Semi-Supervised Federated Peer Learning for Skin Lesion Classification

Tariq M. Bdair t.bdair@tum.de
Computer Aided Medical Procedures, Technical University of Munich, Germany

Nassir Navab nassir.navab@tum.de
Computer Aided Medical Procedures, Technical University of Munich, Germany
The Whiting School of Engineering, Johns Hopkins University, United States

Shadi Albarqouni shadi.albarqouni@ukbonn.de
Clinic for Diagnostic and Interventional Radiology, University Hospital Bonn, Germany
Helmholtz AI, Helmholtz Zentrum München, Germany
Faculty of Informatics, Technical University of Munich, Germany

Abstract

Globally, Skin carcinoma is among the most lethal diseases. Millions of people are diagnosed with this cancer every year. Sill, early detection can decrease the medication cost and mortality rate substantially. The recent improvement in automated cancer classification using deep learning methods has reached a human-level performance requiring a large amount of annotated data assembled in one location, yet, finding such conditions usually is not feasible. Recently, federated learning (FL) has been proposed to train decentralized models in a privacy-preserved fashion depending on labeled data at the client-side, which is usually not available and costly. To address this, we propose FedPerl, a semi-supervised federated learning method. Our method is inspired by peer learning from educational psychology and ensemble averaging from committee machines. FedPerl builds communities based on clients' similarities. Then it encourages communities' members to learn from each other to generate more accurate pseudo labels for the unlabeled data. We also proposed the peer anonymization (PA) technique to anonymize clients. As a core component of our method, PA is orthogonal to other methods without additional complexity, and reduces the communication cost while enhances performance. Finally, we propose a dynamic peer learning policy that controls the learning stream to avoid any degradation in the performance, especially for the individual clients. Our experimental setup consists of 71,000 skin lesion images collected from 5 publicly available datasets. We test our method in four different scenarios in SSFL. With few annotated data, FedPerl is on par with a state-of-the-art method in skin lesion classification in the standard setup while outperforming SSFLs and the baselines by 1.8% and 15.8%, respectively. Also, it generalizes better to an unseen client while being less sensitive to noisy ones.

Keywords: Semi-supervised Federated Learning, Peer Learning, Peer Anonymization, Dynamic Policy, Skin Lesion Classification

1. Introduction

In the recent estimation, an expectation of two hundred thousand fatal invasive and in-situ melanoma cases will be diagnosed in the USA in 2021 (Siegel, 2021). Yearly, millions of people are diagnosed with skin carcinoma (Rogers et al., 2010). Worldwide, skin cancer

https://www.melba-journal.org/papers/2022:011.html.

is considered one of the most expensive and fatal cancers. While most non-melanoma skin cancer cases can be cured, the melanoma ones are curable when detected in the early stages. For example, the 5-year survival rate ranges from 99% in the earliest stage to 27% for the latest stage (Siegel, 2021). Moreover, early detection of skin cancer can reduce the treatment expenses significantly (Esteva et al., 2017). Therefore, several attempts have investigated the automated classification of skin lesions in dermoscopic images (Clark Jr et al., 1989; Binder et al., 1998; Schindewolf et al., 1993). Though, these attempts require handcrafted engineered features and exhausted pre-processing steps.

Yet, huge improvements in computerized methods have been achieved in recent years. For instance, the deep-learning-based methods proved to have a superior (Gessert et al., 2020; Li et al., 2020b; Zhang et al., 2019; Lopez et al., 2017) or a human-level performance (Esteva et al., 2017; Tschandl et al., 2019) when dealing with skin cancer classification. Nevertheless, this success comes at the cost of exhausting pre-processing steps, a prudently designed framework, or a substantial amount of labeled data assembled in one location. In real life, medical data is generated from different scanners and unevenly distributed in multiple centers in raw formats without annotations resulting in heterogeneous data, or so-called Non-IIDness. Unfortunately, building a large repository of annotated medical data is quite challenging due to privacy burdens (Rieke et al., 2020; Kaissis et al., 2020), and labeling cost which is time-consuming and requires domain expert knowledge.

Federated learning (FL) (McMahan et al., 2017) has been recently proposed to learn machine learning models utilizing the ample amounts of labeled data distributed in mobile devices while maintaining clients' privacy *i.e.* without sharing the data. The training process of federated learning starts at the server by broadcasting initial weights of global model parameters to a random set of participating clients, who share the same model architecture with the global model. Each client, afterward, trains locally on its local data before sending back the updated model parameters to the server. Once all clients send their updates, the server aggregates them using `FedAvg` to update the global model weights. Next, the updated global model is broadcasted to a new random set of clients before a new round of local training processes starts. Eventually, the previous steps are repeated until the global model converged. During the training, only model weights are shared while data is kept locally. Note that, the key properties of the FL are data privacy, Non-IIDness, and communication efficiency. Thus, FL goes in line with the nature of the medical setting. Consequently, federated learning has been investigated by several works in the medical domain (Zhu et al., 2019; Li et al., 2020a; Albarqouni et al., 2020) paving the way to training machine learning models in privacy-preserved fashion in real-world applications (Flores et al., 2021; Roth et al., 2020; Sarma et al., 2021). Though, in the previous works, the training demand highly accurate labeled data, e.g., ground-truth confirmed through histopathology, which often is costly and not available.

In a more realistic scenario, the clients may have access to a large amount of unlabeled data along with few annotated ones. Yet, willing to train a reliable model to make use of their data. Fortunately, the above scenario can be addressed by the semi-supervised learning (SSL) paradigm, which is the focus of this paper. In this regard, a very recent work (Yang et al., 2021) has shown the applicability of semi-supervised learning in a federated setting (a.k.a. `SSFL`) for COVID-19 pathology segmentation. The previous work among the firsts who introduced semi-supervised learning to federated learning. Yet, they have straightforwardly applied a semi-supervised learning method, *e.g.* `FixMatch` (Sohn et al., 2020) locally. At first, a local model is trained in a fully supervised fashion using the labeled data. Then, the trained model is used to produce predictions for unlabeled data, where the predictions with high confidence are used to generate pseudo labels. Next, the pseudo labels are attached to the labeled data before a new training process starts. At the server, on the other hand, `FedAvg` was employed to organize the training between different clients, see Sec. 2.2. Another recent work, (Liu et al., 2021) proposed an `SSFL` approach; `FedIRM` for skin lesion classification. They suggested distilling the knowledge from labeled clients to unlabeled ones through building a disease relation matrix, extracted from the labeled clients, and providing it to the unlabeled ones to guide the pseudo-labeling process. In a more challenging situation, which has not been yet investigated thoroughly in the medical images, the labeled data is located at the server side while the clients have access only to unlabeled data. This scenario has been addressed in this paper, see Sec. 3.7.

In `SSFL`, clients are only trained i) globally, where the knowledge is accumulated in global model parameters, and ii) locally, where the knowledge is distilled via the local data. While this is a simple and straightforward approach, we argue that the knowledge gain for generating pseudo labels for the local models is limited. Instead, we hypothesize that gaining extra knowledge by learning from similar clients *i.e.* Peer Learning (PL) is highly significant assuming that peer learning encourages the self-confidence of the clients by sharing their knowledge in a way that does not expose their identities.

Our method is highly inspired by the social science literature, where peer learning is defined as acquiring skills and knowledge through active helping among the companions. It involves people from similar social groups helping each other to learn (Topping, 2005). Peer Learning includes peer tutoring, coaching, mentoring, and others. Though, the distinguishing between these types of PL is out of the scope of this paper. In this work, the link between peer learning and federated learning is direct, where the clients are considered as peers learning from each other.

In the computer science literature, a similar concept to peer learning has been introduced known as Committee Machines (CM) (Tresp, 2001; Aubin et al., 2019; Joksas et al., 2020). In a nutshell, CM is a well-known and active research direction and is defined as an ensemble of estimators, consisting of neural networks or committee members, that cooperate to obtain better accuracy than the individual networks. The committee prediction is generated by ensemble averaging (EA) of the individual members' predictions. CM has shown to be effective in machine learning hardware (Joksas et al., 2020), yet, it has not been investigated in FL. In this work, we employ the ensemble averaging from the committee machine for our *peers anonymization* (PA) technique. PA improves privacy by hiding clients' identities. Moreover, PA reduces the communication cost while preserving performance. To the best of our knowledge, no prior work has proposed the PA technique in the `SSFL` for the medical images.

Peer learning has shown an increase in the performance for the majority of clients, see Sec. 3.4. However, for some clients, peer learning and federated models could harm their performance. Thus, we propose a dynamic peer-learning policy that controls the learning process. Our dynamic learning policy maintains the performance of all clients while boosting the accuracy of the individual ones, see Sec. 3.10.

In this paper, we propose FedPerl, where the key properties are peer learning, peer anonymization, and learning policies. Our approach, in contrast to a very recent work FedMatch (Jeong et al., 2021), is communication-efficient and hides clients' identities where it employs the ensemble averaging method before sharing clients' knowledge with other peers. Our first contributions that include peer learning and peer anonymization in the standard semi-supervised learning setting were presented in (Bdair et al., 2021). In this extended and comprehensive version, we add the following **contributions**:

- We propose a dynamic learning policy that controls the contribution of peer learning in the training process. While our dynamic policy excels the static one, it at the same time helps the individual clients to achieve better performance.
- We show that our peer anonymization is orthogonal and can be easily integrated into other methods without additional complexity.
- We introduce and test our method in a challenging scenario, not been yet investigated thoroughly in the medical images, where the labeled data is located at the server-side. Moreover, we test the ability of FedPerl to generalize to unseen clients. Additionally, we conduct extensive analyses on the effect of committee size on the performance at the client and community levels.
- We introduce additional evaluation metrics to evaluate the calibration of these models and their clinical applicability.
- We validate our method on skin lesion classification, with database consists of more than 71,000 images, showing superior performance over the baselines.

2. Methodology

2.1 Problem Formulation

Given M clients $\mathcal{C}_m$ who have access to their own local dataset $\mathcal{D}_m \in \mathbb{R}^{H\times W\times N_m}$, where H and W are the height and the width of the input images, and N_m is the total number of images. $\mathcal{D}_m$ consists of labeled $\mathcal{S}_L = \{\mathcal{X}_L, \mathcal{Y}_L\}$ and unlabeled data $\mathcal{S}_U = \{\mathcal{X}_U\}$, where $\{\mathcal{X}_L, \mathcal{X}_U\} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_L, \boldsymbol{x}_{L+1}, \ldots, \boldsymbol{x}_{L+U}\}$ are input images; $\boldsymbol{x} \in \mathbb{R}^{H\times W}$, and $\mathcal{Y}_L = \{\boldsymbol{y}_1, \ldots, \boldsymbol{y}_L\}$; $\boldsymbol{y} \in \mathbb{R}^C$ are the corresponding categorical labels for C classes. Given query image $\boldsymbol{x}_q$, our objective is to train a global model $f(\cdot)$ to predict the corresponding label $\tilde{\boldsymbol{y}}_q$ for $\boldsymbol{x}_q$, where labeled and unlabeled data are leveraged in the training in a privacy-preserved fashion.

Definition. We define a model $f(\cdot)$ to be trained in a privacy-preserved fashion, if the following conditions are met

(i) Data can not be transferred across different clients participating in the training process adhering to the General Data Protection Regulation (GDPR) [1].

[1] https://gdpr.eu/

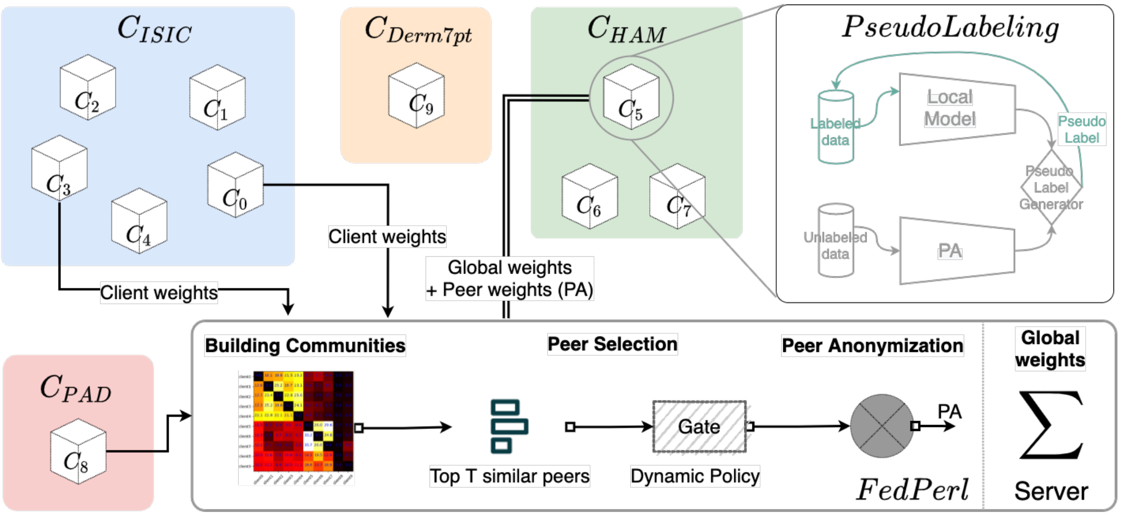

Figure 1: Our semi-supervised federated learning framework (FedPerl). Our method consists of (i) Building communities: similar clients clustered into one community, (ii) Peer Learning: peers are helping in pseudo labeling, and (iii) Peer Anonymization (PA) to hide client identity, improve privacy, and reduce the communication cost. Top Right: Pseudo labeling utilizing an anonymized peer in this diagram. Bottom: Selecting the similar peers, peer learning, peers anonymization, and similarity matrix calculations are performed on the server. FedPerl exploits either static or dynamic learning policies.

(ii) Local models can not be transferred across different clients participating in the training process to avoid privacy breaches (Orekondy et al., 2018), or model inversion (Fredrikson et al., 2015).

2.2 Semi-Supervised Federated Learning (SSFL)

The aforementioned conditions can be met by picking off-the-shelf SoTA SSL models, *e.g.*, `FixMatch` (Sohn et al., 2020), to train the clients locally leveraging the unlabeled data, while employing `FedAvg` (McMahan et al., 2017) to coordinate between the clients in a federated fashion as (Yang et al., 2021),

$$\min_{\phi} \mathcal{L}(\mathcal{D};\phi) \quad \text{with} \quad \mathcal{L}(\mathcal{D};\phi) = \sum_{m=1}^{M} w_m \, \mathcal{L}_{SSL_m}(\mathcal{D}_m;\phi), \tag{1}$$

where $w_m = {}^{N_m}/{\sum_{i=1}^{M} N_i}$ is the respective weight coefficient for each client, and ϕ is the model parameters. The SSL objective function appeared in `FixMatch` (Sohn et al., 2020), can be used to train the client locally utilizing both labeled and unlabeled data as

$$\begin{aligned} \mathcal{L}_{SSL_m}(\mathcal{D}_m;\phi) = \arg\min_{\phi} & \, \mathcal{L}_{CE}(\mathcal{Y}_L, f(\alpha(\mathcal{X}_L);\phi)) \\ & + \beta \mathcal{L}_{CE}(\tilde{\mathcal{Y}}_U, f(\mathcal{A}(\mathcal{X}_U);\phi)), \end{aligned} \tag{2}$$

where $\mathcal{L}_{CE}(\cdot,\cdot)$ is the cross-entropy loss, β is a hyper-parameter that controls the contribution of the unlabeled loss to the total loss, $\tilde{\mathcal{Y}}_U$ is the pseudo labels for the unlabeled data $\mathcal{X}_U$, and $\alpha(\cdot)$ and $\mathcal{A}(\cdot)$ are weak and strong augmentations respectively. For an unlabeled input $\boldsymbol{x}_i$, the pseudo label $\tilde{\boldsymbol{y}}_i \in \tilde{\mathcal{Y}}_U$, is produced by applying a confidence threshold τ on the client' prediction on a weak augmented version of $\boldsymbol{x}_i$ such that

$$\tilde{\boldsymbol{y}}_i = \arg\max(\mathbb{I}(f_{\mathcal{C}_m}(\alpha(\boldsymbol{x}_i);\phi^*) \geq \tau)), \tag{3}$$

where $f_{\mathcal{C}_m}(.)$ is the local model, ϕ^* are frozen model parameters, and $\mathbb{I}(\cdot)$ is the indicator function.

2.3 FedPerl: Peer Learning in SSFL

While the straightforward SSFL is simple, we argue that the learned knowledge for the individual clients could be further improved by involving similar clients in the training. Inspired by peer learning, our method utilizes similar peers to help the target client in the pseudo labeling by sharing their knowledge without exposing their identities through employing the peer anonymization method. Our proposed FedPerl, illustrated in Fig. 1, consists of three components; namely 1) building communities, 2) peer learning, and 3) peer anonymization. While peer learning can be static or dynamic as shown in the following sections.

2.3.1 Building communities

In educational social science (Topping, 2005), "peers" are referred to as two or more persons who share similarities and consider themselves as companions. In this work, we adopt the same concept and describe a group of clients as "peers" if they are similar. Previous work has shown that clustering can be achieved using models updates (Briggs et al., 2020), While other works measure similarities between deep neural networks by comparing the representations between layers (Kornblith et al., 2019). We build upon this and argue that the model weights represent and summarize the learned knowledge for each client from its training data. Thus, to measure the similarities between the clients, we represent each client $\mathcal{C}_m$ by a feature vector $\mathcal{F}_m = \{(\mu_0,\sigma_0),\ldots,(\mu_l,\sigma_l)\} \in \mathbb{R}^{2\cdot l}$, where (μ_l,σ_l) is the first two statistical moments, *i.e.* the mean and the standard deviation, of the model's layer l parameters. Then, we compute the similarity ω_{mk} between clients $\mathcal{C}_m$ and $\mathcal{C}_k$ using the cosine similarity, where $\omega_{mk} = \frac{\mathcal{F}_m^T\mathcal{F}_k}{\|\mathcal{F}_m\|\cdot\|\mathcal{F}_k\|}$. Using the cosine similarity brings the model parameters to the same behaviors without being exact, given that the means might differ, as long as they are in the same direction. Finally, the similarity matrix between all clients is defined as

$$\mathcal{W}_{M\times M} = \begin{bmatrix} \omega_{11} & \ldots & \omega_{1M} \\ \vdots & \ddots & \vdots \\ \omega_{M1} & \ldots & \omega_{MM} \end{bmatrix}. \tag{4}$$

Our method starts with standard federated learning warm-up rounds (*e.g.* ten rounds in our case). In the next training rounds, the feature vectors are extracted after receiving the updates from the participating clients. Then, the similarity matrix is computed and updated accordingly. In `FedPerl`, The communities are formed implicitly based on the similarity matrix where similar clients are clustered into one community (see Sec. 3.3).

2.3.2 Peers Learning

The term "learning" is frequently defined as improved knowledge, experiences, and capabilities (Topping, 2005). In peer learning, "peers" help each other by sharing their knowledge (Topping, 2005). In this regard, we describe "peer learning" as the means of top T alike clients (peers) help each other to generate pseudo labels by sharing their knowledge (model parameters). This is a helpful process since a main property of the medical data is the data heterogeneity. In federated learning, the clients experience different data and class distribution during the training. Thus, accumulating and sharing the distributed knowledge is useful. Particularly, it can help the local client generate pseudo labels for the unlabeled data from experiences that might never have learned from its own labeled data. To realize this, we modify the pseudo label defined in Eq.3 to include the predictions of the similar T peers, *i.e.* $f_t(\cdot;\phi)$ according to the similarity matrix $\mathcal{W}$ as

$$\tilde{\boldsymbol{y}}_i = \arg\max\left(\mathbb{I}\left(f_{\mathcal{C}_m}(\alpha(\boldsymbol{x}_i);\phi^*) + \sum_{t=0}^{T} f_t(\alpha(\boldsymbol{x}_i);\phi_t^*) \geq \tau\right)\right). \tag{5}$$

2.3.3 Peers Anonymization

To improve privacy and adhering to the privacy regulations introduced in 2.1, the knowledge sharing among peers has to be anonymized and regulated. Thus, we propose *peers anonymization* (PA), at the server side, a simple, yet effective technique. Particularly, we create an anonymized peer $f_a(\cdot;\phi_a)$ that assembles the learned knowledge from the top T similar peers where

$$f_a(\cdot;\phi_a) = \frac{1}{|T|}\sum_{t=0}^{T} f_t(\cdot;\phi_t). \tag{6}$$

Then, $f_a(\cdot)$ is shared with the local model to help in pseudo labeling. Accordingly, Eq.5 is modified to

$$\tilde{\boldsymbol{y}}_i = \arg\max\left(\mathbb{I}\left(f_{\mathcal{C}_m}(\alpha(\boldsymbol{x}_i);\phi^*) + f_a(\alpha(\boldsymbol{x}_i);\phi_a^*) \geq \tau\right)\right). \tag{7}$$

Notice that sharing the peers and the anonymized peer are not equivalent (Sec.3.2), *i.e.* $\frac{1}{|T|}\sum_{t=0}^{T} f_t(\alpha(\boldsymbol{x}_i);\phi_t)) \neq f_a(\alpha(\boldsymbol{x}_i);\phi_a)$. Eventually, the anonymized peer is shared only one time for each client at every training round, not at every local update. The advantages of the anonymized peer are i) it reduces the communication cost as sharing the knowledge of one peer is better than sharing 2 or more peers, ii) hides clients' identities by creating an anonymized peer. Finally, to prevent the local model from deviated from its local knowledge, we employ an MSE loss as a consistency-regularization term, which broadly used in semi-supervised learning,

$$\mathcal{L}_{CON_m} = \| f_{\mathcal{C}_m}(\boldsymbol{x}_i;\phi) - f_a(\boldsymbol{x}_i;\phi_a^*) \|^2 . \tag{8}$$

2.3.4 Overall objective:

The overall objective function for client m is the sum of semi-supervised and consistency-regularization losses, and given by

$$\mathcal{L}_m = \mathcal{L}_{SSL_m} + \gamma \mathcal{L}_{CON_m}, \tag{9}$$

where γ is a hyperparameter, and $\mathcal{L}_{SSL_m}$ and $\mathcal{L}_{CON_m}$ are Eq.2 and Eq.8, respectively. Note that the two terms in Eq.9 collaborate to achieve the balance between the local and global knowledge.

2.3.5 Dynamic Learning Policy

Thus far, we have proposed a static learning policy in which the top T similar peers are used to help the local clients in the pseudo labeling process. In peer learning, the clients are divided into groups or communities based on their similarities. A natural result of this step is also individual clients who do not belong to any community. Practically, we may have no control over the effect of applying the static peer learning policy on these clients, which could vary from one client to another where it is beneficial for some clients and not for others. For example, individual clients who do not belong to any community would be forced to learn from top T peers, based on the proposed similarity matrix, however, there is no guarantee that they would be beneficial in the training since they may not belong to the same or similar community. Therefore, we suggest performing a dynamic policy where we could carefully involve the peers based on additional similarities or restricting the peers to a subset who are close enough. Our dynamic learning policy controls the learning stream to the clients where the peers are utilized in the learning process. In short, our goal is to maintain the gain and boost the performance for all clients. In this regard, we propose the following policies.

Validation Policy In this policy, first, the client and its peers are validated on the global validation dataset. Then, only the peers with a validation accuracy equal to or higher than the client's accuracy are utilized. This policy can be applied with or without the peers' anonymization technique. Formally, assume that $V_{acc}(.)$ is a function that measures the accuracy on a global validation dataset, then the set of the peers that participate in peer learning for client $\mathcal{C}_m$ is defined as

$$\Omega = \{\mathcal{C}_n | V_{acc}(\mathcal{C}_n) \geq V_{acc}(\mathcal{C}_m)\}, \tag{10}$$

where $\mathcal{C}_n$ is a peer, $n = 1, 2, ..., T$, and T is the committee size.

Gated Validation Policy As in the previous policy, the peers are validated on the global validation dataset. However, we apply a gateway on their accuracies, such that if it is equal to or higher than a pre-defined gateway threshold ρ, the peer will be involved in the pseudo labeling. Otherwise, it will be discarded from the process. In this policy, the set of the peers that participate in peer learning for client $\mathcal{C}_m$ is defined as

$$\Omega = \{\mathcal{C}_n | V_{acc}(\mathcal{C}_n) \geq \rho\}. \tag{11}$$

Algorithm 1 Semi-Supervised Federated Peer Learning for Skin Lesion Classification

```
StartServer()
initialize global weights $\Phi_G^0$
for each round r=1, 2, ..., $R$ do:
    $n \longleftarrow$ select random $n$ clients from $M$ // i.e. n=3
    for each client $\mathcal{C}_m$ in $1, 2, ..., n$ do in parallel:
        $f_{\mathcal{C}_m} \longleftarrow$ initialize client's weights with global weights
        if $r > 10$ : //Peer learning starts after warm-up rounds
            $f_{\mathcal{C}_{1:T}} \longleftarrow$ GetTopSimilarPeers($T, f_{\mathcal{C}_m}$) // Sec. 2.3.1
            if IsDynamicPolicy // Sec. 2.3.5
                $f_{\mathcal{C}_{1:T}} \longleftarrow$ ApplyPolicy($f_{\mathcal{C}_m}$, $f_{\mathcal{C}_{1:T}}$, $\rho$) //Could return from 0 up to $T$ peers, here we assume all peers passed
            if IsPeerAnonymization
                $f_a \longleftarrow$ AnonymizePeers($f_{\mathcal{C}_{1:T}}$) // Eq.6
                $\Phi_m \longleftarrow$ LocalTraining($f_{\mathcal{C}_m}, f_a$)
            else // No Peer Anonymization
                $\Phi_m \longleftarrow$ LocalTraining($f_{\mathcal{C}_m}, f_{\mathcal{C}_{1:T}}$)
        else // No Peer learning, standard federated learning
            $\Phi_m \longleftarrow$ LocalTraining($f_{\mathcal{C}_m}$)
    end for
    $\Phi_G \longleftarrow \frac{1}{n}\sum_{j=1}^{n} \Phi_j$ // Update global weights i.e. FedAvg
    $\mathcal{F}_m \longleftarrow$ extract features vector for each client
    $\mathcal{W}_{M \times M} \longleftarrow$ update the similarity matrix // Eq.4
end for
```

Gated Similarity Policy Like in the gated validation policy, this policy depends on a gateway that controls peers participation. Yet, no validation set is used, and a peer is allowed to participate if its similarity with the client is equal to or higher than the gateway threshold ρ. Assume that $H_{sim}(.)$ is a function that measures the similarity between two clients, then the set of the peers that participate in peer learning for client $\mathcal{C}_m$ is defined as

$$\Omega = \{\mathcal{C}_n | H_{sim}(\mathcal{C}_m, \mathcal{C}_n) \geq \rho)\}. \tag{12}$$

Note that regardless of the used policy, we first select the top T similar peers based on the similarity matrix. Then, one of the above policies is applied. The only difference between the last two policies is that in the gated validation policy, we used the validation accuracy as a gateway, while in the gated similarity policy, we stick to our similarity matrix. A pseudo-code summarizing our method is shown in Algorithm 1

3. Experiments and Results

We test our method on skin dermoscopic images through a set of experiments. Before that, we show proof of concept results of our method and compare it with current SOTA in `SSFL` for CIFAR10 and FMNIST in image classification tasks in section 3.1. `FedPerl` outperforms the baselines at different settings. Next, in section 3.2, we compare skin image classification results of our method with the baselines. The results show that peer learning enhances the performance of the models, yet applying PA enhances the communication cost in addition to the performance. After that, we show and discuss how FedPerl builds the communities in section 3.3. The results show that FedPerl clusters the clients into main communities and individual clients thanks to our similarity matrix. Besides, FedPerl boosts the overall performance of communities while it has a different effect on the individual clients. Thus, in section 3.4, we comment on the impact of the peer learning on the individual clients. FedPerl shows superiority and less sensitivity to a noisy client. Then, we dig more deeply and present the classification results for each class in section 3.5. Our method enhances the classification for the individual classes, *e.g.* up to 10 times for the DF class. Further, to confirm our findings and for more validation, we present the results using different evaluation metrics in section 3.6. Our method is more calibrated and shows superiority over the SSFL in the area under ROC and Precision-Recall curves, risk-coverage curve, and reliably diagrams. The qualitative results are presented in the same section. In section 3.7, we propose a more challenging scenario in which the clients do not have any labeled data. The classification results show that FedPerl still achieves the best performance with or without PA. We end this part of experiments by showing the ability of FedPerl to generalize to an unseen clients in section 3.8. In section 3.9, we conduct a comparison with FedIRM, a SOTA SSFL method in skin lesion classification, under a fourth scenario where we have few labeled clients. Both models achieve comparable results when participation rate (PR)= 30%, while our method shows a lower performance when PR= 100%. Note that the previous results were obtained when utilizing a static learning policy. Yet, in the last part of our experiments, we show the results of our dynamic peer learning policy in section 3.10. In general, the new policy outperforms the results from the earlier one, while at the same time, it is successfully boosting the performance of the individual clients.

Datasets Our database consists of 71,000 images collected from 5 publicly available datasets as the following. (1) ISIC19 (Codella et al., 2019) which consists of 25K images with 8 classes. The classes are melanoma (MEL), melanocytic nevus (NV), basal cell carcinoma (BCC), actinic keratosis (AK), benign keratosis (BKL), dermatofibroma (DF), the vascular lesion (VASC), and squamous cell carcinoma (SCC). (2) HAM10000 dataset (Tschandl et al., 2018) which consists of 10K images and includes 7 classes. (3) Derm7pt (Kawahara et al., 2019) which consists of 1K images with 6 classes. (4) PAD-UFES (Pacheco et al., 2020) which consists of 2K images and includes 6 classes. The previous datasets are divided randomly into ten clients besides the global model, without overlap between datasets, *cf.* Fig.2. (5) ISIC20 dataset (Rotemberg et al., 2021) which consists of more than 33K images with malignant ($\sim$ 500 images) and benign ($\sim$ 32.5K images) classes. The last dataset is used as testing data to study how FedPerl generalizes to unseen data. Note that testing our method on the ISIC20 is a very challenging task due to the huge class imbalance and class distribution mismatch.

Baselines We conduct our experiments on the following baselines; (i) Local models: which include lower, upper, and SSL (FixMatch (Sohn et al., 2020)) models, these models are trained on their local data without utilizing the federated learning. (ii) Federated learning models: which include lower, upper, and SSFLs similar to (Yang et al., 2021), and FedMatch (Jeong et al., 2021) models, where these models trained locally on their data and

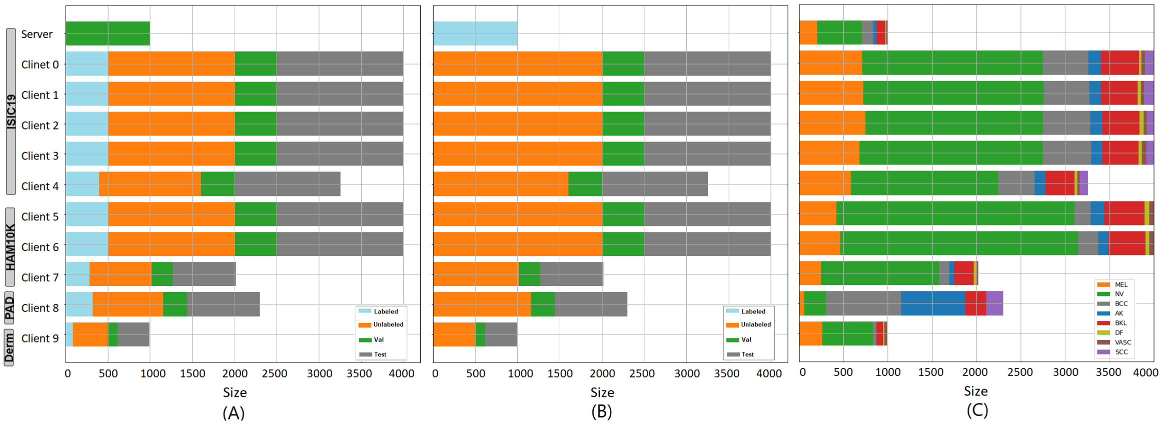

Figure 2: Illustrative diagram shows the distribution of our clients. The datasets are divided randomly into ten clients besides the global model, without overlap between the clients. (A) The standard semi-supervised learning scenario. Each client data is divided into testing (gray), validation (green), labeled (blue), and unlabeled (orange) data. (B) The unlabeled/disjoint clients scenario. The labeled and unlabeled images are combined and used as unlabeled images. (C) The class distribution. The data split is designed to simulate a realistic scenario with severe class imbalance, varying data sizes, and diverse communities. The clients 5-9 missing one or more classes.

utilizing the federated learning globally. (iii) Ablation for our method, namely `FedPerl` with(out) the PA. Note that for ease of implementation, we compare our method with one variant of `FedMatch` that do not implement weights decomposition.

Scenarios Our experiments conducted under four scenarios. In the first scenario, the standard semi-supervised learning, *cf.* Fig.2.(A), each client data is divided into testing (gray), validation (green), labeled (blue), and unlabeled (orange) data. The data split intended to resemble a realistic scenario with varying data size, severe class imbalance, and diverse communities, *e.g.*, the clients 0-4 originated from ISIC19, the clients 5-7 originated from HAM10000, and client 8 and 9 originated from Derm7pt, and PAD-UFES, respectively. We train the lower bounds on the labeled data, while we train `FixMatch`, `SSFLs`, and `FedPerl` on both labeled and unlabeled data. The upper bounds trained akin to SSLs, yet, all labels were exposed. In the second scenario, the unlabeled clients' scenario, we use the global data to train the global model. On the clients' side, however, the labeled and unlabeled images are combined and used as an unlabeled dataset, *i.e.* the labels were excluded from the training, *cf.* Fig.2.(B). While the second scenario is not yet investigated thoroughly in the medical images, we address it in this paper. In the third scenario, we test the ability of our model and the baselines to generalize to an unseen client (ISIC20) with new classes that have never been seen in the training. The fourth scenario proposed by (Liu et al., 2021) in which there are few labeled clients. For this scenario, clients 1 & 9 are selected as labeled clients while the remaining are not, such that they represent the largest community and individual clients, respectively.

Implementation Details We opt for EffecientNet (Tan and Le, 2019) pre-trained on ImageNet(Russakovsky et al., 2015) as a backbone architecture and trained using Adam optimizer (Kingma and Ba, 2014) for 500 rounds. We follow FedVC (Hsu et al., 2020) approach for clients federated learning. The idea of FedVC is to conceptually split large clients into multiple smaller ones, and repeat small clients multiple times such that all virtual clients are of similar sizes. Practically, this is achieved by fixing the number of training examples used for federated learning round to be fixed for every client, resulting in exactly optimization steps. The batch size B and participation rate (PR) were set to 16 & 30% (3 clients each round), respectively. The local training is performed for one epoch. The learning rate investigated in $[0.00001, 0.0001]$ and found best at 0.00005. τ investigated in $[0.5, 0.95]$, and found best at 0.6 & 0.9 for the federated and local models respectively. β investigated in $[0.1, 5]$, and found best at 0.5. γ investigated in $[0.01, 0.1]$, and found best at 0.01. T investigated in $\{2, 3, 4, 5\}$, and found best at $T = 2$. The dynamic learning policy threshold ρ tested at three values 0.75, 0.85, and 0.95, respectively. All images were resized to 224×224, and normalized to intensity values of $[0, 1]$. Random flipping and rotation were considered as weak augmentations, whereas RandAugment (Cubuk et al., 2020) was used as strong augmentation. We opt for PyTorch framework for the implementation hosted on standalone NVIDIA Titan Xp 12 GB machine. As the followed procedures in semi-supervised learning, `FedPerl` starts with warm-up rounds, e.g. 10 rounds in our case. The testing results are reported for the models with best validation accuracy. The average training time takes around 7 hours for each run for `FedPerl` models (w/o PA), about 5.85 hours for `FedPerl` (with PA), about 5.5 hours for `SSFL`, and about 6.25 hours for `FedMatch` shedding the light on the cost effectiveness of our approach. All the hyperparameters tuning was performed on a validation detest. Also, we made our code publicly available at `https://github.com/tbdair/FedPerlV1.0`.

Evaluation Metrics We report the statistical summary of precision, recall, and F1-score. A Relative Improvement (RI) *w.r.t* the baseline is also reported, where RI of a over b is : $(a - b)/b$. To highlight more in the model's performance at various threshold settings, we plot Area Under Receiver Operating Characteristic (AUROC) and Area Under Precision-Recall (AUPR) curves. Note that we follow the One vs ALL methodology for plotting. AUROC shows the model's ability to discriminate between positive examples and negative examples assuming balance data. Yet, AUPRC is a useful performance metric for imbalanced data, such as our case, where we care about finding positive examples. Further, we investigate on the uncertainty evaluation and models confidence. Thus, we report Risk-Coverage (RC) curve (Geifman and El-Yaniv, 2017), Reliability Diagram (RD) (Guo et al., 2017), and Expected and Maximum Calibration errors (Ding et al., 2020), denoted as ECE and MCE respectively. RC curve plots the risk as a function of the coverage. The coverage denotes the percentage of the input processed by the model without rejection, while the risk denotes the level of risk of the model's prediction (Geifman and El-Yaniv, 2017). For a selective model, the mode abstains the prediction of input sample x if the prediction confidence of that sample below a specific threshold e.g. 0.5. The higher coverage with lower risk, the better the model is. We refer the readers to section 2 in (Geifman and El-Yaniv, 2017) for the full definition of RC curve. Reliability Diagram, on the other hand, plots the accuracy as a function of confidence such that in the ideal case *i.e.* a perfect calibrated model, the RD will plot the identity function. For instance, suppose that we have 1000 samples, each with 0.85 confidence, we expect that 850 samples should be correctly classified. RD divides the predictions into different bins of confidence, *i.e.* $B_v; v \in \{1, ..., V\}$, where V is the total number of bins. Then, the average accuracy and the confidence for each bin B_v are calculated as $acc(B_v) = \frac{1}{|B_v|} \sum_{i \in B_v} \mathbf{1}(\tilde{y_i} = y_i)$, $conf(B_v) = \frac{1}{|B_v|} \sum_{i \in B_v} \tilde{p_i}$, respectively, where $\tilde{y_i}$, y_i, and $\tilde{p_i}$ are the prediction, ground truth, and the confidence for sample i, respectively. The difference (gab) between the accuracy and the confidence can be positive when the confidence is higher than the accuracy, and negative when the accuracy is higher than the confidence. These gabs shown in the RD using different colors, *cf.* sec.3.6 and Fig.9. For a perfect calibrated model, $acc(B_v) = conf(B_v)$ for all $v \in \{1, ..., V\}$. However, achieving a perfect calibrated model is impossible (Guo et al., 2017). Likewise, ECE and MCE are calculated, where ECE is defined as the difference in the weighted average of the bins' accuracy and confidence, while MCE represents the maximum difference, see Eq.13 and Eq.14 respectively.

$$ECE = \sum_{v=1}^{V} \frac{|B_v|}{s} \left| acc(B_v) - conf(B_v) \right|, \tag{13}$$

$$MCE = \max_{v \in \{1, ..., V\}} \left| acc(B_v) - conf(B_v) \right|, \tag{14}$$

where s is the number of samples in bin B_v. For a perfect calibrated model ECE and MCE both equal 0. To calculate the reliability diagrams and calibration errors, we adopted an adaptive binning strategy (Ding et al., 2020) that depends on fixable intervals in the calculations. This strategy is more accurate than using fixed intervals (Ding et al., 2020). Practically, we can realize the intervals used from the figure itself. For example, the width of the bars in figures 9 and 7 represents the ranges used to calculate ECE and MCE.

3.1 Proof-Of-Concept

First, we show proof of concept of our method on CIFAR-10 and FMNIST datasets and compare it with `FedMatch` (Jeong et al., 2021); a very recent work of `SSFL`. To have a fair comparison, we follow the codebases and the experimental setup they used. The results, reported in Table 1, show that `FedPerl` outperforms `FedMatch` (Jeong et al., 2021) in all experiments setup indicating the effectiveness of our method on finding the similarity (Sec 2.3.1), without introducing extra complexity, e.g., weight decomposition (Jeong et al., 2021). Additionally, our *peers anonymization* (PA) improves the accuracy and privacy at a low communication cost. Note that PA employs one anonymized peer while `FedMatch` uses two clients in the training. Interestingly, the FMNIST dataset results show that our method outperforms `FedProx-SL`, which is inconsistent with the CIFAR10 dataset results. Although, these results are not comparable because `FedProx-SL` results were taken from the original paper, whereas `FedPerl` results were generated by our environment. Yet, this could be attributed to the fact that FMNIST images are much simpler than the ones in CIFAR10 yielding more robust similarities and hence producing more accurate pseudo labels.

Table 1: The classification accuracy for the Proof-Of-Concept experiment on CIFAR10 & FMNIST datasets. *: Results as in FedMatch (Jeong et al., 2021). SL: Fully supervised. The SSL methods use 10% of the labeled data.

PA	Method (SSFL)	CIFAR10 IID	CIFAR10 NonIID	FMNIST NonIID
*	FedAvg-SL	58.60±0.42	55.15±0.21	-
*	FedProx-SL	59.30±0.31	57.75±0.15	82.06±0.26
*	FedAvg-UDA	46.35±0.29	44.35±0.39	-
*	FedProx-UDA	47.45±0.21	46.31±0.63	73.71±0.17
*	FedAvg-FixMatch	47.01±0.43	46.20±0.52	-
*	FedProx-FixMatch	47.20±0.12	45.55±0.63	62.40±0.43
*	FedMatch	52.13±0.34	52.25±0.81	77.95±0.14
w/o PA	FedMatch (Our run)	53.12±0.65	53.10±0.99	76.48±0.18
w/ PA	FedMatch (Our run)	**53.32±0.59**	**53.80±0.39**	**76.72±0.44**
w/o PA	FedPerl	**53.37±0.11**	**53.75±0.40**	**76.52±0.08**
w/ PA	FedPerl	**53.98±0.06**	**53.50±0.71**	**82.75±0.44**

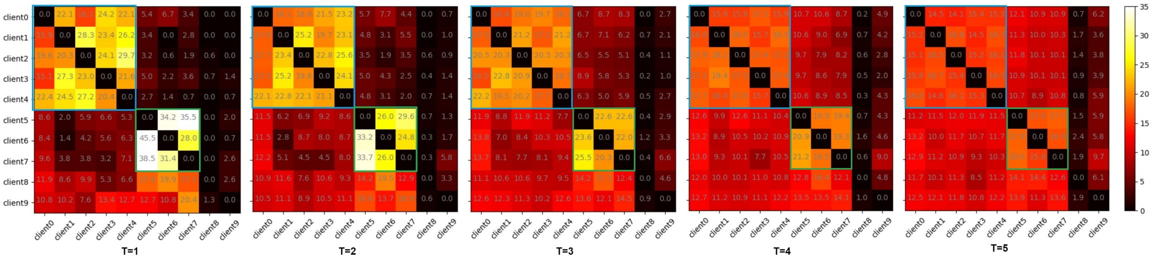

Figure 3: FedPerl clusters clients into two main communities (blue & green rectangles), while clients 8 & 9 do not belong to any community. As we increase the committee size T, the frequency of selecting peers within the same community decreases. The numbers and colors correspond to the frequency, where the brighter colors or higher numbers values represent higher frequencies.

3.2 Skin Lesion Results

Federated Learning Results In this section, we present the federated learning classification results before applying our method *i.e.* without peer learning nor PA. The results in Table 2 proves the current findings that `FedAvg` outperforms the local models significantly. For example, see Local/`FixMatch` vs. `FedAvg`, the obtained F1-score are 0.647 and 0.698, 0.664 and 0.734, and 0.726 and 0.773, respectively, with relative improvement (RI) up to 19.74%. Interestingly, both lower `FedAvg` and `FedAvg` ‡ (`SSFL`) models exceed the local SSL and upper bound models, respectively. That implies aggregating knowledge across different clients is more beneficial than exploiting local unlabeled or labeled data individually. Next, we discuss `FedPerl` results at different values of T.

FedPerl results without PA The results of `FedPerl` without applying peer anonymization is shown in Table 2 (denoted as w/o PA). The first concluding remarks reveal that peer learning enhances the local models. For illustration, our method outperforms the lower

Table 2: The results under the standard semi-supervised learning scenario. Mean (Median) ± Std. of different evaluation metrics. †:∼Yang et al. (2021). ‡:∼ FedMatch(Jeong et al., 2021). RI: Relative Improvement. AC: Additional Cost. The AC is calculated *w.r.t* the baseline (SSFL). For simplicity, we assume the initial cost for the SSFL is 0%. +: with PA.

Setting	Model	F1-score	Precision	Recall	RI(%)	AC(%)
Lower	Local	0.647(0.632)±0.053	0.644(0.622)±0.053	0.666(0.650)±0.053	-	
	FedAvg	0.698(0.690)±0.084	0.711(0.702)±0.072	0.709(0.700)±0.077	7.88	
SSL	FixMatch	0.664(0.636)±0.060	0.666(0.645)±0.063	0.692(0.671)±0.052	2.63	
SSFL	FedAvg†	0.734(0.725)±0.065	0.744(0.730)±0.064	0.739(0.728)±0.061	13.44	0
	FedMatch‡	0.739(0.729)±0.076	0.751(0.745)±0.068	0.744(0.732)±0.071	14.22	200
w/o PA	FedPerl(T=1)	**0.746(0.741)±0.071**	**0.753(0.744)±0.069**	**0.748(0.744)±0.069**	**15.30**	100
w/o PA	FedPerl(T=2)	**0.747(0.736)±0.071**	**0.756(0.741)±0.067**	**0.750(0.739)±0.069**	**15.46**	200
w/o PA	FedPerl(T=3)	**0.746(0.741)±0.072**	**0.757(0.743)±0.066**	**0.747(0.743)±0.070**	**15.30**	300
w/o PA	FedPerl(T=4)	**0.741(0.731)±0.077**	**0.751(0.735)±0.069**	**0.745(0.736)±0.072**	**14.53**	400
w/o PA	FedPerl(T=5)	**0.744(0.734)±0.073**	**0.753(0.744)±0.071**	**0.747(0.739)±0.069**	**15.00**	500
	FedMatch+‡	**0.745(0.737)±0.071**	**0.750(0.737)±0.067**	**0.750(0.746)±0.069**	**15.15**	100
	FedPerl(T=2)	**0.746(0.737)±0.075**	**0.754(0.741)±0.071**	**0.749(0.742)±0.073**	**15.30**	100
	FedPerl(T=3)	**0.746(0.738)±0.066**	**0.756(0.743)±0.060**	**0.748(0.740)±0.065**	**15.30**	100
	FedPerl(T=4)	**0.746(0.736)±0.077**	**0.755(0.745)±0.072**	**0.750(0.740)±0.074**	**15.30**	100
	FedPerl(T=5)	**0.749(0.739)±0.068**	**0.758(0.744)±0.065**	**0.750(0.742)±0.066**	**15.77**	100
Upper	Local	0.726(0.701)±0.044	0.729(0.705)±0.045	0.732(0.710)±0.042	12.21	
	FedAvg	0.773(0.757)±0.068	0.779(0.765)±0.065	0.773(0.759)±0.069	19.47	

model with RI between 14.53% and 15.46%. Further, `FedPerl` exceeds (`SSFL`) `FedAvg`† (Yang et al., 2021) and `FedMatch` (Jeong et al., 2021) by 1.8% and 1.08%, respectively. Moreover, our approach better than the local upper bound by 2.9%. Note that `SSFL` is considered a special case of `FedPerl` when $T = 0$. In addition, `FedPerl` results at a different number of peers T (committee size) are comparable, while the communication cost, comparing to the standard `SSFL`, increases proportionally with the increasing value of T (see AC in Table 2). Note that, the additional cost is calculated with respect to the baseline (`SSFL`). For simplicity, we assume the initial cost for the `SSFL` is 0%. Finally, the results imply that employing one similar peer ($T = 1$) is adequate to obtain remarkable enhancement with minimal communication cost, yet, at the loss of privacy. To address this, we propose **peers anonymization** technique.

FedPerl results After applying the peer anonymization, all models show a similar or slightly better performance when compared to the previous results (*i.e.* w/o PA), *cf.* Table 2. Yet, the new models enhance the baseline's performance, while still being better at hiding clients' identities and reducing the communication cost $O(1)$ regardless of the committee size T. Interestingly, applying peer anonymization not only enhances `FedPerl`, but also the `FedMatch` method. Specifically, the F1-score increases from 0.739 to 0.745, see `FedMatch` vs. `FedMatch+` in Table 2. Note that the additional advantages of `FedMatch`+ over `FedMatch` are the anonymized peer and the communication cost. The improvement of performance is attributed to the carefully designed strategy of creating the anonymized peer, such that the learned knowledge from many models ensembled into a single model. The results confirm the superiority of `FedPerl`, and show that our peer anonymization is orthogonal and can be easily integrated into other methods without additional complexity. In Fig. 4, we show the accuracy performance during the training. While we notice that similar clients have

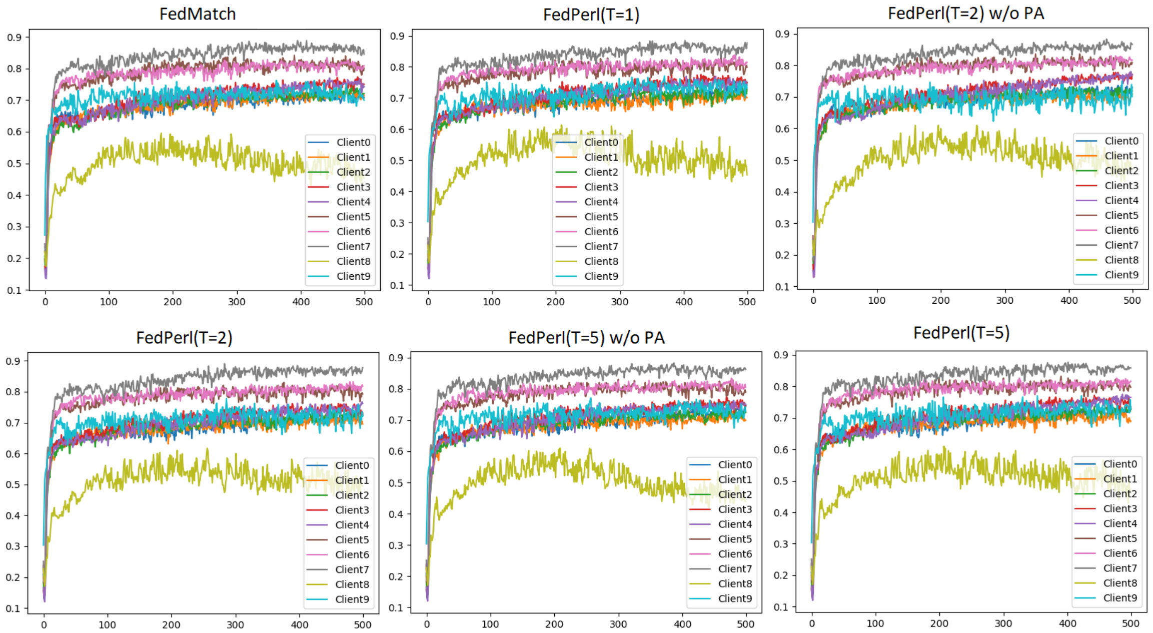

Figure 4: The accuracy performance during the training. Due to the large number of curves that can be presented, we opt for FedMatch and FedPerl at different community sizes. The similar clients have achieved similar training performance.

achieved similar training behavior, no further improvement in the last stages of the training was observed for all clients. For example, the accuracy for the clients 0-4 between 75-65, while it is between 80-90 for the clients 5-7. Client 9 achieved accuracy that is similar to clients 0-4. However, the best accuracy for client 8 was achieved in the middle of the training. This suggests handling Out-of-Distribution clients in federated learning has to be further investigated.

3.3 Building Communities Results

In this experiment, we investigate the importance of the similarity matrix used to rank similar clients and cluster them into communities. In Fig.3 we present the percentage of selecting peers during the training at different T values. To gain more insights, let us consider when the community size (T=2). For instance, the percentage of 33.2% between clients 6 and 5 reflects how often client 5 has chosen as a similar peer for client 6. The blue & green rectangles show that the clients clustered into two main communities. Interestingly, the clustering matches the clients' distribution we designed in our experiment, *cf.* Fig.2. For further analysis on community 1 (blue rectangle), we find the frequency of selecting peers from the same community for each client by calculating the horizontal summations (columns 0-4). The frequencies are $81.6\%, 85.6\%, 89.5\%, 86.4\%$, and 88.1% for the clients 0-4, respectively. That suggest, for example, client 0 learns from its community with a percentage of 81.6% of the training time. On average 86.24% of the time, first community members learn from each other, while it is 57.77% for community 2 (green rectangle). The same clustering also is shown for FedPerl at different committee sizes; $T = \{1, 3, 4, 5\}$. Note that the frequency values are gradually decreasing when a larger committee size is used for communities 1 & 2. The decrease in frequencies is expected because the likelihood of selecting peers from the outside of the community increases as we use a bigger committee size. Hence, the frequencies are distributed among the clients. In contrast, the frequencies for selecting peers for the individual clients (8 & 9) are comparable to each other at different T values. For further analysis on the community results, we average the classification results in each community and report them in Table 3. The first note from the results indicates that peer learning boosts the overall performance of the communities, compare the values of $T = 0$ vs. $T = \{1, 2, 3, 4, 5\}$ for C_{ISIC} and C_{HAM} respectively. Note that peer learning is not applied when $T = 0$. Further, we notice a stable performance for the C_{ISIC} community after applying the peer learning regardless of T values, yet with slight changes. However, an increasing then a decreasing performance is observed for the C_{HAM} at increasing values of T. This performance inconsistency is attributed to the community size. For instance, C_{ISIC} community includes 5 clients, while C_{HAM} community contains 3 clients. At first, let us consider C_{ISIC}. The probability of selecting peers, based on their similarities, from the outside community for different values of $T \geq 1$ is very low, and most likely the peers coming from the same community *i.e.* internal peers. For the case when $T = 5$, selecting an external peer is guaranteed. Yet, its effect is negligible comparing to the other clients, who most likely are internal peers. Now let us consider C_{HAM}. We notice that the performance increases gradually and reaches the best at $T = 2$. Based on our similarity matrix, the peers until this value most likely are internal peers, yielding to enhancement in the performance. Yet, after that (*i.e.* $T > 2$), involving external peers is confirmed. Consequently, the local model is distracted by increasing the number of external peers when using larger T values. Hence, the decrease in the performance. On the other hand, the individual clients' results are interesting (*i.e.* clients 8 & 9). While an enhancement is noticed for client 9, a reduction is observed for client 8. We note that the accuracy of client 9 is increased as the committee size increases thanks to peer learning. In general, the large the committee size, the better the performance. Yet, peer learning harms client 8. One explanation is attributed to the class distribution mismatch between client 8 and the

Table 3: The mean F1-score is reported to show the influence of peer learning on the community and individual clients. M: number of clients. *: SSFL.

Model	C_{ISIC} (M=5)	C_{HAM} (M=3)	8 (M=1)	9 (M=1)
FedPerl(T=0)*	0.718	0.816	0.602	0.703
FedPerl(T=1)	0.738	0.829	0.584	0.699
FedPerl(T=2)	0.736	0.833	0.567	0.717
FedPerl(T=3)	0.735	0.828	0.594	0.725
FedPerl(T=4)	0.735	0.826	0.562	0.727
FedPerl(T=5)	0.737	0.824	0.588	0.731

Table 4: The classification results for each client (mean F1-score). C_{ISIC}, C_{HAM}: the results at the community level. $Avg_{/C8}$: the results after excluding client 8. †:∼Yang et al. (2021). ‡:∼FedMatch(Jeong et al., 2021). Diff.% = $Avg_{/C8}$ - Avg. +: with PA.

Setting	Model/Client	0	1	2	3	4	C_{ISIC}	5	6	7	C_{HAM}	8	9	$Avg_{/C8}$	Avg	Diff.%
Lower	Local	0.581	0.618	0.603	0.622	0.596	0.604	0.742	0.738	0.670	0.717	0.656	0.641	0.646±0.056	0.647±0.053	-
	FedAvg	0.678	0.687	0.667	0.703	0.692	0.685	0.794	0.796	0.787	0.792	0.492	0.684	0.731±0.047	0.698±0.084	3.3
SSL	FixMatch	0.634	0.635	0.608	0.637	0.626	0.628	0.752	0.783	0.716	0.751	**0.650**	0.602	0.666±0.063	0.664±0.060	-
SSFL	FedAvg †	0.727	0.718	0.686	0.723	0.735	0.718	0.812	0.831	0.806	0.816	0.602	0.703	0.764±0.045	0.734±0.065	3.0
	FedMatch‡	0.724	0.724	0.722	0.734	0.751	0.731	0.801	0.850	0.813	0.822	0.553	0.717	0.760±0.046	0.739±0.076	2.1
	FedMatch+‡	**0.733**	**0.740**	**0.729**	**0.734**	0.744	**0.736**	**0.813**	0.843	**0.826**	**0.827**	0.581	0.703	**0.768±0.053**	**0.745±0.071**	2.3
w/o PA	FedPerl(T=2)	**0.735**	**0.731**	**0.725**	**0.737**	0.739	**0.733**	0.805	**0.850**	**0.839**	**0.831**	0.582	**0.729**	**0.769±0.047**	**0.747±0.071**	2.2
	FedPerl(T=2)	**0.737**	**0.737**	**0.724**	0.730	**0.751**	**0.736**	**0.818**	0.846	**0.834**	**0.833**	0.567	**0.717**	**0.765±0.046**	**0.746±0.075**	1.9
Upper	Local	0.698	0.698	0.677	0.700	0.696	0.694	0.806	0.804	0.752	0.787	0.702	0.722	0.728±0.046	0.726±0.044	-
	FedAvg	0.736	0.747	0.735	0.753	0.761	0.746	0.859	0.855	0.861	0.858	0.630	0.789	0.797±0.054	0.773±0.068	2.4

other clients, *cf.* Fig.2.(C). Further analysis is discussed in the next section concerning the individual clients' performance.

Random peers To investigate the importance of peer learning and our similarity matrix, we perform an additional experiment where the peers for the clients are selected randomly. The obtained F1-score is 0.736, with RI equals 13.75% and 0.27% *w.r.t.* the lower bound and SSFL, respectively. These results imply two conclusions. (i) Even with random clients, peer learning is still beneficial to training, compare this experiment results with the SSFL. (ii) Utilizing our similarity matrix brings extra knowledge by picking more accurate peers, compare this experiment results with the FedMatch models.

3.4 The Influence of Peer Learning on Clients

This experiment aims to gain more insights on the individual results and realize the influence of peer learning on clients and compare it with the baselines. The results are shown in Table 4. We observe that FedPerl exceeds the baselines, including the local upper bounds with salient margins, *e.g.*, for client 7 it is about 16.4% (Lower Local vs. FedPerl). In the same direction, FedPerl steadily surpasses FedMatch at the community's level and in all individual clients' results except for client 4. Yet, thanks to our PA, FedMatch+ shows better results than FedMatch at all communities and clients except for clients 4, 6, and 9. Surprisingly, FedPerl excels the upper FedAvg for client 0. The performance improvement is observed for all clients except client 8. One explanation is that FedPerl does not find suitable peers for client 8 to learn from due to the class distribution mismatch (*cf.* Fig. 2.(C)). For further investigation on the impact of client 8, we explore excluding it from the training. Then we compare the new and the previous results, both reported as $Avg_{/C8}$ and Avg respectively in Table 4. The comparison unveils that all federated learning models (*i.e.* FedAvg, FedMatch, and FedPerl) obtain better performance after excluding client 8. Still, the best performance is observed for FedPerl over the local upper and the (SSFL) FedAvg models. Note that the performance reduction after including client 8 in the training (see Avg in Table 4) implies the negative impact of this client. To realize that, we calculate the difference in performance before and after including client 8, *i.e.* $Avg_{/C8} - Avg$, and report the results in column $Diff.$ in Table 4. The resulted values show the negative impact of client 8 on the results. Where the higher the difference is, the higher the negative impact is. For example, it negatively impacted, the smallest on FedPerl (1.9%), moderate on FedPerl w/o PA (2.2%) and on both FedMatch methods (2.1% and 2.3%), and the

Table 5: The classification results for the eight classes (mean F1-score). +: with PA. †:∼Yang et al. (2021). ‡:∼FedMatch(Jeong et al., 2021)

Setting	Model	MEL	NV	BCC	AK	BKL	DF	VASC	SCC
Lower	Local	0.430	0.811	0.502	0.293	0.357	0.099	0.318	0.124
	FedAvg	0.501	0.834	0.646	0.377	0.507	0.173	0.642	0.111
SSL	FixMatch	0.451	0.831	0.540	0.304	0.374	0.052	0.292	0.135
SSFL	FedAvg †	0.565	0.852	0.680	**0.396**	0.570	0.416	0.707	0.253
	FedMatch†	0.573	0.852	0.700	0.366	0.565	0.462	0.701	0.275
	FedMatch+	**0.579**	**0.853**	**0.701**	0.376	**0.574**	**0.506**	**0.708**	**0.302**
w/o PA	FedPerl(T=2)	**0.576**	**0.854**	**0.706**	0.393	**0.589**	**0.552**	0.702	**0.305**
	FedPerl(T=2)	**0.602**	**0.854**	0.687	0.390	**0.592**	**0.493**	**0.712**	**0.315**
Upper	Local	0.551	0.853	0.651	0.428	0.520	0.308	0.654	0.308
	FedAvg	0.617	0.867	0.750	0.510	0.637	0.672	0.804	0.282

largest on `FedAvg` (3%). Such negative behavior could represent a threat in the federated learning, where a noisy and out-of-distribution client might hurt other clients and mislead the global model. Yet, the most interesting observation from this experiment that `FedPerl` is less prone to the negative and noisy impact than `SSFLs`, thanks to the training schema we proposed. We do not claim that `FedPerl` is robust against class distribution mismatch, but rather it is less sensitive to a noisy client. Nevertheless, the inconsistency in behavior between clients 8 & 9 could be further investigated.

On the other side, we notice that the enhancement after applying peer learning also observed at the community level; C_{ISIC} and C_{HAM} with 13.2% and 11.6%, respectively, confirming the finding in the previous section.

Note that our final objective consists of two terms that try to achieve the balance between the local and global benefits. Experimentally, we have shown that client 8 harms the clients. This impact was the minimum on `FedPerl` who is utilizing peer learning. Thus, we argue that involving peers, who influence the local models through participating in the pseudo labeling, has two advantages; (i) it restricts client 8 to send more reliable updates, and (ii) it reduces the negative influence of that client. Also, the T peers learn and coach the local client and guide it to be more accurate, where a noisy client could be fixed by averaging with more reliable clients.

3.5 Class Level Results

Because our setting is heterogeneous and suffers from severe class imbalance (*cf.* Fig.2.(C)), it is of importance to validate our method in that setting. Thus, we report the class level performance in Table 5. `FedPerl` obtains skin lesion classification accuracy better than local models (`FedPerl` vs. Local/`FixMatch`). For example, the improvement reaches ten times in the DF class. Moreover, `FedPerl` enhances the accuracy for BCC, BKL, DF, VASC, and SCC lesions by 16.6%, 21.8%, 50.4%, 42.0%, and 18.0%, respectively, in the SSL setting. The comparison with `FedMatch` reveals the same behavior seen in the previous results. First, our method, in general, outperforms `FedMatch` in all lesions. Second, applying PA to `FedMatch` (denoted as `FedMatch+`) boosts its accuracy. On the other hand, we observe an

Table 6: The area under ROC curve for the eight classes. +: with PA. †:∼Yang et al. (2021). ‡:∼FedMatch(Jeong et al., 2021)

Setting	Model	MEL	NV	BCC	AK	BKL	DF	VASC	SCC
Lower	Local	0.662	0.777	0.760	0.677	0.644	0.529	0.676	0.540
	FedAvg	0.709	0.834	0.827	0.634	0.739	0.575	0.909	0.528
SSL	FixMatch	0.670	0.804	0.785	0.685	0.658	0.515	0.644	0.540
SSFL	FedAvg†	0.737	0.846	0.827	**0.692**	0.777	0.675	0.889	0.583
	FedMatch‡	0.749	0.851	0.843	0.650	0.772	0.690	0.897	0.586
	FedMatch+	**0.751**	**0.854**	**0.847**	0.655	**0.778**	**0.717**	**0.904**	**0.596**
w/o PA	FedPerl(T=2)	**0.750**	**0.859**	**0.853**	0.659	**0.785**	**0.732**	**0.900**	**0.608**
	FedPerl(T=2)	**0.758**	**0.860**	0.838	0.660	**0.791**	**0.717**	**0.907**	**0.608**
Upper	Local	0.728	0.838	0.831	0.733	0.733	0.648	0.824	0.606
	FedAvg	0.773	0.869	0.848	0.750	0.805	0.876	0.958	0.594

Table 7: The area under Precision-Recall curve for the eight classes. +: with PA. †:∼Yang et al. (2021). ‡:∼FedMatch(Jeong et al., 2021)

Setting	Model	MEL	NV	BCC	AK	BKL	DF	VASC	SCC
Lower	Local	0.505	0.864	0.622	0.457	0.409	0.287	0.524	0.164
	FedAvg	0.582	0.894	0.702	0.443	0.556	0.394	0.712	0.313
SSL	FixMatch	0.527	0.879	0.646	0.476	0.453	0.358	0.534	0.216
SSFL	FedAvg†	0.620	0.903	0.730	**0.494**	0.617	0.574	0.762	0.348
	FedMatch‡	0.640	0.906	0.745	0.460	0.615	0.559	0.753	0.344
	FedMatch+	**0.645**	**0.908**	**0.752**	0.479	**0.632**	**0.598**	0.761	**0.349**
w/o PA	FedPerl(T=2)	**0.642**	**0.910**	**0.751**	0.476	**0.630**	**0.618**	0.754	**0.368**
	FedPerl(T=2)	**0.651**	**0.911**	0.744	0.475	**0.629**	**0.627**	**0.769**	**0.356**
Upper	Local	0.596	0.899	0.710	0.561	0.555	0.456	0.690	0.361
	FedAvg	0.668	0.916	0.762	0.574	0.670	0.719	0.847	0.373

insignificant decrease in the accuracy of the AK lesion. The key factor of FedPerl advantage is attributed to the knowledge exchanged through peer learning.

3.6 Additional Evaluation Metrics

Area under ROC & Precision-Recall curves For more validation, we report the area under ROC curve (AUROC) and the area under Precision-Recall curve (AUPRC) in Table 6 and Table 7 respectively. It is clearly shown that FedPerl exceeds SSFLs in all classes results except for the AK class. For instance, in AUROC results, the enhancement of FedPerl over SSFL around 2.1%, 1.4%, 1.1%, 1.4%, 4.2%, 1.8%, and 2.5% for the MEL, NV, BCC, BKL, DF, VASC, and SCC classes respectively. Moreover, the boosting of FedPerl over the lower FedAvg reaches 14% for the DF class. Interestingly, FedPerl outperforms both upper bounds for the SCC class. On the other side, the comparison of the AUPRC results reveals the same observations. Finally, the superiority of our method still found over FedMatch.

Risk Coverage curve We show the Risk-Coverage curves for FedPerl and our baselines in Fig.5. Each plot in the figure depicts a model. Inside each plot, we draw the curves

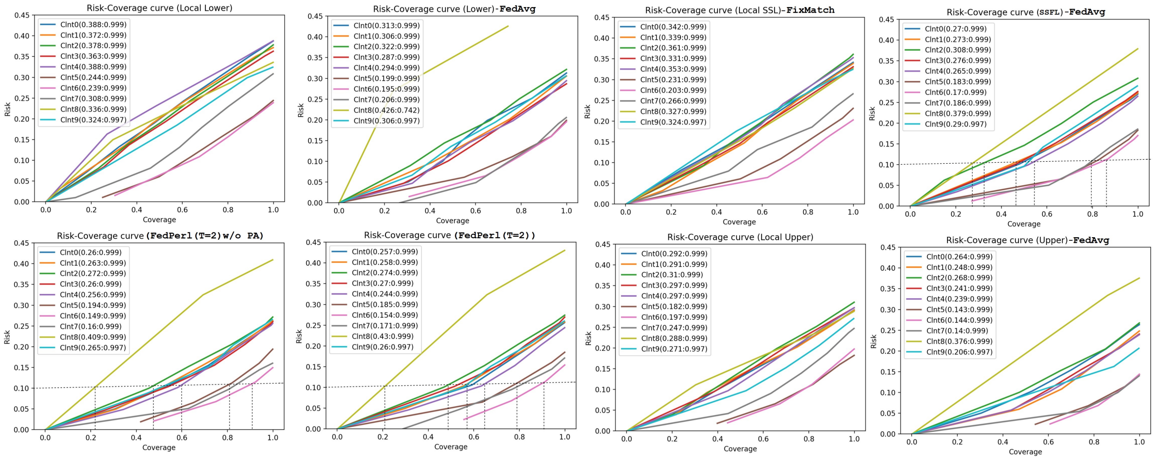

Figure 5: The area under Risk-Coverage curve for FedPerl and the baselines. The numbers that appear next to a client name represent the risk at the full coverage respectively *i.e.* (risk: coverage). In general, FedPerl obtains the lowest risk with the best coverage among all models.

for all clients. The numbers that appear next to a client name represent the risk value at the full coverage of the input data, *i.e.* (risk: coverage). It is shown from the figures that `FedPerl` achieves the lowest risk with the best coverage amongst all models, and this for all clients except for clients 5 & 8. Note that the coverage of client 8 in all federated models is worse than the local models, which is attributed to class mismatch. Please refer to sections 3.3 and 3.4 for more details. Nevertheless, if we consider the clients 0, 4, and 9 as examples, we observe that `FedPerl` obtains the maximum coverage at risks of 25.7%, 24.4%, and 26.0% respectively. These values are better than all local models including the upper local model, and better than `FedAvg` SSL (`SSFL`) model. Though, an insignificant drop in the coverage is noticed for client 5 comparing to `SSFL`. A detailed comparison between `FedPerl` and `SSFL` at 10% risk shows the superiority of `FedPerl` over `SSFL` in all clients, except client 8. For instance, the coverage jumps from (33% to 49%) for client 2, from the range of $(46.5\% - 55\%)$ to the range of $(53\% - 65\%)$ for clients 0, 1, 3, 4, and 9, and from the range of $(79.8\% - 86\%)$ to the range of $(79.5\% - 90\%)$ for clients 5, 6, and 7. Note that the minimum coverage of client 7 in `FedPerl` (at 0 risk) is 30%, while it is 0 coverage at 0 risk for `SSFL`. Client 6, on the other hand, achieves a minimum coverage of 56% at 2% risk. Utilizing our method achieves lower risk and better coverage in skin lesion classification.

Reliability Diagram and Calibration Error To investigate the uncertainty and models' calibration, we draw reliability diagrams and the expected and the maximum calibration errors in Fig.9. We show the results for the federated models include ours. The numbers inside the sub-figures show the calibration error for each client. The numbers next to the model name show the averaged ECE and MCE errors for all clients. In each figure, we present the models' accuracy at different confidence intervals, such that the width of each bin represents the difference between the highest and lowest confidences. The figures show that our method improves the calibration for all models and reduces the errors significantly *cf.* Fig.9 (`FedPerl` vs. `FedAvg` models). While the most interesting and surprising results

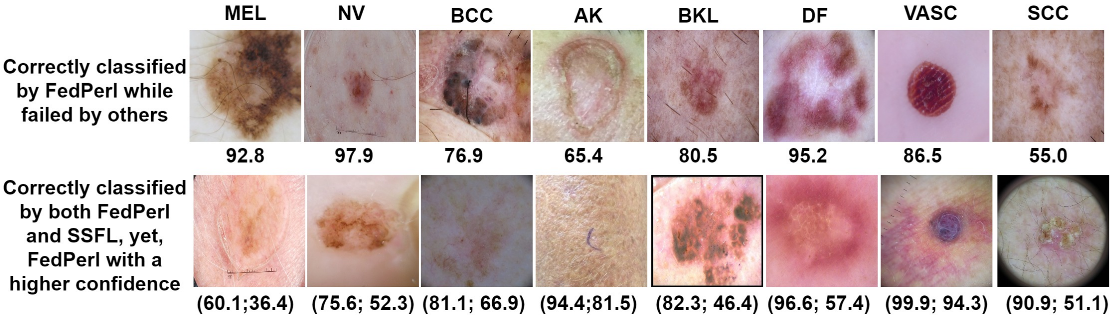

Figure 6: Qualitative results. Sample predictions of FedPerl and SSFL for skin lesion. FedPerl confidence is shown below the images in the first raw, while the second raw shows the confidence for FedPerl and SSFL respectively.

reveal that the lower federated model (Lower `FedAvg`) is the most calibrated model after the upper model (Upper `FedAvg`), such that it is better than `SSFL` and `FedPerl` respectively. We can attribute this issue to the uncertainty of using unlabeled data during the training of both models (`SSFL` and `FedPerl`). In contrast to that, the lower and the upper `FedAvg` models only trained on high-quality labeled data. Nonetheless, our model has better calibration errors than the `SSFL`, where the ECE and MCE are 0.144 and 0.277 for `FedPerl`, and 0.152 and 0.287 for `SSFL`, respectively. Besides, `FedPerl` outperforms the remaining baselines with considerable margins, further results are presented in Fig.**??** in **??**. Such lower calibration errors indicate more reliable and confident predictions for the `FedPerl` over the other methods. Moreover, our experiments showed that applying peer learning produced a more calibrated model than `SSFL`, *cf.* Fig.9 (`FedPerl` vs. `SSFL` models). Yet, after applying peer anonymization, a better calibration error is obtained, *cf.* Fig.9 (`FedPerl(T=2)` vs. `FedPerl(T=2) w/o PA` models). That implicitly means that the used peers are calibrated enough to produce more accurate pseudo labels than the ones generated from the clients individually.

Skin lesion qualitative results Sample predictions of `FedPerl` are shown in Fig.6. The first row shows samples cases were classified correctly by `FedPerl` but miss-classified by the other methods. Below each case, we show the prediction confidence. The first row shows the confidence for `FedPerl`, while the second raw shows the confidence for both `FedPerl` and `SSFL` respectively. It is noticed that there are challenging cases, still, `FedPerl` was able to classify them correctly, e.g. AK and SCC classes. The remaining cases were classified correctly with high confidence by `FedPerl`, while they miss-classified by the others. The second row, on the other hand, shows cases were classified correctly by both `FedPerl` and `SSFL`, yet, `FedPerl` achieves higher confidence. For instance, in BKL, DF, and SCC classes, the confidence margins are 35.9, 39.2, and 39.8, respectively.

3.7 Unlabeled Clients Scenario

Till this experiment, we have trained our models to exploit the labeled and unlabeled data at each client. The previous setting is widely studied in the literature a.k.a the standard semi-supervised learning paradigm. In federated learning, however, a more challenging situation

Table 8: The classification results for unlabeled clients scenario. +: with PA. †:∼Yang et al. (2021). ‡:∼FedMatch(Jeong et al., 2021)

Setting	Model	F1-score	Precision	Recall
SSFL	FedAvg†	0.637(0.649)±0.121	0.647(0.649)±0.099	0.670(0.678)±0.120
	FedMatch‡	0.641(0.662)±0.131	0.653(0.657)±0.099	0.667(0.693)±0.134
w/o PA	FedPerl(T=1)	**0.644(0.662)±0.115**	**0.658(0.660)±0.078**	**0.674(0.688)±0.118**
w/o PA	FedPerl(T=2)	**0.644(0.670)±0.126**	0.651(0.664)±0.100	**0.671(0.691)±0.130**
w/o PA	FedPerl(T=3)	**0.645(0.654)±0.117**	**0.655(0.657)±0.094**	**0.670(0.677)±0.123**
w/o PA	FedPerl(T=4)	**0.644(0.660)±0.124**	**0.655(0.665)±0.103**	0.668(0.678)±0.129
w/o PA	FedPerl(T=5)	**0.641(0.659)±0.129**	**0.655(0.660)±0.098**	0.668(0.681)±0.134
	FedMatch+	**0.649(0.662)±0.118**	**0.655(0.659)±0.102**	**0.677(0.688)±0.121**
	FedPerl(T=2)	**0.645(0.662)±0.119**	**0.654(0.659)±0.103**	**0.673(0.687)±0.119**
	FedPerl(T=3)	**0.648(0.663)±0.118**	**0.660(0.669)±0.102**	**0.678(0.693)±0.120**
	FedPerl(T=4)	**0.649(0.666)±0.124**	**0.656(0.663)±0.102**	**0.678(0.692)±0.125**
	FedPerl(T=5)	**0.645(0.659)±0.114**	0.652(0.653)±0.096	**0.675(0.687)±0.118**

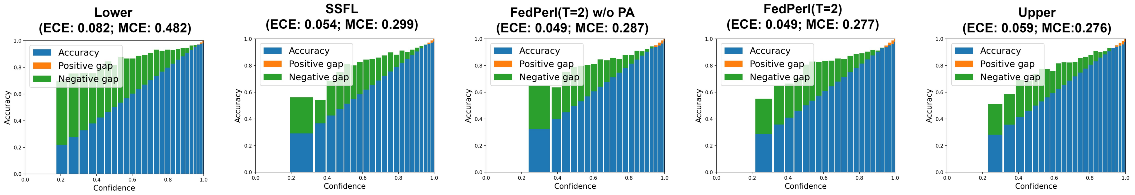

Figure 7: Reliability diagrams and calibration errors on the ISIC20 dataset. FedPerl is more calibrated with lower calibration errors than the baselines.

may appear to the surface in which the clients only have access to unlabeled data without knowing their annotations, see *Scenarios* in sec. 3 for more details. The results of applying this scenario to `FedPerl` and our baselines are reported in Table 8. Thanks to peer learning, our method enhances the performance of the baselines up to 0.8% and 0.4% compared to `FedAvg` and `FedMatch`, respectively. Moreover, an additional improvement of about 1.2% is obtained after applying peer anonymization (see last four rows in Table 8). That also holds for `FedMatch` where `FedMatch+` shows a relative improvement of about 0.8% after applying PA. The better results are attributed to the aggregated knowledge from distributed similar clients who help the local models to overcome the missing of labeled data.

3.8 Generalization to Unseen Client Scenario

The goal of this experiment is to investigate the generalization ability of the federated models to unseen clients. To achieve this, we collect the previously trained global models, including the baselines and `FedPerl`, then we perform inference on the ISIC20 dataset. Note that this dataset consists of more than 33K images with two classes; malignant and benign. Take into consideration that the class distribution is highly imbalanced such that around 500 images contain malignant cases, while the remaining images have benign cases. Also, the models trained to distinguish between 8 classes making the direct inference a very challenging task. To resolve this issue, we perform two steps. First, we generate

Table 9: The unseen client scenario. The global models' classification results for FedPerl and the baselines on the ISIC20 dataset. +: with PA. †:∼Yang et al. (2021). ‡:∼FedMatch(Jeong et al., 2021)

		Malignant			Benign		
Setting	Model	F1-score	Precision	Recall	F1-score	Precision	Recall
Lower	FedAvg	0.131	0.097	0.204	0.976	0.985	0.966
SSFL	FedAvg†	0.161	0.114	0.274	0.974	0.987	0.962
	FedMatch‡	0.160	0.113	0.278	0.972	0.987	0.954
w/o PA	FedPerl(T=1)	0.160	0.112	**0.279**	0.973	0.987	0.960
w/o PA	FedPerl(T=2)	**0.178**	**0.126**	**0.305**	0.974	0.987	0.962
w/o PA	FedPerl(T=3)	**0.166**	0.110	**0.339**	0.969	**0.988**	0.951
w/o PA	FedPerl(T=4)	**0.169**	**0.117**	**0.308**	0.972	0.987	0.958
w/o PA	FedPerl(T=5)	**0.166**	**0.120**	0.269	**0.975**	0.987	**0.965**
	FedMatch+	0.146	0.099	**0.281**	0.970	0.987	0.954
	FedPerl(T=2)	**0.163**	0.113	**0.295**	0.973	0.987	0.959
	FedPerl(T=3)	**0.167**	0.114	**0.308**	0.972	0.987	0.957
	FedPerl(T=4)	**0.170**	**0.115**	**0.324**	0.971	0.987	0.956
	FedPerl(T=5)	0.150	0.099	**0.305**	0.968	0.987	0.950
Upper	FedAvg	0.153	0.095	0.382	0.961	0.988	0.935

the eight-class predictions from the models. Then, we assemble these predictions into two groups. The malignant group contains melanoma, basal cell carcinoma, actinic keratosis, and squamous cell carcinoma classes. The benign group includes melanocytic nevus, benign keratosis, dermatofibroma, and vascular lesions. Then, we generate our metrics as a binary classification task.

The results are reported in Table 9. Interestingly, FedPerl obtains the best malignant-class classification results outperforming the lower, the SSFL including FedMatch, and the upper bounds, with F1-score up to 0.178 for FedPerl models. Note that, for clinical applications, the ability of a model to detect the true positive cases (malignant) is high relevant than detecting the true negative cases (benign) because the early detection of cancerous lesions reduces the treatment cost and the death rate. The ability of FedPerl to classify the malignant and benign classes is also shown in the reliability diagrams and calibration errors, *cf.* Fig.7. We can see from the figure that FedPerl is more calibrated and achieves better expected and maximum calibration errors than SSFL. From these results, we show that FedPerl has a better generalization ability to detect malignant cases than the baselines and FedMatch. While we have seen in all previous experiments that applying PA to FedMatch (denoted as FedMatch+) always boosts its performance, this observation does not hold in this experiment. Specifically, the F1-score drops from 0.160 to 0.146. The same observation is found for some FedPerl models.

3.9 Comparison with SOTA in the Few Labeled Clients Scenario

In this experiment, we conduct a comparison with FedIRM (Liu et al., 2021); very recent work in SSFL for the skin lesion classification. Notice that FedIRM introduced a scenario where some clients are labeled while others are not. In addition, the training paradigm in `FedIRM` assumed that all clients participate in the training in each round, *i.e.* $PR = 100\%$, which is not applicable in many cases. The vast majority of federated learning approaches assume that a random set of clients will participate in the training each round, which was our selection in this paper where the $PR = 30\%$. Thus, to cover both cases, we present the results at $PR = \{30\%, 100\%\}$. For our comparison, we opt `FedAvg` and `FedPerl(T=2)` models. Note that the hyperparameters are kept as in the previous experiments, while the results are reported in Fig.8. First, let us consider when $PR = 30\%$. `FedAvg` obtains F1-score equals 62.3 while `FedIRM`, `FedPerl` w/o PA, and `FedPerl` achieve comparable results at 66.3, 66.1, and 66.1, respectively. Although our method outperforms `FedAvg` ($PR = 100\%$), we observe a slight relative drop in the performance, when we compare to `FedIRM`, by 1.4% and 0.3% for `FedPerl` w/o PA and `FedPerl`, respectively. That could be attributed to that `FedIRM` only transfers the knowledge from labeled to unlabeled clients to guide the pseudo labeling process. However, this is not the case in our method where we utilize similar peers (regardless of their labels). Note that around 80% of the clients are unlabeled in this particular scenario favoring the `FedIRM` method. Still, `FedPerl` outperforms `FedAvg`, and extensive hyperparameters tuning could yield better performance for our method. For the same reasons, `FedPerl` w/o PA, when $PR = 100\%$, achieves the lower results among `FedPerl` models with F1-score equals 65.7, where more unlabeled clients were involved in the training. Yet, averaging the unlabeled peers might cancel their negative impact on the local model, as shown by `FedPerl` with peer anonymization (PA) at F1-score = 68.7.

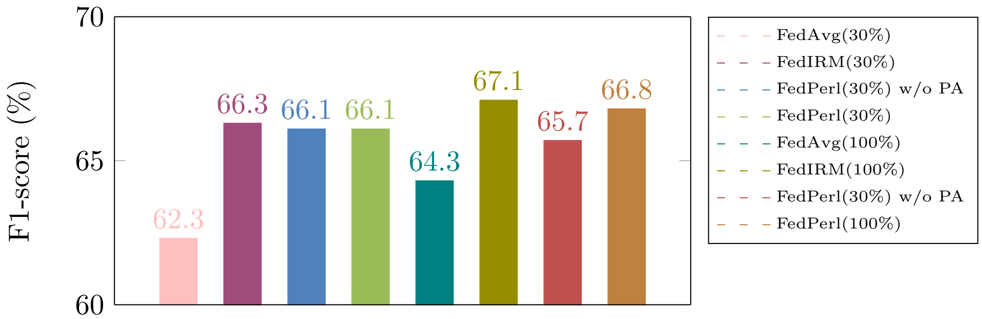

Figure 8: Comparison between our Method and FedIRM. While both methods achieve comparable results when the participation rate =30%, ours show lower performance when 100% of the participation rate. Still, FedPerl outperforms FedAvg.

3.10 Dynamic Learning Policy

Previously, we have shown that the static peer learning policy is constantly beneficial to clients and communities. For instance, see the results in Table 4. Also, we have shown that for the individual clients, who do not belong to any community, our method is still profitable, as for client 9. Yet, for other clients, *i.e.* client 8, we have seen that peer learning, `FedMatch`, and `FedAvg` perform lower than the local model. Even though our model is better than the others. To resolve this issue, we propose, in section 2.3.5, a dynamic learning policy that controls the learning stream on the clients. The results are reported in Table 10. Due

Table 10: Dynamic learning polices results. The classification results under two different settings. C_{ISIC}, C_{HAM}: the results at the community level. (mean F1-score). PA+/-: with/out Peers anonymization.

The classification results when the clients contain labeled and unlabeled data (the standard SSL setting)														
Policy	Model/Client	0	1	2	3	4	C_{ISIC}	5	6	7	C_{HAM}	8	9	Avg
No Policy (baselines)	PA- FedPerl(T=2)	0.735	0.731	0.725	0.737	0.739	0.733	0.805	0.850	0.839	0.831	0.582	0.729	0.747
	PA+ FedPerl(T=2)	0.737	0.737	0.724	0.730	0.751	0.736	0.818	0.846	0.834	0.833	0.567	0.717	0.746
Validation Policy	PA- FedPerl(T=2)	0.729	0.727	0.724	**0.737**	0.746	0.732	0.814	0.845	0.819	0.826	0.571	**0.732**	0.744
	PA+ FedPerl(T=2)	**0.743**	0.729	**0.736**	0.732	0.749	**0.738**	0.806	0.845	0.822	0.824	0.572	0.724	0.746
Gated Validation Policy	PA-(75) FedPerl(T=2)	0.729	0.727	**0.738**	0.732	0.750	0.735	0.814	0.841	0.828	0.827	**0.598**	0.725	**0.748**
	PA+(75) FedPerl(T=2)	**0.746**	0.727	**0.737**	0.731	0.748	**0.738**	0.814	0.844	0.834	0.831	0.571	0.725	**0.748**
	PA-(85) FedPerl(T=2)	0.728	0.734	**0.740**	**0.738**	**0.760**	**0.740**	0.814	0.842	0.830	0.829	0.535	0.710	0.743
	PA+(85) FedPerl(T=2)	**0.738**	0.732	0.723	**0.742**	**0.755**	**0.738**	**0.818**	**0.850**	0.838	**0.835**	**0.596**	**0.729**	**0.752**
	PA-(95) FedPerl(T=2)	**0.739**	0.732	**0.735**	**0.737**	**0.754**	**0.739**	0.815	**0.850**	**0.839**	**0.834**	**0.583**	**0.730**	**0.751**
	PA+(95) FedPerl(T=2)	**0.747**	**0.745**	**0.731**	**0.738**	**0.752**	**0.743**	**0.818**	**0.851**	**0.839**	**0.836**	**0.596**	**0.731**	**0.755**
Gated Similarity Policy	PA-(75) FedPerl(T=2)	0.734	0.721	**0.742**	**0.739**	**0.764**	**0.740**	**0.821**	0.843	0.832	0.832	**0.593**	0.714	**0.750**
	PA+(75) FedPerl(T=2)	**0.740**	0.725	**0.735**	**0.742**	**0.754**	**0.739**	0.811	0.841	0.827	0.826	0.580	0.699	0.745
	PA-(85) FedPerl(T=2)	0.727	**0.741**	**0.731**	**0.742**	**0.751**	**0.739**	0.812	0.846	0.825	0.828	**0.586**	0.716	**0.748**
	PA+(85) FedPerl(T=2)	**0.738**	0.735	**0.735**	**0.742**	**0.752**	**0.741**	**0.820**	**0.850**	**0.839**	**0.836**	**0.588**	**0.730**	**0.753**
	PA-(95) FedPerl(T=2)	0.734	0.728	**0.732**	**0.739**	**0.765**	**0.740**	**0.820**	0.845	**0.839**	**0.835**	**0.617**	**0.731**	**0.755**
	PA+(95) FedPerl(T=2)	**0.737**	**0.740**	**0.731**	**0.739**	**0.764**	**0.742**	**0.819**	**0.853**	**0.836**	**0.836**	**0.618**	**0.732**	**0.757**
The classification results when the labeled data is only available on the server while the clients have no labeled data (the unlabeled clients or the disjoint setting)														
Policy	Model/Client	0	1	2	3	4	C_{ISIC}	5	6	7	C_{HAM}	8	9	Avg
No Policy (baseline)	PA+ FedPerl(T=2)	0.649	0.642	0.671	0.645	0.654	0.652	0.730	0.751	0.729	0.737	0.308	0.670	0.645
Validation Policy	PA+ FedPerl(T=2)	0.642	0.638	0.669	**0.647**	**0.678**	**0.655**	0.721	**0.752**	**0.740**	**0.738**	0.267	0.656	0.641
Gated Validation Policy	PA+(75) FedPerl(T=2)	0.642	0.639	0.667	**0.647**	**0.659**	0.651	**0.740**	0.750	**0.745**	**0.745**	0.303	**0.678**	**0.647**
	PA+(85) FedPerl(T=2)	**0.669**	**0.657**	0.664	0.643	**0.666**	**0.660**	**0.740**	0.740	**0.744**	**0.740**	**0.340**	0.687	**0.655**
	PA+(95) FedPerl(T=2)	**0.653**	**0.650**	0.662	0.630	**0.669**	**0.653**	**0.743**	**0.751**	**0.733**	**0.742**	**0.343**	0.664	**0.650**
Gated Similarity Policy	PA+(75) FedPerl(T=2)	**0.659**	**0.660**	**0.671**	**0.650**	**0.657**	**0.659**	**0.732**	0.749	**0.744**	**0.742**	**0.334**	**0.671**	**0.653**
	PA+(85) FedPerl(T=2)	**0.649**	**0.645**	**0.679**	0.636	**0.656**	**0.653**	0.728	**0.757**	**0.751**	**0.745**	**0.327**	**0.670**	**0.650**
	PA+(95) FedPerl(T=2)	**0.658**	**0.647**	**0.678**	**0.655**	**0.676**	**0.663**	**0.732**	**0.759**	**0.738**	**0.743**	**0.336**	**0.672**	**0.655**

to the enormous amount of models that could be examined in this experiment, we opt for ρ at $\{0.75, 0.85, 0.95\}$ and $T = 2$. We generate the results for the standard semi-supervised and unlabeled clients scenarios. Our baseline in this experiment is our model `FedPerl(T=2)` as our goal is to compare with the static policy, and we do not see any need to include the previous models which already compared with `FedPerl(T=2)`.

3.10.1 The standard semi-supervised learning results

Validation Policy First, by comparing overall results, denoted as *Avg* in Table 10, we notice no significant improvement in the performance for both models; PA($\pm$) `FedPerl(T=2)`. On the other hand, lower results are obtained for the C_{HAM} community. For instance, the F1-score dropped from 0.831 and 0.833 to 0.826 and 0.824, respectively. In contrast, a comparable result at 0.732 or a slight enhancement at 0.738 are obtained for C_{ISIC}. Besides, the clients' results are inconsistent regardless if they belong to a community or not. While we notice boosting for the clients 0, 2, 3, and 9, the remain clients have lower results. Further, we notice no positive influence on the results when applying PA.

Gated Validation Policy While there is not much benefit in the previous policy, the results in this experiment show a consistent improvement as the gateway threshold ρ increases. For instance, the overall results boosted up to 0.2%, 0.6%, and 0.9% when $\rho = 0.75$, 0.85, and 0.95, respectively. The consistent improvement also found at the community level when ρ is larger than 0.75, with better results at $\rho = 0.95$. While an increase reaches 1% is noticed for C_{ISIC} clients starting from $\rho = 0.75$ with PA model *i.e.* PA+(75) `FedPerl(T=2)`, the increase is seen starting from PA+(85) model for C_{HAM} clients with F1-score reaches 0.836. In general, the clients' results get boosted by our gated validation policy. In the beginning, when $\rho = 0.75$, clients 0, 2, and 8, show better performance comparing to the baseline. Then, more clients are included when $\rho = 0.85$ until all clients show improvement with our model PA+(95) with F1-score at 0.752 at client 4. These results confirm the same behavior found in communities' results. A more discussion on the individual clients' results, *i.e.* 8 & 9, reveals that the combination of PA with values of $\rho = \{0.85, 0.95\}$ achieves more reliable F1-scores. Even though our model PA-(75) obtains the highest score for client 8, the results for other clients are not of the same quality. In summary, we present in this experiment that our gated validation policy improves the overall, communities', and clients' results demonstrating its advantage. More importantly, the results of client 8 were boosted from 0.567 to 0.596 at PA+(95) model.

Gated Similarity Policy This policy is different from the earlier one in using the similarity between the client and its peers as a gateway to control peers' participation instead of using the global validation dataset. We notice that the general behavior is similar to the preceding one. Though, better results are obtained at different levels, especially for client 8, whose reported F1-scores are equal to 0.617 and 0.618 on models PA±(95), which are better than the former ones by 1.9% and 4.7% respectively. The similarity in the results is justified because both policies proposed to manage the learning stream on the clients, especially the individual ones, which has been shown in both strategies. A gated similarity policy brings more stability to all clients and better accuracy for client 8.

3.10.2 The unlabeled clients' results

We have shown in the past section that the validation policy has no potential improvement, while the combination of the gated methods with PA usually obtains the best performance. Therefore, and for simplicity, we opt to report only the results with the PA technique.

After analyzing the second part of Table 10. We notice that the results of validation policy are improved by F1-score equals to 0.655 and 0.738, for C_{ISIC} and C_{HAM} respectively. Yet, the overall results decreased by 0.4%. The individual results, on the other hand, vary between the clients. While clients $3, 4, 6$, and 7 show an enhancements, clients' $0, 1, 2, 5, 8$, and 9 accuracies are decreased. In contrast to the previous results, we observe a constant improvement of gated policies in the overall accuracy from 0.647 to 0.655 for the validation with PA+(75) to similarity with PA+(95) gated models, respectively. Note that all models from both policies accomplish better results than the baseline model. The communities' results, on the other hand, show comparable results, yet better than the baseline, for both strategies with some advantages for the similarity models. While the individual improvement is distributed among the clients in the gated validation approach except for client 2, it is intelligible in similarity models, especially in PA+(95) model. Moreover, both individual clients; 8 & 9, show steady improvements in all similarity models, yet, client 9 suffers from lower performance in gate validation with ρ larger than 0.75. However, the maximum gain appears for client 8 in the PA+(95) gated validation policy with F1-score equals 0.343.

4. Discussion

Our method; `FedPerl` compiles many concepts such as semi-supervised learning, federated learning, peer learning, committee machine, and learning policies to devise a novel framework for skin lesion classification tasks. We show through extensive experiments and evaluation metrics that our method has superior performance over the baselines in the standard semi-supervised labeled and unlabeled clients settings.

FedPerl simplicity & performance. A key feature of our method is simplicity. Implementing and applying our method is direct and can be implemented with a few lines of code. The computational cost to calculate the similarity between the clients is negligible, thanks to our strategy which computes the similarity on extracted features rather than on the whole weight parameters. Such that for a model with l layers and ω weights parameters, where $\omega \ggg l$, the cost of our similarity is $\mathrm{O}(l) \lll \mathrm{O}(\omega)$, note that ω could be million of parameters. From another perspective, the experiments show that `FedPerl` is more calibrated and outperforms the baselines including the `SSFL`, thanks to the peer learning we propose, where `FedPerl` exploits other clients by interacting with their experiences. As a core component in our method, peer anonymization reduces the communication cost while enhances performance. Additionally, it improves the clients' privacy by hiding their identities. Yet, a non-avoidable cost is still property in peer learning.

Similarity. Clients' communities are shaped implicitly based on the similarities between the clients. To measure the similarities, we exploited models parameters to profile the clients. Yet another approach to quantify the similarity is to use a server-side validation set as it has been utilized in `FedMatch`. While we have shown through different experiments that our method of finding the similarities outperformed the one that depends on validation set *i.e.* `FedMatch`, another drawback is that the availability of validation datasets at the server side is a challenging task. Further, we have shown the importance of peer learning and our similarity in the random peers experiment. Still, the representational similarity is an open research direction in federated learning.

Orthogonality. Another main property of the PA technique is that it can be implemented directly to other methods, which are similar to ours, with negligible effort. We have shown through different experiments that applying PA to `FedMatch` is resulted in a better model, *i.e.* `FedMatch+`. While the new model achieves better accuracy, it also reduces the communication cost comparing to the original one.

Privacy. Our anonymized peer is designed by aggregating/averaging the model parameters of the top T similar peers. This process generates a virtual model that is not related to a specific client and offers a harder target for attackers seeking information about individual training instances (McMahan et al., 2017; Orekondy et al., 2018). Nevertheless, a privacy guarantee for aggregated models (not individuals) is an open issue and has not been thoroughly investigated in the community and mathematical analysis is yet to be proven.

Local Updates. While the local models' weights are continually updated during the training, the peers' ones remain intact. A natural question could be if such a procedure might poison the models, especially with larger iteration updates? While such concern is of high importance, we have designed our method to alleviate this problem by training the local model and keeping the peer models intact to avoid any poisoning. Also, we employed an MSE loss as a consistency-regularization, `FedVC` approach in the federated learning, and our dynamic policy, especially if the local model is quite different from the peers and has been trained for more local iterations.

FedPerl communities & committee size. Figure 3 shows that `FedPerl` clusters the clients into communities based on their similarities. The overall performance for each community gets boosted by `FedPerl`, *cf.* Tables 3 and 4, which is attributed to the knowledge sharing. We have noticed that the community performance is related to the committee size *i.e.* T. While changing T has an insignificant effect on C_{ISIC} performance, its effect is clear on C_{HAM}. Thus, a natural question would be, what is the ideal committee size? Our experiments show that as long as T below the actual community size, the overall performance is rather stable, *cf.* C_{ISIC} in Table 3. Once T exceeds the community size, the performance starts decreasing, *cf.* C_{HAM} in Table 3. We associate this with the probability of including external peers as we increase T, which might have a negative influence on the local models/sites of the community, see sec.3.3. While the cluster/community size can be defined by the cardinality of the spectral clustering of the similarity matrix, yet in more practical scenarios, setting T to a value larger than the community size is impractical. The trade-off between the committee size and the performance needs further investigation.

FedPerl clustering. The clustering in the literature means grouping similar data and assigning labels to them. Because we use this word frequently in our paper and to resolve any ambiguity, we provide the following interpretation. First, we do not use any clustering method nor it is defined heuristically or fixed at the beginning of the federated learning. Hence, we do not assign labels to the clusters, but rather we want to highlight that our similarity matrix works effectively to force similar peers to learn from each other. At the beginning of the training, the clusters, *i.e.* learning from similar peers, are dynamically changed, which is explained by the small numbers in each row in Fig. 3, where the darker colors or smaller numbers values represent lower frequencies. However, as the training proceeds, these clusters are evolved to force the similar peers to learn from each other more frequently, which is shown by the brighter colors or higher numbers values in the same figure.

FedPerl & individual clients. The clustering produces individual clients who do not belong to a specific community *i.e.* clients 8 & 9, which confirms the reality. The effect of `FedPerl` is diversified between those two clients. While client 9 makes use of `FedPerl`, a drastic drop in the performance of client 8 was noticed, which could be attributed to the class distribution mismatch. This indicates that `FedPerl` may not fit non-iid scenarios. Yet, combining `FedPerl` with works that are handling the distribution mismatch (non-iid) problem would be a promising direction of research (Li et al., 2018; Zhao et al., 2018; Li et al., 2021; Zhang et al., 2021). On the other side, one nice property has been shown by our experiments that `FedPerl` is less sensitive to the noisy clients than the standard `SSFL` and `FedAvg` methods (*cf.* Table. 4), which could be attributed to the learning schema of selecting similar peers in `FedPerl`. In our experiments, we found out that inductive bias coming from similar in-distribution clients did not hurt the global model, it rather improved the global model performance. Having said that, Out-of-Distribution (OOD) client, e.g., client 8, has shown to harm the model's performance. If there is a strong inductive bias from a couple of OOD clients, this potentially might hurt the global model. One might need to consider a smarter way of aggregation for such OOD clients. However, this is out of the scope of this manuscript.

FedPerl & unlabeled clients (Scenario #2). In a more challenging experiment, which is unique in federated learning, we trained our models utilizing labeled global data and unlabeled local ones. `FedPerl` also shows the best results comparing to the baselines thanks to our peer learning strategy, which enforces additional knowledge to the clients besides the global one exploited via federated learning. Further, applying PA produced more stable results and higher accuracy.

FedPerl & unseen clients (Scenario #3). In another part of our experiments, we have tested our method on the ISIC20 dataset. The low performance for all models can be attributed to two things; i) Class Mismatch: the models were trained on 8 classes while ISIC20 contains only two classes with severe class imbalance (500 malignant vs. 32.5K benign), and ii) Domain Shift: none of the models proposed to address the domain shift problem between ISIC19 and ISIC20. In this experiment, we tried to show how SSFL models perform in such a challenging situation. The results showed that our model is still better than all baseline models in skin cancer classification shedding the light on the generalization capability. This is attributed to the `FedPerl` is learning more powerful and discriminative representations for the minority class by aggregating the peers' knowledge and experiences. While we attributed the better performance of our model than `FedMatch` to the similarity matrix that we utilized such that our method picks more accurate peers to the local model than the `FedMatch` approach. In the current version, `FedPerl` does not have a specific property that handles the class imbalance.

Learning from few labeled clients (Scenario #4). The comparison with a SOTA method reveals that our method is on par with `FedIRM` when part of the unlabeled clients participate in the training. Yet, that is not the case when all clients are involved, which is a rare setup. We attributed that to the quality of the pseudo labels generated with the help of unlabeled peers to ones generated with the help of labeled clients. Nevertheless, combining both approaches in a joint or a co-training setup could be an interesting research direction and might lead to better performance.

Learning Policy. Our first strategy depends on a static peer learning policy that involves best T peers based on their similarities. While this policy is effective in the communities and clients, it suffers in performance when countered by an ODD client. To resolve this issue, we proposed, in this paper, more dynamic and adaptive policies. Specifically, the successful policies employed a gateway to control the learning rate from peers. The participation is measured based on either a global dataset or how similar the peer is to the client. Only the clients who pass a predefined threshold can participate in the training. We found that the results of the two policies are somehow similar with advantages to the one based on similarity. Yet, and most importantly, the performance of the OOD client gets boosted from both policies. Because we do not have control or can not anticipate the in-distribution from out-of-distribution clients, the selection between the static and dynamic methods goes toward the dynamic ones. Even if we know the clients, the results show that the dynamic policy betters the static one. On the other hand, our preference between the validation or similarity gated policies goes toward the similarity. In most cases, the global validation data is not available, which prevents us from applying the validation policy. Further, the gated similarity policy produces more consistent and stable results. Though, the trade-off between the global and local benefits could be the decision-maker in real-life scenarios. Our dynamic learning policies are considered heuristic ones, however, they were proposed to address a problem that we noticed in our previous work Bdair et al. (2021), where the performance of some individual clients has not improved by the federated learning. We could achieve that by utilizing the global validation dataset or the client similarities. However, to provide a comprehensive study addressing any potential questions from the reader, we tested three different policies. Note that the three policies are separate and work independently. Besides, they have shown to be effective (*cf.* Table 10).

5. Conclusion

In this paper, we propose a semi-supervised federated learning framework for skin lesion classification in dermoscopic images. Our method; `FedPerl` overcomes the limitations of the previous works, inspired by peer learning from educational science and ensemble averaging from committee machines. We show a real-life application of our method that fits the complexity of the medical data *i.e.* data heterogeneity, severe class imbalance, and an abundant amount of unlabeled data. Our database consists of 71,000 skin lesion images obtained from 5 public datasets. The testing environment consists of the standard semi-supervised setting and a more challenging and less investigated scenario where clients have access just to the unlabeled data. Our method is on par with the state-of-the-art method in skin classification in the standard federated learning, *i.e.* random set of clients participating in the training and outperforms the baselines and other `SSFL` by 15.8%, and 1.8%, respectively. Moreover, `FedPerl` demonstrates less sensitivity to noisy clients and has better generalization ability to unseen data. Besides, we propose the peer anonymization (PA) technique. PA is a simple and efficient approach to create an anonymized peer and hide clients' identities. PA enhances performance while reduces communication costs. We show that our method is orthogonal and easy to implement to other methods without additional complexity. In this paper, we investigate two learning policies; a fixed policy that selects the top similar peers, and a dynamic and more adaptive one that controls the learning stream on the clients. We have shown that both strategies are effective with advantages to the dynamic one. Thus far, we exploited the model parameters as similarity measurement, while we could employ different techniques to profile the clients. Further, we could investigate the privacy guarantee for aggregated models as future work.

Acknowledgments

T.B. is financially supported by the German Academic Exchange Service (DAAD).

Ethical Standards

The work follows appropriate ethical standards in conducting research and writing the manuscript. This work presents computational models trained with publicly available data, for which no ethical approval was required.

Conflicts of Interest

The authors declare no conflicts of interest.

References

Shadi Albarqouni, Spyridon Bakas, Konstantinos Kamnitsas, M Jorge Cardoso, Bennett Landman, Wenqi Li, Fausto Milletari, Nicola Rieke, Holger Roth, Daguang Xu, et al. *Domain Adaptation and Representation Transfer, and Distributed and Collaborative Learning: Second MICCAI Workshop, DART 2020, and First MICCAI Workshop, DCL 2020, Held in Conjunction with MICCAI 2020, Lima, Peru, October 4-8, 2020, Proceedings*, volume 12444. Springer Nature, 2020.

Benjamin Aubin, Antoine Maillard, Jean Barbier, Florent Krzakala, Nicolas Macris, and Lenka Zdeborová. The committee machine: Computational to statistical gaps in learning a two-layers neural network. *Journal of Statistical Mechanics: Theory and Experiment*, 2019(12):124023, 2019.

Tariq Bdair, Nassir Navab, and Shadi Albarqouni. Fedperl: Semi-supervised peer learning for skin lesion classification. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 336–346. Springer, 2021.

M Binder, H Kittler, A Seeber, A Steiner, H Pehamberger, and K Wolff. Epiluminescence microscopy-based classification of pigmented skin lesions using computerized image analysis and an artificial neural network. *Melanoma research*, 8(3):261–266, 1998.

Christopher Briggs, Zhong Fan, and Peter Andras. Federated learning with hierarchical clustering of local updates to improve training on non-iid data. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pages 1–9. IEEE, 2020.

Wallace H Clark Jr, David E Elder, DuPont Guerry IV, Leonard E Braitman, Bruce J Trock, Delray Schultz, Marie Synnestvedt, and Allan C Halpern. Model predicting survival in stage i melanoma based on tumor progression. *JNCI: Journal of the National Cancer Institute*, 81(24):1893–1904, 1989.

Noel Codella, Veronica Rotemberg, Philipp Tschandl, M Emre Celebi, Stephen Dusza, David Gutman, Brian Helba, Aadi Kalloo, Konstantinos Liopyris, Michael Marchetti, et al. Skin lesion analysis toward melanoma detection 2018: A challenge hosted by the international skin imaging collaboration (isic). *arXiv preprint arXiv:1902.03368*, 2019.

Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical automated data augmentation with a reduced search space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 702–703, 2020.

Yukun Ding, Jinglan Liu, Jinjun Xiong, and Yiyu Shi. Revisiting the evaluation of uncertainty estimation and its application to explore model complexity-uncertainty trade-off. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 4–5, 2020.

Andre Esteva, Brett Kuprel, Roberto A Novoa, Justin Ko, Susan M Swetter, Helen M Blau, and Sebastian Thrun. Dermatologist-level classification of skin cancer with deep neural networks. *nature*, 542(7639):115–118, 2017.

Mona Flores, Ittai Dayan, Holger Roth, Aoxiao Zhong, Ahmed Harouni, Amilcare Gentili, Anas Abidin, Andrew Liu, Anthony Costa, Bradford Wood, et al. Federated learning used for predicting outcomes in sars-cov-2 patients. 2021.

Matt Fredrikson, Somesh Jha, and Thomas Ristenpart. Model inversion attacks that exploit confidence information and basic countermeasures. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, pages 1322–1333, 2015.

Yonatan Geifman and Ran El-Yaniv. Selective classification for deep neural networks. *arXiv preprint arXiv:1705.08500*, 2017.

Nils Gessert, Maximilian Nielsen, Mohsin Shaikh, René Werner, and Alexander Schlaefer. Skin lesion classification using ensembles of multi-resolution efficientnets with meta data. *MethodsX*, 7:100864, 2020.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *International Conference on Machine Learning*, pages 1321–1330. PMLR, 2017.

Tzu-Ming Harry Hsu, Hang Qi, and Matthew Brown. Federated visual classification with real-world data distribution. *arXiv preprint arXiv:2003.08082*, 2020.

Wonyong Jeong, Jaehong Yoon, Eunho Yang, and Sung Ju Hwang. Federated semi-supervised learning with inter-client consistency and disjoint learning. *https://openreview.net/forum?id=ce6CFXBh30h*, 2021.

Dovydas Joksas, P Freitas, Z Chai, WH Ng, M Buckwell, C Li, WD Zhang, Q Xia, AJ Kenyon, and A Mehonic. Committee machines—a universal method to deal with non-idealities in memristor-based neural networks. *Nature communications*, 11(1):1–10, 2020.

Georgios A Kaissis, Marcus R Makowski, Daniel Rückert, and Rickmer F Braren. Secure, privacy-preserving and federated machine learning in medical imaging. *Nature Machine Intelligence*, 2(6):305–311, 2020.

Jeremy Kawahara, Sara Daneshvar, Giuseppe Argenziano, and Ghassan Hamarneh. Seven-point checklist and skin lesion classification using multitask multimodal neural nets. *IEEE Journal of Biomedical and Health Informatics*, 23(2):538–546, mar 2019. ISSN 2168-2194.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. In *International Conference on Machine Learning*, pages 3519–3529. PMLR, 2019.

Daiqing Li, Amlan Kar, Nishant Ravikumar, Alejandro F Frangi, and Sanja Fidler. Federated simulation for medical imaging. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 159–168. Springer, 2020a.

Haoliang Li, YuFei Wang, Renjie Wan, Shiqi Wang, Tie-Qiang Li, and Alex C Kot. Domain generalization for medical imaging classification with linear-dependency regularization. *arXiv preprint arXiv:2009.12829*, 2020b.

Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. *arXiv preprint arXiv:1812.06127*, 2018.

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

Quande Liu, Hongzheng Yang, Qi Dou, and Pheng-Ann Heng. Federated semi-supervised medical image classification via inter-client relation matching. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 325–335. Springer, 2021.

Adria Romero Lopez, Xavier Giro-i Nieto, Jack Burdick, and Oge Marques. Skin lesion classification from dermoscopic images using deep learning techniques. In *2017 13th IASTED international conference on biomedical engineering (BioMed)*, pages 49–54. IEEE, 2017.

Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial Intelligence and Statistics*, pages 1273–1282. PMLR, 2017.

Tribhuvanesh Orekondy, Seong Joon Oh, Yang Zhang, Bernt Schiele, and Mario Fritz. Gradient-leaks: Understanding and controlling deanonymization in federated learning. *arXiv preprint arXiv:1805.05838*, 2018.

Andre GC Pacheco, Gustavo R Lima, Amanda S Salomão, Breno A Krohling, Igor P Biral, Gabriel G de Angelo, Fábio CR Alves Jr, José GM Esgario, Alana C Simora, Pedro BC Castro, et al. Pad-ufes-20: a skin lesion benchmark composed of patient data and clinical images collected from smartphones. *arXiv preprint arXiv:2007.00478*, 2020.

Nicola Rieke, Jonny Hancox, Wenqi Li, Fausto Milletari, Holger R Roth, Shadi Albarqouni, Spyridon Bakas, Mathieu N Galtier, Bennett A Landman, Klaus Maier-Hein, et al. The future of digital health with federated learning. *NPJ digital medicine*, 3(1):1–7, 2020.

Howard W Rogers, Martin A Weinstock, Ashlynne R Harris, Michael R Hinckley, Steven R Feldman, Alan B Fleischer, and Brett M Coldiron. Incidence estimate of nonmelanoma skin cancer in the united states, 2006. *Archives of dermatology*, 146(3):283–287, 2010.

Veronica Rotemberg, Nicholas Kurtansky, Brigid Betz-Stablein, Liam Caffery, Emmanouil Chousakos, Noel Codella, Marc Combalia, Stephen Dusza, Pascale Guitera, David Gutman, et al. A patient-centric dataset of images and metadata for identifying melanomas using clinical context. *Scientific data*, 8(1):1–8, 2021.

Holger R Roth, Ken Chang, Praveer Singh, Nir Neumark, Wenqi Li, Vikash Gupta, Sharut Gupta, Liangqiong Qu, Alvin Ihsani, Bernardo C Bizzo, et al. Federated learning for breast density classification: A real-world implementation. In *Domain Adaptation and Representation Transfer, and Distributed and Collaborative Learning*, pages 181–191. Springer, 2020.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015.

Karthik V Sarma, Stephanie Harmon, Thomas Sanford, Holger R Roth, Ziyue Xu, Jesse Tetreault, Daguang Xu, Mona G Flores, Alex G Raman, Rushikesh Kulkarni, et al. Federated learning improves site performance in multicenter deep learning without data sharing. *Journal of the American Medical Informatics Association*, 2021.

T Schindewolf, Wilhelm Stolz, Rene Albert, Wolfgang Abmayr, and Harry Harms. Classification of melanocytic lesions with color and texture analysis using digital image processing. *Analytical and Quantitative Cytology and Histology*, 15(1):1–11, 1993.

Rebecca L. Siegel. Cancer statistics, 2021. published early online january 12, 2021 in ca cancer journal for clinicians. mph, american cancer society, atlanta, ga., 2021.

Kihyuk Sohn, David Berthelot, Chun-Liang Li, Zizhao Zhang, Nicholas Carlini, Ekin D Cubuk, Alex Kurakin, Han Zhang, and Colin Raffel. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *arXiv preprint arXiv:2001.07685*, 2020.

Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International Conference on Machine Learning*, pages 6105–6114. PMLR, 2019.

Keith J Topping. Trends in peer learning. *Educational psychology*, 25(6):631–645, 2005.

Volker Tresp. Committee machines. *Handbook for neural network signal processing*, pages 1–18, 2001.

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.

Philipp Tschandl, Noel Codella, Bengü Nisa Akay, Giuseppe Argenziano, Ralph P Braun, Horacio Cabo, David Gutman, Allan Halpern, Brian Helba, Rainer Hofmann-Wellenhof, et al. Comparison of the accuracy of human readers versus machine-learning algorithms for pigmented skin lesion classification: an open, web-based, international, diagnostic study. *The Lancet Oncology*, 20(7):938–947, 2019.

Dong Yang, Ziyue Xu, Wenqi Li, Andriy Myronenko, Holger R Roth, Stephanie Harmon, Sheng Xu, Baris Turkbey, Evrim Turkbey, Xiaosong Wang, et al. Federated semi-supervised learning for covid region segmentation in chest ct using multi-national data from china, italy, japan. *Medical Image Analysis*, page 101992, 2021.

Jianpeng Zhang, Yutong Xie, Yong Xia, and Chunhua Shen. Attention residual learning for skin lesion classification. *IEEE transactions on medical imaging*, 38(9):2092–2103, 2019.

Lin Zhang, Yong Luo, Yan Bai, Bo Du, and Ling-Yu Duan. Federated learning for non-iid data via unified feature learning and optimization objective alignment. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4420–4428, 2021.

Yue Zhao, Meng Li, Liangzhen Lai, Naveen Suda, Damon Civin, and Vikas Chandra. Federated learning with non-iid data. *arXiv preprint arXiv:1806.00582*, 2018.

Wentao Zhu, Maximilian Baust, Yan Cheng, Sébastien Ourselin, M Jorge Cardoso, and Andrew Feng. Privacy-preserving federated brain tumour segmentation. In *Machine Learning in Medical Imaging: 10th International Workshop, MLMI 2019, Held in Conjunction with MICCAI 2019, Shenzhen, China, October 13, 2019, Proceedings*, volume 11861, page 133. Springer Nature, 2019.

Remainder omitted in this sample.

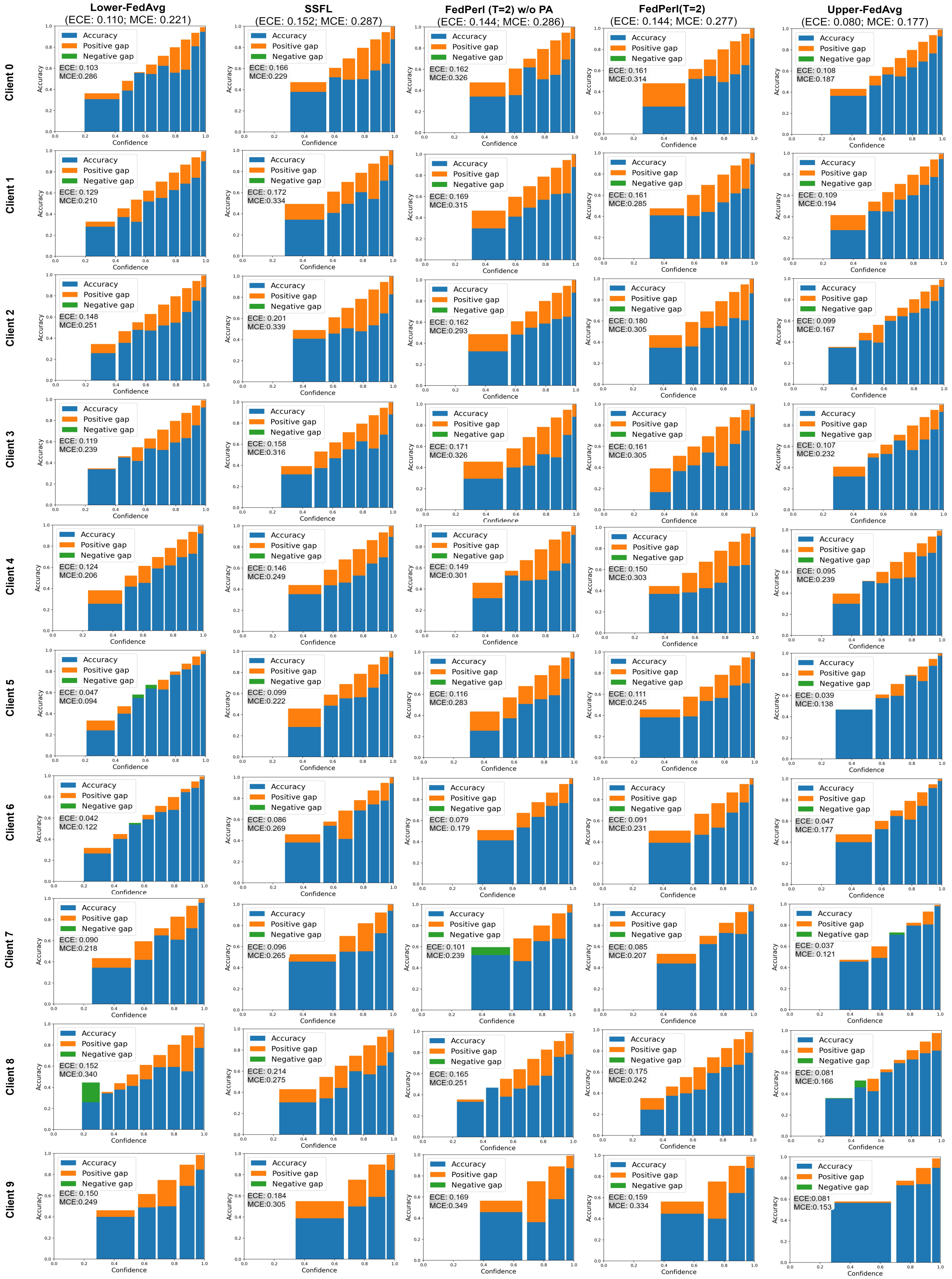

Figure 9: Reliability diagrams and calibration errors. FedPerl is more calibrated than SSFL and local upper models indicating better and more confident predictions. The local models are shown in the supplementary materials.