HHS Public Access

Author manuscript

Med Image Anal. Author manuscript; available in PMC 2023 May 01.

Published in final edited form as:

Med Image Anal. 2022 May ; 78: 102424. doi:10.1016/j.media.2022.102424.

Handling Data Heterogeneity with Generative Replay in Collaborative Learning for Medical Imaging

Liangqiong Qu[a], **Niranjan Balachandar**[a], **Miao Zhang**[a], **Daniel Rubin**[b,*]

[a]Department of Biomedical Data Science at Stanford University, Stanford, CA 94305, USA

[b]Department of Biomedical Data Science and Department of Radiology at Stanford University, Stanford, CA 94305, USA.

Abstract

Collaborative learning, which enables collaborative and decentralized training of deep neural networks at multiple institutions in a privacy-preserving manner, is rapidly emerging as a valuable technique in healthcare applications. However, its distributed nature often leads to significant heterogeneity in data distributions across institutions. In this paper, we present a novel generative replay strategy to address the challenge of data heterogeneity in collaborative learning methods. Different from traditional methods that directly aggregating the model parameters, we leverage generative adversarial learning to aggregate the knowledge from all the local institutions. Specifically, instead of directly training a model for task performance, we develop a novel dual model architecture: a primary model learns the desired task, and an auxiliary "generative replay model" allows aggregating knowledge from the heterogenous clients. The auxiliary model is then broadcasted to the central sever, to regulate the training of primary model with an unbiased target distribution. Experimental results demonstrate the capability of the proposed method in handling heterogeneous data across institutions. On highly heterogeneous data partitions, our model achieves ~4.88% improvement in the prediction accuracy on a diabetic retinopathy classification dataset, and ~49.8% reduction of mean absolution value on a Bone Age prediction dataset, respectively, compared to the state-of-the art collaborative learning methods.

Keywords

Collaborative learning; federated learning; data heterogeneity; Generative Adversarial Networks (GAN); autoencoder

1. Introduction

Deep artificial neural networks (DNNs) have led to state-of-the-art performances in a wide range of computer vision tasks. One major issue in DNNs is the requirement for large quantities of the annotated image data to train robust models. However, large quantities of the data are inherently decentralized and are widely distributed in multiple centers, especially for medical imaging, where cohort sizes at single institutions are often small.

*Corresponding author: dlrubin@stanford.edu (Daniel Rubin).

Gathering data from multiple centers is often hindered by barriers to data sharing and regulatory and privacy concerns. Collaborative learning (also termed federated learning, distributed learning, or decentralized learning) allows collaboratively training a shared global model on multiple centers while keeping the personal data decentralized, and is thus an attractive alternative (McMahan et al., 2016; Chang et al., 2018; Kairouz et al., 2019).

Recently, collaborative learning has been highlighted as a foundational research area in medical domain (Langlotz et al., 2019). Numerous collaborative learning methods have been proposed in past decades (Su and Chen, 2015; McMahan et al., 2016; Lin et al., 2017; Chang et al., 2018; Vepakomma et al., 2018), such as Federated Averaging (FedAVG) (McMahan et al., 2016) and Cyclical weight transfer (CWT) (Chang et al., 2018). Despite the promising progress, collaborative learning still faces the data heterogeneity challenge, where data across institutions is usually non-independent and identically distributed (non-iid) (Qu et al., 2021; Roth et al., 2020; Dayan et al., 2021; Liu et al., 2021), which impedes the model convergence and cause performance drops. Numerous efforts have been devoted to solving the performance drops on data heterogeneity challenge (Hsu et al., 2019; Hsieh et al., 2019; Balachandar et al., 2020), such as applying different types of optimization heuristic to stabilize the local model update (Li et al., 2020; Hsu et al., 2019, 2020; Wang et al., 2020), and using knowledge distillation to learn a more powerful global model (Gong et al., 2021; Zhang et al., 2021). While effective, these methods may require heuristic parameters tune (Hsu et al., 2019; Hsieh et al., 2019; Li et al., 2020), suffer from usage restriction for layer-wise mapping optimization methods (Zhang et al., 2021), or may not be optimal for highly heterogeneous settings.

In this paper, we propose a novel and flexible generative replay strategy to address the data heterogeneity challenge in collaborative learning and to meet gaps in the prior approaches. Different from traditional methods that directly aggregating the model parameters, we leverage generative adversarial learning (Shin et al., 2017; van de Ven and Tolias, 2018; Choi et al., 2018) to aggregate the knowledge from the data distribution of all the local institutions. Specifically, we develop a novel dual model architecture consisting of a primary classifier for the desired task (e.g., a classification or regression task), and an auxiliary "generative replay model" for either generating artificial examples or extracting latent features from local clients. Our generative replay strategy helps solve the challenge of data heterogeneity in collaborative learning by training the primary model using the aggregated knowledge collected by generative replay model from the heterogenous clients. We further incorporate differential privacy (Abadi et al., 2016) into the training of the auxiliary model to prevent the exposure of sensitive or private patient information.

Our strategy is flexible to deploy, can either be 1) incorporated into a well-deployed collaborative learning methods to improve their capability of handling data heterogeneity across institutions (left image in Fig. 1), or 2) be used to construct a novel and individual collaborative learning framework FedReplay (right image in Fig. 1) for communication efficiency and better privacy protection. When incorporated into an existing collaborative learning framework—for example CWT (Chang et al., 2018)—we first train an auxiliary and universal image generation network on local institutional data, share the generator between local institutions, and then train a standard CWT model based on the local institutional data

and the augmented replayed data from the shared generator (referred to CWT+Replay). The training of auxiliary generator in CWT+Replay is time-consuming and the replayed data may reveal patient privacy. We further apply our generative replay technique to conduct an individual collaborative learning framework, termed FedReplay, to avoid the possibility of revealing patient privacy from the replayed images. Similar to CWT+Replay, FedReplay also has a dual model architecture, a primary model that learns the desired task, and an auxiliary encoder network that extracts latent variables. Our FedReplay is robust to various types of heterogeneity in data across institutions, as the primary model is directly trained on the union of the latent variables collected from the auxiliary generators of all the local institutions, and the latent variables contain most important features in the original data. The main contributions of our paper are summarized as follows:

1. We propose a novel generative replay technique to address the challenge of data heterogeneity among institutions that participate in collaborative learning. As opposed to existing methods that either provide sophisticated ways to control the optimization strategy or share partial global data to mitigate the performance drops from data heterogeneity, our generative replay technique provides a new insight for the development of collaborative learning methods, which is easy to implement, and can be applied to any kind of deep learning task (e.g., classification, regression, etc.).
2. Our generative replay technique is flexible to use. It can either be directly incorporated into a existing federated learning framework to increase their capability of handling data heterogeneity across institutions with minimal modifications (left image in Fig. 1), or be used as a novel and individual collaborative learning framework to reduce communication cost and mitigate privacy cost (right image in Fig. 1).
3. While previous collaborative learning methods require frequent communication between local institutions and the central server, the proposed FedReplay only requires a one-time communication between local institutions and the central server, which is time-efficient. The training of primary model is performed solely on the central server, without restriction to hardware and network speeds among institutions.

2. Related Works

Collaborative Learning

Prior collaborative learning methods can be generally classified into two categories, parallelized collaborative learning methods and serial collaborative learning methods.

Parallelized methods involve training each individual institution on the local institutional data for several iterations/epochs, transferring the gradients/weights from individual institutions to a central server for averaging, and then transferring the averaged weights/gradients back to individual institutions. FedAVG (McMahan et al., 2016) and Federated stochastic gradient descent (FedSGD) (Su and Chen, 2015; McMahan et al., 2016) are two of the most popular parallelized methods. FedAVG (McMahan et al., 2016) involves

frequent transferring of model weights between individual institutions and central server, while FedSGD (Su and Chen, 2015; McMahan et al., 2016) involves frequent transferring of gradients between individual institutions and central server, which is an extreme case of FedAVG (McMahan et al., 2016).

Parallelized methods are computationally efficient due to their parallel processing. However, they may be suboptimal if institutions have very different network connection speeds or deep learning hardware (a common situation among medical institutions). Bonawitz et al. (2019) relaxed the strict system condition by dropping devices that fail to compute the predefined epochs within a specified time window. Beyond the heterogeneity in system, heterogeneity in data across institutions is another critical concern for parallelized methods. The training examples at each local institution are sampled from institution-specific distribution for parallelized methods, which is a biased estimator of the central target distribution if heterogeneity exists. The learned weights in different institutions will diverge severely when high skewness exists in the data, thus the synchronized averaged central model will lose accuracy or even completely diverge (Hsu et al., 2019). Several recent efforts have been devoted to solving the performance drops in statistically data heterogeneous settings. For example, Zhao et al. (2018) proposed FedAVG+Share to improve the performance of FedAVG on non-IID data by distributing a small amount of data for globally sharing between all the institutions. Unlike FedAVG that simply updates the weights on the server, Hsu et al. (2019) accumulated the model updates with momentum and used an exponentially weighted moving average as the model update, to improve its robustness to non-IID data. While effective, these methods either require 1) all the participated institutions are active (Khaled et al., 2019; Wang and Joshi, 2018), 2) additional convexity assumptions (Wang et al., 2019), 3) need sharing partial institutional data (Zhao et al., 2018), or 4) only work well in data distributions with mild heterogeneity (see the comparison results in Fig. 5).

Compared to the parallelized methods, serial methods involve training in a serial and cyclical way, which may be less computationally efficient than parallel processing, but may be more flexible to variations in hardware and network speeds among institutions. CWT (Chang et al., 2018) is one of the typical serial methods, which involves updating weights at one institution at a time, and cyclically transferring weights to the next training institution until convergence. Split learning (SplitNN) (Vepakomma et al., 2018) can be also considered as a serial collaborative learning method. In SplitNN (Vepakomma et al., 2018), each institution trains a partial deep network up to the cut layer, sends the intermediate output feature maps at the cut layer to the server. The server then completes the rest of the training without looking at raw data from clients. In SplitNN, in addition to the model weights, the intermediate feature maps and gradients are communicated between different institutions. Similar to parallelized methods, heterogeneity in data across institutions is also a big concern in serial methods. Serial methods always suffer from catastrophic forgetting problems when heterogeneity exists in data distributions. The model tends to abruptly forget what was previously learned information when it transfers to the next institution. The higher the skewness of heterogeneity in data, the more severe the catastrophic forgetting tends to be in serial methods, thus resulting in performance drops. Balachandar et al. (2020) introduced cyclically weighted loss to mitigate the performance loss for label distribution skewness in CWT, but this classification label based weighted loss only works for image classification

tasks. By contrast, our generative replay strategy is more scalable and can work well on various types of tasks.

Adversarial Attacks

Recent works have shown that collaborative learning is vulnerable to gradient inversion attacks (Zhu and Han, 2020; Huang et al., 2021; Geiping et al., 2020) or model inversion attacks (Yin et al., 2021; Fredrikson et al., 2015; He et al., 2019). Zhu and Han (2020) demonstrated that it was able to reconstruct a client's private data with the shared gradients. But this work is limited to shallow network and low resolution images. Geiping et al. (2020) substantially improved the reconstructed image quality by exploiting a cosine distance loss together with the optimization problem. Similar to gradient inversion attacks, model inversion attacks are first introduced by Fredrikson et al. (2015) and aim to reconstructing private data from the output (or intermediate output) through the inference of a well-trained regression model. Recent works have improved and extended the approach to more complex setting (Yin et al., 2021; Fredrikson et al., 2015; He et al., 2019), e.g., extending to modern DNNs (He et al., 2019) on collaborative learning. In this paper, we study the privacy protection capability of the different collaborative learning methods under gradient inversion attacks and model inversion attacks.

Differential Privacy

Differential privacy is a mathematical system for publicly sharing information while not revealing private information about individuals. With differential privacy, users should learn useful information about a population as a whole but not about a particular individual. Early works on differential privacy mainly focus on convex optimization problems (Chaudhuri et al., 2011; Bassily et al., 2014). In 2016, Abadi et al. (2016) introduced an algorithm with Gaussian mechanism for training non-convex deep learning models with strong differential privacy supporting. They further introduced the moments accountant mechanism to keep track of the overall privacy budget, guaranteeing a tighter estimation for privacy loss compared to the standard composition theorem (Dwork et al., 2010). Mironov (Mironov, 2017) proposed Rényi Differential Privacy (RDP) as a natural relaxation of standard differential privacy for easy composition, and introduced RDP accountant to keep track of the accumulative privacy loss, which further provides tighter bound for privacy loss compared to moments accountant. Several recent works have also applied differential privacy to the training of collaborative learning methods (Shokri and Shmatikov, 2015; Beaulieu-Jones et al., 2018; Li et al., 2019). For example, Li et al. (2019) applied differential privacy techniques to protect the patient data in a federated brain tumour segmentation. Beaulieu-Jones et al. (2018) incorporated the differential privacy into the training of CWT (Chang et al., 2018), which demonstrated provable differential privacy guarantee without large performance sacrifices. In this paper, we incorporate differential privacy into our generative replay strategy and train our auxiliary model in a differentially private manner following the technique described by Abadi et al. (2016). We further apply RDP accountant (Mironov, 2017) for privacy loss analysis.

3. Method

In this section, we 1) first describe the details of how to incorporate the proposed generative replay strategy into existing collaborative learning networks, 2) present the details of the proposed individual collaborative method FedReplay, 3) provide the privacy analysis under different adversarial attacks and illustrate how to improve the patient privacy by deploy our generative replay strategy with differential privacy.

3.1. Optimizing Collaborative Learning Methods with Generative Replay

In this section, we use CWT (Chang et al., 2018) as an example to illustrate how to apply the proposed generative replay strategy to the existing collaborative learning methods. We use an image classification task as the desired task. We apply the Vector Quantized Variational AutoEncoder (VQ-VAE-2) (Razavi et al., 2019) as our auxiliary image generation network. The complete pseudo-code of CWT+Replay is shown in Algorithm 1 and its detailed architecture is shown in Fig. 1.

CWT+Replay consists of a primary model (classifier) for the learned classification task, and an auxiliary generative replay model for synthesizing images that closely resemble input images. In CWT+replay, the auxiliary generator (VQ-VAE-2 (Razavi et al., 2019)) is first trained before the standard CWT training of primal model. VQ-VAE-2 is originated from Vector Quantized Variational AutoEncoder (VQ-VAE) (Van Den Oord et al., 2017), which consists of an encoder for mapping the observations to discrete latent variables, and a decoder for reconstructing observations from these latent variables, and its prior is learnt rather than static. Similar to CWT (Chang et al., 2018), the generator is also trained in a serial way, i.e., first updating generator weights θ_G at one institution at a time, and serially transferring weights to the next training institution. During the training of generator in institution k, the generator aims to learn a mixed data distribution of both the real data $\mathbf{x}_k$ in institution k and the replayed data from previous generator. Here total K institutions are involved in the collaborative learning framework.

Once the generator is trained successfully, we then distribute the decoders, the latent variables[1], and their corresponding labels to all the institutions for standard primary classifier training. Specifically, during the training of the classifier in institution k, the classifier is trained based on both the real data $\mathbf{x}_k$ in institution k and the replayed data from the generator. With the auxiliary replayed data, all the institutions now aim at optimizing the primary classifier based on both the replayed data which resemble the data distribution from other institutions and the institutional specific data, rather than only the institution-specific data distribution, thus do not suffer from catastrophic forgetting as in CWT (Chang et al., 2018).

The application of generative replay strategy to other collaborative learning methods can be also deployed similarly. For example, when incorporating into FedAVG (McMahan et al., 2016), similar to CWT+Replay, we first train an auxiliary generative replay model

[1]We can also train a prior generator with PixelCNN (Oord et al., 2016) to simulate the latent variables, and then distribute the trained PixelCNN instead of the latent variables.

to synthesize images that closely resemble the input images, we then train a standard parallelized FedAVG model based on both the real data from local institutions and the replayed data from the generator.

Algorithm 2: FedReplay. The K institutions are indexed by k. $\mathbf{x}_k$ is training data for institution k. Model weights θ_C, loss function L, learning rate η, and total training epochs N for primary classifier, respectively. E is the auxiliary pre-trained encoder function.

Extract latent variables in institution (k, E) **:**
  ▷ Extract latent variables
  $\mathbf{e}_k \leftarrow E(\mathbf{x}_k)$
  ▷ Return latent variables to server
  return $\mathbf{e}_k$ to server

Server executes:
  ▷ Extracting the latent variables from all the institutions
  for each institution in $k = 1, 2, ..., K$ **do in parallel**
    $\mathbf{e}_k \leftarrow$ **Extract latent variables in institution** (k, E_{top}, E_{bottom})
  ▷ Union of all latent variables
  $\mathbf{e} \leftarrow \{\mathbf{e}_1, \mathbf{e}_2, ..., \mathbf{e}_K\}$
  ▷ Updating primary classifier in server with latent variables
  Initialize parameters θ_C
  for each batch e in $\mathbf{e}$ in total epochs N **do**
    $\theta_C \leftarrow \theta_C - \eta \nabla L(\theta_C; e)$

3.2. The FedReplay Collaborative Learning Approach

Even though CWT+Replay works well on non-IID situations, it still requires frequent transfer of the model weights between local institutions and the central server. In addition, the training of auxiliary image generation network for CWT+Replay is time-consuming and the replayed data may reveal patient privacy. To optimize the communication cost and to prevent privacy leakage from replayed data, we introduce a new collaborative learning approach, FedReplay, based on our generative replay technique in this section. Similar to CWT+Replay, FedReplay also uses a dual model architecture, where an auxiliary encoder extracts the latent variables from local institutions, and a primary model learns the task based on the extracted latent variables.

The framework of our FedReplay is shown in Fig. 1 and its pseudo-code is depicted in Algorithm 2. FedReplay assumes the server has high storage capacity, which allows storage of the relevant latent variables, and training of the primary model based on the extracted latent variables. Our FedReply follows a three-stage training phase: 1) Train a universal and auxiliary encoder network on the data at one of the local institutions and distribute the unique pre-trained encoder network to all of the other institutions; 2) Apply the pre-trained encoder network to local institutions, compress the raw data into latent variables, and upload the latent variables and their corresponding labels to the central server; 3) Train a global primary model purely on the central server with the union of all the latent variables. As we do not need to know the decoder part for data reconstruction in FedReplay, we can simply apply several convolutional and ReLu (the nonlinear rectified linear unit function) layers as the encoder network. In addition, the training of the auxiliary encoder network can be also easily achieved on one of the arbitrary institutions with the learned task (see Section 4.1 for more details).

Our FedReplay is robust to various types of heterogeneity in data across institutions, as the primary model is directly trained on the union of the latent variables from all the institutions, and the latent variables contain most important features in the original data. FedReplay is also robust to the system heterogeneity, since the training of primary model is totally performed on the central sever, and it only requires a one-time communication between institutions and the central server.

3.3. Privacy Analysis

3.3.1. Adversarial Attacks—Recent works have shown that it is able to reconstruct pixel-wise level private data from the shared gradients or latent variables in collaborative learning (He et al., 2019; Zhu and Han, 2020). We evaluate the privacy protection capability of FedReplay by comparing the quality of recovered image in FedReplay and the baseline FedAVG under model inversion attacks (He et al., 2019) and gradient inversion attacks (Geiping et al., 2020), respectively.

FedAVG shares the gradients of the whole network, while FedReplay shares the intermediate latent variables extracted from the auxiliary encoder network. Given a neural network with parameters θ, the real private image x that we aim to recover, the shared gradient $\nabla L_\theta(x)$, and the dummy input $\tilde{x}$, we formulate the image recovery task of FedAVG as an optimization problem following (Geiping et al., 2020):

$$\arg\min_{\tilde{x}}\left(1-\frac{\langle \nabla_\theta L_\theta(\tilde{x}),\, \nabla_\theta L_\theta(x)\rangle}{\|\nabla_\theta L_\theta(\tilde{x})\|\,\|\nabla_\theta L_\theta(x)\|}\right)+\alpha \mathrm{TV}(\tilde{x})\,. \tag{1}$$

The former part aims to match the gradient of the recovered input $\tilde{x}$ with the target transmitted gradient $\nabla_\theta L_\theta(x)$, which is measured as a cosine distance following (Geiping et al., 2020). The latter part regularizes the recovered image with a total variation (Rudin et al., 1992), which encourages $\tilde{x}$ to be piece-wise smooth. The hyperparameter α balances the effects of the two terms.

Similarly, malicious attacker may also be able to recover the private data from the shared latent variables (denoted as $E_\theta(x)$) in FedReplay (He et al., 2019; Zhang et al., 2020). Specifically, we consider the white-box setting for this model inversion attack, as each local client shares the same auxiliary encoder network. We formulate the model inversion attacks as the following optimization problem:

$$\arg\min_{\tilde{x}}\|E_\theta(\tilde{x})-E_\theta(x)\|^2+\alpha \mathrm{TV}(\tilde{x}), \tag{2}$$

where the former part is Euclidean Distance which aims to match the latent variables $E_\theta(\tilde{x})$ of dummy input to the shared latent variables $E_\theta(x)$. Following (He et al., 2019), we apply regularized Maximum Likelihood Estimation to solve this problem. We will qualitatively and quantitatively measure the privacy protection capability of FedAVG and FedReplay according to Eq. 1 and Eq. 2 in the 4.4.2.

3.3.2. Differential Privacy—To provide a privacy guarantee for our generative replay strategy, we incorporate differential privacy (Dwork et al., 2006) into our generative replay strategy and train our auxiliary model in a differentially private manner.

Algorithm 3: Training generator in institution k under differential privacy

Input: Examples data $\mathbf{x}_k = \{x_k^1, x_k^2, ...\}$ in institution k, loss function $L_G(\theta_G)$, train epochs M, batch size b, learning rate η_G, noise scale σ, and gradient clip norm C.

Initialize generator parameters θ_G

for each local batch x_k in $\mathbf{x}_k$ over epochs N **do**

  for each sample x_k^i in batch x_k **do**

    ▷ Compute gradient for each sample

    $\mathbf{g}_t(x_k^i) \leftarrow \nabla L_G(\theta_G; x_k^i)$

    ▷ Clip the ℓ_2 norm of gradient

    $\overline{\mathbf{g}}_t(x_k^i) \leftarrow \mathbf{g}_t / \max(1, \frac{\|\mathbf{g}_t\|_2}{C})$

  ▷ Add Gaussian noise

  $\mathbf{g}_t \leftarrow \frac{1}{b}\sum_i (\overline{\mathbf{g}}_t(x_k^i) + \mathcal{N}(0, \sigma^2 C^2 \mathbf{I}))$

  ▷ Update parameters

  $\theta_G \leftarrow \theta_G - \eta_G \mathbf{g}_t$

Output: θ_G and the computed the overall privacy cost (ε, δ) with RDP (Mironov, 2017)

Differential privacy (Dwork et al., 2006) is formally defined as an algorithm $\mathscr{A}: D \rightarrow Range(\mathscr{A})$ with domain D and the output range $Range(\mathscr{A})$ is (ε, δ)-differential private, if for all neighboring datasets $D_1 \in D$, $D_2 \in D$ that differ only on one individual subject, and for all subset S $\subseteq Range(\mathscr{A})$ the following holds:

$$\mathrm{P_r}[\mathscr{A}(D_1) \in S] \leq \exp(\varepsilon) \cdot \mathrm{P_r}[\mathscr{A}(D_2) \in S] + \delta\,. \tag{3}$$

Here smaller ε indicates strong bind for the above constraints and provides strong privacy guarantees, but usually accompanied with decreased performance. δ relaxes the inequality by allowing a δ additive probability term, which is preferably set to smaller than $1/|D_1|$. $|D_1|$ indicates the train size of dataset D_1. The differential privacy guarantees that the algorithm $\mathscr{A}$ on two neighboring datasets D_1, D_2 will produce similar results, thus ensuring that the algorithm $\mathscr{A}$ learns from the population but not from the individuals.

We follow the work of (Abadi et al., 2016) to protect the privacy of the training data in each institution with Gaussian mechanisms, i.e., clipping the gradient norm and adding with Gaussian noise in each stochastic gradient descent computation. Algorithm 3 illustrates the detailed privacy training procedure of the generator with parameters θ_G and loss function $L_G(\theta_G)$. In each batch size training step, we first 1) compute the gradient $\mathbf{g}_t\left(x_k^i\right) = \nabla L_G\left(\theta_G; x_k^i\right)$ for each individual sample x_k^i, 2) clip the ℓ_2 norm of each gradient, i.e., replacing the gradient vector $\mathbf{g}_t$ with $\mathbf{g}_t / \max\left(1, \frac{\|\mathbf{g}_t\|_2}{C}\right)$ for a given clipping threshold C, 3) add the gradient with Gaussian noise (noise scale σ), and then 4) update parameters θ_G with the calculated gradients. We continue the training until model convergence or we approach the specified privacy budget. Here we only describe how we provide privacy supporting for the training of the auxiliary generator, the training procedure of the primary classifier and the auxiliary encoder network (for FedReplay) with privacy supporting can be also deployed similarly.

We apply RDP (Mironov, 2017) to keep track of the cumulative privacy loss parameters (ε, δ) during the training stage. The privacy accountant in RDP (Mironov, 2017) takes the sub-sampling rate ($b/|D|$), and the applied noise scale σ as input, and is also dependent on the overall training epochs.

4. Experiments

We applied our generative replay approach to CWT (Chang et al., 2018), and compared the proposed CWT+Replay and FedReplay, with several state-of-the-art collaborative learning methods (Abadi et al., 2016; Chang et al., 2018; Vepakomma et al., 2018; Hsu et al., 2019; Zhao et al., 2018). In this section, we will describe the studied dataset, detailed simulated data partitions, the compared methods, and quantitative results to demonstrate the effectiveness of the proposed generative replay strategy. A privacy analysis for the proposed generative replay strategy is also provided at the end of this section.

4.1. Experimental Setup

Dataset.—We evaluated the performance of different collaborative learning methods on both medical image classification and regression tasks.

- **Image classification task on Diabetic Retinopathy (Retina) Dataset** (Kaggle, 2017). Retina dataset consists of 17563 pairs of right and left color digital retinal fundus images. Each image was rated as a scale of 0 to 4 according to the presence of diabetic retinopathy, where 0 to 4 indicates no, mild, moderate, severe and proliferative diabetic retinopathy, respectively. For simplicity, the labels were binarized to Healthy (scale 0) and Diseased (scale 2, 3 or 4) in our study, while the mild diabetic retinopathy images (scale 1) between healthy and diseased status were excluded. Additionally, only left eye images were used, to remove the confusion from using multiple images for the same patient. We randomly selected 6000 images (3000 positive (Healthy) and 3000 negative (Diseased) images) for training, 3000 images (1500 positive and 1500 negative images) for validation, and 3000 images (1500 positive and 1500 negative images) for testing. We applied a similar technique that Kaggle Diabetic Retinopathy Competition winner Benjamin Graham proposed (Graham, 2015) to pre-process these images, i.e., first rescaling images to the same eye radius of 300, subtracting the local average color of each image, then cropping to remove the image boundaries, and finally resizing to image resolution 256×256 to serve as the input of deep neural network. Similar to (Chang et al., 2018), the prediction accuracy was applied as the evaluation metric.
- **Image recognition task on RSNA Pediatric Bone Age Prediction with X-Ray image (BoneAge dataset)** (Halabi et al., 2019). BoneAge dataset is used to estimate the bone age of pediatric patients based on radiographs of their hand. It consists of 14236 (mean age, 127 months) labeled and deidentified hand radiographs. Only male hands were used in our study, to remove the discrepancy between male and female hands. Specifically, we randomly selected 1000 images for validation, used 4575 images for training, and applied the original test dataset

from BoneAge dataset as the global test dataset. The image resolution of the bone age images were resized to 256 × 256, and their intensity values were normalized to [0,1]. The mean absolute distance in months was used as the evaluation metric.

Data Partitions.—We simulated three sets of data partitions on both Retina dataset (Kaggle, 2017) and BoneAge dataset (Halabi et al., 2019), i.e., IID data partitions, non-IID data partitions with mild degree of label distribution skewness, and non-IID data partitions with high degree of label distribution skewness. We simulated 4 local institutions in each set of experiments. The detailed data partitions are shown in Fig. 2 and Fig. 3. We applied the mean Kolmogorov-Smirnov (KS) statistics between every two institutions to measure the degree of label distribution skewness. KS=0 means IID data partitions, while KS=1 indicates identically different label distributions across institutions.

Comparison Methods.—We compared our CWT+Replay and FedReplay with the following baseline methods for collaborative learning:

- FedAVG (McMahan et al., 2016), one of the most popular parallelized methods for collaborative learning. We set the fractions of institutions that were selected for computation in each round to 1, i.e., all the local institutions were involved in the computation in each round. In addition, we set the number of iterations trained on local institution to $\lfloor |D_k|/b \rfloor$ in each round, where $|D_k|$ is the quantity of training samples in local institution k and b is the local minibatch size.
- CWT (Chang et al., 2018), a type of serial collaborative learning. Following (Balachandar et al., 2020), we set the number of iterations trained on local institution k to $\lfloor |D_k|/b \rfloor$ to prevent performance drops from sample size variability.
- SplitNN (Vepakomma et al., 2018), can be considered as a serial collaborative learning. We choose the peer-to-peer mode version in (Vepakomma et al., 2018) for collaborative training, where one institution first starts one round of training, then cyclically sends the partial of network model weights to the next institution for another round of training until model convergence. In SplitNN, in addition to the model weights, the intermediate feature maps and gradients are also communicated between the institutions and the central server. Similar to SplitNN, our CWT+Replay and FedReplay also require transferring the intermediate feature maps (latent variables) between the institutions and the central server.
- FedAVGM (Hsu et al., 2019), an optimization method for FedAVG (McMahan et al., 2016), which applies momentum optimizer on the server (Hsu et al., 2019) to improve its robustness to non-IID data.
- FedAVG+Share (Zhao et al., 2018), an optimization method for FedAVG (McMahan et al., 2016). FedAVG+Share aims to improve the performance of FedAVG on non-IID data by globally sharing a small amount of data between all

the institutions. Following (Zhao et al., 2018), we distributed 5% globally shared data among each institution in our experiments.

- FedProx (Li et al., 2020), an optimization method for FedAVG (McMahan et al., 2016), which introduces a proximal term to the local objective to help stabilize the model training. We tune the penalty constant μ in the proximal term from the candidate set {0.001, 0.01, 0.1, 1} on Split-2 for both Retina and BoneAge datasets, and apply the same penalty constant for all the remaining data partitions. Specifically, μ is set to 0.001 for both Retina and BoneAge dataset.

Implementation Details.—All the methods described in this paper were implemented with Pytorch and optimized with SGD. We used the ResNet34 (He et al., 2016) as the backbone network architecture for both the classification and regression tasks in all collaborative learning methods. For SplitNN (Vepakomma et al., 2018), we applied the data partitions with Split 2 as example, and run multiple experiments on both Retina (Kaggle, 2017) and BoneAge dataset (Halabi et al., 2019) to choose the optimized cut layer. Specifically, conv1 was applied as the cut layer since it provides the best performance in our experiments. We set batch size b to 32, the learning rate η to 0.001 and progressively decreased with scale 10 every 35 epochs. For auxiliary generator of CWT+Replay: we followed work in (Razavi et al., 2019) to set the hyper parameters for VQ-VAE-2. All the images in VQ-VAE-2 were pre-processed to resolution 256×256, and thus the latent layers of VQ-VAE-2 were with resolution 32 × 32 and 64 × 64 for top levels and bottom levels, respectively. Unlike in (Razavi et al., 2019), we set the batch size to 32, and the learning rate η_G to 0.0004 (halved every 150 epochs). Adam (Kingma and Ba, 2014) was applied as the optimizer.

We used ResNet34 (He et al., 2016) as the backbone network for FedReplay. Shown in Fig. 4, we split the ResNet34 (He et al., 2016) into two parts, where the first part (consisting of first convolutional, batch normalization, ReLU and max pooling layer) was used as the encoder network, while the remaining part was used as the primary classifier network. The training of the encoder network is achieved on one of the arbitrary institution with the institutional dataset and the desired task. We applied the whole ResNet34 (He et al., 2016) to help train the encoder network. Specifically, we trained the ResNet34 (He et al., 2016) with the attached institutional dataset on the learned task (classification task for Retina (Kaggle, 2017) dataset, and regression task for BoneAge dataset (Halabi et al., 2019)) until model convergence. The first part of the trained ResNet34 (He et al., 2016) were then used as the auxiliary encoder network. The training parameters for the auxiliary encoder network and primary classifier were the same to the primary classifier model in CWT+Replay.

4.2. Performance Evaluation

In this section, we evaluated the performance of different collaborative learning methods, assessing prediction accuracy for Retina dataset (Kaggle, 2017), and mean absolute error (MAE) loss for BoneAge dataset (Halabi et al., 2019). The performance on the centrally hosted data was applied as the benchmark performance.

As shown in Fig. 5, all the compared collaborative learning methods show comparable performance to the benchmark centrally hosting method in the case where data at different institutions is homogenous (Split 1). However, the performance of the standard CWT (Chang et al., 2018), SplitNN (Vepakomma et al., 2018), and FedAVG (McMahan et al., 2016) drop significantly with the increasing degree of label distribution skewness in data across institutions. For example, the MAE loss of CWT (Chang et al., 2018), SplitNN (Vepakomma et al., 2018), and FedAVG (McMahan et al., 2016) on BonAge dataset (Halabi et al., 2019) increase from 6.88 ± 0.04, 6.80 ± 0.05, 7.26±0.07 on data Split 1 to 27.88±2.82, 30.28±1.47, 30.87±0.07 on data Split 3, respectively.

The serial collaborative learning methods, such as CWT (Chang et al., 2018) and SplitNN (Vepakomma et al., 2018), always suffer from catastrophic forgetting when heterogeneity exists in data distributions across institutions. The model tends to abruptly forget what was previously learned information when it transfers to next institution. For example, in the Split 3 experiment of Retina dataset (Kaggle, 2017), when transferring the model trained in Inst2 to the next Inst3, the model learned in Inst 3 tends to forget what was previously learned information from Inst2, and the prediction accuracy on Inst2 is dropped from original 87.7% to 69.0% (see the second row in Table 1). While for parallelized methods, such as FedAVG (McMahan et al., 2016), the data used for training at each local institution is sampled from an institution-specific distribution, which is a biased estimator of the central target distribution if heterogeneity exists. The learned weights in different institutions will diverge severely when high skewness exists in data across institutions, and the synchronized averaged central model will lose accuracy or even completely diverge (Zhao et al., 2018). The existing optimized methods, such as FedAVGM (Hsu et al., 2019), FedProx (Li et al., 2020) and FedAVG+Share (Zhao et al., 2018) may help alleviate the model divergence problem, but they still suffer from performance drops in highly skewed non-IID data partitions, such as the MAE loss 21.81 ± 3.71 and 15.63 ± 3.43 of FedAVGM (Hsu et al., 2019) and FedAVG+Share (Zhao et al., 2018) on Split 3, compared to the MAE loss 7.49 ± 0.032, and 7.99 ± 0.14 on homogenous Split 1. As a comparison, our generative replay strategy helps generate synthetic images closely resembling the studied participants (CWT+Replay) or provide accessible to the union of the latent variables (FedReplay), thus can avoid the performance drops even on highly skewed heterogeneity cases.

4.3. Application to a Real-World Federated Dataset

We further evaluated our method on a real-world federated dataset, International Brain Tumor Segmentation (BraTS) 2017 challenge (Menze et al., 2014; Bakas et al., 2017), and compared it to the state-of-the-art collaborative learning methods CWT (Chang et al., 2018), FedAVG (McMahan et al., 2016), and FedAVG+Share (Zhao et al., 2018). The performance on the centrally hosted data was applied as the benchmark performance.

BraTs 2017 is a multi-institutional pre-operative multimodal MRI scans of glioblastoma (GBM/HGG) and lower grade glioma (LGG). Each subject consists of four modal MRI scans: a) native (T1), b) post-contrast T1-weighted, c) T2-weighted, and d) T2 Fluid Attenuated Inversion Recovery (T2-FLAIR) volumes, and an associated voxel-level annotation of “tumor core”, “enhancing tumor”, and “whole tumor”. We used HGG patients

in our study, applied the T2-FLAIR modality as the input, and evaluated on the binary whole tumor segmentation task. We used the training dataset of BraTs 2017 (consists of 210 subjects collected from 10 different institutions[2]) to form our institutional training and test datasets. Specifically, we random selected 45 subjects as the test dataset, used the remaining 165 subjects for the training dataset. We run with three trials and applied the Dice Similarity Coefficient (DSC) (Dice, 1945) to measure similarity between the predicted results and the ground truth mask label.

We used U-net (Ronneberger et al., 2015) pre-trained on brain MRI segmentation dataset (Buda et al., 2019) as the baseline network. U-net consists of four levels of blocks and a bottleneck layer with 512 convolutional filters. Each block contains an encoding part with two convolutional layers (followed by batch normalization and ReLU activation function) and one max pooling layer, and a decoding part with two convolutional layers (followed by batch normalization and ReLU activation function) and an up-convolutional layer. Skip connections are applied between each encoding layer and its corresponding layer in the decoding part. For FedReplay, we split the U-net at the first encoding block, applied the first encoding as the auxiliary encoder network, used the remaining part as the primary classifier. We trained the auxiliary encoder on a institution that contains the largest number of subjects, and then distributed the auxiliary encoder to all the rest institutions for feature extraction. We extracted three consecutive axial slices as the input, all the images were resized to 256 × 256 and then cropped to 224 × 224. All our experiments used a batch size of 32 and learning rate 1e-4 with Adam optimizer. All the models were trained with total 35 communication rounds.

We report mean DSC over 10 institutions trained with CWT, FedAVG, FedAVG+Share, and the proposed FedReplay on BraTs 2017 in Figure 6. FedReplay outperforms all the competing collaborative learning methods on the real-world federated dataset, achieves comparable performance to the benchmark method (Central) on the centrally hosted data, i.e., 85.4% of our FedReplay as a comparison to 85.3% of Central.

4.4. Privacy Analysis

4.4.1. Privacy Analysis with Differential Privacy—In this section, we applied CWT+Replay as an example to demonstrate the capability of the proposed strategy in providing a provable differential privacy guarantee without too much sacrifice in performance. We leveraged RDP (Mironov, 2017) to keep track of the cumulative privacy loss parameters (ε, δ) during the training stage. In RDP (Mironov, 2017), the privacy accountant is directly relevant to the sub-sampling rate ($b/|D|$), the applied noise rate σ, and the total training epochs. In our experiments, we set the noise rate to 0.7. Fig. 7 depicts the ε value as a function of epochs for different train sample sizes and various target δ. Here we set the target δ to $1/|D|$ following (Abadi et al., 2016).

As seen in Fig. 7, with larger institutional training sample size, e.g., when $|D| = 10000$, a maximum ε value of 5 allows for 300 epochs of training and $\delta = 1^{-5}$, while with decreasing training sample size, the privacy accountant is also increasing. The training sample size

[2]We thank Spyridon Bakas, one of the authors of BraTs challenge, for providing real institutional details of BraTs 2017.

$|D|$ in our BoneAge prediction experiment (Halabi et al., 2019) for each institution is 1144, and thus $\delta = 8.7^{-4}$. We set the maximum training epochs to 225 in our BoneAge image generation task. Thus, when a maximum δ of value 8.7^{-4} was required, we obtained an ε value of 17.8 in our image generation task. Table 2 shows the MAE loss of the CWT+Replay with and without differential privacy on Split 3 of Bone Age dataset (Halabi et al., 2019), respectively. For fair comparison, we only applied differential privacy on the training of our image synthesis network. Incorporated with differential privacy, our image generation network is able to synthesize images that resembling the studied participants while not revealing information from the real studied participants. Table 2 indicates that our CWT+Replay still achieves promising results even when the auxiliary image synthesis network is developed under a provable differential privacy guarantee, e.g., 9.84±0.02 for our result compared to the state-of-the-art result 15.63 ± 3.43 on Split 3 of FedAVG+Share (Zhao et al., 2018).

4.4.2. Data Leakage—We further evaluate the privacy protection capability of FedReplay by comparing the quality of the recovered images from FedAVG and FedReplay according to Eq. 1 and Eq. 2. We apply the Peak Signal-to-Noise Ratio (PSNR), a ratio between the maximum value (power) of an image and the mean squared error between the target image and recovered image. We measure the PSNR of the reconstruction of 224 × 224 Retina images over a random selected 50 images of the test dataset. The higher the PSNR value, the better the reconstructed image quality. We apply AdamW to optimization Eq. 1 and Eq. 2, and all optimization runs up to 20,000 iterations.

We use ResNet34 as the baseline network. ResNet34 consists of several sequential Residual blocks and can be listed as {Conv1, Layer1, Layer2, Layer3, Layer4, FC}. We evaluate three sets of auxiliary encoder network (extracted from ResNet34) for FedReplay: 1) Conv1: a Convolutional layer following up a ReLu, a batch normalization and a max pooling layer, 2) Layer2: Conv1 with two residual blocks, and 3) Layer 3: Conv1 with three Residual blocks.

Table 3 compares the prediction accuracy and the privacy protection capability of FedAVG and the proposed FedReplay on data partition Split-3 of Retina dataset. It is not easy to reconstruct a recognizable image from both FedAVG and FedReplay under current attacker technique when the input image has high resolution(224×224), as the low PSNR shown in Table 3 and no clear patterns shown in Fig. 8. In addition, the flexibility design of the dual model architecture of our FedReplay allows the room for better privacy protection capability by applying a deeper network as the auxiliary encoder while only slightly impeding the final prediction performance. See the nearly unrecognizable results from Layer3 shown in Fig. 7 and the decent prediction performance 75.00% shown in Table 3 (compared to 70.65% of FedAVG).

5. Conclusion

In this paper, we present a novel generative replay strategy to address the challenge of data heterogeneity across institutions in collaborative learning. Our generative replay strategy is flexible to use. It can either be incorporated into existing collaborative learning methods to improve their capability of handling data heterogeneity across institutions, or be used as a

novel and separate collaborative learning framework (FedReplay) to reduce communication cost and mitigate the privacy cost. Unlike existing methods that either design sophisticated ways to control the optimization strategy or share partial global data to mitigate the performance drops from data heterogeneity across institutions, our generative replay strategy with differential privacy supporting is a flexible and straightforward alternative solution, which provides new insight for the development of collaborative learning methods in real applications.

One concern of our generative replay technique is that the task performance is heavily dependent on the quality of the generator when it is applied to existing collaborative learning methods. However, this is also largely alleviated by the progress of training generative models. In addition, our FedReplay relaxes this constraint by training a primary model based on the extracted latent variables, and thus do not need to train a high quality generator for synthesizing images that closely resembles the input images. As a contrast, the training of the auxiliary encoder network for FedReplay can be easily achieved on one of the arbitrary institutions with the learned task.

Acknowledgment

This work was supported in part by a grant from the NCI, U01CA242879.

References

Abadi M, Chu A, Goodfellow I, McMahan HB, Mironov I, Talwar K, Zhang L, 2016. Deep learning with differential privacy. In: ACM on CCS. pp. 308–318.

Bakas S, Akbari H, Sotiras A, Bilello M, Rozycki M, Kirby JS, Freymann JB, Farahani K, Davatzikos C, 2017. Advancing the cancer genome atlas glioma mri collections with expert segmentation labels and radiomic features. Scientific data 4 (1), 1–13.

Balachandar N, Chang K, Kalpathy-Cramer J, Rubin DL, 2020. Accounting for data variability in multi-institutional distributed deep learning for medical imaging. J. Am. Med. Inform. Assoc

Bassily R, Smith A, Thakurta A, 2014. Private empirical risk minimization: Efficient algorithms and tight error bounds. In: Proc. IEEE FOCS IEEE, pp. 464–473.

Beaulieu-Jones BK, Yuan W, Finlayson SG, Wu ZS, 2018. Privacy-preserving distributed deep learning for clinical data. arXiv preprint arXiv:1812.01484.

Bonawitz K, Eichner H, Grieskamp W, Huba D, Ingerman A, Ivanov V, Kiddon C, Konečnỳ J, Mazzocchi S, McMahan HB, et al. , 2019. Towards federated learning at scale: System design. arXiv preprint arXiv:1902.01046.

Buda M, Saha A, Mazurowski MA, 2019. Association of genomic subtypes of lower-grade gliomas with shape features automatically extracted by a deep learning algorithm. Computers in biology and medicine 109, 218–225. [PubMed: 31078126]

Chang K, Balachandar N, Lam C, Yi D, Brown J, Beers A, Rosen B, Rubin DL, Kalpathy-Cramer J, 2018. Distributed deep learning networks among institutions for medical imaging. J. Am. Med. Inform. Assoc 25 (8), 945–954. [PubMed: 29617797]

Chaudhuri K, Monteleoni C, Sarwate AD, 2011. Differentially private empirical risk minimization. J. Mach 12 (3).

Choi Y, Choi M, Kim M, Ha J-W, Kim S, Choo J, 2018. Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. In: Proc. IEEE CVPR pp. 8789–8797.

Dayan I, Roth HR, Zhong A, Harouni A, Gentili A, Abidin AZ, Liu A, Costa AB, Wood BJ, Tsai C-S, et al. , 2021. Federated learning for predicting clinical outcomes in patients with covid-19. Nature medicine 27 (10), 1735–1743.

Dice LR, 1945. Measures of the amount of ecologic association between species. Ecology 26 (3), 297–302.

Dwork C, McSherry F, Nissim K, Smith A, 2006. Calibrating noise to sensitivity in private data analysis. In: TCC. Springer, pp. 265–284.

Dwork C, Rothblum GN, Vadhan S, 2010. Boosting and differential privacy. In: IEEE on FOCS. IEEE, pp. 51–60.

Fredrikson M, Jha S, Ristenpart T, 2015. Model inversion attacks that exploit confidence information and basic countermeasures. In: Proceedings of the 22nd ACM SIGSAC conference on computer and communications security. pp. 1322–1333.

Geiping J, Bauermeister H, Dröge H, Moeller M, 2020. Inverting gradients–how easy is it to break privacy in federated learning? arXiv preprint arXiv:2003.14053.

Gong X, Sharma A, Karanam S, Wu Z, Chen T, Doermann D, Innanje A, October 2021. Ensemble attention distillation for privacy-preserving federated learning. In: Proc. IEEE ICCV pp. 15076–15086.

Graham B, 2015. Kaggle diabetic retinopathy detection competition report. University of Warwick.

Halabi SS, Prevedello LM, Kalpathy-Cramer J, Mamonov AB, Bilbily A, Cicero M, Pan I, Pereira LA, Sousa RT, Abdala N, et al. , 2019. The rsna pediatric bone age machine learning challenge. Radiology 290 (2), 498–503. [PubMed: 30480490]

He K, Zhang X, Ren S, Sun J, 2016. Deep residual learning for image recognition. In: Proc. IEEE CVPR pp. 770–778.

He Z, Zhang T, Lee RB, 2019. Model inversion attacks against collaborative inference. In: Proceedings of the 35th Annual Computer Security Applications Conference. pp. 148–162.

Hsieh K, Phanishayee A, Mutlu O, Gibbons PB, 2019. The non-iid data quagmire of decentralized machine learning. arXiv preprint arXiv:1910.00189.

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

Hsu T-MH, Qi H, Brown M, 2020. Federated visual classification withreal-world data distribution. In: Proc. ECCV Springer International Publishing, Cham, pp. 76–92.

Huang Y, Gupta S, Song Z, Li K, Arora S, 2021. Evaluating gradient inversion attacks and defenses in federated learning 34.

Kaggle, 2017. Diabetic retinopathy detection. https://www.kaggle.com/c/diabetic-retinopathy-detection.

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

Khaled A, Mishchenko K, Richtárik P, 2019. First analysis of local gd on heterogeneous data. arXiv preprint arXiv:1909.04715.

Kingma DP, Ba J, 2014. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.

Langlotz CP, Allen B, Erickson BJ, Kalpathy-Cramer J, Bigelow K, Cook TS, Flanders AE, Lungren MP, Mendelson DS, Rudie JD, et al. , 2019. A roadmap for foundational research on artificial intelligence in medical imaging: from the 2018 nih/rsna/acr/the academy workshop. Radiology 291 (3), 781–791. [PubMed: 30990384]

Li T, Sahu AK, Zaheer M, Sanjabi M, Talwalkar A, Smith V, 2020. Federated optimization in heterogeneous networks. Proceedings of Machine Learning and Systems 2, 429–450.

Li W, Milletarì F, Xu D, Rieke N, Hancox J, Zhu W, Baust M, Cheng Y, Ourselin S, Cardoso MJ, et al., 2019. Privacy-preserving federated brain tumour segmentation. In: Proc. MLMI Springer, pp. 133–141.

Lin Y, Han S, Mao H, Wang Y, Dally WJ, 2017. Deep gradient compression: Reducing the communication bandwidth for distributed training. arXiv preprint arXiv:1712.01887.

Liu Q, Chen C, Qin J, Dou Q, Heng P-A, 2021. Feddg: Federated domain generalization on medical image segmentation via episodic learning in continuous frequency space. In: Proc. IEEE CVPR pp. 1013–1023.

McMahan HB, Moore E, Ramage D, Hampson S, et al. , 2016. Communication-efficient learning of deep networks from decentralized data. arXiv preprint arXiv:1602.05629.

Menze BH, Jakab A, Bauer S, Kalpathy-Cramer J, Farahani K, Kirby J, Burren Y, Porz N, Slotboom J, Wiest R, et al. , 2014. The multimodal brain tumor image segmentation benchmark (brats). IEEE Trans. on medical imaging 34 (10), 1993–2024. [PubMed: 25494501]

Mironov I, 2017. Rényi differential privacy. In: Proc. CSF IEEE, pp. 263–275.

Oord A. v. d., Kalchbrenner N, Kavukcuoglu K, 2016. Pixel recurrent neural networks. arXiv preprint arXiv:1601.06759.

Qu L, Balachandar N, Rubin DL, 2021. An experimental study of data heterogeneity in federated learning methods for medical imaging. arXiv preprint arXiv:2107.08371.

Razavi A, van den Oord A, Vinyals O, 2019. Generating diverse high-fidelity images with vq-vae-2. In: Advances in Neural Information Processing Systems. pp. 14866–14876.

Ronneberger O, Fischer P, Brox T, 2015. U-net: Convolutional networks for biomedical image segmentation. In: Proc. MICCAI Springer, pp. 234–241.

Roth HR, Chang K, Singh P, Neumark N, Li W, Gupta V, Gupta S, Qu L, Ihsani A, Bizzo BC, et al., 2020. Federated learning for breast density classification: A real-world implementation. In: Domain Adaptation and Representation Transfer, and Distributed and Collaborative Learning. Springer, pp. 181–191.

Rudin LI, Osher S, Fatemi E, 1992. Nonlinear total variation based noise removal algorithms. Physica D: nonlinear phenomena 60 (1–4), 259–268.

Shin H, Lee JK, Kim J, Kim J, 2017. Continual learning with deep generative replay. pp. 2990–2999.

Shokri R, Shmatikov V, 2015. Privacy-preserving deep learning. In: Proc. ACM SIGSAC pp. 1310–1321.

Su H, Chen H, 2015. Experiments on parallel training of deep neural network using model averaging. arXiv preprint arXiv:1507.01239.

van de Ven GM, Tolias AS, 2018. Generative replay with feedback connections as a general strategy for continual learning. arXiv preprint arXiv:1809.10635.

Van Den Oord A, Vinyals O, et al., 2017. Neural discrete representation learning. pp. 6306–6315.

Vepakomma P, Gupta O, Swedish T, Raskar R, 2018. Split learning for health: Distributed deep learning without sharing raw patient data. arXiv preprint arXiv:1812.00564.

Wang H, Yurochkin M, Sun Y, Papailiopoulos D, Khazaeni Y, 2020. Federated learning with matched averaging. arXiv preprint arXiv:2002.06440.

Wang J, Joshi G, 2018. Cooperative sgd: A unified framework for the design and analysis of communication-efficient sgd algorithms. arXiv preprint arXiv:1808.07576.

Wang S, Tuor T, Salonidis T, Leung KK, Makaya C, He T, Chan K, 2019. Adaptive federated learning in resource constrained edge computing systems. IEEE J. Sel. Areas Commun 37 (6), 1205–1221.

Yin H, Mallya A, Vahdat A, Alvarez JM, Kautz J, Molchanov P, 2021. See through gradients: Image batch recovery via gradinversion. In: Proc. IEEE CVPR pp. 16337–16346.

Zhang L, Luo Y, Bai Y, Du B, Duan L-Y, October 2021. Federated learning for non-iid data via unified feature learning and optimization objective alignment. In: Proc. IEEE ICCV pp. 4420–4428.

Zhang Y, Jia R, Pei H, Wang W, Li B, Song D, 2020. The secret revealer: Generative model-inversion attacks against deep neural networks. In: Proc. IEEE CVPR pp. 253–261.

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

Zhu L, Han S, 2020. Deep leakage from gradients. In: Federated learning. Springer, pp. 17–31.

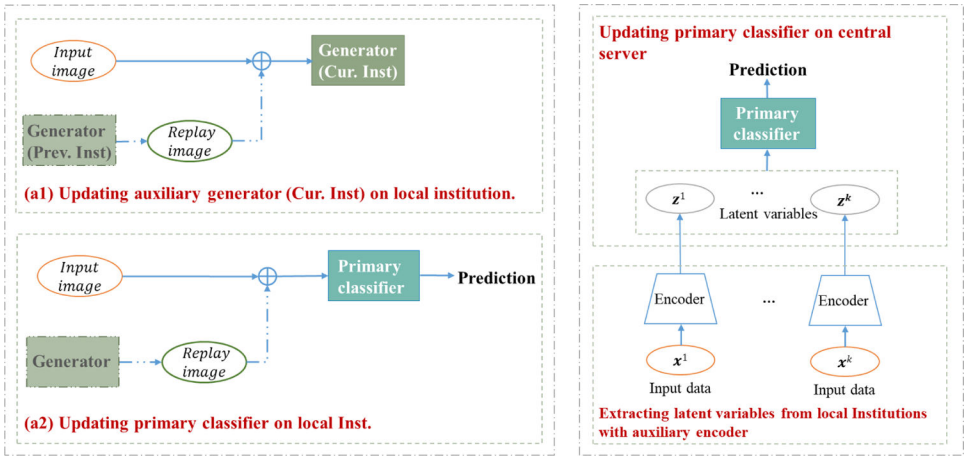
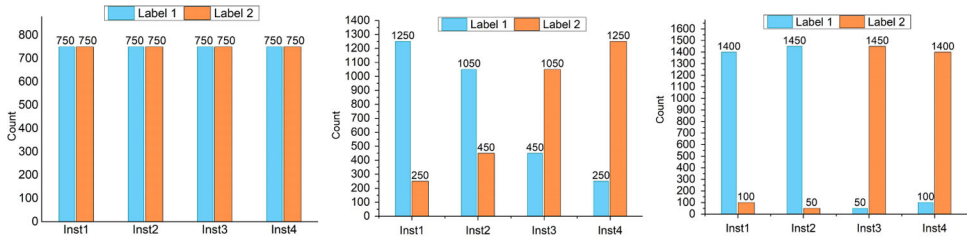


Figure 1:

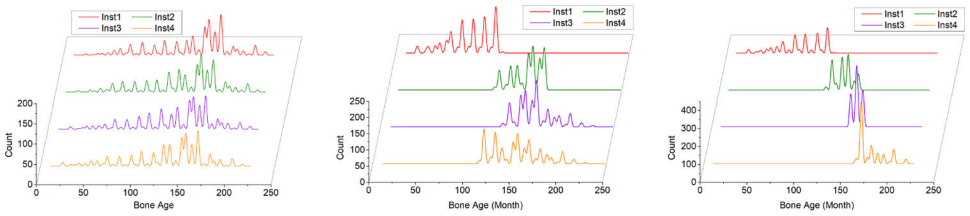
Two ways of applying the proposed generative replay strategy to address the challenge of data heterogeneity among institutions that participate in collaborative learning. We apply an image classification task as the desired task. Left: Illustration of incorporating the proposed generative replay strategy into existing collaborative method CWT (Chang et al., 2018) (referring to CWT+Replay). Right: Illustration of applying the proposed generative replay strategy to construct an individual collaborative learning framework FedReplay.



(a) Split 1, KS-0 (b) Split 2, KS-0.40 (c) Split 3, KS-0.61

Figure 2:

Three sets of simulated data partitions on Retina dataset (Kaggle, 2017). Here labels 1 and 2 indicate positive (Healthy) and negative (Diseased) labels, respectively. Large KS indicates high degree of label distribution skewness.



(a) Split 1, KS-0.02 (b) Split 2, KS-0.63 (c) Split 3, KS-0.97

Figure 3:

Three sets of simulated data partitions on BoneAge dataset (Halabi et al., 2019). The histogram of labels for each institution is shown.

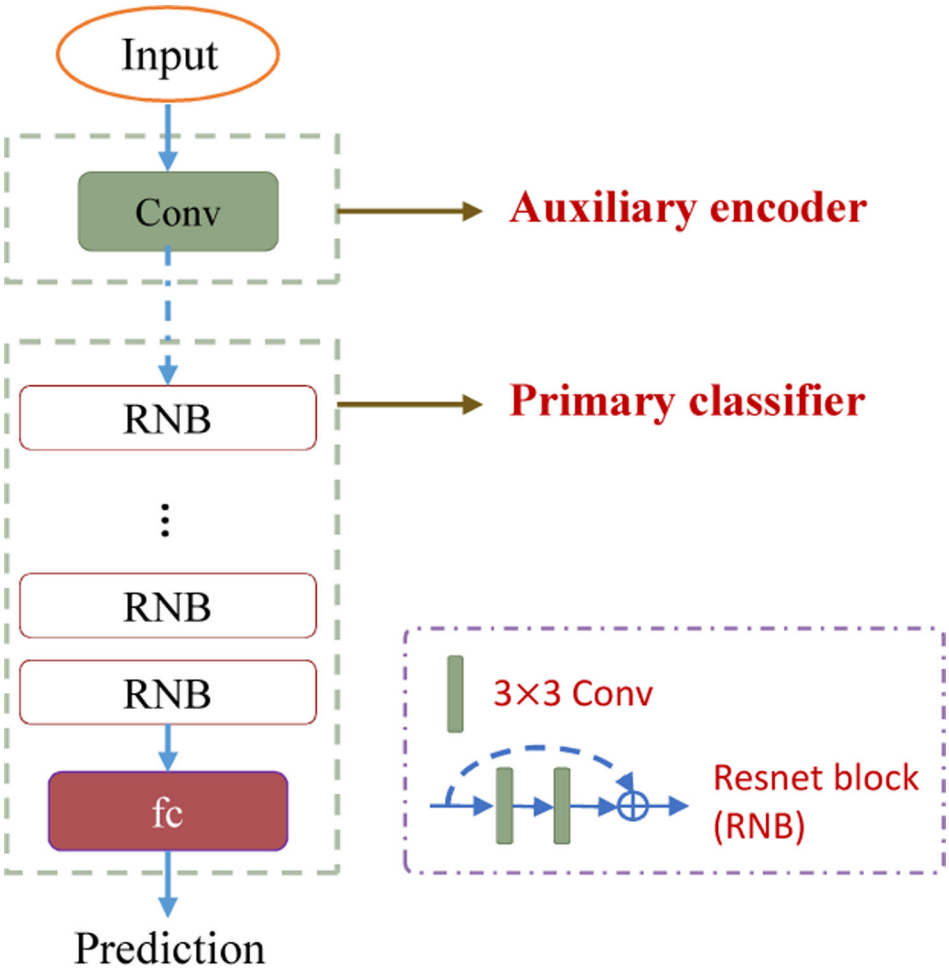
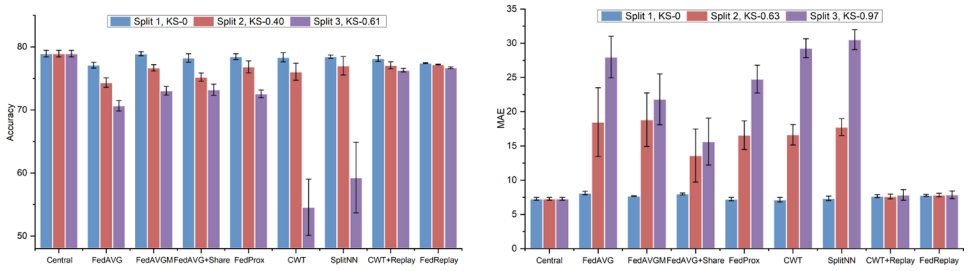


Figure 4:

Detailed network architecture of FedReplay when ResNet34 (He et al., 2016) is used as the backbone network. We omit ReLu, batch normalization and max pooling layers that follow each convolution layer. We split the ResNet34 (He et al., 2016) into two parts, where the first part was used as the auxiliary encoder network, and the remaining part was used as the primary classifier network.



(a) Predicted accuracy on Retina dataset (b) Predicted MAE on BoneAge dataset

Figure 5:

Performance evaluation on Retina dataset (Kaggle, 2017) and BoneAge dataset (Halabi et al., 2019). Mean and standard deviation test accuracies (the higher the better) and MAE loss (the lower the better) were obtained with 4 runs. We use the same setting for the following experiments.

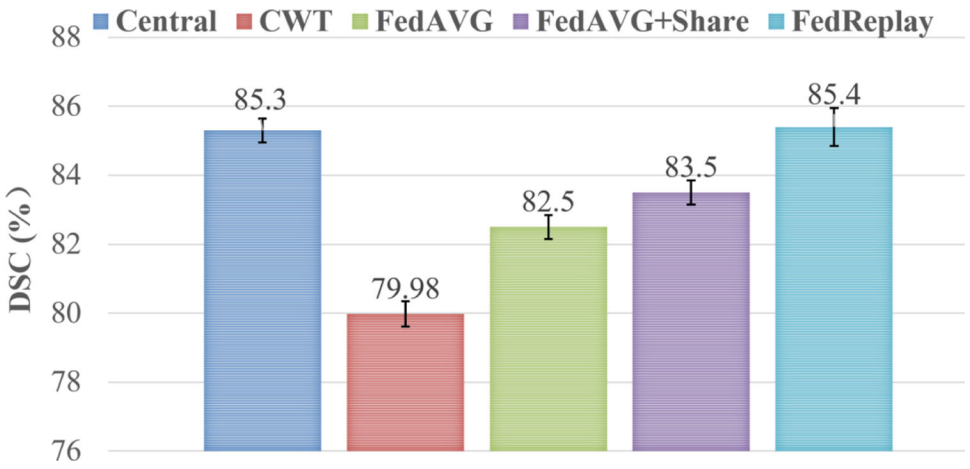


Figure 6:

Mean predicted dice similarity coefficient (DSC) over 10 institutions trained with CWT, FedAVG, FedAVG+Share, and the proposed FedReplay on BraTs 2017, as a comparison to the benchmark method (Central) on the centrally hosted data. Our FedReplay outperforms all the competing collaborative learning methods on the real-world federated dataset.

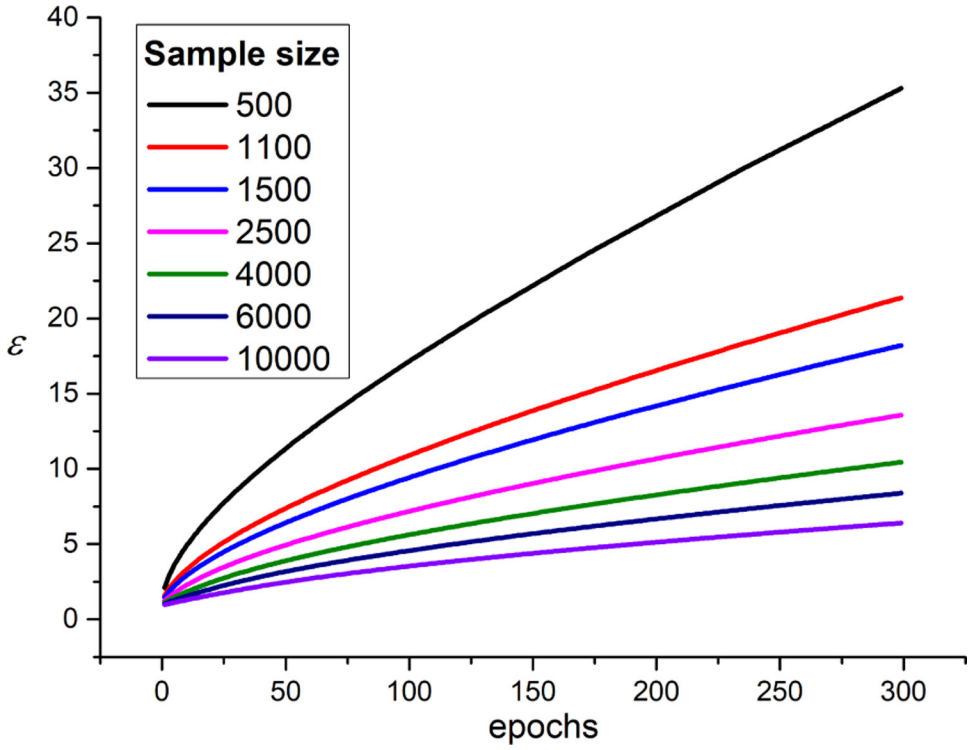


Figure 7:

The ε value as a function of epochs for different train sample sizes. The target δ is set to $1/|D|$ following (Abadi et al., 2016), and $|D|$ is the train sample size.

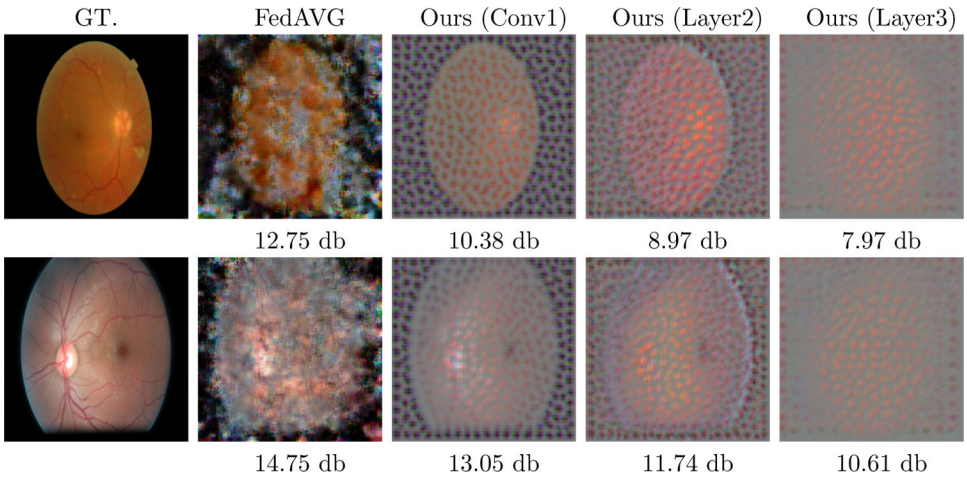


Figure 8:

Reconstruction of two images from FedAVG with deep gradient inversions and our FedReplay with model inversion attacks. We experiment with three neural network architectures for the auxiliary encoder of FedReplay, from simple Conv1 layer, to several Residual blocks. FedReplay shows similar privacy protection capability to the baseline FedAVG when shallow network is applied as the encoder network, and superior protection capability when deeper neural network is applied (Ours Layer3).

Algorithm 1: CWT+Replay. The K institutions are indexed by k, and $\mathbf{x}_k$ is training data for institution k. N and M are the training epochs, θ_C and θ_G are the model weights, L and L_G are the loss functions, and η and η_G are the learning rate for primary classifier and auxiliary generator, respectively. E_{top} and E_{bottom} are encoder functions, D is the decoder function, and $\mathbf{e}$ is the encoded latent variables.

Updating classifier in institution $(k, D, \mathbf{e}, \theta_C)$ **:**
  for local epoch i in epochs N **do**
    for each local batch x_k and e in $\mathbf{x}_k$ and $\mathbf{e}$ **do**
      $\widetilde{x^r} \leftarrow D(e)$ ▷ `Synthesizing replayed images`
      $\mathbf{x_U} \leftarrow \{\widetilde{x^r}, x_k\}$ ▷ `Union of synthetic and real images`
      $\theta_C \leftarrow \theta_C - \eta \nabla L(\theta_C; \mathbf{x_U})$ ▷ `Updating classifier`
  return θ_C to server
Updating generator in institution $(k, E_{top}, E_{bottom}, D, \mathbf{e})$ **:**
  $D_{old} \leftarrow D$
  for local epoch i in epochs M **do**
    for each local batch x_k and e in $\mathbf{x}_k$ and $\mathbf{e}$ **do**
      $\widetilde{x^r} \leftarrow D_{old}(e)$ ▷ `Synthesizing replayed images`
      $x_U \leftarrow \{\widetilde{x^r}, x_k\}$ ▷ `Union of synthetic and real images`
      $\mathbf{h}_{top} \leftarrow E_{top}(x_U)$
      $\mathbf{e}_{top} \leftarrow Quantize(\mathbf{h}_{top})$ ▷ `Quantize`
      $\mathbf{h}_{bottom} \leftarrow E_{bottom}(x_k, \mathbf{e}_{top})$
      $\mathbf{e}_{bottom} \leftarrow Quantize(\mathbf{h}_{bottom})$ ▷ `Quantize`
      $\widetilde{x_U} \leftarrow \mathrm{D}(\mathbf{e}_{top}, \mathbf{e}_{bottom})$
      $\theta_G \leftarrow \theta_G - \eta_G \nabla L_G(\theta_G; x_U, \widetilde{x_U})$ ▷ `Updating generator`
      $\mathbf{e} \leftarrow \{\mathbf{e}_{bottom}, \mathbf{e}_{top}\}$
    return $E_{top}, E_{bottom}, D, \mathbf{e}$ to server
Server executes:
  Initialize parameters θ_C and parameters θ_G
  for each round $t = 1, 2, ...$ **do**
    for each institution in $k = 1, 2, ..., K$ **do**
      if $t == 1$ **then**
        $(E_{top}, E_{bottom}, D, \mathbf{e}) \leftarrow$ **Updating generator in institution** $(k, E_{top}, E_{bottom}, D, \mathbf{e})$
      else
        $(D, \mathbf{e}, \theta_C) \leftarrow$ **Updating classifier in institution** $(k, D, \mathbf{e}, \theta_C)$

Table 1:

Catastrophic forgetting phenomenon of CWT (Chang et al., 2018) on non-IID data partitions Split 3 of Retina dataset (Kaggle, 2017). The prediction accuracy on 4 institutional train dataset from models achieved after one round of training on each institution is shown. In this experiment, the model was trained in cyclical order: Inst1→ Inst2 → Inst3 → Inst4 → Inst1 …

Data \ Model	Inst1	Inst2	Inst3	Inst4
Inst1	95.6%	91.0%	69.9%	51.7%
Inst2	83.4%	87.7%	69.0%	60.0%
Inst3	23.4%	53.7%	93.0%	91.0%
Inst4	9.4%	41.8%	95.7%	97.7%

Table 2:

MAE loss of CWT+Replay on BoneAge dataset (Halabi et al., 2019) with and without differential privacy constrain during image synthesis training.

	Split 1	Split 2	Split 3
No privacy	7.60 ± 0.03	7.55 ± 0.06	7.84 ± 0.09
With privacy	7.89 ± 0.05	8.39 ± 0.26	9.84 ± 0.02

Table 3:

Comparison of FedReplay and FedAVG in terms of prediction accuracy and average restoration performance on Split-3 of Retina datasets. FedReplay achieves better prediction accuracy on heterogenous data partitions while at the same time shows similar or even better patient privacy capability (worse reconstruction image quality) than FedAVG.

	FedAVG	Ours (Conv1)	Ours (Layer 2)	Ours (Layer 3)
ACC	70.65 ± 0.83	**77.20** ± **0.13**	76.48 ± 0.63	75.00 ± 0.98
PSNR	**11.82** ± **1.86**	11.64 ± 0.98	10.48 ± 1.23	9.22 ± 1.28