Experience Replay as an Effective Strategy for Optimizing Decentralized Federated Learning

Matteo Pennisi, Federica Proietto Salanitri, Giovanni Bellitto, Concetto Spampinato, Simone Palazzo
PeRCeiVe Lab, University of Catania, Catania, Italy
www.perceivelab.com

Bruno Casella, Marco Aldinucci
Parallel Computing group, University of Turin, Turin, Italy
https://alpha.di.unito.it/

Abstract

Federated and continual learning are training paradigms addressing data distribution shift in space and time. More specifically, federated learning tackles non-i.i.d data in space as information is distributed in multiple nodes, while continual learning faces with temporal aspect of training as it deals with continuous streams of data. Distribution shifts over space and time is what it happens in real federated learning scenarios that show multiple challenges. First, the federated model needs to learn sequentially while retaining knowledge from the past training rounds. Second, the model has also to deal with concept drift from the distributed data distributions. To address these complexities, we attempt to combine continual and federated learning strategies by proposing a solution inspired by experience replay and generative adversarial concepts for supporting decentralized distributed training. In particular, our approach relies on using limited memory buffers of synthetic privacy-preserving samples and interleaving training on local data and on buffer data. By translating the CL formulation into the task of integrating distributed knowledge with local knowledge, our method enables models to effectively integrate learned representation from local nodes, providing models the capability to generalize across multiple datasets.
We test our integrated strategy on two realistic medical image analysis tasks — tuberculosis and melanoma classification — using multiple datasets in order to simulate realistic non-i.i.d. *medical data scenarios. Results show that our approach achieves performance comparable to standard (non-federated) learning and significantly outperforms state-of-the-art federated methods in their centralized (thus, more favourable) formulation.*

1. Introduction

The significance of continual learning for supporting federated learning is underscored by the recent advancements in deep learning in high-stake application domains such as medical image analysis. Indeed, while data-driven approaches have shown promise in these domains, the availability of large-scale datasets is crucial for the reliability and effectiveness of resulting models. Nevertheless, curating large datasets is complex, with slow data collection, integration challenges, and privacy concerns hindering progress. These limitations impact the quality, generalizability, and bias of models trained on local datasets, limiting their ability to handle future data distribution shifts.

To address the lack of large-scale datasets and emerging data privacy concerns, federated learning has emerged as an effective and viable solution. It enables distributed training across multiple nodes, each with its private dataset, without explicitly sharing data. However, federated learning techniques perform best when dataset distributions are approximately independent and identically distributed (i.i.d.), a condition rarely met in practice due to variations in data acquisition and characteristics. Moreover, the standard FL formulation, foreseeing the presence of a central coordinating node, raises privacy concerns and potential single points of failure.

To overcome these challenges, in this paper, we propose a learning strategy leveraging experience replay and generative models for supporting decentralized training. Our approach introduces a principled method for training local models that ultimately converge to similar decisions, without relying on a shared model architecture or central coordination. Additionally, despite not being the primary focus of this work, privacy preservation is achieved through the transmission of synthetic data generated in a way to obfuscate real data patterns.

In the proposed approach, multiple nodes initially train their local models and a generative adversarial network (GAN) on their respective datasets. The GAN is employed to generate privacy-preserving synthesized versions of the datasets. Once local training is complete, a node sends its model and the generated synthetic data buffer to a random node in the network. The receiving node adapts the incoming model using its own data and the received buffer data to limit forgetting.

We test our approach on two medical tasks, simulating non-i.i.d. scenarios, such as tuberculosis classification from X-ray data and melanoma classification using skin images. The experimental results demonstrate that it achieves performance comparable to centralized training on all real data, outperforming existing federated learning methods.In summary, the contributions of this work are the following:

- A decentralized federated learning strategy, based on continual learning principles, outperforming centralized approaches and achieving performance similar to standard training settings.
- A principled strategy for knowledge transfer that takes into account feature semantics during model merging, avoiding interference and enabling the reuse of important features from models received by each node.
- The integration of continual learning into federated frameworks improves adaptability and performance in non-i.i.d. data scenarios. This integration paves the way for more efficient distributed systems, reducing the need for central coordination and model homogeneity.

2. Related Work

Federated Learning (FL) [22] has recently emerged as a collection of distributed learning approaches that enable nodes to maintain the privacy of their training data while collaboratively creating a shared model. Typically, in a standard FL scenario, a central server distributes a model to a group of client nodes; each node fine-tunes the model on its local data and then sends the updated model back to the server. The server aggregates these updates from all nodes to form the global model, which is then iteratively sent back to the nodes until convergence. The medical domain, with its specific constraints on data sharing, serves as an ideal testing ground for evaluating federated learning methods [19, 25, 5, 7]. The most common way to aggregate information from multiple nodes is by averaging the local models of each client, as proposed in FedAvg [22] and FedProx [18]. However, statistical data heterogeneity poses a challenge as it may lead to catastrophic forgetting [14, 10]. To address this issue, FedCurv [26] introduces a penalty term to the loss function to drive the local models towards a shared optimum. FedMA [30], on the other hand, constructs a shared global model in a layer-wise manner by matching and averaging hidden elements with similar feature extraction signatures. Our approach differs from existing feature integration methods in that it doesn't involve averaging model updates or gradients, which could be vulnerable to input reconstruction attacks [9, 32, 34]. Instead, each node seeks to learn features that perform well on its local dataset while retaining knowledge from other nodes in a more principled manner than parameter averaging. Recently, federated personalized methods, like FedBN [20], aim to fit the global model to local data while keeping certain components, such as batch normalization layers, private, and aggregating other model parameters at the central node. However, using a central node for aggregating local updates eases communication with many clients but has downsides: it is a single failure point, it can become a bottleneck with more clients [21], and might not be suitable in all collaborative learning contexts [14]. Our paper focuses on *decentralized federated learning*, where the central node is replaced by peer-to-peer communication between clients. In this setup, there is no global shared model as in standard FL, but the communication protocol is designed so that all local models approximately converge to the same solution. Decentralized learning is particularly suitable for applications in the medical domain, where the number of nodes (i.e., institutions) is relatively low. However, research in this area is ongoing, and no definitive solutions have been established. Some existing works propose Bayesian approaches to learn a shared model over a graph of nodes by aggregating information from local data and each node's one-hop neighbors [17]. Others, like the secure weight averaging algorithm [31], ensure that model parameters are not shared between nodes, but they all converge to the same numerical values, with the disadvantages associated with parameter averaging when dealing with *non-i.i.d.* data distributions. Other approaches implement different communication strategies based on parameter sharing, such as decentralized variants of FedAvg [28, 22]. While many existing solutions do not specifically target the medical domain and often use toy datasets like MNIST and CIFAR10, there are two works [25, 8], similar in the decentralized learning spirit to ours, that present use cases of decentralized and swarm learning for medical image segmentation. However, like other approaches, they also adopt simple parameter averaging to integrate features or predictions from multiple nodes.

3. Method

3.1. Overview

An overview of our proposed method is shown in Fig. 1. A *federation* can be defined as a set of N peer nodes, each

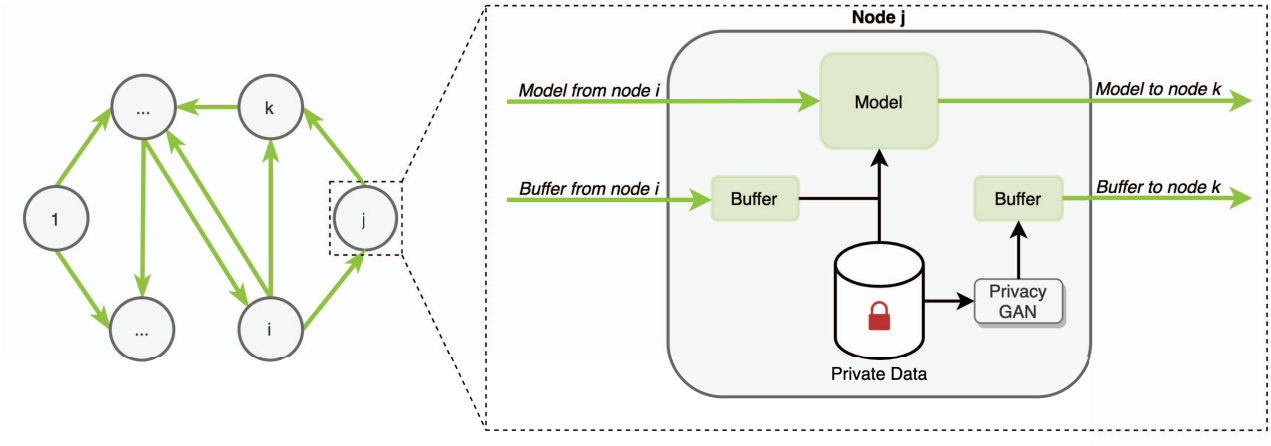

Figure 1: **Overview of the proposed learning strategy.** Each node initially trains a *privacy-preserving GAN*, that is used to sample synthetic data from the local distribution, without retaining features that may be used to identify patients. Then, each node iteratively receives the local model and a buffer of synthetic samples from a random node, and fine-tunes the received model on its own private data, using the buffer to prevent forgetting of previously-learned features.

with a private data set. Starting from the private dataset, each node creates a synthetic *privacy-preserving* version thereof, which is then used as a basis for the replay strategy of the experience.

During each iteration of the decentralized federated training, every node within the federation is provided with a model and a buffer of synthetic samples from a random node of the federation. The model at each node is then fine-tuned using both its private dataset and the buffer data, employing an experience replay method that is widely used in continual learning settings (e.g., [1]). The objective is to learn features that are transferable across nodes and capable of handling non-independent and identically distributed (non-i.i.d.) distributions. At the conclusion of each round, after completing multiple training iterations, the locally-trained model is sent to a successor node chosen randomly, along with a buffer containing local synthetic samples. The entire process is then repeated until convergence.

The proposed approach is tested on the task of federated learning for medical image classification. Consequently, the presented method is described in the context of this task, but the overall approach can be applied to any other task without sacrificing its generality.

3.2. Federated learning with experience replay

Existing methods in federated learning rely primarily on parameter averaging, such as FedAvg [22]. However, this approach presents major concerns when it comes to effectively integrating knowledge from multiple sources. One key challenge is the misalignment of feature locations across different models, which can be further disrupted by updates. Consequently, the models tend to converge slowly towards consensus. In theory, two models could potentially learn the same set of features, but at different locations within the same layer, leading to their cancellation when averaged. This problem becomes even more evident in decentralized scenarios where there is no central authority to enforce global agreement on node features.

In our methodology, we tackle this issue by employing continual learning strategies [6], which enable us to perform a task using a non-*i.i.d.* data stream while preserving previously-acquired knowledge. This ensures that local models can reutilize and adapt features effectively, enabling them to serve both current and prior tasks. Similarly, in the context of federated learning, the goal is to train a global model using distinct non-*i.i.d.* data distributions sourced from various nodes.

In our ER-based federated learning strategy, a node receives another node's model and surrogate data (generated through a privacy-preserving GAN) — the *"previous task"* — and fine-tunes that model on its own private date — the *"current task"* — while using received synthetic data to retain/adapt from the knowledge learned by the previous node. Let $\mathcal{T} = (T_1, T_2, \ldots, T_N)$ be a *set of N tasks*, where T_i is the task to be solved on node n_i.
Task T_i aims at optimizing a model M_i, with parameters $\boldsymbol{\theta}_i$, on dataset $\mathcal{D}_i$ of node n_i.
The buffer $\mathcal{B}_i$ is the set of synthetic images, generated by a privacy-preserving GAN $\mathcal{G}_i$ using data $\mathcal{D}_i$ available on node n_i.
Training is organized in parallel *rounds*. At the end of round r, each node n_i produces a model M_i^r trained on dataset $\mathcal{D}_i$ and on a buffer $\mathcal{B}_j$, received from another node n_j, to optimize an objective $\mathcal{L}$, i.e., to find $\arg\min_{\boldsymbol{\theta}_i^r} = \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim\mathcal{D}_i\cup\mathcal{B}_j}[\mathcal{L}(M_i^r(\mathbf{x}, \boldsymbol{\theta}_i^r), \mathbf{y})]$. For each training round, all nodes in parallel share to/receive from

other nodes, buffer of synthetic images, and trained models.

In the following, we describe our method (whose graphical representation is given in Fig. 1) from the point of view of a single node n_j. At round r, the node n_j is trained on a new task T_j using the dataset $\mathcal{D}_j$, in a continual learning setting by fine-tuning the model $\mathcal{M}_i^{r-1}$ (with parameters $\boldsymbol{\theta}_i^{r-1}$) received by another model of the federation at previous round $r-1$ on the node data ,$\mathcal{D}_j$, and on the incoming buffer $\mathcal{B}_i$. Therefore, conversely to alternative federated learning methods, individual nodes do not possess their own local models. Instead, during the decentralized learning process, a node continually receives a model from another node, incorporates its local information while retaining previously-acquired knowledge, and then forwards the updated model to the subsequent node.

The loss function for the model $\mathcal{M}_j^r$ in node n_j at round r is thus defined as follows:

$$\begin{aligned}\mathcal{L}(\boldsymbol{\theta}_j^r) = \lambda\, \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim\mathcal{D}_j}[\mathcal{L}(M_j^r(\mathbf{x},\boldsymbol{\theta}_j^r),\mathbf{y})]+ \\ +(1-\lambda)\, \mathbb{E}_{(\mathbf{x}',\mathbf{y}')\sim\mathcal{B}_i}[\mathcal{L}(M_j^r(\mathbf{x}',\boldsymbol{\theta}_j^r),\mathbf{y}')]\end{aligned} \tag{1}$$

where λ balances the ratio between real samples from the local dataset D_i and replayed synthetic samples from node n_i.

After optimizing the $\mathcal{L}(\boldsymbol{\theta}_j^r)$ objective through mini-batch gradient descent, the model $M_j^r(\boldsymbol{\theta}_j^r)$, with updated parameters $\boldsymbol{\theta}_j^r$, is sent to a random node n_k of the federation, along with a buffer $\mathcal{B}_j$ of locally-generated synthetic samples.

Then, the general federated model $\mathcal{M}$, after all training rounds, is given by the union of all the N node models, i.e., $\mathcal{M} = M_1 \cup M_2 \cup \cdots \cup M_N$. However, experimental results, reported below in Sect. 4, demonstrate that all models converge to similar decisions, thus each node model can be considered as a general model for the entire network.

3.3. Privacy-preserving GAN

In the proposed method, nodes exchange both models and data, implementing a knowledge transfer procedure based on experience replay (see Sect. 3.2). Of course, sharing real samples would go against federated learning policies; hence, exchanged samples are generated so that they are representative of the local data, while taking precautions against privacy violations — which may happen, for instance, if the generative model overfits the source dataset.

Formally, we assume that each node n_i, from a set of N nodes, owns a private dataset $\mathcal{D}_i = \{(\mathbf{x}_1,\mathbf{y}_1),(\mathbf{x}_2,\mathbf{y}_2),\ldots,(\mathbf{x}_M,\mathbf{y}_M)\}$, where each $\mathbf{x}_j \in \mathcal{X}$ represents a sample in the dataset, and each $\mathbf{y}_j \in \mathcal{Y}$ represents the corresponding target. The local dataset is used to train a conditional GAN [23], consisting of a generator G, that synthesizes samples for a given label by modeling $P(\mathbf{x}|\mathbf{y},\mathbf{z})$), where $\mathbf{z} \in \mathcal{Z}$ is a random vector sampled from the generation latent space, and a discriminator D, which outputs the probability of an input sample being real, modeling $P(\text{real}|\mathbf{x},\mathbf{y})$.

While theoretical proofs demonstrate that, upon convergence, the distribution learned by the generator matches and generalizes from the initial data distribution [10], GAN architectures may, unfortunately, encounter training anomalies such as mode collapse and overfitting. As a result, the standard GAN formulation could generate samples that closely resemble the original ones, a situation deemed unacceptable within the context of federated learning. In order to mitigate this risk, we employ a *privacy-preserving strategy*, similar to the one proposed in [24], that aims at generating samples that do not retain potentially sensitive information, but still include features that are relevant to target tasks. Specifically, in the GAN training, we integrate a privacy-preserving loss, which penalizes the model based on the similarity between real and synthetic sample pairs. This similarity is quantified using the LPIPS metric [33], a measure that captures perceptual similarity by evaluating the distance between feature vectors extracted from a pre-trained VGG model [27]. In practice, given a batch of real samples $\mathbf{x}^{(r)}$ and a batch of synthetic samples $\mathbf{x}^{(s)}$, the privacy-preserving loss is computed as:

$$\mathcal{L}_{\text{PP}} = \frac{1}{b}\sum_{\mathbf{x}^{(r)}}\sum_{\mathbf{x}^{(s)}} d_L\left(\mathbf{x}^{(r)},\mathbf{x}^{(s)}\right), \tag{2}$$

where d_L is the LPIPS distance.

The resulting new loss for the Generator is a combination between standard generator loss with the one in Eq. 2.

4. Experimental Results

We test our learning strategy on two applications simulating real-case scenarios with multiple centers holding, and not sharing, their own data: 1) tuberculosis classification from X-ray images using two different datasets, and 2) skin lesion classification with three different datasets. In this section, we present the employed benchmarks, the training procedure and report the obtained results to demonstrate the advantages of the proposed approach w.r.t. the state-of-the-art.

4.1. Datasets

X-ray image datasets for tuberculosis classification. We assume that the federation comprises two distinct nodes: one contains the Montgomery County X-ray set, while the other contains the Shenzhen Hospital X-ray set [2, 13, 12]. The Montgomery Set includes 138 frontal chest X-ray images, with 80 negatives and 58 positives. These images were captured using a Eureka stationary machine (CR) at either 4020×4892 or 4892×4020 pixel resolution. On the other hand, the Shenzhen dataset was collected using

a Philips DR Digital Diagnostic system. It contains 662 frontal chest X-ray images, consisting of 326 negatives and 336 positives, with a variable resolution of approximately 3000×3000 pixels.

Skin lesion classification. We use the ISIC 2019 challenge dataset, which comprises 25,331 skin images categorized into nine different diagnostic categories. The federation is organized with three nodes because the data comes from three distinct sources: 1) BCN20000 [4] dataset, which contains 19,424 images of skin lesions captured between 2010 and 2016 at the Hospital Clínic in Barcelona. 2) HAM10000 dataset [29] includes 10,015 skin images collected over a 20-year period from two sites: the Department of Dermatology at the Medical University of Vienna, Austria, and the skin cancer practice of Cliff Rosendahl in Queensland, Australia.3) MSK4 [3] dataset, an anonymous dataset that consists of 819 skin lesion samples.

To streamline the problem, we focus only on the melanoma class, converting it into a binary classification task.

For each task and dataset, 80% of the available data is used to train both the privacy-preserving GAN and the classification model. The remaining 20% of each dataset is reserved for the test set. To prevent performance biases due to class imbalance, the test sets are balanced with respect to the labels. In evaluating the federated methods, including state-of-the-art ones, model selection is performed using 5-fold cross-validation on the training set. A grid search is conducted on various hyperparameters, such as the number of training rounds, number of rounds per epoch, learning rate, and for FedProx [18], the μ hyperparameter is also included.

4.2. Training procedure and metrics

4.2.1 Federated training

In all settings, we employ the ResNet-18 model as the classification model. The model is trained by minimizing the cross-entropy loss using the Adam optimizer with mini-batch gradient descent. The mini-batch size is set to 32 for the Shenzhen dataset, 8 for the Montgomery dataset, and 64 for the skin lesion datasets. Through cross-validation, we determined that the optimal learning rate is 10^{-4}. To enhance the training process and increase data diversity, data augmentation techniques are applied: random horizontal flip is employed for all datasets, while for skin lesion images, we additionally applied random 90-degree rotations. All images are resized to a standard size of 256×256 pixels. In order to balance the real and synthetic samples used during training, we control the ratio between them through the parameter λ in Eq. 1. In this case, λ is set to 0.5 for all experiments, meaning that each mini-batch contains an equal number of real and synthetic images. This approach ensures that our method performs the same number of optimization steps as other conventional approaches that do not utilize synthetic data.

The node federation is trained for R rounds. In our implementation, at each round, nodes are randomly ordered to establish each node's predecessor and successor: given our focus on medical applications, we can assume that the number of nodes is low enough that synchronization is not an issue. However, asynchronicity can be achieved by assuming that nodes can store incoming data in a queue: if the distribution of successor nodes is uniform and computation times are similar for all nodes, this is on average equivalent to the synchronous case. The number of rounds R and epochs E for our method on tuberculosis and melanoma classification tasks are set both to 100, according to the 5-fold cross-validation. Buffer size is set for all experiments to 512.

4.2.2 GAN training

We would like to emphasize that GAN training is conducted **prior to** the federated learning process, using only training data while excluding test samples, as mentioned in Section 4.1. For our privacy-preserving GAN, we employ StyleGAN2-ADA [16] as the backbone due to its suitability in low-data regimes and its exceptional generation capabilities. The training is carried out in two steps:

1. Initially, the GAN is trained without any privacy-preserving loss to facilitate learning high-quality visual features. This training phase is stopped if the FID does not improve for 10,000 iterations.

2. Then, we introduce the privacy-preserving loss and fine-tune the model to limit the embedding of patient-specific patterns in the GAN latent space. In this case, we employ a criterion to stop training if the FID increases by a factor of 2.5 compared to its value obtained in the first step.

For classification purposes, the GANs are trained in a label-conditioned manner, with a mini-batch size of 32 and a learning rate of 0.0025 for both the generator and the discriminator. Early-stopping criteria are based on the Frêchet Inception Distance (FID) [11] between real and synthetic distributions. Regarding the α parameter in Eq.2, we experimented with various values of α (0, 0.5, 1, 1.5, 2, and 3) and found that a value of 1 provides the best trade-off between image generation quality and pairwise LPIPS distance[33] across all tested datasets. To quantitatively evaluate privacy preservation, we compute the average LPIPS distance between each real image and its closest synthetic sample through latent space projection (as described in Sect. 4.4). A higher LPIPS value indicates a lower possibility of reconstructing real images from the generator, thereby indicating better privacy preservation.

Table 1: **Comparison between proposed method, centralized baselines and state-of-the-art methods**.

Methods	Tuberculosis			Melanoma			
	Shenzhen	**Montgomery**	**Mean**	**BCN**	**HAM**	**MSK4**	**Mean**
Standalone	82.31	90.00	86.16	82.90	82.55	69.75	78.40
Centralized training	82.77	77.67	80.22	78.80	82.90	71.23	77.64
Centralized training with synthetic data only	76.92	79.33	78.13	60.71	61.09	61.23	61.01
Centralized training with synthetic data and real data	85.38	86.67	86.03	81.53	80.44	73.46	78.48
FedAvg [22]	72.31	83.33	77.82	77.55	75.15	67.28	73.33
FedProx [18]	78.46	76.67	77.56	78.80	81.87	64.81	75.16
FedBN [20]	63.08	70.00	66.54	**82.19**	81.12	59.26	74.19
Ours	**80.15**	**86.67**	**83.41**	82.11	**84.58**	**68.40**	**78.36**

Table 2: **Accuracy convergence among distributed node models**. Each local model is evaluated on all test sets of the federation in order to measure convergence and generalization (lower standard deviation corresponds to higher convergence).

	Dataset	Ours	Standalone
Tuberculosis	**Shenzhen**	80.54 ± 1.20	66.15 ± 22.84
	Montgomery	85.67 ± 2.36	70.00 ± 28.28
Melanoma	**BCN**	82.87 ± 1.22	65.06 ± 19.68
	HAM	84.45 ± 0.75	59.94 ± 20.47
	MSK4	67.78 ± 1.28	65.43 ± 5.05

4.3. Federated learning performance

We evaluate the performance (in terms of classification accuracy) in the non-*i.i.d.* setting, and compare it to several centralized baselines, namely:

- **Centralized training**: all datasets are merged in a single node where all training happens. In this setting, no federated learning constraints are applied.
- **Centralized training with synthetic data only**. In this setting, each node trains a privacy-preserving GAN model and shares a synthetic version of its own data with the central node, where global training is performed. In this case, we aim to assess how much information is retained by synthetic data to support classification.
- **Centralized training with synthetic and real data.** This setting is a combination of the previous two: real and synthetic samples are centrally merged and used for training a global classifier. This scenario measures the contribution of synthetic data as a data augmentation approach.

We also evaluate our learning strategy against the standard training of individual node models, referred to as "Standalone" training. Classification accuracy is determined by assessing individual node models using their own distinct datasets. The outcomes, detailed in Table 1, reveal that Standalone training seems to yield the best results. Centralized strategies generally fall short compared to Standalone training, due to the non-independent and identically distributed (non-*i.i.d.*) nature of the data. However, when the centralized approach is trained with original data augmented with synthetic samples, its classification accuracy is on par with the Standalone training, possibly due to the learned generative latent spaces that likely tend to smooth different modes of non-*i.i.d.* data. Our method, on the other hand, surpasses its centralized counterpart but trails slightly behind Standalone training (with a difference of 1.5 percentage points). While this might seem like a drawback initially, it is crucial to remember that the aim of federated learning is to construct a model capable of better generalization, by harnessing the various data distributions present within the federation to potentially accommodate future data drifts. To assess the generalization capacity of the trained models, we analyze how a local node model performs on test datasets from other nodes. These findings are detailed in Table 2, and indicate a high average accuracy, coupled with a low standard deviation, suggesting that each node model performs just as well on its own dataset as it does on others' datasets (i.e., all node models converge to similar decisions). In contrast, Standalone training results in significantly lower accuracy and higher standard deviation than our method, making it a less effective strategy for achieving the desired generalization capabilities.

Furthermore, we evaluate our approach against state-of-the-art federated learning methods, such as: a) centralized federated methods like FedAvg [22] and FedProx [18], which are generally known to outperform decentralized methods [28, 17], and b) a personalized method, FedBN [20]. For a fair evaluation, we employ the official code repository[1] for FedBN [20] and the selection of hyperparameters for the tested datasets was conducted through a grid search on training rounds/epochs, learning rate, and μ for FedProx [18], using a 5-fold cross-validation as to our method.

[1] https://github.com/med-air/FedBN

Table 3: **Classification Accuracy w.r.t. buffer size**. Each local model is evaluated on all test sets of the federation in order to measure convergence and generalization (lower standard deviation corresponds to higher convergence).

	Node Convergence	
Buffer	**Shenzhen**	**Montgomery**
0	70.62 ± 11.97	80.33 ± 10.84
256	80.46 ± 2.96	81.67 ± 4.24
512	80.54 ± 1.20	85.67 ± 2.36
1024	82.23 ± 1.31	86.00 ± 3.01
2048	82.08 ± 1.39	88.67 ± 2.97

The outcomes for the tuberculosis and melanoma tasks, presented in Table 1, demonstrate that our proposed method surpasses all other techniques in comparison. Notably, our learning strategy outperforms: a) *centralized methods*, such as FedAvg [22] and FedProx [18], indicating that experience replay serves as a more efficient feature aggregation method than basic parameter averaging; and b) personalized methods like FedBN [20], which affects a restricted portion of feature representation (i.e., input layer distributions), whereas our strategy adjusts the entire model to suit both local and remote tasks.

Experience replay proves effective in federated learning, aiding in merging features from varied data distributions. We also evaluated its influence using different buffer sizes. Results for the tuberculosis task, shown in Table 3, highlight the buffer's role in performance and model consistency. Without it, performance drops and variability increases. While performance rises with buffer size, gains plateau after 512. Thus, considering data sharing and communication costs, we opted for a 512-size buffer.

4.4. Privacy-preserving performance

In this section, we evaluate the amount of information from real samples that our privacy-preserving Generator retains using the projection method suggested in [15]. Given an actual image $\mathbf{x}$, we identify an intermediate latent point $\mathbf{w}$ such that the generated image $G(\mathbf{w})$ closely resembles $\mathbf{x}$. This is achieved by optimizing $\mathbf{w}$ to reduce the LPIPS distance [33] between $\mathbf{x}$ and $G(\mathbf{w})$.

In practical terms, we perform backprojection for each image in the dataset used for GAN training to identify its closest synthetic counterpart, and measure the LPIPS distance between the original and projected images. Fig. 2 presents the histograms of the resultant distances on the Shenzhen dataset, using GAN models trained both with and without our proposed privacy-preserving loss (for fairness, both models start the backprojection from the same $\mathbf{w}$). The histograms reveal that standard GAN training tends to yield distances nearer to 0, substantiating that real images are indeed incorporated into the generator latent space. However, our model significantly counters this issue, by generating samples that are considerably different from the original ones.

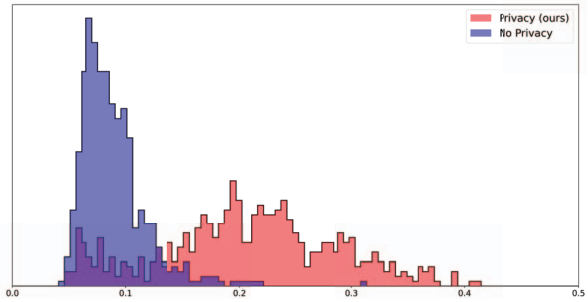

Figure 2: **Quantitative analysis of privacy-preserving generation**. In blue, LPIPS distance histogram between real images and the corresponding images obtained through latent space projection using a GAN trained without the proposed privacy-preserving loss. In red, LPIPS distance histogram between real images and the closest images generated with the proposed approach.

5. Conclusion

In this study, we introduce a decentralized federated learning framework that supersedes the conventional parameter averaging with a more methodical feature integration strategy leveraging the combination of experience replay and privacy-preserving generative models. Nodes share information by exchanging local models and buffers of synthetic samples; local model updates are performed in such a way as to promote the adoption and modification of features learned by other nodes, thus mitigating any potentially disruptive effects brought about by indiscriminate feature averaging. Our empirical results indicate that our method surpasses current state-of-the-art centralized strategies in non-*i.i.d.* scenarios, a common setting in the medical field. In future work, we plan to investigate some unexplored features of our method. For instance, unlike existing methods based on parameter averaging, our strategy does not require that all nodes adopt the same model architecture. Model heterogeneity could be used to establish a shared ensemble and fuse diverse feature learning capabilities.

Acknowledgements

This research was supported by the PNRR MUR project PE0000013-FAIR.Matteo Pennisi is a PhD student enrolled in the National PhD in Artificial Intelligence, XXXVII cycle, course on Health and life sciences, organized by Università Campus Bio-Medico di Roma.

References

[1] Pietro Buzzega et al. Dark experience for general continual learning: a strong, simple baseline. *NeurIPS*, 2020. 3

[2] Sema Candemir et al. Lung segmentation in chest radiographs using anatomical atlases with nonrigid registration. *IEEE TMI*, 33(2):577–590, 2013. 4

[3] Noel CF Codella et al. Skin lesion analysis toward melanoma detection. In *IEEE ISBI*, 2018. 5

[4] Marc Combalia et al. Bcn20000: Dermoscopic lesions in the wild. *arXiv:1908.02288*, 2019. 5

[5] Ittai Dayan et al. Federated learning for predicting clinical outcomes in patients with COVID-19. *Nature medicine*, 27, 2021. 2

[6] Matthias Delange et al. A continual learning survey: Defying forgetting in classification tasks. *IEEE PAMI*, 2021. 3

[7] Ines Feki et al. Federated learning for COVID-19 screening from chest x-ray images. *Applied Soft Computing*, 106:107330, 2021. 2

[8] Zheyao Gao, Fuping Wu, Weiguo Gao, and Xiahai Zhuang. A new framework of swarm learning consolidating knowledge from multi-center non-iid data for medical image segmentation. *IEEE TMI*, pages 1–1, 2022. 2

[9] Jonas Geiping et al. Inverting gradients-how easy is it to break privacy in federated learning? *NeurIPS*, 2020. 2

[10] Ian J Goodfellow et al. An empirical investigation of catastrophic forgetting in gradient-based neural networks. *arXiv:1312.6211*, 2013. 2, 4

[11] Martin Heusel et al. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *NeurIPS*, 2017. 5

[12] Stefan Jaeger et al. Automatic tuberculosis screening using chest radiographs. *IEEE TMI*, 33(2):233–245, 2013. 4

[13] Stefan Jaeger et al. Two public chest x-ray datasets for computer-aided screening of pulmonary diseases. *Quantitative imaging in medicine and surgery*, 4(6):475, 2014. 4

[14] Peter Kairouz et al. Advances and open problems in federated learning. *Foundations and Trends in Machine Learning*, 14, 2021. 2

[15] Tero Karras et al. Analyzing and improving the image quality of stylegan. In *CVPR*, 2020. 7

[16] Tero Karras et al. Training generative adversarial networks with limited data. In *NeurIPS*, 2020. 5

[17] Anusha Lalitha et al. Peer-to-peer federated learning on graphs. *arXiv:1901.11173*, 2019. 2, 6

[18] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. *Proceedings of Machine Learning and Systems*, 2:429–450, 2020. 2, 5, 6, 7

[19] Wenqi Li et al. Privacy-preserving federated brain tumour segmentation. In *International workshop on machine learning in medical imaging*, pages 133–141. Springer, 2019. 2

[20] Xiaoxiao Li, Meirui Jiang, Xiaofei Zhang, Michael Kamp, and Qi Dou. Fedbn: Federated learning on non-iid features via local batch normalization. *arXiv:2102.07623*, 2021. 2, 6, 7

[21] Xiangru Lian et al. Can decentralized algorithms outperform centralized algorithms? a case study for decentralized parallel stochastic gradient descent. *NeurIPS*, 2017. 2

[22] Brendan McMahan et al. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pages 1273–1282. PMLR, 2017. 2, 3, 6, 7

[23] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *arXiv:1411.1784*, 2014. 4

[24] Matteo Pennisi, Federica Proietto Salanitri, Giovanni Bellitto, Simone Palazzo, Ulas Bagci, and Concetto Spampinato. A privacy-preserving walk in the latent space of generative models for medical applications, 2023. 4

[25] Abhijit Guha Roy et al. Braintorrent: A peer-to-peer environment for decentralized federated learning. *arXiv:1905.06731*, 2019. 2

[26] Neta Shoham et al. Overcoming forgetting in federated learning on non-iid data. *arXiv:1910.07796*, 2019. 2

[27] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015. 4

[28] Tao Sun, Dongsheng Li, and Bao Wang. Decentralized federated averaging. *arXiv:2104.11375*, 2021. 2, 6

[29] Philipp Tschandl, Cliff Rosendahl, and Harald Kittler. The HAM10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions. *Scientific data*, 5(1):1–9, 2018. 5

[30] Hongyi Wang et al. Federated learning with matched averaging. *arXiv:2002.06440*, 2020. 2

[31] Tobias Wink and Zoltan Nochta. An approach for peer-to-peer federated learning. In *2021 51st Annual IEEE/IFIP DSN-W*, 2021. 2

[32] Liyang Xie, Kaixiang Lin, Shu Wang, Fei Wang, and Jiayu Zhou. Differentially private generative adversarial network. *arXiv:1802.06739*, 2018. 2

[33] Richard Zhang et al. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018. 4, 5, 7

[34] Ligeng Zhu et al. Deep leakage from gradients. *NeurIPS*, 2019. 2