















## Research Article

Enabling end-to-end secure federated learning in biomedical research on heterogeneous computing environments with APPFL<sub>x</sub>

Trung-Hieu Hoang<sup>a, , \*</sup>, Jordan Fuhrman<sup>b, </sup>, Marcus Klarqvist<sup>g, </sup>, Miao Li<sup>c, e, </sup>, Pranshu Chaturvedi<sup>c, d, </sup>, Zilinghan Li<sup>c, d, </sup>, Kibaek Kim<sup>c, </sup>, Minseok Ryu<sup>f, </sup>, Ryan Chard<sup>c, </sup>, E.A. Huerta<sup>c, </sup>, Maryellen Giger<sup>b, </sup>, Ravi Madduri<sup>c, , \*</sup>

<sup>a</sup> Department of Electrical and Computer Engineering and Coordinated Science Laboratory, University of Illinois at Urbana-Champaign, Urbana, 61801, IL, USA

<sup>b</sup> University of Chicago, Chicago, IL, USA

<sup>c</sup> Data Science and Learning Division, Argonne National Laboratory, Lemont, IL, USA

<sup>d</sup> Department of Computer Science, University of Illinois at Urbana-Champaign, Street, Urbana, 61801, IL, USA

<sup>e</sup> School of Industrial and Systems Engineering Georgia Institute of Technology, Atlanta, GA, USA

<sup>f</sup> School of Computing and Augmented Intelligence Arizona State University, Tempe, AZ, USA

<sup>g</sup> Broad Institute of Harvard and MIT, Cambridge, MA, USA

## ARTICLE INFO

## Keywords:

Federated learning for science  
Biomedical research  
Function as a service  
Privacy-preserving  
Identity management  
Cloud computing  
Electrocardiogram  
Chest radiographs  
COVID-19 detection

## ABSTRACT

Facilitating large-scale, cross-institutional collaboration in biomedical machine learning (ML) projects requires a trustworthy and resilient federated learning (FL) environment to ensure that sensitive information such as protected health information is kept confidential. Specifically designed for this purpose, this work introduces APPFL<sub>x</sub> - a low-code, easy-to-use FL framework that enables easy setup, configuration, and running of FL experiments. APPFL<sub>x</sub> removes administrative boundaries of research organizations and healthcare systems while providing secure *end-to-end communication*, *privacy-preserving* functionality, and *identity management*. Furthermore, it is completely agnostic to the underlying computational infrastructure of participating clients, allowing an instantaneous deployment of this framework into existing computing infrastructures. Experimentally, the utility of APPFL<sub>x</sub> is demonstrated in two case studies: (1) predicting participant age from electrocardiogram (ECG) waveforms, and (2) detecting COVID-19 disease from chest radiographs. Here, ML models were securely trained across heterogeneous computing resources, including a combination of on-premise high-performance computing and cloud computing facilities. By securely unlocking data from multiple sources for training without directly sharing it, these FL models enhance generalizability and performance compared to centralized training models while ensuring data remains protected.

In conclusion, APPFL<sub>x</sub> demonstrated itself as an easy-to-use framework for accelerating biomedical studies across organizations and healthcare systems on large datasets while maintaining the protection of private medical data.

## 1. Introduction

In biomedical research, access to many types of data, such as electronic health records, medical images, and electrocardiogram (ECG) readings, is strictly regulated by law and federal guidelines like HIPAA in the United States and GDPR in the European Union. Data access committees and Institutional Review Boards (IRB) manually manage access controls to ensure that research is ethical and that the privacy of research subjects is safeguarded. Historically, these necessary, but cumbersome, access restrictions have had the undesirable effect of siloing

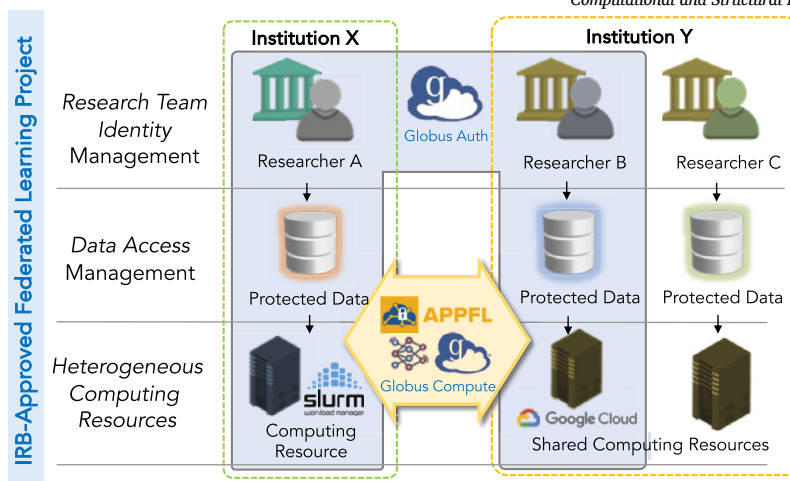
data in organizations which in turn has stymied collaborative research. Federated learning (FL) [1] has been proposed as a viable framework to make protected health data available for training machine learning models across institutions without explicitly sharing any sensitive data (using differential privacy mechanisms) [2–5]. Such a distributed and privacy-preserving framework can address, and potentially overcome, the institutional and policy-based restrictions limiting the exchange of data that is required to train cross-institutional machine learning models that are robust and are resistant to model drift.

\* Corresponding authors.

E-mail addresses: [hthieu@illinois.edu](mailto:hthieu@illinois.edu) (T.-H. Hoang), [madduri@anl.gov](mailto:madduri@anl.gov) (R. Madduri).

<https://doi.org/10.1016/j.csbj.2024.12.001>

Received 14 August 2024; Received in revised form 18 October 2024; Accepted 1 December 2024



(a) Workflow of cross-institute collaborative federated learning using APPFLx.



(b) APPFLx versus other federated learning frameworks.

**Fig. 1.** (a) Our APPFLx enhances cross-institute collaborative federated learning workflow. Globus authentication service [20,28] and our APPFLx jointly provide researcher identity, data and computing resource management. In this diagram, while sharing the same computing and data storage resource at Institution X, only researchers A and B are approved to join the FL experiment. (b) Compared to other federated learning frameworks, APPFLx provides a comprehensive solution for biomedical research with differential privacy-preserving, secure identity management, and high-performance computing friendly.

Empowered by these capabilities, FL has recently garnered considerable attention in the research community and several solutions have been proposed [6–10]. For instance, Flower [7] introduced a large-scale framework that can support up to 15 million parallel clients. While FATE [11] focuses on scalability and performance, suitable for industrial applications, OpenFL [6] positions itself as a flexible, extensible, and easily usable framework for data scientists. Additional efforts have focused on ensuring data privacy using various privacy-preserving methods [4,12–14]. In addition to these, we previously proposed *Argonne Privacy Preserving Federated Learning* (APPFL) framework [8], a comprehensive end-to-end secure FL framework that includes multiple FL algorithms, differential privacy schemes, and communication protocols. Extensive experiments have been carried out to demonstrate its performance and communication efficiency on different biomedical datasets and computing environments [15,16].

**Challenges of FL Framework in Biomedical Research.** Despite significant progress, implementing a Federated Learning framework in multi-institute biomedical studies presents additional challenges. First, under IRB regulations, it is essential to guarantee the trustworthiness of participating clients in the FL environment. Access should be restricted to authorized members only, and this requires a secure and trusted identity and access management (IAM) mechanism embedded into the framework. The second challenge is in managing heterogeneous compute

resources — such as on-premise computing clusters at one location, and cloud-computing services like Amazon Web Services or Google Cloud at another location — that use different job schedulers, like Slurm [17] or Load Sharing Facility (LSF) [18] (Fig. 1(a)). A mature FL framework should address these challenges and be able to operate regardless of the underlying infrastructure and guarantee that participation is limited to trusted and authorized partners. To address these challenges, we extended and leveraged APPFLx [19] that *emphasizes the modular design and privacy-preserving aspects* of APPFL [8] with *new capabilities* (Fig. 1(b)) which *further eases the establishing of cross-institute biomedical research*.

**Identity and Access Management in FL.** In order to ascertain that participation in an FL experiment is limited to trusted and authorized partners in the federation, it is imperative to authenticate users through their institutional identities that are integrated with their respective organizational Identity and Access Management (IAM) services. Globus Auth [20,21] provides one such secure service via a single sign-on solution, enabling researchers to easily access secure data and resources; this service is currently used in multiple large-scale initiatives across different scientific domains such as astronomy and genomics [22,23]. Globus Auth service reduces the time and effort required to control access to data while ensuring the security and integrity of the shared resource. For these reasons, we integrated Globus Auth in APPFLx to set up secure

federations using industry-standard identity and access management capabilities that enforce authentication and access control.

### FL on Heterogeneous High-performance Computing Resource.

There are additional practical requirements to ensure end-to-end security and provide a simple-to-use framework. For example, it is very likely that the participating institutions in the federation have heterogeneous computing capabilities such as on-premise high-performance computing or cloud computing facilities from different cloud providers. In an FL environment, there is generally a dedicated server that receives information from a series of participating clients that are collaboratively training a shared machine learning model. Clients independently perform local computations on their private data and communicate that result to the centralized server which in turn updates the joint model. Function-as-a-service (FaaS) or federated functions are used to execute model training at clients and model aggregation at the server. As a result, we were able to leverage Globus Compute [24] service, which provides flexible distributed task execution mechanisms. This is enabled by exposing secure *endpoints* on the participating client machines that are accessible to the centralized server. Globus Compute has been applied to high-performance computing tasks in many types of research [25,26]. Since Globus Compute simply communicates data between a client and the server, it makes the federation completely agnostic to the individual computing platforms of the participating clients. This setup permits the client to leverage their institutionally available compute (e.g., number of compute nodes, cores per worker, allocation account) which removes the burden of managing system-detailed configurations on the centralized server resulting in considerably easier and faster deployment times. This is among the first implementations of Globus Compute as the communication backbone of an FL framework, together with FLoX [27]. Nonetheless, the focus of FLoX and APPFLx is different. FLoX is designed for easy deployment of single-user multi-device cross-device FL experiments and does not need to support multi-user IAM provided by Globus Auth.

**Aims and Contributions.** Here, we describe the integration of Globus Auth [20] with Globus Compute [24] into our previously proposed APPFL framework [8] to create APPFLx. The key focus of this study is introducing APPFLx to expedite secure FL in biomedical research, demonstrated through two real-world, large-scale collaborations across multiple research institutions with protected health data. In short, the main contributions of this study are provided as follows:

- APPFLx streamlines the FL deployment process and provides both secure end-to-end communication and ensures strong identity and data access management controls that are compatible with organizational identity and access management services and with federal requirements.
- APPFLx introduces a modular design that empowers users to develop customized FL algorithms that are agnostic to the underlying compute infrastructures.
- An FL-as-a-service web platform, APPFLx enables non-experts to quickly set up and deploy secure federated learning experiments.
- To illustrate the pliability of this framework, we conducted extensive experiments on two real-world biomedical research domains in 1) estimating human biological aging from ECG waveforms and, 2) detecting COVID-19 disease from chest radiographs (CXR). The results demonstrate that federated learning studies can be easily established across four distinct research facilities to jointly train machine-learning models in a privacy-preserving fashion.
- Finally, we conducted model inversion attacks to highlight the necessity of using privacy-preserving technologies in FL for biomedical use cases and demonstrate the effect of applying the differential privacy protection scheme implemented by APPFLx framework.

**Section 2** briefly summarizes the key aspects of APPFLx, model inversion attacks and other relevant details on experimental setup. **Section 3** walks through the two case studies, and provides details on model

inversion attacks that we use to showcase the benefit of APPFLx in privacy preservation. The main discussions of this study are provided in **Section 4**. Finally, in **Section 5** we provide the main conclusions.

## 2. Methods

The methodologies of this study encompass three key aspects: (1) The introduction of APPFLx for facilitating FL in biomedicine research (**Section 2.1**), which will be demonstrated using two real-world case studies in the subsequent section; (2) The design choices and architecture of APPFLx (**Section 2.2**); and (3) The model inversion attacks via gradient data leakage study (**Section 2.3**).

### 2.1. Enabling secure federated learning in biomedicine using APPFLx

**Key Steps for Establishing FL Experiment.** APPFLx is an extensible FL framework with built-in support for several FL algorithms, making it independent of any specific aggregation scheme. Users can easily experiment different federation strategies and select the right algorithm depending on their use case and resource constraints. Researchers can easily set up an FL experiment in a simple 4-step process. We briefly summarize the workflow here:

1. **Identity Verification:** Participating institutions in FL sign-in to the federation using their institutional identity that is integrated with a Globus identity [20]. Globus Auth is integrated with identities of most US universities and national labs. The scientist setting up the FL experiment (the *orchestrator*) organizes the participating Globus identities into a Globus group and assigns roles to each member. A secure communication channel for moving data between only these parties is established in this way.
2. **System Setup:** All clients install and configure a Globus Compute-endpoint on the target compute and register the address of this endpoint with the centralized FL server.
3. **Configuration:** The researcher can then define any model architecture and data loader in PyTorch [29] for their project. Additionally, training hyper-parameters, FL aggregation scheme, and privacy-preserving settings on the participating clients are provided through a simple configuration file.
4. **Running FL Experiment:** Finally, the researcher establishes and monitors the FL experiment. Besides regular training tasks, APPFLx also (optionally) performs cross-site validation, in which a model is sent and evaluated on the local private datasets at clients without sharing the data.

**Web-based User Interface for Rapid FL Setup.** In order to further lower the barrier in conducting FL experiments, we created APPFLx as a service (FL-as-a-Service) and a web application [19]. Using a user-friendly dashboard, a researcher can interactively complete all the required steps outlined above, such as managing Globus groups, registering new client endpoints, uploading model descriptions, and configuring FL parameters for both the server and the clients. The platform also has the ability to automatically initiate and deploy an APPFLx server, with appropriate access management policies expressed as AWS IAM rules, to an ECS container to act as the orchestration server. For started experiments, it is easy to monitor the training progress by leveraging existing visualization tools such as Tensorboard [30] or inspecting the real-time client logs. The service is made publicly available at <https://appflx.link>.

In the following sections, we describe setting up FL experiments across organizational boundaries using heterogeneous computing capabilities for the training of ML models and fast iteration of FL experiments. We aim to demonstrate the flexibility and power of the APPFLx service while demonstrating the importance of FL in developing robust AI models under undesirable distribution shifts.

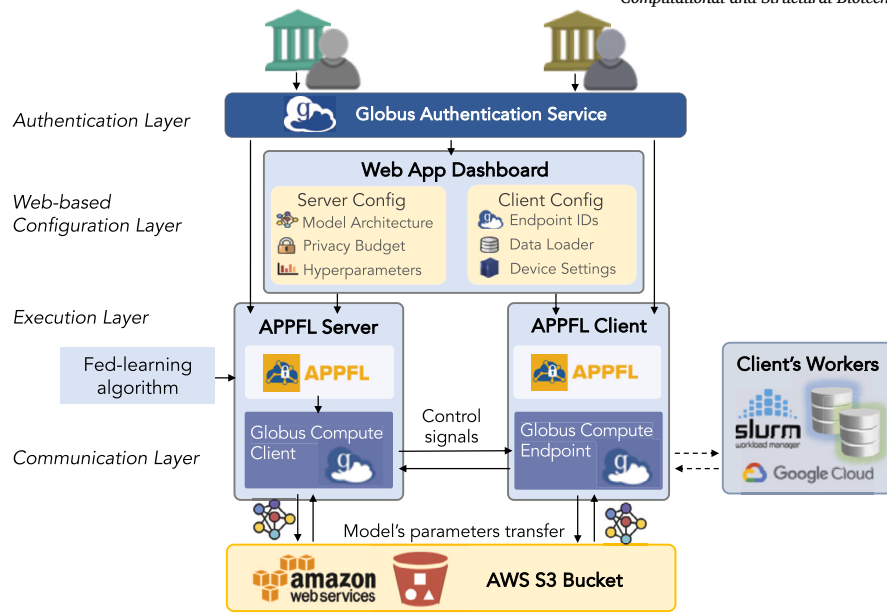


Fig. 2. Main components of the APPFLx. Modules can be classified into four layers: authentication, web-based configuration, execution and communication.

## 2.2. Architecture of APPFLx

Taking advantage of modular design of APPFL [8], we integrated Globus Compute [24] and Globus Auth to establish a mechanism for handling remote task execution, identity management, and performing federated learning at multiple distributed clients. Fig. 2 illustrates the main components of our system, classified into four main layers, namely *authentication*, *web-based configuration*, *execution* and *communication* layer.

**Authentication Layer.** Globus Authentication (Globus Auth) [28] - a secure service for end-user identity management is used extensively in our framework. Globus Auth(GA) serves as a broker for authentication, identity provider, and managing interaction between end-users. After the authentication step, all verified credentials will be passed through the execution layer where the workers execute the training process on behalf of the researcher. Fig. 1(a) gives an illustration. Suppose researchers A and B are working together on a project that does not include researcher C. With GA, only A and B are approved to be members of the group. In this case, A and B have the right to access their institution's computing and data management resources. During the FL, APPFLx acts as a proxy for the two researchers, where it is now authenticated to have the same right as its owner to perform FL tasks. We utilize Globus Groups (part of Globus Auth) to implement authorization through a group membership mechanism. In this model, the creation of a federation is represented as a Globus Group, where the group administrator invites participants to join. Invitations are sent to participants' institutional email addresses, requiring them to authenticate with their institutional credentials. Upon successful authentication, users are authorized to participate in the federation based on their membership in the Globus Group established by the federation administrator.

**Web-based Configuration Layer.** The purpose of the web application is to streamline the configuration process for users and enhance the user experience. We used Python Flask framework [31] for setting up the backend, and Globus Auth for authentication. Using a unified dashboard, users can provide server-side and client-side configurations. Client-side configuration includes data loaders, customized for their training data, and an identity string of Globus Compute-endpoint for registering their computing resource. The group administrator can then initiate the server-side settings, which include a list of registered clients,

the model architecture, FL algorithm, and specify the privacy-preserving budget and other training hyperparameters. We also provide helper functions that help users check the clients' status and aggregate training logs from either a regular text file or Tensorboard. Optionally, users can select an automated APPFLx server deployment on an ECS container (provided by AWS [32]), to quickly set up an experiment.

**Execution Layer.** The main challenge when working with multiple HPC systems is the heterogeneity of their cluster management and job scheduler. HPC administrators may or may not adopt a platform like Slurm [17]. To successfully execute jobs, users also need to specify worker configurations (e.g., default job queue, charging budget, etc.) which creates a great burden on the FL framework. To this end, we utilize the advantage of Globus Compute [24], making it becomes the main backbone for network communication and executable jobs management. The benefit of using Globus Compute is that this burden of handling job execution over diverse platforms is done automatically through a universal programming interface. Globus Compute is demonstrated to work with wide range of computing resources, from personal computers to supercomputing facilities, like the Delta supercomputer (NCSA), or Theta (ALCF). Another benefit of Globus Compute is that the HPC configuration is dedicated to the client when setting up the APPFLx clients which removes the need to continuously communicate with the server when updating configurations.

We briefly summarize how Globus Compute is adopted in APPFLx. Globus Compute consists of two main components: Globus Compute-Client and Globus Compute-endpoint, which are deployed at APPFLx server and client, respectively. Endpoints are abstract representations of computing resources that run in the background on the client's machine on behalf of the authenticated user and handle dispatch job execution from Globus Compute clients on request. Meanwhile, Globus Compute client at APPFLx server serves as a controlling node for managing a set of distributed APPFLx clients, sequentially assigning training tasks during an FL experiment. We require all servers and clients must have an outbound Internet connection.

**Communication Layer.** While Globus Compute can effectively communicate the structure of machine learning models, and training configurations from server to all clients, it is not optimized for transferring large binary files, such as model weights. To this end, we leverage Amazon Web Services S3 service (AWS) [32] to accommodate this task.



**Table 1**

(a) Basic info about the participating sites of Case Study 1: Biological aging prediction from ECG signal experiment. (b) Statistic of the datasets used in this case study. (c) Testing mean square error (MSE) of the biological aging prediction from ECG signal models. The average column computes the average MSE across two datasets, weighted by the number of testing samples.

(a) Participating sites of Case Study 1: Biological aging Prediction from ECG Signal.

Site	Role	Location	Computing Infrastructure	Dataset
ANL	Client	Lemont, IL	GPU Computing Cluster	ECG-ANL
Broad	Client	Cambridge, MA	Google Cloud Compute Engine	ECG-Broad
UIUC	Server	Urbana, IL	CPU Computing Cluster	N/A

(b) Statistics of the two datasets used in the ECG case study.

Dataset	Train	Val	Test	Total
ECG-ANL	64518	7905	7905	80328
ECG-Broad	33140	4143	4143	41426

(c) The mean square error of FL and local models.

Training Dataset	Testing Set		
	ECG-ANL	ECG-Broad	Average
ECG-ANL ( <i>local training</i> )	109.95	224.48	149.33
ECG-Broad ( <i>local training</i> )	225.41	38.93	161.28
ECG-ANL+Broad - FedAvg[1]	125.00	41.70	96.35

### 2.3. Model inversion attack experiment

**Inversion Attack via Gradient Data Leakage.** According to [3,13,33], gradient data from the early iterations of training are generally *more susceptible to training data leakage*. This can be explained by the fact that during training, the magnitude of the gradient updates generally converges to zero regardless of the underlying training data. Because of this tendency, at later training steps, the gradient may only contain a small amount information from the training sample while the earlier ones contain a sufficient amount of information to reconstruct the private training data on the client (since the gradients are generally still far from zero). Therefore, with a fixed number of training epochs, we investigate using different dataset sizes as larger datasets typically require more training steps of the neural networks.

Meanwhile, larger training batch size can also adversely impact the reconstruction quality as the gradient update representing a batch of training examples is an average of the training batch. Hence, inspecting an individual gradient corresponding to each training image for a single-image inversion attack in a large training batch is more challenging. Previous works have shown that both the amount of training and batch size reduces the effectiveness of the gradient inversion attack [12,13,34–37].

**Inversion Attack Baseline.** The most susceptible scenario for an inversion attack is when the training dataset size, training batch size, and number of training rounds are all set to one (referred as *baseline* attack). Hence, this setup is utilized to compare the effects of increasing the level of differential privacy, amount of training, and batch size (the results are provided in Section 3.3). When varying the level of differential privacy, the clip value of the Laplacian mechanism [38] is set to one while the value of  $\epsilon$  gradually decreases (increasing the scale of the Laplacian noise, equivalently). When examining the influence of the training amount, we train the network model using 20 images and then perform a gradient inversion attack over this *lightly trained* model. The same process is repeated with 150 images in the *extra training* case. Finally, regarding the effect of training batch size, we compare the reconstruction results for various batch size choices (1, 10, and 50).

**Details of the Attack Implementation.** We use the inversion attack model given by [13] with an additional modification which includes a batch normalization penalty introduced in [36]. The general strategy is to initialize the inversion attack algorithm with a placeholder image,

which will be updated continuously through an iterative inverting gradient algorithm and eventually converged to the private training data sample. Four different choices of initialization are experimented with: sampling pixel values from a random Gaussian distribution [13], uniform distribution [3], and creating a new image by averaging several images over a non-overlapping data set [33]. We also investigate two common optimizer choices for the attack algorithm, including Adam [45] (used in [13]) and AdamW [39] (used in [3]). Extensive grid search amongst all combinations of the aforementioned image initialization, scale of total-variation penalty [13], BN penalty [36] is necessary for each setup to produce pronounced reconstruction results.

## 3. Results

In this section, we describe APPFLx- our end-to-end, privacy-preserving FL framework in two distinct biomedical research studies. Particularly, we will describe the ease of setting up secure federations and designing and running FL experiments, and then report the overall model performance.

### 3.1. Case study 1: biological aging prediction from ECG signal

**Task Description.** The electrocardiogram (ECG) is the most popular, simplest, and fastest exam used for the evaluation of various cardiovascular diseases. Predicting biological age from the raw ECG waveform is beneficial for revealing an individual’s cardiovascular health [40,41]. In this study, we investigate the task of regressing human biological aging from 12-lead ECG waveform. A deep learning model is trained to reduce the mean squared error (MSE) between the predicted age and the subjects’ age at the date of ECG reading as ground-truth.

**Datasets and FL Sites.** This case study is a collaborative research project between Argonne National Laboratory (ANL), the Broad Institute (Broad), and the University of Illinois at Urbana-Champaign (UIUC). Details of each FL site are provided in Table 1(a). Table 1(b) summarizes the statistics of the two datasets used by each client in the ECG experiment. At ECG-ANL, we adopted the publicly available PhysioNet dataset [42] while the dataset at ECG-Broad is composed of ECG signals from the UK Biobank [43].

An FL experiment across multiple sites is established with APPFLx. In this study, a global server is hosted on a conventional CPU machine (specifically, Intel Core i7-6700K CPU @ 4.00 GHz) at UIUC. The first

**Table 2**

(a) Basic info about the participating sites of Case Study 2: COVID-19 Detection on Chest Radiographs (CXR). (b) Statistics of the datasets used in the COVID-19 chest X-ray image recognition experiment. Numbers in parentheses indicate the number of positive (+) and negative (–) samples. (c) AUC score of the COVID-19 chest X-ray image recognition models.

(a) Participating sites of Case Study 2: COVID-19 Detection on Chest Radiographs (CXR).

Site	Role	Location	Computing Infrastructure	Dataset
ANL	Client	Lemont, IL	GPU Computing Cluster	MIDRC
UChicago	Client	Chicago, IL	GPU Computing Cluster	UChicago
UIUC	Server	Urbana, IL	CPU Computing Cluster	N/A

(b) Statistics of the two datasets in the COVID-19 detection on CXR case study.

Dataset	Train	Val	Test	Total
MIDRC	9867 (4226+/5641-)	2056 (925+/1131-)	2081 (932+/1149-)	14004
UChicago	26047 (4226+/23226-)	5569 (587+/4982-)	5619 (637+/4982-)	37235

(c) AUC comparison between local and FL models.

Training Dataset	Testing Set	
	MIDRC	UChicago
MIDRC ( <i>local training</i> )	0.80 [0.78, 0.82]	0.56 [0.54, 0.58]
UChicago ( <i>local training</i> )	0.59 [0.57, 0.62]	0.67 [0.65, 0.69]
MIDRC+UChicago - FedAvg [1]	0.69 [0.67, 0.71]	0.67 [0.65, 0.69]
MIDRC+UChicago - FedAvg + Fine Tuning	0.79 [0.77, 0.80]	0.84 [0.83, 0.84]

client utilizes a computing cluster with a single NVIDIA GeForce RTX 3090, hosting and performing training on the ECG-ANL dataset. Meanwhile, the second client at Board Institute deploys our platform on a GPU Google Cloud Compute Engine.<sup>1</sup> This setup demonstrates the flexibility of APPFLX in leveraging a wide range of heterogeneous computing environments (e.g., local computing clusters and cloud computing services).

**Regression Model and FL Setup.** As a baseline model, we employ a ResNet34-styled [44] architecture. Each ECG channel is normalized to have zero mean and unit standard deviation. For the FL procedure, we use FedAvg [1] in our experiment for a total of 30 federation rounds. During each training round, the local models of all clients are optimized in 2 local epochs before being aggregated to the global model. We use Adam optimizer [45] with an initial learning rate of 0.003 and a decay scheduler by a factor of 0.975.

**Result - FL versus Local Training.** Table 1(c) gives the MSE on the test set of models trained on individual (rows 1-2) and combined (row 3) datasets. We observe that *models trained on the combined dataset remarkably outperform ones trained on the individual datasets* in average. This highlights the benefit of FL which, allows the training of a machine learning model on multiple datasets. Another noticeable outcome is that FL models typically achieve higher generalization rate when being cross-evaluated on unseen samples from out-of-distribution datasets.

### 3.2. Case study 2: COVID-19 detection on chest radiographs

**Task Description.** One of the domains that may require increased security or privacy when building AI models is medical imaging. Here, allowing for open access to medical images can be impossible, particularly considering privacy concerns related to attack models and potentially inconsistent or ineffective data de-identification requirements across different local sites (e.g., face shearing technology). The COVID-19 pandemic served as a prime example of a use case for APPFLX implementation; many institutions were interested in contributing to aggregated datasets for use in developing medical imaging-based AI models for COVID-19 detection, differential disease diagnosis, and other radiological tasks. Despite good intentions, several complications arose

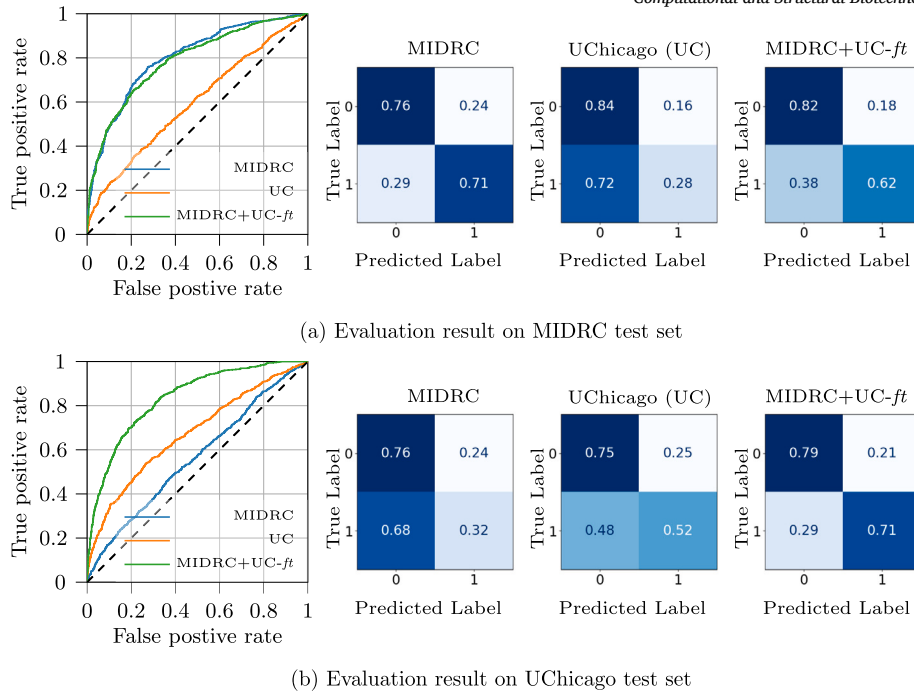
through “Frankenstein” datasets, biased algorithms, and required an extensive time to open source [46]. Many of these obstacles could have been alleviated if a privacy-preserving FL system were available. Thus, we provide COVID-19 detection on chest x-ray images (CXR) as a use case to evaluate our proposed system. Clinically, the most common methods of COVID-19 detection are non-imaging exams (e.g., antigen or RT-PCR exams); however, imaging could play a role in differential disease detection upon image acquisition in the future if non-imaging tests are less readily available or no longer standard practice.

**Datasets and FL Sites.** We set up a two-site training for this case study, detailed in Table 2(a). The first client hosts the publicly available CXR dataset from the Medical Imaging and Data Resource Center (MIDRC).<sup>2</sup> Initiated in 2020 to combat the pandemic, MIDRC is a multi-institutional collaborative initiative in medical imaging through data sharing, having already scraped over 300,000 medical imaging studies. This comprehensive dataset contains digital radiograph images, COVID test results, and demographic information collected from multiple hospitals. The second client holds a private dataset that was collected at the University of Chicago (UChicago). The UChicago dataset was collected as part of the University of Chicago Center for Research Informatics (CRI) COVID-19 Datamart in conjunction with the Human Imaging Research Office (HIRO). The two datasets’ train-test splitting scheme, statistics, and the number of positive and negative samples used in this experiment are reported in Table 2(b).

**Classification Model and FL Setup.** This case study demonstrates a simple transfer learning approach to CXR data. We fine-tune a ResNet18 [44] model pre-trained on ImageNet [47]. The last softmax layer is modified to match the binary classification task (i.e., COVID-19 positive and negative). Further, other recent publications have investigated the relevance of *personalized FL* [48–50], or the development/improvement of FL models for performance by individual clients. We incorporate personalized FL in this study through additional local *fine tuning* on a small subset of labeled data (here, we adopted the validation set) from both the MIDRC and UChicago clients for an additional 40 epochs. Notably, for this fine-tuning step, only the trainable parameters of the batch normalization layers [51] are updated. This adjustment specifically targets the sensitivity of these layers to local data

<sup>1</sup> <https://cloud.google.com/products/compute>.

<sup>2</sup> <https://www.midrc.org/>.



**Fig. 3.** Cross-site evaluation result of the COVID-19 detection on chest radiographs (CXR) case study. The ROC curve (left) and confusion matrices (right) of various COVID-19 CXR detection models: local training on a single MIDRC and UChicago (UC) dataset; *fine tuned* model with a baseline comes from FL (FedAvg [1]) on the two datasets (MIDRC+UC-*ft*) using our APPFLx. We demonstrated a superior performance when an FL model serves as a foundation model for fine tuning a client-specific dataset.

statistics. Similar strategies have been implemented in other domain adaptation approaches [52,53].

Using APPFLx, we establish the federation across multiple institutes. For simplicity, we adopt the common FedAvg [1] algorithm for the global aggregation. Two Globus Compute endpoints (one for each dataset, representing two distinct clients) are installed at UChicago while the global server is hosted at UIUC. The clients securely store their dataset, and only reveal the data loaders to the server. Once the federation is created, the server can automatically facilitate the training and cross-site evaluation process. During training, each client performs two local updates before being integrated into the global model. We repeat this process for a total of 40 global FL aggregation rounds. For hardware requirements, the center server only requires a CPU machine, while most of the heavy-computing tasks are performed locally on two GPU clusters of the University of Chicago (HPE Superdome Flex NUMA computation server with 2 NVIDIA Tesla V100 32 GB GPUs).

**Result - FL versus local training.** All models are evaluated on the clients' test set using the cross-site validation feature of APPFLx. We separately plot the ROC curves (Fig. 3-left) and confusion matrices (Fig. 3-right) then compute the corresponding areas under the ROC curve (AUC), confidence interval (Table 2(c)) when testing them on the two datasets in the task of positive and negative COVID-19 CXR detection. We compute the AUC as the main evaluation metric for discussion. Over the two datasets, we observe that while models trained on a single dataset (rows 1-2) achieve satisfactory performance when evaluating on test sets with the same distribution, the performance *drops significantly on the test set of other sites* (e.g.,  $0.80 \rightarrow 0.56$  for the model trained on MIDRC dataset and  $0.67 \rightarrow 0.56$  for the model trained on UChicago). Meanwhile, the model trained in FL settings with a combined MIDRC+UChicago dataset (row 3) can overcome this challenge in single-dataset models and provide a more stable performance. Notably, for the MIDRC dataset, the performance of the FL model (0.69) was significantly reduced in comparison to the local model (0.80). This result indicates joint training with the UChicago dataset with a large distribution gap can possibly downgrade the performance of the FL model.

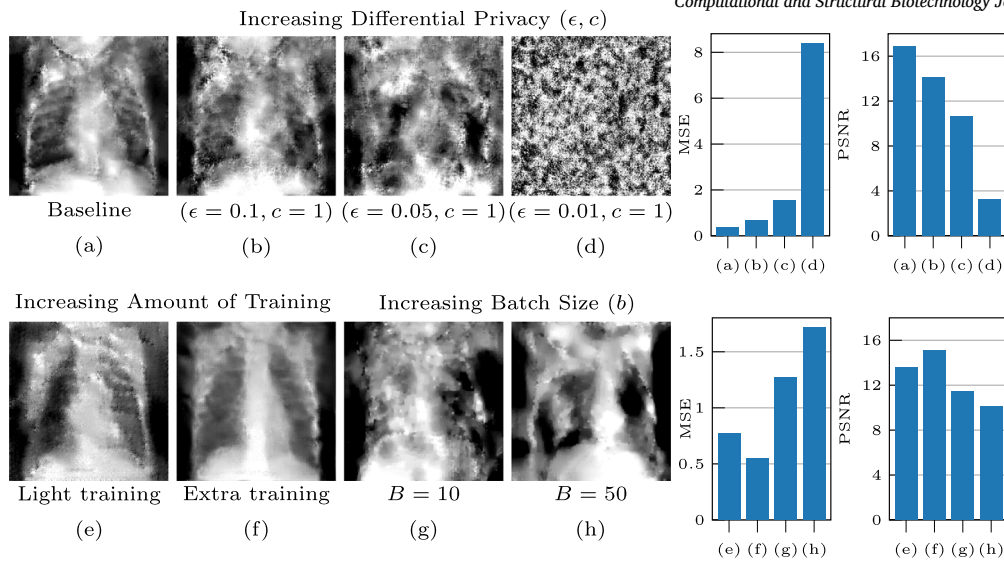
However, our investigation of *fine tuning* on a small subset of the labeled dataset at each site for personalized FL through APPFLx (row 4) demonstrated *superior performance*, achieving comparable performance to the locally-trained MIDRC model and significantly outperforming the local UChicago model. This suggests that the baseline model provided by the FL algorithm serves as a better foundation than the original ImageNet-trained [44,47] model for addressing our task of COVID-19 detection. This has potential practical implications in how FL is deployed, particularly in scenarios with differing data distributions across FL clients as in our case study. Specifically, in such scenarios when a single client cannot effectively achieve high performance simultaneously across clients, personalized FL as deployed in our APPFLx scheme could play a significant role in the development of improved models.

### 3.3. Model inversion attacks and privacy-preserving demonstration on case study 2

Privacy-preserving, an essential part of a trustworthy and secure FL, is one of the most important components in our APPFLx. In this section, we take Case Study 2 (Section. 3.2) to demonstrate the risk of user privacy leakage via an inversion attack simulation and the advantage of APPFLx privacy-preserving scheme to mitigate this critical issue.

**Model Inversion Attacks and Privacy-Preservation.** Recent works [12,13,34–37] have examined a scenario where an attacker has access to the *local client gradient* updates during FL processes. Alarming, the gradients information communicated between the clients and the server during training or even after training can reveal information about the training set from which an iterative gradient inversion algorithm can reasonably reconstruct private local data. Thus, we evaluate the privacy-preserving scheme implemented by the APPFLx framework to qualitatively and quantitatively measure the extent of information leakage through gradients by performing an inversion attack during the FL process with and without our privacy-preserving scheme.

**Experimental Setup.** Following the same setting, we study a ResNet18 [44] model pre-trained on ImageNet [47], which consumes a grayscale input image of size  $224 \times 224$  as input. For a fair comparison with previ-



**Fig. 4.** Reconstruction results of gradient inversion attacks with various differential privacy budgets (top row) and training settings (bottom row). (a) is the baseline attack result. (b)-(d) show the effect of applying increasing levels of differential privacy, from (b), the weakest level of differential privacy to the strongest in (d). (e) and (f) show the effect of using gradients from models with increasing amounts of training. (g) and (h) show the impact of increasing the batch size on the reconstruction quality. We also compare the corresponding MSE and PSNR for all settings (a)-(h) (bar charts, last two columns). Both metrics indicate a clear decline in reconstruction quality when increasing the differential privacy, while increasing the training amount or the batch size has little effect on the reconstruction quality.

**Table 3**

A summary of the key technologies incorporated in APPFLx.

Technology	Role
APPFL	Supporting a variety of synchronous and asynchronous FL algorithms, aggregation, and privacy-preserving schemes
Globus Authentication	Seamlessly authorizing user's identities, based on their institute's credentials before establishing a federation
Globus Compute	Executing computational functions on diverse heterogeneous computing environments (e.g., laptops, clusters, clouds, or supercomputers, etc.)
AWS/Globus Transfer	Securely transferring model's parameters across sites

ous work [3], the publicly available dataset [54] is used in our inversion attack experiments. Other setups follow the setting described in [12], and the effect of differential privacy when varying the clip value ( $c$ ), noise level ( $\epsilon$ ), amount of training, and the training batch size ( $b$ ) on the reconstructed image is investigated.

**Result - Model Inversion Attack.** Our model inversion attack results with *increasing differential privacy preserving budget* are illustrated in Fig. 4-top. (a) provides the baseline result (described in Section 2.3), as the most vulnerable case, and without our privacy-preserving scheme on a single-sample training dataset with batch size  $b = 1$ . (b) uses the same setting as (a) except Laplacian noise with clip value  $c = 1$  and  $\epsilon = 0.1$  is added to tackle the inversion attack (privacy-preserving scheme). We observe that reconstruction quality is significantly downgraded compared to (a). In (c) and (d) we further increase the Laplacian noise (by decreasing  $\epsilon = 0.05$  to  $\epsilon = 0.01$ ), respectively. Here, the reconstructed image is completely unrecognizable as CXR. For quantitative evaluation, the mean squared error (MSE) and peak signal-to-noise ratio (PSNR) for the increasing privacy-preserving levels (top row, right bar charts) showcases a *clear decline in reconstruction quality*, indicating the benefits of *incorporating the privacy-preserving scheme* of APPFLx.

Meanwhile, (Fig. 4-bottom) studies the effects of other training factors on the inversion attack *without* applying the differential privacy scheme. In (e) and (f), light training and extra training of the same ResNet 18 [44] model are compared. In the light training mode, the model explores 20 images, while this number increases to 150 in the extra training case. In both scenarios, the amount of gradient information leakage is still remarkable (in contrast with (c) and (d)), (e) and (f) have comparatively low MSE and high PSNR values, and are only

worse than (a)). Additionally, we further study the effect of increasing training batch size with  $b = 10$  (g) and  $b = 50$  (h). The MSE and PSNR metrics (bottom row, right bar charts) verify the effect of increasing batch size. Surprisingly, in both cases, the image reconstruction is still recognizable given that the gradient update from a large batch of images contains much less image-specific information than a single-image batch (as in (a)). In sum, these results indicate that increasing the amount of training or changing the batch size, up to some extent, can reduce the reconstruction quality. However, that in itself cannot completely eliminate the role of privacy-preserving schemes, and privacy-preserving is *always important at every stage* of the training process.

## 4. Discussion

### 4.1. APPFLx - a platform for FL in biomedical research

**Rapid FL Experiment Deployment.** As far as we are aware, APPFLx is the only FL framework that has end-to-end security and is compatible with the workflow of research organizations (Section 2.1). In the two case studies (Section 3.1, 3.2), we showcase a rapid deployment of this platform for conducting FL experiments across four research institutes (ANL, UChicago, UIUC, and Broad). Table 3 briefly summarizes the key technologies that drive the main functionality of APPFLx. There are no trade-offs in performance when using APPFLx compared to other popular FL frameworks. In terms of functionality, there are no requirements for advanced knowledge in AI or distributed computing. The framework provides many training utilities to lower the barrier for non-technical users, and easy to use privacy-preserving ability for rapid deployment and experimentation.



**Noticing the Cross-dataset Generalizability with FL.** Understanding model generalizability is an important task in machine learning. Using APPFLx, we demonstrated this process in action by assessing the single-dataset model's generalizability on *cross-site evaluation* and how FL comes into place in two real-world case studies. By enabling an easy and trustworthy data-sharing mechanism, models trained with FL achieve better generalizability. Our result further highlights the *need to adopt FL techniques* when experimenting with machine learning models in biomedical research. APPFLx, flexible design framework, becomes a convenient tool for researchers to quickly set up FL experiments, analyzing and improving the model generalizability across multiple sites.

**FL on Clients with Severe Distribution-shift.** Utilizing a large pool of training data, FL models are *typically expected* to be well generalized and capable of achieving supreme performance on datasets from both cross-site and same-site distributions. However, with large distribution-shift, and a limited number of participating FL sites demonstrated in our case study, their performance still lags behind the locally trained models when evaluating on same-site testing sets. Notice that this is a common phenomenon in FL, which is even more likely to happen in biomedical research. Fortunately, this shortcoming has been partially addressed in several previous personalized FL research [48,49,55,56], and in the Case study 2 (Section 3.2), a fine-tuning scheme has been adopted to overcome this drawback.

#### 4.2. Key advantages of APPFLx

**Privacy-preserving for FL training.** Qualitative and quantitative results across all three inversion attack experiments show that adopting a differential privacy scheme is crucial all the time. Using a simple attack model on the leakage of gradient communication between server and client can easily violate the privacy of user data. We see that in a realistic setting, where the data batch size gets larger and after some training has been conducted on the target network, stealing private training information from clients is still possible. Fortunately, with a differential privacy scheme, even the most susceptible setting of batch size one and using one training image only can tear down the inverse relation of gradient sent by the client and the underlying training data, thus eliminating the chance of data leakage. The capability of APPFLx framework allows all FL participants to enable the privacy-preserving scheme when needed, creating a *trustworthy FL environment for conducting cross-institute biomedical research*.

**Latency and Communication Efficacy.** The authentication step is performed once, at the beginning of creating the federation. Hence, the latency for this step is minimal, compared to the entire FL process. With APPFLx, users can freely change the aggregation algorithm, depending on the use case and resource constraints. For instance, one can use asynchronous FL [57,58] when there is a computing power mismatch between participating clients. The communication between server/participant clients (or computing nodes) is dedicated to the Globus Compute. The function execution at endpoints via FaaS (Function as a Service) structure is an industrial standard and has been popularly adopted in many use cases.<sup>3</sup> See [59] for a recent analysis. For a deeper technical comparison of latency and communication efficacy, we refer the readers to visit Section IV.A. of [60].

**Comparison to Prior Work.** From the aspect of identity management, many FL frameworks [6,8] verify digital entities involved in FL process using a common trusted certificate authority (CA) and an HTTPS server for listening and signing requests. Other methods also extend it into a blockchain-based and decentralized system using smart-contract [61]. While these approaches can verify the ownership of a public key by a named subject, there is a lack of correspondence between virtual and real-world identity. Our APPFLx takes a step further by coupling digital

FL clients with researchers' identities. Note that this single-time authentication governs all aspects of an FL process, including granting access to sensitive data, and allocating computing resources, which are naturally controlled by the researcher's identity. To the best of our knowledge, this is the first FL framework that enables this ability. Readers are encouraged to visit Table I of [60] for an in-depth comparison of APPFLx and other popular open-source FL frameworks.

Most of the aforementioned FL frameworks typically target edge devices [62] as their main deployment target. Some frameworks [9,10,63] support leveraging GPUs for hardware acceleration which supports the development of FL applications that require real-time response and cater to applications that compute intensive. However, there is an ongoing need to develop FL technologies that can leverage heterogeneous high-performance computing (HPC) resources, similar to [10,63]. We address these requirements with Globus Compute integration to the APPFL toolkit. The Globus Compute integration allows to *seamless* leverage of traditional tightly-coupled high-performance computing resources and cloud computing resources that increase the overall adoption scenarios.

#### 4.3. Limitations and future work

The initial setup of training sites for clients requires experience in configuring Globus Compute endpoints and going through several platforms like Globus, which may be an obstacle in some situations. Client-server communication is tightly intertwined with Globus Compute, creating a substantial dependence on this service and may limit the scalability of APPFLx when increasing the number of clients. While Globus has been adopted by several organizations, some organizations may be restricted to using products that use more common authentication standards. Future work includes exploring alternative authentication standards prevalent in the industry such as OAuth2 or OpenID. In addition, a more complete investigation of the system's security is necessary to recognize possible system vulnerabilities and thoroughly assess the privacy-preserving capabilities of APPFLx.

APPFLx is one of the few FL frameworks that offer a web user interface, simplifying the initial steps of the FL setup process. The integration of Globus Auth enhances user-friendliness by allowing users to log in with their institutional credentials directly through the same user interface. Despite our best efforts, some steps may still pose challenges for users without technical expertise. We will keep improving the UI/UX and documentation of APPFLx in the future as the project evolves and more user feedback is available. In short, the web platform can handle user authentication and automatically launch the server for clients, streamlining the most challenging steps of the setup process. The APPFLx documentation has been continuously updated to reflect the latest changes in the framework.

## 5. Conclusion

In this paper, we propose the integration of Globus compute to APPFL, namely APPFLx. This is a free, open-source high-performance federated learning platform that is specialized for facilitating cross-institutional collaboration on sensitive medical data. The two key novelties of our framework include *organizational identity management* and *task execution strategy on heterogeneous computing resources*. We further demonstrate the use of APPFLx in two real-world case studies: biological aging prediction from ECG signal and COVID-19 chest x-ray image recognition. In sum, our framework successfully coordinates a federated learning process across four research institutes.

#### CRedit authorship contribution statement

**Trung-Hieu Hoang:** Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Resources, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Jordan**

<sup>3</sup> <https://www.globus.org/user-stories>.

**Fuhrman:** Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Resources, Methodology, Investigation, Data curation, Conceptualization. **Marcus Klarqvist:** Visualization, Validation, Supervision, Software, Resources, Investigation, Data curation, Conceptualization. **Miao Li:** Writing – original draft, Visualization, Validation, Software, Methodology, Formal analysis, Data curation. **Pran-shu Chaturvedi:** Writing – original draft, Visualization, Software, Investigation, Data curation. **Zilinghan Li:** Writing – review & editing, Validation, Software, Investigation. **Kibaek Kim:** Writing – review & editing, Validation, Supervision, Investigation, Conceptualization. **Min-seok Ryu:** Validation, Investigation, Conceptualization. **Ryan Chard:** Validation, Software, Resources. **E.A. Huerta:** Supervision, Resources, Project administration, Investigation, Funding acquisition, Conceptualization. **Maryellen Giger:** Writing – review & editing, Validation, Supervision, Resources, Methodology, Data curation, Conceptualization. **Ravi Madduri:** Writing – review & editing, Validation, Supervision, Resources, Project administration, Methodology, Investigation, Funding acquisition, Data curation, Conceptualization.

### Declaration of generative AI and AI-assisted technologies in the writing process

None.

### Declaration of competing interest

None.

### Acknowledgement

The imaging and associated clinical data downloaded from MIDRC (The Medical Imaging and Data Resource Center) and used for research in this publication was made possible by the National Institute of Biomedical Imaging and Bioengineering (NIBIB) of the National Institutes of Health under contract 75N92020D00021. This manuscript is based upon work supported by the U.S. Department of Energy Office of Science, under contract number DE-AC02-06CH11357.

### References

- [1] McMahan B, Moore E, Ramage D, Hampson S, Arcas BAy. Communication-efficient learning of deep networks from decentralized data. In: Proceedings of the 20th international conference on artificial intelligence and statistics, vol. 54. 2017. p. 1273–82. Available from: <https://proceedings.mlr.press/v54/mcmahan17a.html>.
- [2] Kaissis G, Makowski MR, Rückert D, Braren RF. Secure, privacy-preserving and federated machine learning in medical imaging. *Nat Mach Intell* 2020;2:305–11.
- [3] Kaissis G, et al. End-to-end privacy preserving deep learning on multi-institutional medical imaging. *Nat Mach Intell* 2021;3:473–84.
- [4] Froelicher D, et al. Truly privacy-preserving federated analytics for precision medicine with multiparty homomorphic encryption. *Nat Commun* 2021;12(1):5910. <https://doi.org/10.1038/s41467-021-25972-y>.
- [5] Bai X, et al. Advancing COVID-19 diagnosis with privacy-preserving collaboration in artificial intelligence. *Nat Mach Intell* 2021;3(12):1081–9. <https://doi.org/10.1038/s42256-021-00421-z>. Publisher: Nature Publishing Group. Available from: <https://www.nature.com/articles/s42256-021-00421-z>.
- [6] Foley P, et al. OpenFL: the open federated learning library. *Phys Med Biol* 2022. <https://doi.org/10.1088/1361-6560/ac97d9>. Available from: <http://iopscience.iop.org/article/10.1088/1361-6560/ac97d9>.
- [7] Beutel DJ, et al. Flower: a friendly federated learning research framework. *arXiv preprint*. Available from: [arXiv:2007.14390](https://arxiv.org/abs/2007.14390), 2020.
- [8] Ryu M, Kim Y, Kim K, Madduri RK. Appfl: open-source software framework for privacy-preserving federated learning. In: 2022 IEEE international parallel and distributed processing symposium workshops (IPDPSW); 2022. p. 1074–83. Available from: <https://doi.ieeecomputersociety.org/10.1109/IPDPSW55747.2022.00175>.
- [9] Roth HR, et al. NVIDIA FLARE: federated learning from simulation to real-world. In: Workshop on federated learning: recent advances and new challenges (in conjunction with NeurIPS 2022); 2022. Available from: <https://openreview.net/forum?id=hD9QaIQTLf>.
- [10] Ibm federated learning. Available from: <https://www.ibm.com/docs/en/cloud-paks/cp-data/4.5.x?topic=models-federated-learning>. [Accessed 22 December 2022].

- [11] An industrial grade federated learning framework. Available from: <https://fate.readthedocs.io/en/latest/>.
- [12] Zhu L, Liu Z, Han S. Deep leakage from gradients. *Adv Neural Inf Process Syst* 2019;32. Available from: [https://proceedings.neurips.cc/paper\\_files/paper/2019/file/60a6c4002cc7b29142def8871531281a-Paper.pdf](https://proceedings.neurips.cc/paper_files/paper/2019/file/60a6c4002cc7b29142def8871531281a-Paper.pdf).
- [13] Geiping J, Bauermeister H, Dröge H, Moeller M. Inverting gradients - how easy is it to break privacy in federated learning? *Adv Neural Inf Process Syst* 2020;33. Available from: [https://proceedings.neurips.cc/paper\\_files/paper/2020/hash/c4ede56bbd98819ae6112b20ac6bf145-Abstract.html](https://proceedings.neurips.cc/paper_files/paper/2020/hash/c4ede56bbd98819ae6112b20ac6bf145-Abstract.html).
- [14] Wei K, et al. Federated learning with differential privacy: algorithms and performance analysis. *IEEE Trans Inf Forensics Secur* 2020;15:3454–69. <https://doi.org/10.1109/TIFS.2020.2988575>.
- [15] Gorre N, et al. MIDRC CRP10 AI interface - an integrated tool for exploring, testing and visualization of ai models. *Phys Med Biol* 2023. Available from: <http://iopscience.iop.org/article/10.1088/1361-6560/acb754>.
- [16] Xie Y, et al. Federatedscope: a flexible federated learning platform for heterogeneity. *Proc VLDB Endow* 2023;16(5):1059–72. <https://doi.org/10.14778/3579075.3579081>.
- [17] Yoo AB, Jette MA, Grondona M. SLURM: simple Linux utility for resource management. In: *Job scheduling strategies for parallel processing*. Revised papers, vol. 2862. 2003. p. 44–60.
- [18] Zhou S, Zheng X, Wang J, Delisle P. Utopia: a load sharing facility for large, heterogeneous distributed computer systems. *Softw Pract Exp* 1993;23(12):1305–36. <https://doi.org/10.1002/spe.4380231203>. Available from: <https://onlinelibrary.wiley.com/doi/abs/10.1002/spe.4380231203>. <https://onlinelibrary.wiley.com/doi/pdf/10.1002/spe.4380231203>.
- [19] Li Z, et al. Appflx: providing privacy-preserving cross-silo federated learning as a service. In: 2023 IEEE 19th international conference on e-science (e-science); 2023. p. 1–4. Available from: <https://doi.ieeecomputersociety.org/10.1109/e-Science58273.2023.10254842>.
- [20] Foster I. Globus online: accelerating and democratizing science through cloud-based services. *IEEE Internet Comput* 2011;15(3):70–3. <https://doi.org/10.1109/MIC.2011.64>.
- [21] Allen B, et al. Software as a service for data scientists. *Commun ACM* 2012;55(2):81–8. <https://doi.org/10.1145/2076450.2076468>.
- [22] La Plante P, et al. A real time processing system for big data in astronomy: applications to hera. *Astron Comput* 2021;36:100489. <https://doi.org/10.1016/j.ascom.2021.100489>. Available from: <https://www.sciencedirect.com/science/article/pii/S2213133721000433>.
- [23] Madduri RK, et al. Experiences building globus genomics: a next-generation sequencing analysis service using galaxy, globus, and Amazon web services. *Concurr Comput Pract Exp* 2014;26(13):2266–79.
- [24] Chard R, et al. funcX: a federated function serving fabric for science. In: *Proceedings of the 29th international symposium on high-performance parallel and distributed computing*; 2020.
- [25] Huerta E, et al. Accelerated, scalable and reproducible ai-driven gravitational wave detection. *Nat Astron* 2021;5(10):1062–8. <https://doi.org/10.1038/s41550-021-01405-0>.
- [26] Wilamowski M, et al. 2-O methylation of RNA cap in sars-cov-2 captured by serial crystallography. *Proc Natl Acad Sci* 2021;118(21):e2100170118. <https://doi.org/10.1073/pnas.2100170118>. Available from: <https://www.pnas.org/doi/abs/10.1073/pnas.2100170118>.
- [27] Baughman M, et al. Tournament-based pretraining to accelerate federated learning. In: *SC '23: proceedings of the SC '23 workshops of the international conference on high performance computing, network, storage, and analysis*; 2023. p. 109–15.
- [28] Tuecke S, et al. Globus auth: a research identity and access management platform. In: 2016 IEEE 12th international conference on e-science (e-science); 2016. p. 203–12.
- [29] Paszke A, et al. Pytorch: an imperative style, high-performance deep learning library. In: *Proceedings of the 33rd international conference on neural information processing systems*; 2019.
- [30] Abadi M, et al. TensorFlow: large-scale machine learning on heterogeneous systems. Software available from: <https://www.tensorflow.org/>, 2015.
- [31] Grinberg M. Flask web development: developing web applications with python. O'Reilly Media, Inc.; 2018.
- [32] Cloud computing services - Amazon web services. Available from: <https://aws.amazon.com/>. [Accessed 6 February 2023].
- [33] Hatamizadeh A, et al. Do gradient inversion attacks make federated learning unsafe? *IEEE Trans Med Imaging* 2023.
- [34] Zhao B, Mopuri KR, Bilal H. iDLG: improved deep leakage from gradients. *CoRR*. Available from: [arXiv:2001.02610](https://arxiv.org/abs/2001.02610), 2020. <http://arxiv.org/abs/2001.02610>.
- [35] Huang Y, Gupta S, Song Z, Li K, Arora S. In: Ranzato M, Beygelzimer A, Dauphin Y, Liang P, Vaughan JW, editors. Evaluating gradient inversion attacks and defenses in federated learning. *Advances in neural information processing systems*, vol. 34. Curran Associates, Inc.; 2021. p. 7232–41. Available from: [https://proceedings.neurips.cc/paper\\_files/paper/2021/file/3b3fff6463464959dcd1b68d0320f781-Paper.pdf](https://proceedings.neurips.cc/paper_files/paper/2021/file/3b3fff6463464959dcd1b68d0320f781-Paper.pdf).
- [36] Yin H, et al. See through gradients: image batch recovery via gradinversion. In: 2021 IEEE/CVF conference on computer vision and pattern recognition (CVPR); 2021. p. 16332–41. Available from: <https://doi.ieeecomputersociety.org/10.1109/CVPR46437.2021.01607>.
- [37] Geng J, et al. Improved gradient inversion attacks and defenses in federated learning. *IEEE Trans Big Data* 2023;1–13. <https://doi.org/10.1109/TBDATA.2023.3239116>.

- [38] Dwork C, McSherry F, Nissim K, Smith A. Calibrating noise to sensitivity in private data analysis. In: Theory of cryptography: third theory of cryptography conference, TCC 2006. Proceedings, vol. 3. 2006. p. 265–84.
- [39] Loshchilov I, Hutter F. Decoupled weight decay regularization. In: 7th international conference on learning representations, ICLR 2019; May 2019. p. 6–9. Available from: <https://openreview.net/forum?id=Bkg6RiCqY7>, 2019.
- [40] Lima EM, et al. Deep neural network-estimated electrocardiographic age as a mortality predictor. *Nat Commun* 2021;12(1):5117. <https://doi.org/10.1038/s41467-021-25351-7>.
- [41] Ball RL, Feiveson AH, Schlegel TT, Starc V, Dabney AR. Predicting “heart age” using electrocardiography. *J Pers Med* 2014;4(1):65–78. <https://doi.org/10.3390/jpm4010065>. Available from: <https://www.mdpi.com/2075-4426/4/1/65>.
- [42] Alday EAP, et al. Classification of 12-lead ECGs: the physionet/computing in cardiology challenge 2020. *Physiol Meas* 2020;41(12):124003. <https://doi.org/10.1088/1361-6579/abc960>. Available from: <https://dx.doi.org/10.1088/1361-6579/abc960>.
- [43] Bycroft C, et al. The uk biobank resource with deep phenotyping and genomic data. *Nature* 2018;562(7726):203–9.
- [44] He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. In: Proceedings of 2016 IEEE conference on computer vision and pattern recognition; 2016.
- [45] Kingma DP, Ba J. Adam: a method for stochastic optimization. In: 3rd international conference on learning representations, ICLR 2015. Conference track proceedings; 2015. Available from: <http://arxiv.org/abs/1412.6980>.
- [46] Roberts M, et al. Common pitfalls and recommendations for using machine learning to detect and prognosticate for covid-19 using chest radiographs and ct scans. *Nat Mach Intell* 2021;3:199–217. <https://doi.org/10.1038/s42256-021-00307-0>.
- [47] Deng J, et al. Imagenet: a large-scale hierarchical image database. In: 2009 IEEE conference on computer vision and pattern recognition; 2009. p. 248–55.
- [48] Tan AZ, Yu H, Cui L, Yang Q. Towards personalized federated learning. *IEEE Trans Neural Netw Learn Syst* 2022;1–17. <https://doi.org/10.1109/TNNLS.2022.3160699>.
- [49] Kulkarni V, Kulkarni M, Pant A. Survey of personalization techniques for federated learning. In: 2020 fourth world conference on smart trends in systems, security and sustainability (WorldS4); 2020. p. 794–7.
- [50] Hu Q, Drukker K, Giger ML. Role of standard and soft tissue chest radiography images in deep-learning-based early diagnosis of COVID-19. *J Med Imag* 2021;8(S1):014503. <https://doi.org/10.1117/1.JMI.8.S1.014503>.
- [51] Ioffe S, Szegedy C, Bach F, Blei D. Batch normalization: accelerating deep network training by reducing internal covariate shift. In: Bach F, Blei D, editors. Proceedings of the 32nd international conference on machine learning. Proceedings of machine learning research, vol. 37. Lille, France: PMLR; 2015. p. 448–56. Available from: <https://proceedings.mlr.press/v37/ioffe15.html>.
- [52] Li Y, Wang N, Shi J, Liu J, Hou X. Revisiting batch normalization for practical domain adaptation. In: International conference on learning representations workshop; 2017. Available from: <https://openreview.net/forum?id=BJuysoFeg>.
- [53] Wang D, Shelhamer E, Liu S, Olshausen B, Darrell T. Tent: fully test-time adaptation by entropy minimization. Available from: <https://openreview.net/forum?id=uXl3bZLkr3c>, 2021.
- [54] Asraf Amanullah, Islam Zabirul. Pneumonia and normal chest X-ray PA dataset, mendeley data, v1; 2021.
- [55] Chen Y, Qin X, Wang J, Yu C, Gao W. Fedhealth: a federated transfer learning framework for wearable healthcare. *IEEE Intell Syst* 2020;35(4):83–93. <https://doi.org/10.1109/MIS.2020.2988604>.
- [56] Jiang L, Lin T. Test-time robust personalization for federated learning. In: The eleventh international conference on learning representations; 2023. Available from: <https://openreview.net/forum?id=3aBuJEza5sq>.
- [57] Xie C, Koyejo S, Gupta I. Asynchronous federated optimization. Available from: [arXiv:1903.03934](https://arxiv.org/abs/1903.03934), 2020. <https://arxiv.org/abs/1903.03934>.
- [58] Li Z, et al. Fedcompass: efficient cross-silo federated learning on heterogeneous client devices using a computing power-aware scheduler. In: The twelfth international conference on learning representations; 2024. Available from: <https://openreview.net/forum?id=msXxrttLOi>.
- [59] Bauer A, et al. The globus compute dataset: an open function-as-a-service dataset from the edge to the cloud. *Future Gener Comput Syst* 2024;153:558–74. <https://doi.org/10.1016/j.future.2023.12.007>.
- [60] Li Z, et al. Advances in appfl: a comprehensive and extensible federated learning framework. Available from: <https://arxiv.org/abs/2409.11585>, 2024. [arXiv:2409.11585](https://arxiv.org/abs/2409.11585).
- [61] Geng J, Kanwal N, Jaatun MG, Rong C. DID-EFed: facilitating federated learning as a service with decentralized identities. In: Evaluation and assessment in software engineering; 2021. p. 329–35.
- [62] Brecko A, Kajati E, Koziorek J, Zolotova I. Federated learning for edge computing: a survey. *Appl Sci* 2022;12(18). <https://doi.org/10.3390/app12189124>. Available from: <https://www.mdpi.com/2076-3417/12/18/9124>.
- [63] AI-powered solutions for healthcare. Available from: <https://www.nvidia.com/en-us/clarra/>. [Accessed 22 December 2022].