: 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06), vol. 2. IEEE, pp. 1735–1742.

He, K., Fan, H., Wu, Y., Xie, S., Girshick, R., 2020. Momentum contrast for unsupervised visual representation learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 9729–9738.

Huang, Y., Song, Z., Li, K., Arora, S., 2020. Instahide: Instance-hiding schemes for private distributed learning. In: International Conference on Machine Learning. PMLR, pp. 4507–4518.

Kairouz, P., McMahan, H.B., Avent, B., Bellet, A., Bennis, M., Bhagoji, A.N., Bonawitz, K., Charles, Z., Cormode, G., Cummings, R., et al., 2019. Advances and open problems in federated learning. arXiv preprint arXiv:1912.04977.

Li, T., Sahu, A.K., Zaheer, M., Sanjabi, M., Talwalkar, A., Smith, V., 2018. Federated optimization in heterogeneous networks. arXiv preprint arXiv:1812.06127.

Li, T., Sahu, A.K., Zaheer, M., Sanjabi, M., Talwalkar, A., Smith, V., 2020. Federated optimization in heterogeneous networks. Proc. Mach. Learn. Syst. 2, 429–450.

McMahan, B., Moore, E., Ramage, D., Hampson, S., y Arcas, B.A., 2017. Communication-efficient learning of deep networks from decentralized data. In: Artificial Intelligence and Statistics. PMLR, pp. 1273–1282.

Milletari, F., Navab, N., Ahmadi, S.-A., 2016. V-net: Fully convolutional neural networks for volumetric medical image segmentation. In: 2016 Fourth International Conference on 3D Vision (3DV). IEEE, pp. 565–571.

Misra, I., Maaten, L.v.d., 2020. Self-supervised learning of pretext-invariant representations. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 6707–6717.

Pace, D.F., Dalca, A.V., Geva, T., Powell, A.J., Moghari, M.H., Golland, P., 2015. Interactive whole-heart segmentation in congenital heart disease. In: International Conference on Medical Image Computing and Computer-Assisted Intervention. Springer, pp. 80–88.

Rieke, N., Hancox, J., Li, W., Milletari, F., Roth, H.R., Albarqouni, S., Bakas, S., Galtier, M.N., Landman, B.A., Maier-Hein, K., et al., 2020. The future of digital health with federated learning. NPJ Digit. Med. 3 (1), 1–7.

Ronneberger, O., Fischer, P., Brox, T., 2015. U-net: Convolutional networks for biomedical image segmentation. In: International Conference on Medical Image Computing and Computer-Assisted Intervention. Springer, pp. 234–241.

Sheller, M.J., Edwards, B., Reina, G.A., Martin, J., Pati, S., Kotrotsou, A., Milchenko, M., Xu, W., Marcus, D., Colen, R.R., et al., 2020. Federated learning in medicine: facilitating multi-institutional collaborations without sharing patient data. Sci. Rep. 10 (1), 1–12.

Sheller, M.J., Reina, G.A., Edwards, B., Martin, J., Bakas, S., 2018. Multi-institutional deep learning modeling without sharing patient data: A feasibility study on brain tumor segmentation. In: International MICCAI Brainlesion Workshop. Springer, pp. 92–104.

Sun, J., Li, A., Wang, B., Yang, H., Li, H., Chen, Y., 2021. Soteria: Provable defense against privacy leakage in federated learning from representation perspective. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 9311–9319.

Tian, Y., Krishnan, D., Isola, P., 2019. Contrastive multiview coding. arXiv preprint arXiv:1906.05849.

Truong, N., Sun, K., Wang, S., Guitton, F., Guo, Y., 2020. Privacy preservation in federated learning: An insightful survey from the GDPR perspective. arXiv preprint arXiv:2011.05411.

Van Berlo, B., Saeed, A., Ozcelebi, T., 2020. Towards federated unsupervised representation learning. In: Proceedings of the Third ACM International Workshop on Edge Systems, Analytics and Networking. pp. 31–36.

Wang, T., Isola, P., 2020. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In: International Conference on Machine Learning. PMLR, pp. 9929–9939.

Wu, Y., Wang, Z., Zeng, D., Shi, Y., Hu, J., 2021a. Enabling on-device self-supervised contrastive learning with selective data contrast. In: 2021 58th ACM/IEEE Design Automation Conference (DAC). IEEE, pp. 655–660.

Wu, Z., Xiong, Y., Yu, S.X., Lin, D., 2018. Unsupervised feature learning via non-parametric instance discrimination. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 3733–3742.

Wu, Y., Zeng, D., Wang, Z., Shi, Y., Hu, J., 2021b. Federated contrastive learning for volumetric medical image segmentation. In: International Conference on Medical Image Computing and Computer-Assisted Intervention. Springer, pp. 367–377.

Xu, X., Wang, T., Shi, Y., Yuan, H., Jia, Q., Huang, M., Zhuang, J., 2019. Whole heart and great vessel segmentation in congenital heart disease using deep neural networks and graph matching. In: International Conference on Medical Image Computing and Computer-Assisted Intervention. Springer, pp. 477–485.

Yang, Q., Liu, Y., Chen, T., Tong, Y., 2019. Federated machine learning: Concept and applications. ACM Trans. Intell. Syst. Technol. (TIST) 10 (2), 1–19.

Zeng, D., Wu, Y., Hu, X., Xu, X., Yuan, H., Huang, M., Zhuang, J., Hu, J., Shi, Y., 2021. Positional contrastive learning for volumetric medical image segmentation. In: International Conference on Medical Image Computing and Computer-Assisted Intervention. Springer, pp. 221–230.

Zhang, F., Kuang, K., You, Z., Shen, T., Xiao, J., Zhang, Y., Wu, C., Zhuang, Y., Li, X., 2020. Federated unsupervised representation learning. arXiv preprint arXiv:2010.08982.

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

Zhuang, W., Gan, X., Wen, Y., Zhang, S., Yi, S., 2021. Collaborative unsupervised visual representation learning from decentralized data. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 4912–4921.

Zhuang, W., Wen, Y., Zhang, S., 2022. Divergence-aware federated self-supervised learning. arXiv preprint arXiv:2204.04385.