Using Personalized Federated Learning to Train Diffusion Models

Marios Hadjigeorgiou

Supervisors: Bart Cox, Jérémie Decouchant

EEMCS, Delft University of Technology, The Netherlands

A Thesis Submitted to EEMCS Faculty Delft University of Technology,
In Partial Fulfilment of the Requirements
For the Bachelor of Computer Science and Engineering
June 25, 2023

Name of the student: Marios Hadjigeorgiou
Final project course: CSE3000 Research Project
Thesis committee: Jérémie Decouchant, Bart Cox and Qing Wang

Abstract

Federated Learning (FL) is widely favoured in the training of machine learning models due to its privacy-preserving and data diversity benefits. In this research paper, we investigate an extension of FL referred to as Personalized Federated Learning (PFL) for the purpose of training diffusion models. We explore the personalization technique of Transfer Learning (TL) and analyse evaluation metrics to capture personalization scores. Transfer Learning has been proven to produce good personalization results under IID and non-IID data distributions. We explore the impact of specific hyperparameters and data distribution techniques and examine how the personalization results can be improved even further. We demonstrate that the learning rate and the number of base layers of the convolutional neural network (CNN) form a normal distribution in terms of per-user improvement results. Increasing the number of users introduces variance in the convergence process, with the per-user personalization scores experiencing an overall improvement over the pre-trained model independent of the number of users. Our evaluations show that tuning to the optimal hyperparameter values for specific non-IID data distributions produces better personalization scores than other PFL methods.

1 Introduction

There is a significant amount of decentralized data generated on a growing number of users' devices. A machine learning approach that can use this data to train machine learning models with reduced communication costs and in a privacy-preserving manner is Federated Learning (FL).

The field of FL has garnered significant attention in the past few years due to its capability of using multiple clients in coordination to train machine learning models in a decentralized setting. This approach has brought substantial advantages to the domain of machine learning by enhancing privacy measures and reducing the need for the transfer of sensitive data [22]. Applications of this methodology can be found in technologies of paramount significance like Google's GBoard mobile keyboard and Apple's vocal classifier for ́ıts virtual assistant [13].

Personalized Federated Learning (PFL) is an extension of FL that aims to achieve personalization for each individual user while maintaining a decentralized training approach. The field of PFL has been heavily studied lately and there exists a vast quantity of research on different techniques and approaches to training models in a personalized manner [15, 17, 19].

Conversely, training diffusion models using PFL remains relatively unexplored and requires further investigation. Diffusion models are probabilistic generative models that can capture the characteristics of data by diffusing information. The benefits of such research would significantly aid the field of machine learning as they will give new insights into the architectural designs of PFL algorithms and contribute to the advancement of more accurate personalized diffusion models.

In this study, we concentrate on the implementation of diffusion models using PFL in the PyTorch framework. The contributions of this study can be summarized as follows.

We study the personalization method of Transfer Learning (TL) and focus on tuning specific hyperparameters. We analyse the evaluation methodology to capture personalization results and the concluding remarks after the application of the penalization strategies [8]. The goal of this research is to investigate the impact of different hyperparameters of TL on the personalization score of diffusion models.

The main contributions of this work can be summarized as follows:

- Tuning specific hyperparameters to observe the difference in personalization scores for TL
- Comparing the personalization scores on IID and non-IID datasets
- Comparing the evaluation results with those of other personalization techniques.
- Evaluating the results of the global pre-trained model and the fine-tuned model.

The rest of the paper is constructed as follows: Section 2 introduces the pre-requisite background information on diffusion models, FL and PFL. Section 3 summarizes the relevant work that exists in the field of PFL and Section 4 describes the research methodology employed, including the algorithm for Transfer Learning, the selection of hyperparameters used to investigate and the personalization metrics. Section 5 represents the findings of the study, with statistical and performance analysis of the effect of the different hyperparameters on the personalization results. Section 6 provides a comprehensive analysis of the results and Section 7 analyses the responsible and ethical practices throughout the research process. Finally, Section 8 summarizes the conclusions of the study and suggests ideas for future research.

2 Background

In this section, we introduce some of the fundamental terms and methodologies needed to comprehend the rest of the research paper. In the ensuing steps, we provide a foundational description of Diffusion models, Federated Learning and Personalized Federated Learning.

2.1 Diffusion Models

Diffusion models are probabilistic models that can generate new samples of data using a diffusion process. This process is done through a series of iterative updates, where a gradual diffusion process is applied to a Markov chain model (a sequence of states where the next state depends only on the previous state). The Markov chained transitions initially follow a process called noise injection where Gaussian noise is injected into every state of the data. In Figure 1 the reverse sampling process is displayed where the diffusion process is reversed on each state of the Markov chain model and the original data is recovered. This is based on the theorem that the true reverse process will have the same functional form as the forward process [10]. During training, Bayesian inference is applied to the model to capture the dynamics of data that simulate best the data distribution. These dynamics simulate the likely distribution of the true state of the data given the noisy observations. Through this process, the model learns to produce samples that resemble the original data after a finite time. It has been shown that diffusion models are efficient to train and capable of producing high-quality samples [12].

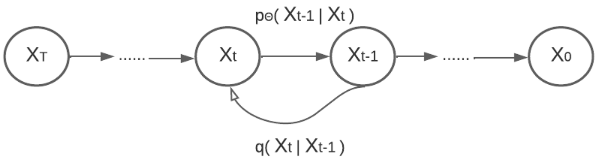

Figure 1: During the Denoising process, the previous and next states are used to calculate the posterior probability of the true state of the data, given a noisy sample. Calculating the posterior probability of the forward step is done through Bayesian inference.

2.2 Federated Learning

FL is a collaborative method where different devices work together to train a machine-learning model. This procedure is overseen by a central server which then receives the model updates from each device and proceeds to aggregate the data into a new global model. FL is a machine learning methodology that allows users to benefit from a global model trained on a vast amount of data without the requirement of storing them centrally [18]. The goal of FL is to train a global model that uniformly achieves good performance over the majority of clients while preserving the privacy of the client's data.

Due to the large number of devices a central server interacts with as well as the diversity of data that each device holds FL faces several challenges. The number of communication rounds as well as the size of transmitted messages at each round can be a critical factor in how expensive the communication is. In addition, each device may differ in terms of hardware and reliability leading to system heterogeneity challenges with some common examples being that of straggler mitigation and fault tolerance [16].

2.3 Personalized Federated Learning

A methodology that is used to mitigate some of these challenges is an extension of FL called Personalized Federated Learning (PFL). PFL is a layer on top of FL that aims to learn a personalized model for each client which is an optimal combination of a local model trained in isolation by a client and a global model trained in a collaborated fashion by all the clients. The challenges that PFL is designed to mitigate are the lack of personalization and the poor convergence on heterogeneous data [19]. Lack of personalization relates to the scenario where the global model does not generalize well for the distribution of a specific client's data. Optimizing for the global model accuracy may result in a lower performance level for a client's local model due to the very different distributions that the two models have. PFL aims to identify the diverging patterns among each client and customize a model that fits the user's needs. The second challenge which PFL strives to alleviate is that of statistical heterogeneity. Devices may generate data in a non independently and identically distributed manner (Non-IID) [3] that may lead to potential variations in the number of data points [16]. These variations may add complexity to the process of modelling and evaluation along with convergence issues. To address this challenge, PFL aims to enhance learning stability through the utilization of hyper-parameter tuning techniques. [19]. The overall objective of PFL is to mitigate these challenges by balancing the exchange between the clients' collaboration and the diversity in statistical patterns among user domains [5].

3 Related Work

Two common subcategories of PFL methodologies are the data-based and the model-based approach [19].

Data-Based

Some examples of the data-based approach include the data augmentation and client selection techniques which both come with their own challenges. Data augmentation relies on the concept of enhancing the statistical heterogeneity of the data of each client. That can be achieved using over-sampling and under-sampling techniques. Unfortunately, augmenting the private dataset of each client requires a formulation of a data-sharing structure or the availability of a global proxy dataset [19]. Data-based approaches require the modification of the data distribution. As a result, the diversity of information formed by each individual user is undermined, indicating subpar personalization results.

Model based

Model-based techniques focus on the implementation of individual models for each client by adapting a global model to the characteristics of each individual user. Common model-based techniques include Regularized Local Loss [19] , Meta-Learning [20] and Transfer Learning [19]. Meta-Learning focuses on developing a learning strategy that improves its performance with experience. It is designed to combine its existing knowledge with the limited amount of new input information while avoiding the risk of overfitting. With traditional base-learning techniques, learning quality improves with more examples from a single task and the continuous application of the same data over the learner always produces the same hypothesis. Conversely, Meta-Learning is not limited to refining a hypothesis based on examples related to only one task but it continuously adapts across diverse tasks. Therefore, in the case where the learner performs poorly on a task, the learning mechanisms will adapt when the same task is presented again.

Regularized Local Loss is a loss function that combines techniques for minimizing the error for the local sample while promoting the generalization of the global model. This technique is based on penalizing the loss function and adjusting the parameter importance, mitigating the issue of weight divergence between the local and global model while alleviating the model from overfitting to the local data.

Transfer Learning
One of the main benefits of TL over Meta-Learning and Regularized Local Loss is that it does not require training a model from scratch and as a result, personalized model training becomes less resource-intensive. There has been extensive research in the field of Transfer Learning that shows promising results [1]. Arivazhagan et al. compare the Transfer Learning algorithm consisting of base and personalization layers with the traditional Federated Learning algorithm FedAvg [18]. When tested on a CNN model with an unbalanced image classification dataset and 210 users, the results showed that the TL model obtains significantly better results in convergence speed and the client's average test accuracy scores compared to the FedAvg model. An important result in this study is that the personalized model's performance becomes more identical to that of FedAvg, when the data partition becomes more identical.

Taking into consideration the importance of evaluating the performance of Transfer Learning it is crucial that we apply personalized metrics to compare the final results. To the best of our knowledge, there is only one study that introduces personalized metrics thus far [8]. Instead of using the traditional method of comparing the individual user's generative model quality, this study introduces new types of metrics that resemble a general image for the overall user personalization performance. In the same study, TL is compared with other personalization techniques based on these metrics. The results show that the TL algorithm was the second-best algorithm out of 4 on a non-IID data distribution with every user having the same number of samples but a different number of classes and samples per class. On the other hand, the TL algorithm performed the worst on a non-IID data distribution where each user has samples from all classes but a different number of total samples and samples per class.

4 Methodology

In this section, we describe in more detail the approach towards the implementation of a structure to train the diffusion model. First, we describe the U-shaped network (UNet) model that characterizes the diffusion model and the features that it carries. Secondly, we will go into detail regarding the diffusion process of the model including the loss function and the propagation step. Furthermore, we will elucidate the PFL methodology that we choose to train the diffusion model with and finally explain the evaluation strategy that we follow to assess the quality of the results.

4.1 The UNet Model Architecture

The model that we chose to be trained is based on the UNet architecture that consists of an encoder-decoder structure for image segmentation tasks. The model consists of contracting and expanding paths. Contracting paths are responsible for capturing the contextual information of the input image and learning a high-level representation of the input. They do so by gradually decreasing the spatial resolution of the image and increasing the feature channels of the image.

Conversely, the expanding paths recover the spatial resolution lost in the encoding process. They consist of gradually increasing the spatial resolution of the features and reducing the number of channels. Using the contracting paths combined with the recovered spatial details it creates the image segmentation. In between the two paths, skip connections are employed to alter the flow of information in order to capture both local and global information.

During the encoding and decoding processes, up-sampling and down-sampling are used to alter the resolution of the feature map while residual blocks extract the features and detect distant relationships. The final convolution layer transforms the extracted features into the desired output dimension

4.2 The Training Process

The training process of TL begins with each client retrieving images and labels from their own dataset. Instead of comparing the predicted output with the respectable labels, in diffusion models noise images and predicting noise images are produced. The goal of this process is to acquire the loss value by measuring the difference between the two sets of images. The optimiser then updates the weights of the model to catch the underlying patterns by adjusting the model's parameters accordingly.

This process is repeated for a number of iteration steps before the new model is returned to the server to undergo the aggregation process.

4.3 PFL Approach

The traditional FL setting generates a common model for all users without considering personalization. However, when data heterogeneity is present, the global model will have suboptimal performance [19]. On the other hand, training the local model strictly on the local dataset and without any collaboration with other clients may lead to poor generalisation performance. With the goal of achieving a balance between generalization and personalization performance, PFL is positioned in between the conventional FL setting and the local approach.

Despite the enhancement of convergence in the global FL model, data-based approaches often need to adjust the local data distributions. This may have as a result the loss of valuable information related to the diversity of each client's behaviour. As a result, this may affect negatively the personalization of the global model for each client.

Transfer Learning
A more promising methodology for PFL is Transfer Learning (TL). Transfer Learning (also known as Fine-Tuning) falls under the category of Model-Based techniques [19] where the knowledge acquired from a pre-trained model is used to improve the performance of a different but similar task. In the deep learning and machine learning settings, TL is responsible for adjusting some parameters of the pre-trained global model. This allows the model to benefit from the knowledge extracted from the global model without the need for training the model from scratch. By using a pre-trained model, the computational cost of training the new personalized model is reduced. The algorithm that we use is inspired by Arivazhagan et al. [1] and is described in Algorithm 1 and Algorithm 2. In the same paper, the algorithm is constructed with

each client saving locally their personalization layers. The base layers are trained in a federated way and the aggregated model is then fine-tuned by each client using the personalization layers. PerFT works in an alternative way. The base layers are frozen and each client trains the personalized layers of the model locally. The aggregated step involves the averaging of the personalized weights of the clients while the base layers are left intact. For each client, the number of Local epochs (N) and the learning rate (α) are kept the same.

The architecture of TL requires a neural network that can be split into two components: base layers and personalization layers. The concept behind this architecture is for the lower layers of the convolutional base to learn the general characteristics, while the personalization layers specialize in learning specific features. This can be achieved by freezing the convolutional base and proceeding to train only the personalized layers. This indicates that the model utilizes the weights of the lower layers that contain the general features while simultaneously learning dataset-specific features.

Algorithm 1 PerFT - Server Side

```
**Input**:
pre-trained model weights $W_{\text{pre}}$,
number of clients $C$,
dataset $D_{\text{j}}$
for user j = 1 to C, Learning rate $\alpha$, Number of epochs $E$,
    Number of base(Frozen) layers $N$
**Output**: fine-tuned model weights $W_{\text{fine}}$

Initialize model weights $W_{\text{fine}} \leftarrow W_{\text{pre}}$
**for** User $j = 1$ to $C$ **do**
    Compute $\gamma_{\text{j}} \leftarrow \frac{|D_{\text{j}}|}{\sum_{k=1}^{C} |D_{\text{k}}|}$
**end for**

**Freeze base layers:** $W_{\text{fine}} \leftarrow W_{\text{pre}}.\text{keys()}[N_{\text{frozen}} :]$
**for** $e = 1$ to $E$ **do**
    Receive $W_{\textbf{pers,j}}$ from each Client $j$=1 to $C$
    Aggregate $W_{\text{fine}} \leftarrow \sum_{j=1}^{C} W_{\text{pers,j}} \cdot \gamma_{\text{j}}$
    Send $W_{\text{fine}}$ to each Client
**end for**
**Return** $W_{\text{fine}}$
```

Algorithm 2 PerFT - Client Side

```
**Input**:
aggregated model weights $W_{\text{fine}}$,
number of local epochs $K$,
fine-tuning dataset $D_{\text{fine}}$,
learning rate $\alpha$
**Output**: personalized model weights $W_{\text{fine}}^{(t)}$

Initialize model weights $W_{\text{pers}} \leftarrow W_{\text{fine}}$
**for** k = 1 to K **do**
    **for** each batch $B$ in $D_{\text{fine}}$ **do**
        Compute loss $L$ using batch $B$ and current model weights
            $W_{\text{pers}}^{(k)} = W_{\text{pers}}^{(k-1)} - \alpha \cdot \nabla_{\text{pers}} L(W_{\text{pers}}^{(k-1)})$
    **end for**
**end for**
**Return** $W_{\text{pers}}^{(k)}$
```

Hyperparameter Optimization

The hyperparameters are various network structural elements that govern the process by which the network is trained. These elements are common for the FL architecture, with the most important ones being the learning rate, batch size and number of epochs.

In the case of TL the hyperparameter optimization that we will test encompasses the following hyperparameters:

- Number of base layers
- Learning rate
- Number of users

The number of base layers is the fundamental hyperparameter for Transfer Learning. An increasing number of base layers indicates a smaller number of personalization layers applied to the model. We investigate the effect of changing the number of base layers on the personalization metrics.

A lower learning rate will initiate smaller-grained updates, helping the model prevent overfitting on the training data [11]. In the same paper, the author proposes a fine-tuning architecture that utilizes a 'slow start, fast decay' learning rate strategy where the learning rate is initially small, and then gradually increases throughout the epochs of the Federated Learning process. The idea behind this strategy is initially to prevent the model from catastrophically forgetting the previously learned distributions when exposed to new samples. Subsequently, the learning rate is increased to allow the model to reach a stable performance without exposing it to excessive overfitting on the training data. This algorithm has been shown to reduce the training time by 10 times and offer higher robustness.

In a PFL setting the goal is to personalize a model for every client. Therefore, we will investigate whether the number of users during the Fine-Tuning procedure plays a significant role in the model's performance.

4.4 Evaluation Metrics

Diffusion Score

The Frechet Inception Distance (FID) [21] is a commonly used evaluation metric in the field of generative models. The FID metric evaluates the quality of generated samples by mapping them to a feature space. It calculates the mean and covariance of the generated and real data. The distance metric it uses to assess the quality of the generated samples is the dissimilarity between the two Gaussian distributions.

Personalization Score

A personalization model with the highest accuracy results may not necessarily be the fairest [8]. In an experiment constructed in the same research paper, a different local, global and personalised model was created for each user. The results showed that the personalized model that achieved the highest average accuracy across all clients, only had 2 out of 9 users experience an increase in accuracy. This leads to the conclusion that a personalization method that yields the best accuracy results on average, may not necessarily yield the best results in terms of per-user personalized accuracy.

For the performance metrics, we will use the Percentage of User-models Improved (PUI) metric. It consists of the percentage of users with a personalized model that produces better results than the global model. The mean shows to be the best metric when the data forms a normal distribution [8]. However, it is impossible to know whether the performance of the users will form a normal distribution and therefore we will also apply the Median Percentage of Improvement (MPI) and Average Percentage of Improvement (API) metrics that show accordingly the median and average percentage improvement of the users who had an increased performance over their global model. The MPI and API metrics provide information about the central tendency and average improvement in the FID scores of the users who have an enhanced local model compared to the global model. By considering both the median and average improvement, we obtain a better understanding of the distribution of users who obtained a better model.

5 Experimental Setup and Results

We evaluate the personalized algorithm of Transfer Learning to observe its performance results after fine-tuning. We also show the penalization results of the algorithm after tuning the base layers, learning rate and number of users.

5.1 Experimental setting

We test the TL algorithm on the FMNIST dataset which consists of 60,000 training images and 10,000 testing images of various clothing items. It contains 10 different classes and every image is of size 28x28 pixels.

The UNet model we are training consists of 260 connected hidden layers. We have pre-trained our model under the FedAvg algorithm, using the Adam optimizer [14] with 5 local epochs, 100 global aggregations and 5 users. The pre-trained model generates a total of 1000 samples and gives an FID score of 31.33.

We test the algorithm's performance on non-IID data. In the non-IID setting, every user gets assigned randomly an unequal number of samples from the total distribution, with samples from an unequal number of classes.

Metrics	**PersFL**	**FedPer**	**pFedMe**	**perFed**
PUI(%)	100	100	100	100
MPI(%)	11.23	6.59	10.81	8.55
API(%)	10.83	6.41	10.47	8.85

Table 1: The personalization metrics applied to different personalized FL methods on CIFAR-10 dataset with a non-IID data distribution from the paper Divi et al [2].

The baseline that we are comparing against is shown in Table 1. It consists of 4 different methodologies of PFL that include PersFL [7], FedPer [1], pFedMe [6], and PerFed [9]. The non-IID data structure that was used involves users acquiring samples from every class with the total number of samples being different for each client.

5.2 Experimental Results

We evaluate the result of the TL model by tuning the different hyperparameters. The evaluation focuses on the personalization metrics PUI, MPI and API.

	User 1	**User 2**	**User 3**	**User 4**	**User 5**
IID data	141.13	131.84	126.70	152.20	119.54
non-IID data	189.81	274.24	202.21	184.39	190.41

Table 2: Per-User FID score using the traditional global model trained with FedAvg

When the global model is tested on the entire dataset, it shows an FID score of 31.33. As shown in Table 2, the per-user score results are more than 4X lower when the global model is applied to each user's limited dataset on IID data and 6X lower when applied to non-IID data. The non-IID data distribution shows clearly that data heterogeneity has a significant role in the performance score of the algorithm.

Number of Base-Layers	**API(%)**	**MPI(%)**	**PUI(%)**
0	0.83	4.19	20
50	2.38	5.95	40
100	2.65	**6.63**	40
150	**6.19**	5.36	**100**
200	1.92	3.00	**60**
250	1.38	3.47	40

Table 3: Personalization results for each user, using a 0.0001 learning rate with a different number of base layers on a non-IID data distribution.

As shown in Table 3, the PUI score and API score achieve the highest scores when the number of base layers is 150. The number of PUI improvements compared to API and MPI is significantly higher, indicating that a big proportion of users experience an increase of performance compared to their global model, but the performance they experience is not significant. The best PUI and API score is reached when the number of base layers is 150. This indicates that 5 out of 5 users experience an increase of 6.19% on average in their FID scores when the number of base layers is 150. Considering that the total number of connected layers in the utilised UNet model is 260, this shows that the number of base layers and the accuracy scores form a normal distribution. For the rest of the experiments, we are using 150 base layers, as it provides the best results.

Learning rate	**API(%)**	**MPI(%)**	**PUI(%)**
0.01	0.0	0.0	0
0.001	**13.66**	**20**	**100**
0.0001	6.19	5.36	**100**
0.00001	3.06	15.30	20
SSFD 1	2.24	11.21	20
SSFD 2	0.0	0.0	0
SSFD 3	0.0	0.0	0

Table 4: Personalization results for each user, using 150 base layers on non-IID dataset with a different number of learning rates

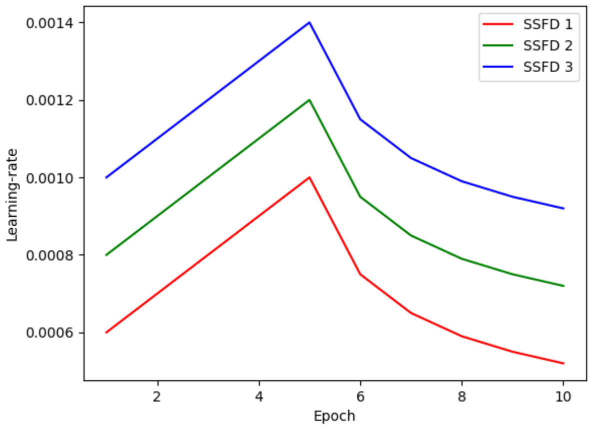

Figure 2: The slow start-fast decay algorithm gradually increases the learning rate and then it decreases exponentially. We try 3 different implementations of this algorithm with different start values.

As shown in Table 4, the learning rate forms a uniform distribution where the best personalization results are reached when the learning rate is 0.001. This shows that the learning rates below that value are suffering from underfitting whereas the learning rates above this value suffer from overfitting.

The ‘slow start-fast decay’ algorithms shown in Figure 2 perform the worst. This shows that this learning rate methodology is not effective for this algorithm. For the rest of the experiments, we use a learning rate of 0.001 as this produces the best personalization results.

Number of Users	**API(%)**	**MPI(%)**	**PUI(%)**
5	**13.66**	**20.00**	**100**
10	5.95	4.74	70
15	11.32	9.20	**100**
20	12.53	14.40	90

Table 5: Personalization results for each user using 150 base layers, 0.001 learning rate on non-IID data and a different number of participants

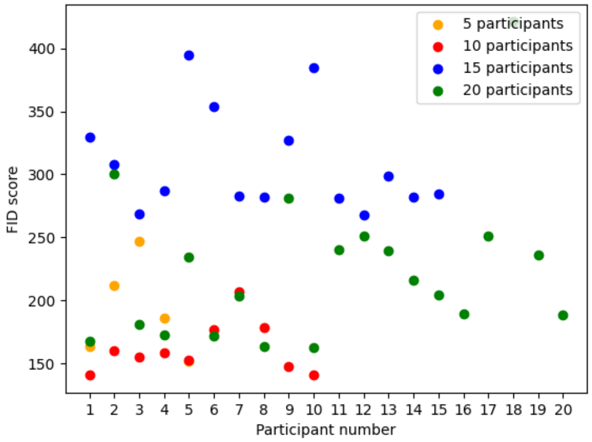

Figure 3: Global model performance scores for different numbers of participants on non-IID dataset

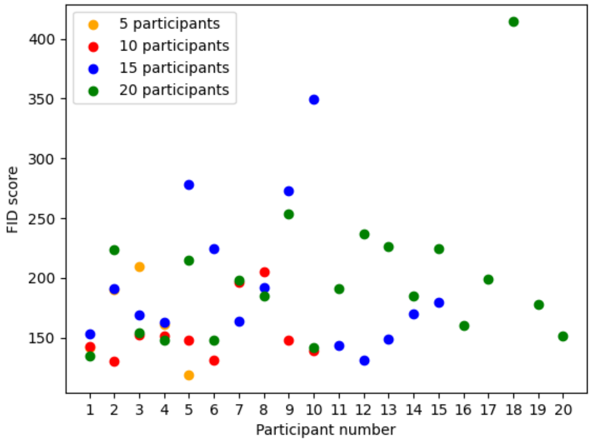

Figure 4: Local model performance score for different number of participants non-IID dataset

As shown in Table 5, the personalization results show fluctuating results when the number of users increases. Similarly, the FID scores of each user show inconsistent results. As demonstrated in Figure 3 and Figure 4, the FID scores of each user in the scenario with 20 participants are significantly better than the scenario with 15 participants. Even though, the personalization metrics experience a similar inconsistency, the Fine-Tuned model on every scenario outperforms the global model for the majority of clients.

	API(%)	MPI(%)	PUI(%)
TL in isolation	5.59	7.53	**100**
TL in FL setting	**13.66**	**20.00**	**100**

Table 6: Comparison of best personalization results for TL in isolation and TL in a FL setting. We use 150 base layers, 0.001 learning rate for 5 users on a non-IID data distribution

The results in Table 6 show that training the diffusion model under a FL setting converges more accurately than training in isolation. Both API and MPI scores are 2X improved.

	Global Model	Personalized Model
User 1	189.81	137.14
User 2	274.24	187.08
User 3	202.21	136.29
User 4	184.39	127.33
User 5	190.41	124.08

Table 7: FID score comparison of global pre-trained model and model after fine-tuning for 5 users, using a non-IID data distribution.

6 Discussion

We have investigated whether the TL model can improve its personalization results after tuning its hyperparameters. We have explored the personalizaton results after tuning the number of base layers, learning rate and number of participants.

The results show that the number of base layers and the learning rate form a normal distribution. A probable cause for this is the fact that a smaller number than the optimal value may result in the model underfitting the training data and failing to capture the user-specific characteristics. On the other hand, a bigger value than the optimal may result in the model overfitting where the model becomes too personalized and as a result fails to generalize well on unseen data.

The 'slow start-fast decay' algorithm has shown low performance results compared to other learning rate values. According to Jeddi et al. [11], this learning rate methodology reduces the training time by 10x and outperforms other adversarial training algorithms. A possible reason for the underperformance of the 'slow start-fast decay' algorithm in our setting is that the pre-trained model we chose has been trained with 100 global epochs while the pre-trained model in the paper has been trained with 200 global epochs. Also, the dataset used for this experiment is the FMNIST while in the research paper they use the CIFAR-10 dataset.

Compared to previous research [8], our results in Table 7 agree that TL improves the personalization results compared to the global model. We highlight that out algorithm PerFT achieves better results on the MPI and API scores in this non-IID data setting as shown in Table 1.

The number of participants affects both the performance and personalization scores of the TL model. As shown in Figure 3 and Figure 4, the FID scores of both the pre-trained global model and the TL model indicate an unstable performance with the addition of more participants in the training process. A possible reason for this behaviour is that the amount of training data assigned to each client is decreased when more users participate in the training process. In the research paper by de Goede [4], it is shown that the generative model convergence with better FID scores when the number of local epochs is increased along with the number of participants. Overall, the personalization scores were better in all user scenarios.

7 Responsible Research

The responsible and ethical principles for our research are expressed in the transparency of the methodologies that we use. The algorithms and methods used are described with clarity and the dataset used is publicly accessible so that the experiments can be easily reproduced.

Further research needs to be done to prevail the existing inequalities under PFL methodologies. Users' limited access to data and their representation in the training process can create biased or inaccurate predictions that can lead to the reinforcement of biases and amplify social disparities. Therefore it is of great importance to mitigate these challenges and use PFL to promote equity, inclusiveness and unbiased results.

8 Conclusions and Future Work

We have demonstrated that Federated Learning can train diffusion models and achieve high converge results on a global setting but result in low converge scores on a per-user level. This study has investigated the personalization methodology of Transfer Learning by implementing the FedTL algorithm and focused on tuning its hyperparameters to analyse their effect on the personalization scores. We have introduced personalization metrics to capture the performance of each individual user and we have applied transfer learning in an IID and non-IID setting.

The results show that the number of base layers and the learning rate form a normal distribution where any value above or below the optimal option results in overfitting and underfitting respectively and a less optimal personalization score. The number of participants showed unstable performance in terms of both converge and personalization scores. FerFT obtains 100% PUI, 13.66% API and 20% MPI score, outperforming other Personalized Federated Learning methods, specifically in non-IID data settings.

Lastly, we identify some remarks for future work. Our implementation is limited to that of the FMNIST dataset. The personalization metrics of the Transfer Learning algorithm could be further explored under more variations of datasets. In addition, personalization methods of Meta-Learning [20] and Regularized Local Loss [19] could also be explored for the effect of hyperparameters on the personalized evaluation metrics.

References

[1] Manoj Ghuhan Arivazhagan and Vinay Aggarwal. Federated learning with personalization layers. *Adobe Research*, page 5, 2020.

[2] Ali Borji. Pros and cons of gan evaluation measures. 2018.

[3] Bart Cox, Lydia Y Chen, and Jérémie Decouchant. Aergia: leveraging heterogeneity in federated learning systems. In *Proceedings of the 23rd ACM/IFIP International Middleware Conference*, pages 107–120, 2022.

[4] Matthijs de Goede. Training diffusion models with federated learning: A communication-efficient model for cross-silo federated image generation. page 6, 2023.

[5] Yuyang Deng, Mohammad Mahdi Kamani, and Mehrdad Mahdavi. Adaptive personalized federated learning, 2020.

[6] Canh T. Dinh, Nguyen H. Tran, and Tuan Dung Nguyen. Personalized federated learning with moreau envelopes, 2022.

[7] Siddharth Divi, Habiba Farrukh, and Berkay Celik. Unifying distillation with personalization in federated learning, 2021.

[8] Siddharth Divi, Yi-Shan Lin, Habiba Farrukh, and Z. Berkay Celik. New metrics to evaluate the performance and fairness of personalized federated learning. pages 3–6, 2023.

[9] Alireza Fallah, Aryan Mokhtari, and Asuman Ozdaglar. Personalized federated learning: A meta-learning approach, 2020.

[10] W. Feller. On the theory of stochastic processes, with particular reference to applications. In Neyman and Jerzy, editors, *Berkeley Symposium on Mathematical Statistics and Probability*, pages 403–432, 1949.

[11] Ahmadreza Jeddi, Mohammad Javad Shafiee, and Alexander Wong. A simple fine-tuning is all you need: Towards robust deep learning via adversarial fine-tuning, 2020.

[12] Ho Jonathan, Jain Ajay, and Abbeel Pieter. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, page 5, 2020.

[13] Peter Kairouz, H. Brendan McMahan, et al. Advances and open problems in federated learning. *Foundations and Trends® in Machine Learning*, page 44, 2021.

[14] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017.

[15] Viraj Kulkarni, Milind Kulkarni, and Aniruddha Pant. Survey of personalization techniques for federated learning. 2020.

[16] Tian Li, Anit Kumar Sahu, Ameet Talwalkar, and Virginia Smith. Federated learning: Challenges, methods, and future directions. *Carnegie Mellon University & Determined AI*, pages 3–4, 2021.

[17] Yishay Mansour, Mehryar Mohri, Jae Ro, and Ananda Theertha Suresh. Three approaches for personalization with applications to federated learning. 2020.

[18] H. Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera y Arcas. Communication-efficient learning of deep networks from decentralized data, 2023.

[19] Alysa Ziying Tan, Han Yu, Lizhen Cui, and Qiang Yang. Towards personalized federated learning. *IEEE Transactions on Neural Networks and Learning Systems*, pages 2–6, 2022.

[20] Ricardo Vilalta and Youssef Drissi. A perspective view and survey of meta-learning. *Artificial Intelligence Review*, 18:77–95, 2002.

[21] Yu Yu, Weibin Zhang, and Yun Deng. Frechet inception distance (fid) for evaluating gans. 2021.

[22] Chen Zhang, Yu Xie, Hang Bai, Bin Yu, Weihong Li, and Yuan Gao. A survey on federated learning. *Knowledge-Based Systems*, 216, 2021.