DECENTRALIZED FEDERATED LEARNING VIA OVERLAPPING DATA AUGMENTATION

Anonymous authors
Paper under double-blind review

ABSTRACT

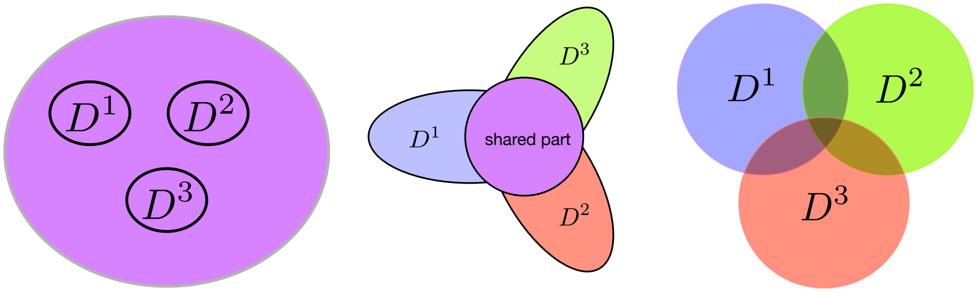
Recently, there have been rising concerns about the heterogeneity among local clients in federated learning, which could lead to inefficient utilization of the data from other clients. To mitigate the adverse effects of heterogeneity, FL research has mostly focused on learning a globally shared initialization under the assumption that the shared information is consistent among all clients. In this paper, we consider a more general scenario, Selective Partial Sharing (SPS), where each pair of clients may share different patterns or distribution components. We propose a novel FL framework named `Fed-SPS` to exploit the shared knowledge by a partial and pairwise collaboration. Meanwhile, to reduce data traffic and improve computing efficiency, we realize a decentralized learning paradigm for our framework. Due to privacy concerns, one cannot obtain the overlapped distribution components with direct access to the raw data. While the learned personalized model is an approximation of local distribution, we propose to identify the selective sharing structure by exploring the vulnerability overlap between local models. With the detected sharing structure, we propose an overlapping data augmentation, which efficiently boosts the leveraging of the overlapped data between clients. Comprehensive experiments on a suite of benchmark data sets and a real-world clinical data set show that our approach can achieve better generalization compared with existing methods.

1 INTRODUCTION

Federated learning (FL) is an effective paradigm that enables the decentralization of learning from fragmented data without sacrificing privacy (McMahan et al., 2017). Recently, FL has gained growing interest for its capability to facilitate real-world applications, including recommendation (Muhammad et al., 2020), finance (Yang et al., 2019) and healthcare (Xu et al., 2021), etc. Classical FL, usually denoted as FedAvg (McMahan et al., 2017) is designed to learn an aggregated global model using i.i.d. data from local clients. However, non-i.i.d. data typically emerge in federated learning scenarios because data could be gathered from a heterogeneous group of users in reality. A global model learned by averaging local gradients can suffer severe performance degradation when the local distributions drift dramatically (Deng et al., 2020a; Cui et al., 2021). This limits the deployment of FL techniques and stimulates lots of research on addressing statistical heterogeneity.

A variety of efforts have been made to tackle this challenge with a prior hypothesis that the shared information is consistent across all clients as shown in Figure 1(b). A recent line of research proposes to achieve a trade-off between local and global training by regulating the deviation of local models from a global model (Li et al., 2020a; Dinh et al., 2020; Karimireddy et al., 2020). These approaches may not fully leverage the knowledge from other clients, whose diversity suggests informative structural differences in their local data (Zhu et al., 2021). Other research aims to facilitate local training by learning a common representation (Liang et al., 2020; Collins et al., 2021). In this method, it is usually assumed that all clients share a common representation and the aim of federated training is to approach such a global representation.

However, the aforementioned hypothesis may not always be true in reality. One example is the clinical research network (CRN) involving multiple hospitals (Fleurence et al., 2014), where each hospital has its own patient population and protected patient data. Assuming the private data from all clients share a common representation, if we learn a risk prediction model within the CRN by taking



(a) i.i.d. data (b) consistently partial sharing (c) selective partial sharing

Figure 1: Illustrations of the three assumptions of information sharing (a) i.i.d. data; all clients share the same implying distribution. (b) consistent partial sharing; there is a global shared part across all clients. (c) selective partial sharing; the overlaps between each pair of clients are not consistent.

advantage of the common representation, the expectation from each hospital is that a better model can be obtained in the CRN compared to one learned from its own data. In this scenario, there has been a prior study showing that the model performance can even decrease due to negative transfer (Pan & Yang, 2009) induced by severe distribution discrepancies (Deng et al., 2020a).

With these considerations, we study a more general scenario where we do not presume a global shared pattern but inconsistently shared overlaps among local clients. As shown in Figure 1(c), different clients may share different overlaps with others and the overlaps cannot be identified by accessing the raw data directly. These issues raise a noticeable question: *How can we fully exploit the distribution overlaps to improve the local model performance in a federated network?*

To address this question, we provide a novel framework `Fed-SPS`, which aims to learn a personalized model for each client through a pairwise collaboration. Specifically, we argue that local limited data may not fully reflect the underlying true local distribution and the key challenge is the leveraging of overlaps between clients. While it is not allowed to have direct access to the raw data, we propose to determine the selective sharing structure by detecting the vulnerability overlap of the learned local models. To maximize the benefit from other clients, we propose to learn models with overlapping data augmentation of each pair of clients for efficient utilization of the informative overlaps.

It is worthwhile to summarize our key contributions as follows:

1. we question the practicality of common information sharing by suggesting a more general scenario that there are distributional overlaps among local clients. Meanwhile, the shared overlaps are inconsistent across all clients;

2. we explore a new direction that the distribution shift between local models could be mitigated by strengthening the shared components. We propose to fully take advantage of the informative overlaps by overlapping data augmentation without privacy compromise;

3. we realize a pairwise collaboration by proposing a decentralized framework `Fed-SPS`. Experiments on benchmark data sets and a real-world clinical data set show that our framework achieves a better generalization performance.

2 Related Work

2.1 Federated Learning with Statistical Heterogeneity

Federated learning (McMahan et al., 2017) has gained significant attention in machine learning community because of its extensive practical values. In the meantime, federated learning has also raised several concerns, including communication efficiency (Konečnỳ et al., 2016), privacy (Agarwal et al., 2018), and statistical heterogeneity (Karimireddy et al., 2020; Cui et al., 2022), and they have been the topic of multiple research efforts (Mohri et al., 2019). Recently, to handle statistical heterogeneity, there are researchers focusing on learning distributionally robust models. For example, Mohri et al. (2019) pursue a consistent performance in federated learning, and propose to optimize a global model for any possible mixed distribution. Deng et al. (2020b) inherit this idea and introduce a more communication-efficient framework. Reisizadeh et al. (2020) put forward a Gradient Descent Ascent (GDA) method to defend against affine distribution shifts. Instead of focusing on defending against a specific shift as in previous research, which may harm the generalization during learning, we show that distributional robustness could benefit the generalization, because boosting the robustness on the overlapped distribution encourages a more precise collaboration during model training.

2.2 Personalized Federated Learning

A global model (e.g., FedAvg) could harm certain clients when there are severe distribution discrepancies (Deng et al., 2020a), and this stimulates the study of personalized federated learning. A wealth of work focused on a better balance between global and local training. For example, there are research (Dinh et al., 2020; Li et al., 2020a; Karimireddy et al., 2020) proposing to stabilize local training by regulating the deviation from the global model over the parameter space. A few works (Wang et al., 2019; Yu et al., 2020) point out that fine-tuning the learned global model could benefit local adaptation. Some studies (Khodak et al., 2019; Fallah et al., 2020) focus on obtaining a personalized variant from an initial shared model across all clients in a meta-learning manner. In these methods, when learning for a local client, the global model could lose some applicable message implied in other clients.

Another line of research aims to learn a global feature extractor with specific classifiers for all clients (Collins et al., 2021; Liang et al., 2020). For example, Collins et al. (2021) propose to use a multi-head network to model a common feature embedding and provide a theoretical justification for the algorithm convergence in a linear setting. However, we are afraid that the shared representation may not always be consistent across all clients. A shared overlap between two clients may not overlap with others. Hence, we suggest a refined leveraging of the informative overlaps for all clients during learning.

3 Problem Definition

3.1 Notations

Suppose there are K local clients in a federated network. Each client is associated with a specific data distribution $D^k := \left\{(X^k, Y^k)\right\}$, $k \in \{1, 2, ..., K\}$, where the input space X^k and the output space Y^k are shared across all K clients. In the following, we will also use D^k to denote the k-th client without causing further confusion.

Note that we use $\hat{D}^k$ to denote the data points sampled from D^k, which is the private dataset with n^k samples in the k-th client $\hat{D}^k := \left\{(x_p^k, y_p^k)\right\}_{p=1}^{n^k}$. Each client tries to learn a model with others to predict the label $\hat{y}$ by maximizing its utility (i.e., the prediction accuracy). The classical federated learning algorithm assumes that the data of all clients are i.i.d. as shown in Figure 1(a). It learns a global model h for all clients by minimizing the empirical risk over the samples from all clients,

$$\min_{h \in \mathcal{H}} \frac{1}{\sum_{k=1}^{K} n^k} \sum_{k=1}^{K} \sum_{p=1}^{n^k} l(h(x_p^k), y_p^k), \tag{1}$$

where $\mathcal{H}$ denotes the model space and l is the loss function.

3.2 Statistical Heterogeneity among All Clients

In reality, the underlying data distribution D^k of local clients is mostly agnostic and may be substantially different from each other.

Consistently Partial Sharing. A straightforward assumption is that there is a common shared part across all clients as shown in Figure 1(b). Under this assumption, the goal of federated learning is that local clients collaboratively learn such a global initialization/representation.

Selective Partial Sharing. In contrast to global sharing, we consider a scenario where the shared parts are not consistent among all clients as illustrated in Figure 1(c). For example, suppose the distributions D^i and D^j have a common component $P_{share}(D^i, D^j)$, $P_{share}(D^i, D^j)$ may not overlap to other clients. For the example in Figure 2(b), ① $= P_{share}(D^1, D^T)$ is not overlapped with other clients. To collaborate to improve the utility of the learned models, D^1 and D^T expect to make full use of the data sampled from ① in each other. Due to data confidentiality, it is not allowed to determine the shared parts between clients by accessing the local data directly.

In this paper, we consider the scenario of selective partial sharing. Due to data confidentiality, it is not allowed to determine the shared parts between clients by accessing the local data directly. Our goal is to exploit the sharing structure among clients to learn a personalized model h^k for each client D^k in a privacy-preserving manner.

4 Methodology

In this section, we recall the concept of Wasserstein distance and distributional robustness. Then, we will introduce the significance of data augmentation in a single client. For the utilization of the data from other clients, we propose to identify the shared distributions by exploring the vulnerability overlap of the learned models between clients. The identified vulnerability overlap encourages an overlapping data augmentation. We enhance the model utility by learning from the data augmented selectively and give a decentralized learning framework for the realization.

4.1 Preliminaries: Wasserstein Distance and Distributional Robustness

Wasserstein Distance. Suppose $Z \triangleq X \times Y$ is the sample space where X and Y are the spaces of input and output. For simplicity, we assume that the learning task is a classification task and Y is discrete. Before introducing the Wasserstein distance, we firstly define the distance between two samples $z_p := (x_p, y_p)$ on sample space Z, i.e.,

$$d_z(z_p, z_q) \triangleq d_x\left(x_p, x_q\right) + \mathbb{1}_{y_p \neq y_q}, \tag{2}$$

where d_x denotes the distance between the two inputs (e.g., Euclidean distance) and d_z measures the dissimilarity of two samples. Moreover, d_z denotes the transport cost function on the sample space Z. Wasserstein distance measures the dissimilarity of two distributions D^i and D^j,

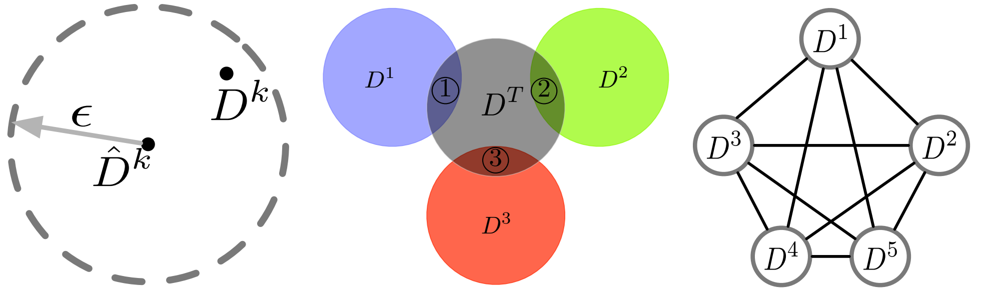
$$W(D^i, D^j) = \inf_{\Pi \in \mathcal{C}(D^i, D^j)} \int_{Z \times Z} d_z\left(z_p, z_q\right) d\Pi\left(z_p, z_q\right), \tag{3}$$

where $\mathcal{C}(D^i, D^j)$ is the set of couplings between D^i and D^j.

Distributional Robustness. Most real-world machine learning problems have uncertain factors in the data, which may be due to limited observability, noisy measurements, and implementation errors (Rahimian & Mehrotra, 2019). For a learned model h, to deal with the distribution shift between the true distribution D^k and the empirical $\hat{D}^k$, distributionally robust risk is introduced to measure the maximal loss of h over the data distributions around $\hat{D}^k$:

$$\widetilde{D}^j(\epsilon, h) = \arg \max_{D: W(D, \hat{D}^k) \leq \epsilon} \mathbb{E}_{z \in D}\, \ell(z, h), \tag{4}$$

where $\widetilde{D}^j(\epsilon, h)$ is called adversarial distribution and ϵ is a pre-defined tolerance parameter, which controls the scale of the uncertainty about the data distribution.



(a) distribution misspecification (b) overlapping distribution (c) decentralized collaboration

Figure 2: (a) the true distribution D^k may be around the empirical distribution $\hat{D}^k$ because of the possible shift. (b) Illustration of the overlapping distributions between the target client and other clients. D^T denotes the target distribution. ⓚ is the overlapping component of D^T and D^k. (c) Illustration of our decentralized collaboration; there is no main server and all local clients collaborate with others in a pairwise manner.

4.2 Data Augmentation on Local Clients

In federated learning, the true distribution D^k on each client is usually agnostic. Local training for each client is hard to obtain a satisfactory model, as insufficient data with potential shift can cause a significant difference between the true distribution D^k and the empirical data $\hat{D}^k$.

Data augmentation may be an effective method for addressing the problem above. Suppose the true distribution D^k is around the empirical data $\hat{D}^k$ as shown in Figure 2(a), one alternative is enhancing the distributional robustness by augmenting data. In particular, we minimize a weighted sum of the empirical risk and the distributionally robust risk as follows:

$$\min_h \mathbb{E}_{z\in\hat{D}^k}\,\ell(z,h) + \alpha\cdot \sup_{D:W(D,\hat{D}^k)\le\epsilon} \mathbb{E}_{z\in D}\ell(z,h), \tag{5}$$

where $\alpha > 0$ is the pre-defined regularization parameter, which controls the trade-off between the empirical risk and the robust risk. In the following, we will give an upper bound of the objective of Eq.(5) compared with the single empirical risk. While the distributionally robust risk for a model h could be infinite, if h satisfies Lipschitz continuity, we have the following proposition.

Proposition 1. *Suppose h is L-Lipschitz, i.e., $|h(z_p) - h(z_q)| \le L\cdot d_z(z_p,z_q)$, $\forall z_p, z_q \in Z$, then the objective in Eq.(5) satisfies:*

$$\begin{aligned}\mathbb{E}_{z\in\hat{D}^k}\,\ell(z,h) + \alpha\cdot \sup_{D:W(D,\hat{D}^k)\le\epsilon} \mathbb{E}_{z\in D}\ell(z,h) &\le (1+\alpha)\cdot \sup_{D:W(D,\hat{D}^k)\le\epsilon} \mathbb{E}_{z\in D}\ell(z,h)\\ &\le (1+\alpha)\cdot(\mathbb{E}_{z\in\hat{D}^k}\,\ell(z,h) + 2L\epsilon).\end{aligned} \tag{6}$$

Blanchet & Murthy (2019) prove that Eq.(5) could be solved by appealing to duality and the second term in Eq.(5) satisfies:

$$\sup_{D:W\left(D,\hat{D}^k\right)\le\epsilon} \mathbb{E}_{z\in D}[\ell(z,h)] = \inf_{\lambda\ge 0}\left\{\lambda\epsilon + \mathbb{E}_{z_p\in\hat{D}^k}\left[\ell_\lambda^{d_z}(z_p,h)\right]\right\} \tag{7}$$

$$\ell_\lambda^{d_z}(z_p,h) \triangleq \sup_{z\in Z}\ell(z,h) - \lambda d_z(z,z_p). \tag{8}$$

From the above derivation, the original infinite-dimensional optimization problem needs first maximizing the objective $\ell(z,h) - \lambda d_z(z,z_p)$ in the sample space Z shown in Eq.(8).

Since the empirical distribution $\hat{D}^k$ is discrete, we build a projection T for each sample $z_p \in \hat{D}^k$ to achieve a local data augmentation:

$$\widetilde{D}^j(\epsilon,h) = \{T_\lambda(z_p,h^j)|z_p\in\hat{D}^j\},\; T_\lambda(z_p,h) = \arg\max_{z\in Z}\,\ell(z,h) - \lambda\cdot d_z(z,z_p). \tag{9}$$

Then we minimize the objective $\left\{\lambda\epsilon + \mathbb{E}_{z\in\hat{D}^k}\left[\ell_\lambda^{d_z}(z,h)\right]\right\}$ using an optimal $\lambda \ge 0$ shown in Eq.(7).

Augmenting data for local clients. From Eq.(9), the projection T augments adversarial data which could fool the learned model. While learning from limited local data could cause overfitting, data augmentation by T mitigates this issue by encouraging a more smooth model. A detailed data augmentation realization about optimizing T and λ are summarized in Algorithm 1 in Appendix.

4.3 Overlapping Data Augmentation between Clients

To improve the model performance, a better choice is fully using the shared knowledge while isolating the excluded components to avoid the possible negative transfer. For example, suppose we use D^T to denote the target client, (since we learn a personalized model for each client, all clients will be target clients during learning), one needs to learn from the data in ①, ② and ③ shown in Figure 2(b), as they share the same distributional components with D^T. Meanwhile, this avoids potential negative transfer by neglecting the data not in the overlaps.

However, due to data confidentiality and privacy concerns, it's hard to determine where the two clients overlap directly. To motivate our intuition, we propose to capture the shared parts of the learned model, which is an approximation of the local distribution. In particular, we propose the concept *overlapping data augmentation*, and the formal definition is as follows:

Definition 1 (Overlapping Data Augmentation (ODA))**.** *Suppose h^i and h^j denote the learned models for the i-th and the j-th local clients, overlapping data augmentation (ODA) is the data distribution that satisfies:*

$$\mathrm{ODA}((h^i,D^i),(h^j,D^j)) = \arg\max_{D:W(D,D^i)\le\epsilon\text{ and }W(D,D^j)\le\epsilon} \mathbb{E}_{z\in D}(\ell(z,h^i)+\ell(z,h^j)). \tag{10}$$

Augmenting overlapping data between clients. From Definition 1, ODA refers to the distribution around $\hat{D}^i$ and $\hat{D}^j$ that could worsen the performance of the two models simultaneously. Considering that ODA associates with a common decision boundary of h^i and h^j, it captures the transferable and shared distributions of the learned models. Therefore, we propose to augment the overlapping data from ODA to boost the model learning.

4.4 An Efficient Realization of Fed-SPS

We propose to learn from the augmented data of all clients from the two perspectives. For the augmented local data shown in Sec. 4.2, our framework advances the generalization by learning a smooth model to alleviate overfitting; for the data augmented from other clients in Sec. 4.3, our framework achieves a selective and efficient utilization of shared components.

Combining the Eq.(5) and the Definition 1, we firstly provide a straightforward way that incorporates the ODA into the training objective. The formulation is as follows:

$$\min_{h^i} \mathbb{E}_{z\in\hat{D}^i}\ \ell(z,h) + \alpha\ \mathbb{E}_{z\in\widetilde{D}(\epsilon,h^i,\hat{D}^i)}\ell(z,h^i) + \beta \sum_{j=1,j\neq i}^{K} \mathbb{E}_{z\in\mathrm{ODA}((h^i,D^i),(h^j,D^j))}\ell(z,h^i), \tag{11}$$

where α and β are pre-defined parameters.

Improving optimization efficiency. For the third term in Eq.(11), in every iteration, each client h^i needs to determine ODA between itself and every other client, which has the time complexity $O(K^2)$ with a significant communication cost. Moreover, it's hard to obtain a precise adversarial distribution around two distributions simultaneously. To reduce the computation cost, we propose to approximate the ODA by reusing the obtained adversarial distribution $\widetilde{D}^j$ in Eq.(9):

$$\widehat{\mathrm{ODA}}((h^i,D^i),(h^j,D^j)) = \{T_\lambda(z_p,h^j)|\ell(T_\lambda(z_p,h^j),h^i) - \ell(z_p,h^i) > 0, z_p \in \hat{D}^j\}. \tag{12}$$

where $T_\lambda(z_p,h^j)$ is the data generated by the projection T from Eq.(9).

Since the sample $T_\lambda(z_p,h^j)$ in $\left\{T_\lambda(z_p,h^j)|z_p\in\hat{D}^j\right\}$ could fool the model h^j, we identify the subgroup that increases the loss of h^i. This means that the identified subgroup in Eq.(12) fools the model h^i and h^j simultaneously. So $\widehat{\mathrm{ODA}}((h^i,D^i),(h^j,D^j))$ is an approximation of ODA defined in Eq.(10).

From Eq.(12), it is worth noting that the approximated $\widehat{\mathrm{ODA}}((h^i,D^i),(h^j,D^j))$ is different from $\widehat{\mathrm{ODA}}((h^j,D^j),(h^i,D^i))$. We revise the training objective as follows:

$$\min_{h^i} \mathbb{E}_{z\in\hat{D}^i}\ \ell(z,h) + \alpha\ \mathbb{E}_{z\in\widetilde{D}(\epsilon,h^i,\hat{D}^i)}\ell(z,h^i) + \beta \sum_{j=1,j\neq i}^{K} \mathbb{E}_{z\in\widehat{\mathrm{ODA}}((h^i,D^i),(h^j,D^j))}\ell(z,h^i), \tag{13}$$

Reducing communication overhead. To exploit the inconsistently shared knowledge between clients, we develop a decentralized FL framework to realize our analysis shown in Figure 2(c). While a standard decentralized FL could impose great traffic overhead, we propose a "exploration-exploitation" strategy to reduce communication overhead. In particular, in each iteration, we assign a higher sampling probability to the clients which provides more performance gain. In this way, we reduce unnecessary communication cost while maintaining the model performance. Algorithm 2 in Appendix summarizes the whole pipeline of Fed-SPS.

4.5 Theoretical Analysis about the Metric of Statistical Heterogeneity

From the analysis above, we propose to mitigate the distribution discrepancies by augmenting the shared data between clients. However, we usually have no information about the uncertainty in the data. The true shift among clients is mostly agnostic. Therefore, the true sample distance which characterizes the shift may not be Euclidean distance as defined in Eq.(2).

While there may exist deviations when measuring the agnostic shifts, we are interested in a theoretical analysis of the effect of the deviations on the distributionally robust risk. We study the upper bound of the uniform convergence error δ_n formulated as follows:

$$\sup_{h\in\mathcal{H}} \left\{ \left| \sup_{D:W^*(D,D^*)\le\epsilon} \mathbb{E}_{z\in D}[\ell(z,h)] - \sup_{D:W\left(D,\hat{D}\right)\le\epsilon} \mathbb{E}_{z\in D}[\ell(z,h)] \right| \right\}, \tag{14}$$

where D^* and $\hat{D}$ denote the true data distribution and the empirical data set. W^* is the Wasserstein distance with the optimal transportation cost function d_z^*. d_z^* may be different from d_z defined through human intuition. We begin by imposing the following assumptions.

Assumption 1. *the sample space is bounded:* $\sup_{z,z' \in Z} d_z(z, z') < \infty$*;*

Assumption 2. *the model* h *is upper semicontinuous and uniformly bounded* $0 \leq h(z) \leq M < \infty$, $\forall z \in Z$ *and L-Lipschitz with respect to* d_z*:*

$$\sup_{h \in \mathcal{H}} \sup_{z,z' \in Z} |l(z,h) - l(z',h)| \leq L \cdot d_z(z,z'). \tag{15}$$

Assumption 3. *the deviation of the sample distance is uniformly bounded:*

$$\sup_{z,z' \in Z} |d_z^*(z,z') - d_z(z,z')| \leq B. \tag{16}$$

The above Assumption 1 and 2 are standard regularity assumptions (Lee & Raginsky, 2017). From Assumption 3, the metric d_z has a limited deviation from the optimal metric d_z^*.

For the model space $\mathcal{H}$, we use entropy integral (Talagrand, 2014) to measure its complexity:

$$\mathfrak{C}(\mathcal{H}) := \int_0^\infty \sqrt{\log \mathcal{N}_\infty(\mathcal{H}, u)} \mathrm{d}u, \tag{17}$$

where $\mathcal{N}_\infty(\mathcal{H}, u)$ denotes the covering number of $\mathcal{H}$ in the uniform metric.

Theorem 1. *Suppose* $\mathcal{H}$, d_z *satisfy Assumption 1, 2 and 3, for any* $t > 0$*, we have*

$$\delta_n \leq \frac{48\mathfrak{C}(\mathcal{H})}{\sqrt{n}} + \frac{LB}{\sqrt{\epsilon}} + M\left(\frac{\log\frac{2}{t}}{2n}\right)^{\frac{1}{2}}, \tag{18}$$

with probability at least $1 - t$*.*

Theorem 1 cares the uniform convergence of δ_n when the metric of distribution shift d_z is agnostic. From Theorem 1, if d_z has a limited deviation from d_z^*, the convergence of δ_n, which characterizes the maximum of the deviation of the distributionally robust risk defined in Eq.(14), is bounded.

4.6 More Discussion

As our framework achieves a selective data augmentation, it could be used in other SOTA methods in a pluggable way. We summarize the properties of `Fed-SPS` in the following, and more details could be found in Appendix.

Decentralization. Centralized model learning may cause data traffic and the waste of local computing resources. To mitigate these challenges, we realize a pairwise collaboration among clients, which reduces data congestion and makes full use of local computing power.

Optimization. The original objective has a $O(K^2)$ time complexity during each iteration. In our approach, we approximate the ODA using Eq.(12), which reuses generated data $\widetilde{D}(\epsilon, h^j, \hat{D}^j)$ and has a $O(K)$ complexity.

Communication overhead. While decentralized FL methods could bring additional communication compared with centralized FL methods, we propose an efficient sampling scheme that reduces meaningless communication burdens. Experiments in Appendix verify that our method has a comparable time consumption with others.

Privacy. Decentralized FL methods send models among clients to update models, which is more likely to leak privacy than centralized methods, `Fed-SPS` is no exception. Our framework is compatible with the techniques aiming to alleviate this issue, such block-chain (Li et al., 2020b), differential privacy (Chen et al., 2022), etc. More details could be found in Appendix.

5 Experiments

In this section, we validate the effectiveness of `Fed-SPS` from the three aspects: 1) we first demonstrate that our method can learn a more robust feature on synthetic data; 2) we further verify its superiority by conducting experiments on three benchmark image datasets; 3) we show its practicability on a real-world clinical dataset eICU (Pollard et al., 2018) in federated setting. More implementation details can be found in Appendix.[1]

[1] The source codes are made publicly available in `https://github.com/fedips/Fed-SPS`.

5.1 Synthetic Experiment

We start by conducting experiments on *Adult* (Kohavi et al., 1996). Adult contains more than 40000 adult records, including age, gender, income, etc. The task is to predict whether an individual earns more than 50K/year. We follow the setting in (Mohri et al., 2019). In particular, we split the data into two clients with significant statistical heterogeneity. One is PhD client in which all individuals are PhDs and the other is non-PhD client. In this experiment, we use linear models following the setting in (Mohri et al., 2019) and compare the performance with FedAvg, local training and AFL (Mohri et al., 2019). From Table 1, since non-PhD client has plenty of samples($>$ 30000), non-PhD achieves similar performances (82.3) in local training and FedAvg. However, since the two clients have different data distributions (e.g., 23% positive samples in non-PhD client and 74% positive samples in PhD client), weighted averaging the model gradients to obtain a global model cannot satisfy the two clients simultaneously as FedAvg and AFL do. Our method collaboratively learns personalized models by focusing on the overlaps and achieves better performance for both clients.

Table 1: synthetic experiments on Adult

Methods	average(%)	non-PhD(%)	PhD(%)
local training	82.3$_{\pm.0}$	82.5$_{\pm.0}$	66.9 $_{\pm1.0}$
FedAvg	82.3$_{\pm.1}$	82.4$_{\pm.1}$	72.8$_{\pm.3}$
AFL	82.5$_{\pm.5}$	82.6$_{\pm.5}$	73.0$_{\pm2.2}$
Fed-SPS	**82.9**$_{\pm.3}$	**83.0**$_{\pm.3}$	**73.8**$_{\pm.3}$

5.2 Benchmark Experiments

Datasets. We use three real-world image datasets: CIFAR10, CIFAR100 (Krizhevsky et al., 2009) and FEMNIST (Caldas et al., 2018; Cohen et al., 2017) in our experiments. We simulate non-i.i.d. environment for CIFAR10 and CIFAR100 following the work in (McMahan et al., 2017). In particular, we control the heterogeneity by randomly assigning several classes to each client. Note that S denotes the number of classes per client and K denotes the number of clients. For example, 50000 samples in CIFAR10 belong to ten classes. $\{S = 2, K = 100\}$ means that we randomly assign two classes of images for 100 clients, so each client will have $50000/K$ samples. For FEMNIST, we follow the setting in (Collins et al., 2021) and restrict the dataset to 10 handwritten letters. The samples to local clients are assigned according to a log-normal distribution (Li et al., 2019).

Baselines. We compare our method with various recent personalized FL methods. In particular, the baselines are: 1) FedPer (Arivazhagan et al., 2019), which learns a global representation with local heads simultaneously; 2) Fed-MTL (Smith et al., 2017), which models the relationship among local tasks using a regularized method; 3) LG-FedAvg (Liang et al., 2020), which learns a global classifier with multiple feature extractors; 4) FedRep (Collins et al., 2021), which learns a global representation for all clients.

Ablation Studies. Note that local training and FedAvg are two special cases of our method. For the ablation study, we compare our method with the following three methods: 1) local training; 2) FedAvg (McMahan et al., 2017); 3) local + α, local training with distributionally robust training as formulated in Eq.(5) (with a proper α). More discussions about ablation studies could be found in Appendix.

Table 2: CIFAR10 ($S = 2$)

Methods	$K = 100$	$K = 1000$
local training	87.28$_{\pm.6}$	72.17$_{\pm.3}$
local training + α	85.27$_{\pm.2}$	71.26$_{\pm.1}$
FedAvg	44.29$_{\pm.4}$	41.11$_{\pm1.6}$
FedPer	83.52$_{\pm.5}$	73.07$_{\pm.5}$
Fed-MTL	70.44$_{\pm.2}$	56.90$_{\pm.1}$
LG-FedAvg	78.12$_{\pm1.1}$	68.32$_{\pm1.4}$
FedRep	86.08$_{\pm.2}$	76.41$_{\pm.5}$
Fed-SPS	**87.82**$_{\pm.2}$	**77.75**$_{\pm.3}$

Table 3: CIFAR10 ($S = 5$)

Methods	$K = 100$	$K = 1000$
local training	67.76$_{\pm.5}$	47.56$_{\pm.4}$
local training + α	69.54$_{\pm.3}$	48.63$_{\pm.3}$
FedAvg	58.14$_{\pm1.5}$	50.40$_{\pm.9}$
FedPer	**72.26**$_{\pm1.0}$	52.27$_{\pm.8}$
Fed-MTL	54.21$_{\pm.3}$	34.58$_{\pm.5}$
LG-FedAvg	57.83$_{\pm1.2}$	41.05$_{\pm1.3}$
FedRep	72.19$_{\pm.9}$	53.41$_{\pm.7}$
Fed-SPS	71.02$_{\pm.3}$	**54.11**$_{\pm.5}$

We show the accuracy of methods on CIFAR10 in Table 2 and Table 3. Our method achieves comparable or better results on the two settings. From the results in Table 2 and Table 3, since there is a severe distribution discrepancy, FedAvg is hard to learn a satisfying global model compared with other personalized federated methods. When the heterogeneity decreases ($S = 5$), FedAvg achieves better performance (58.14%). Compared with a lower heterogeneity setting ($S = 5$ in Table 3), our method outperforms all baselines when $S = 2$ in Table 2, which illustrates that our method effectively models the overlapped information during training.

To further demonstrate the superiority of our method, we also compare our method with baselines on CIFAR100 dataset, which has 100 classes with 50000 samples. In particular, we use a more complex model following the setting in (Collins et al., 2021). We conduct experiments under the two setting $S = 20, K = 100$ and $S = 20, K = 500$, and the results are shown in Table 4.

Table 4: CIFAR100 ($S = 20$)

Methods	$K = 100$	$K = 500$
local training	32.04$_{\pm .7}$	21.06$_{\pm .6}$
local training + α	34.25$_{\pm .3}$	21.44$_{\pm .1}$
FedAvg	15.74$_{\pm .8}$	19.43$_{\pm .7}$
FedPer	37.41$_{\pm .5}$	21.92$_{\pm .6}$
Fed-MTL	23.38$_{\pm .3}$	10.56$_{\pm .3}$
LG-FedAvg	31.06$_{\pm .5}$	14.83$_{\pm .4}$
FedRep	38.17$_{\pm .8}$	21.88$_{\pm .6}$
`Fed-SPS`	**39.52**$_{\pm .5}$	**24.11**$_{\pm .4}$

Table 5: performance on FEMNIST and on eICU

Methods	FEMNIST (accuracy)	eIUC (AUC)
local training	29.12$_{\pm .9}$	82.93$_{\pm 1.2}$
local training + α	31.26$_{\pm .3}$	83.12$_{\pm .7}$
FedAvg	60.71$_{\pm 1.8}$	84.42$_{\pm .9}$
FedPer	39.21$_{\pm .6}$	83.22$_{\pm .8}$
Fed-MTL	32.34$_{\pm .7}$	77.45$_{\pm .1}$
LG-FedAvg	29.78$_{\pm .5}$	81.30$_{\pm .7}$
FedRep	65.52$_{\pm .8}$	82.26$_{\pm .3}$
`Fed-SPS`	**67.19**$_{\pm .3}$	**86.55**$_{\pm .4}$

From Table 4, `Fed-SPS` surpasses the baselines in the two settings. Compared with the setting $S = 20, K = 100$, the performances of all personalized methods degrade as the number of clients increases. When $K = 500$, all clients own fewer samples and our method achieves a more significant utility gain.

For the experiment on FEMNIST, we show the results of all methods in Table 5. In this experiment, we set $K = 150$ in which each client has about $100 \leq n \leq 300$ samples. Local training has relatively poor performance (29.12%) compared with other methods learning from the data of other clients. Compared with baselines, our method learns a robust representation with higher transferability and achieves better performance (67.19%).

5.3 Real-world Clinical Experiments

We also evaluate the strength of our method on a real-world clinical dataset eICU (Pollard et al., 2018), which collects data about their admissions to ICUs with hospital information. Each instance is a specific ICU stay. We follow the data pre-processing procedure in (Sheikhalishahi et al., 2019) and naturally treat different hospitals as local clients. The hospitals located in different places can have different patient populations, which induces statistical heterogeneity in the federated hospital network. We conduct the task of predicting in-hospital mortality, which is defined as the patient's outcome at the hospital discharge. The in-hospital mortality prediction is a binary classification task, where each data sample spans a 1-hour window.

In this experiment, we select 14 hospitals and each hospital has more than 500 samples. For all methods, we use a Bi-LSTM as the learning model. Due to label imbalance (more than 90% data are negative samples), we use AUC to measure the performance of all methods.

The results of all methods are in Table 5. Because of the existence of noise in the data eICU (e.g., missing value, error in measurement and mistaken recording, etc.), a learned model with higher robustness contributes to the mortality prediction. From Table 5, our method learns more robust and shared features and achieves a higher AUC compared with baselines.

6 Conclusion

In this paper, we investigate the scenario of selective partial sharing in federated learning. We propose to explore the shared components and utilize the informative data from the overlaps by a scalable data augmentation method. Comprehensive empirical evaluation results measured by quantitative metrics demonstrate the effectiveness and reliability of our method. Our study suggests several interesting topics for future explorations, including applying our framework to defend against Byzantine attacks (Lamport et al., 2019), a secure sample selection scheme, and the precise identification of information sharing in a federated network.

REFERENCES

Naman Agarwal, Ananda Theertha Suresh, X Yu Felix, Sanjiv Kumar, and Brendan McMahan. cpsgd: Communication-efficient and differentially-private distributed sgd. In *NeurIPS*, 2018.

Manoj Ghuhan Arivazhagan, Vinay Aggarwal, Aaditya Kumar Singh, and Sunav Choudhary. Federated learning with personalization layers. *arXiv preprint arXiv:1912.00818*, 2019.

Jose Blanchet and Karthyek Murthy. Quantifying distributional model risk via optimal transport. *Mathematics of Operations Research*, 44(2):565–600, 2019.

Sebastian Caldas, Sai Meher Karthik Duddu, Peter Wu, Tian Li, Jakub Konečnỳ, H Brendan McMahan, Virginia Smith, and Ameet Talwalkar. Leaf: A benchmark for federated settings. *arXiv preprint arXiv:1812.01097*, 2018.

Shuzhen Chen, Dongxiao Yu, Yifei Zou, Jiguo Yu, and Xiuzhen Cheng. Decentralized wireless federated learning with differential privacy. *IEEE Transactions on Industrial Informatics*, 18(9): 6273–6282, 2022.

Gregory Cohen, Saeed Afshar, Jonathan Tapson, and Andre Van Schaik. Emnist: Extending mnist to handwritten letters. In *2017 International Joint Conference on Neural Networks (IJCNN)*, pp. 2921–2926. IEEE, 2017.

Liam Collins, Hamed Hassani, Aryan Mokhtari, and Sanjay Shakkottai. Exploiting shared representations for personalized federated learning. 2021.

Sen Cui, Weishen Pan, Jian Liang, Changshui Zhang, and Fei Wang. Addressing algorithmic disparity and performance inconsistency in federated learning. *Advances in Neural Information Processing Systems*, 34:26091–26102, 2021.

Sen Cui, Jian Liang, Weishen Pan, Kun Chen, Changshui Zhang, and Fei Wang. Collaboration equilibrium in federated learning. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 241–251, 2022.

Yuyang Deng, Mohammad Mahdi Kamani, and Mehrdad Mahdavi. Adaptive personalized federated learning. *arXiv preprint arXiv:2003.13461*, 2020a.

Yuyang Deng, Mohammad Mahdi Kamani, and Mehrdad Mahdavi. Distributionally robust federated averaging. *Advances in Neural Information Processing Systems*, 33, 2020b.

Canh T Dinh, Nguyen H Tran, and Tuan Dung Nguyen. Personalized federated learning with moreau envelopes. *arXiv preprint arXiv:2006.08848*, 2020.

Alireza Fallah, Aryan Mokhtari, and Asuman Ozdaglar. Personalized federated learning with theoretical guarantees: A model-agnostic meta-learning approach. *Advances in Neural Information Processing Systems*, 33:3557–3568, 2020.

Rachael L Fleurence, Lesley H Curtis, Robert M Califf, Richard Platt, Joe V Selby, and Jeffrey S Brown. Launching pcornet, a national patient-centered clinical research network. *Journal of the American Medical Informatics Association*, 21(4):578–582, 2014.

Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank Reddi, Sebastian Stich, and Ananda Theertha Suresh. Scaffold: Stochastic controlled averaging for federated learning. In *International Conference on Machine Learning*, pp. 5132–5143. PMLR, 2020.

Mikhail Khodak, Maria-Florina Balcan, and Ameet Talwalkar. Adaptive gradient-based meta-learning methods. *arXiv preprint arXiv:1906.02717*, 2019.

Ron Kohavi et al. Scaling up the accuracy of naive-bayes classifiers: A decision-tree hybrid. In *Kdd*, volume 96, pp. 202–207, 1996.

Jakub Konečnỳ, H Brendan McMahan, Felix X Yu, Peter Richtárik, Ananda Theertha Suresh, and Dave Bacon. Federated learning: Strategies for improving communication efficiency. *arXiv preprint arXiv:1610.05492*, 2016.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

Leslie Lamport, Robert Shostak, and Marshall Pease. The byzantine generals problem. In *Concurrency: the Works of Leslie Lamport*, pp. 203–226. 2019.

Jaeho Lee and Maxim Raginsky. Minimax statistical learning with wasserstein distances. *arXiv preprint arXiv:1705.07815*, 2017.

Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smithy. Feddane: A federated newton-type method. In *2019 53rd Asilomar Conference on Signals, Systems, and Computers*, pp. 1227–1231. IEEE, 2019.

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

Yuzheng Li, Chuan Chen, Nan Liu, Huawei Huang, Zibin Zheng, and Qiang Yan. A blockchain-based decentralized federated learning framework with committee consensus. *IEEE Network*, 35(1): 234–241, 2020b.

Paul Pu Liang, Terrance Liu, Liu Ziyin, Nicholas B Allen, Randy P Auerbach, David Brent, Ruslan Salakhutdinov, and Louis-Philippe Morency. Think locally, act globally: Federated learning with local and global representations. *arXiv preprint arXiv:2001.01523*, 2020.

Wenjing Lou, Wei Liu, and Yuguang Fang. Spread: Enhancing data confidentiality in mobile ad hoc networks. In *IEEE INFOCOM 2004*, volume 4, pp. 2404–2413. IEEE, 2004.

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*, pp. 1273–1282. PMLR, 2017.

Mehryar Mohri, Gary Sivek, and Ananda Theertha Suresh. Agnostic federated learning. In *International Conference on Machine Learning*, pp. 4615–4625. PMLR, 2019.

Khalil Muhammad, Qinqin Wang, Diarmuid O'Reilly-Morgan, Elias Tragos, Barry Smyth, Neil Hurley, James Geraci, and Aonghus Lawlor. Fedfast: Going beyond average for faster training of federated recommender systems. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 1234–1242, 2020.

Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359, 2009.

Tom J Pollard, Alistair EW Johnson, Jesse D Raffa, Leo A Celi, Roger G Mark, and Omar Badawi. The eicu collaborative research database, a freely available multi-center database for critical care research. *Scientific data*, 5(1):1–13, 2018.

Hamed Rahimian and Sanjay Mehrotra. Distributionally robust optimization: A review. *arXiv preprint arXiv:1908.05659*, 2019.

Amirhossein Reisizadeh, Farzan Farnia, Ramtin Pedarsani, and Ali Jadbabaie. Robust federated learning: The case of affine distribution shifts. In *NeurIPS*, 2020.

Abhijit Guha Roy, Shayan Siddiqui, Sebastian Pölsterl, Nassir Navab, and Christian Wachinger. Braintorrent: A peer-to-peer environment for decentralized federated learning. *arXiv preprint arXiv:1905.06731*, 2019.

Seyedmostafa Sheikhalishahi, Vevake Balaraman, and Venet Osmani. Benchmarking machine learning models on eicu critical care dataset. *arXiv preprint arXiv:1910.00964*, 2019.

Virginia Smith, Chao-Kai Chiang, Maziar Sanjabi, and Ameet Talwalkar. Federated multi-task learning. *arXiv preprint arXiv:1705.10467*, 2017.

Michel Talagrand. *Upper and lower bounds for stochastic processes: modern methods and classical problems*, volume 60. Springer Science & Business Media, 2014.

Cédric Villani. *Topics in optimal transportation*, volume 58. American Mathematical Soc., 2021.

Kangkang Wang, Rajiv Mathews, Chloé Kiddon, Hubert Eichner, Françoise Beaufays, and Daniel Ramage. Federated evaluation of on-device personalization. *arXiv preprint arXiv:1910.10252*, 2019.

Stefanie Warnat-Herresthal, Hartmut Schultze, Krishnaprasad Lingadahalli Shastry, Sathyanarayanan Manamohan, Saikat Mukherjee, Vishesh Garg, Ravi Sarveswara, Kristian Händler, Peter Pickkers, N Ahmad Aziz, et al. Swarm learning for decentralized and confidential clinical machine learning. *Nature*, 594(7862):265–270, 2021.

Jie Xu, Benjamin S Glicksberg, Chang Su, Peter Walker, Jiang Bian, and Fei Wang. Federated learning for healthcare informatics. *Journal of Healthcare Informatics Research*, 5(1):1–19, 2021.

Qiang Yang, Yang Liu, Tianjian Chen, and Yongxin Tong. Federated machine learning: Concept and applications. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 10(2):1–19, 2019.

Tao Yu, Eugene Bagdasaryan, and Vitaly Shmatikov. Salvaging federated learning by local adaptation. *arXiv preprint arXiv:2002.04758*, 2020.

Ligeng Zhu, Zhijian Liu, and Song Han. Deep leakage from gradients. *Advances in Neural Information Processing Systems*, 32, 2019.

Zhuangdi Zhu, Junyuan Hong, and Jiayu Zhou. Data-free knowledge distillation for heterogeneous federated learning. *arXiv preprint arXiv:2105.10056*, 2021.

A THEORETICAL PROOFS

A.1 PROOF OF PROPOSITION 1

Firstly, we will prove the following lemma,

Lemma 1. *Suppose h is L-Lipschitz, i.e. $\|h_{z_p} - h_{z_q}\| \le L * d_z(z_p, z_q)$, $\forall\ z_p, z_q \in Z$, then, for any $Q :\in W(Q, \hat{D}) \le \epsilon$,*

$$\mathbb{E}_{z \in Q}\ell(z,h) \le \sup_{Q' :\in W(Q',\hat{D}) \le \epsilon} \mathbb{E}_{z \in Q'}\ell(z,h) \le \mathbb{E}_{z \in Q}\ell(z,h) + 2L\epsilon. \tag{19}$$

From Kantorovich dual representation of W in (Villani, 2021), suppose $F = h/L$ we have

$$W\left(Q, Q'\right) = \sup\left\{ \left|\mathbb{E}_Q F - \mathbb{E}_{Q'} F\right| \,\middle|\, \sup_{\substack{z, z' \in z \\ z \neq z'}} \frac{\left|F(z) - F\left(z'\right)\right|}{d_z\left(z, z'\right)} \le 1 \right\}. \tag{20}$$

And $W\left(Q, Q'\right) \le 2\epsilon$ by triangle inequality. So Lemma 1 holds.

From Lemma 1, we select Q as $\hat{D}$ and prove Proposition 1.

A.2 PROOF OF THEOREM 1

Suppose $\lambda_n \in \arg\min_{\lambda \ge 0} \lambda\epsilon + \mathbb{E}_{z \in \hat{D}}\left[\ell_{\lambda}^{d_z}(z,h)\right]$, we will firstly give the bound of $\left|\ell_{\lambda_n}^{d_z^*}(z,h) - \ell_{\lambda_n}^{d_z}(z,h)\right|$.

Suppose $\left|\ell_{\lambda_n}^{d_z^*}(z,h) \le \ell_{\lambda_n}^{d_z}(z,h)\right|$, and $z_0 = \arg\max_{z' \in Z} \ell\left(z', h\right) - \lambda_n d_z^*\left(z, z'\right)$. We have

$$\begin{aligned}
&\left|\ell_{\lambda_n}^{d_z^*}(z,h)-\ell_{\lambda_n}^{d_z}(z,h)\right| \\
&=\left|\sup_{z'\in Z}\left(\ell\left(z',h\right)-\lambda_n d_z^*\left(z,z'\right)\right)-\left(\sup_{z'\in Z}\ell\left(z',h\right)-\lambda_n d_z\left(z,z'\right)\right)\right| \\
&\leq\left(\ell\left(z_0,h\right)-\lambda_n d_z^*\left(z_0,z\right)\right)-\left(\ell\left(z_0,h\right)-\lambda_n d_z\left(z_0,z\right)\right) \\
&\leq\sup_{z'\in Z}\lambda_n\left|d_z^*\left(z',z\right)-d_z\left(z',z\right)\right| \\
&\leq\lambda_n\cdot B.
\end{aligned} \tag{21}$$

So we have

$$\mathbb{E}_{z\in\hat{D}}(\ell_{\lambda_n}^{d_z^*}(z,h)-\ell_{\lambda_n}^{d_z}(z,h))\leq\lambda_n\cdot B. \tag{22}$$

Recall that we have

$$\begin{aligned}
&\sup_{D:W^*(D,D^*)\leq\epsilon}\mathbb{E}_{z\in D^*}[\ell(z,h)]-\sup_{D:W\left(D,\hat{D}\right)\leq\epsilon}\mathbb{E}_{z\in\hat{D}}[\ell(z,h)] \\
&=\inf_{\lambda\geq 0}\left\{\lambda\epsilon+\mathbb{E}_{z\in D^*}\left[\ell_{\lambda}^{d_z^*}(z,h)\right]\right\}-(\lambda_n\epsilon+\mathbb{E}_{z\in\hat{D}}\left[\ell_{\lambda_n}^{d_z}(z,h)\right] \\
&\leq\mathbb{E}_{z\in D^*}\left[\ell_{\lambda_n}^{d_z^*}(z,h)\right]-\mathbb{E}_{z\in\hat{D}}\left[\ell_{\lambda_n}^{d_z}(z,h)\right].
\end{aligned} \tag{23}$$

Combining Eq.(22) and Eq.(24), we have

$$\begin{aligned}
&\sup_{D:W^*(D,D^*)\leq\epsilon}\mathbb{E}_{z\in D^*}[\ell(z,h)]-\sup_{D:W\left(D,\hat{D}\right)\leq\epsilon}\mathbb{E}_{z\in\hat{D}}[\ell(z,h)] \\
&\leq\mathbb{E}_{z\in D^*}\left[\ell_{\lambda_n}^{d_z^*}(z,h)\right]-\mathbb{E}_{z\in\hat{D}}\left[\ell_{\lambda_n}^{d_z^*}(z,h)\right]+\lambda_n B.
\end{aligned} \tag{24}$$

Then, we introduce the following lemma which bounds the optimal λ in Eq.(7) in the maintext.

Lemma 2. *(Lee & Raginsky, 2017) Suppose $\lambda_n\in\arg\min_{\lambda\geq 0}\lambda\epsilon+\mathbb{E}_{z\in D}\left[\ell_{\lambda}^{d_z}(z,h)\right]$ in Eq.(7) in the maintext. If the model $h\in\mathcal{H}$ satisfies Assumption 2 in the maintext, then λ_n is bounded:*

$$\lambda_n\leq\frac{L}{\sqrt{\epsilon}}. \tag{25}$$

From Lemma 2, combining Eq.(24), we have

$$\sup_{D:W^*(D,D^*)\leq\epsilon}\mathbb{E}_{z\in D^*}[\ell(z,h)]-\sup_{D:W\left(D,\hat{D}\right)\leq\epsilon}\mathbb{E}_{z\in\hat{D}}[\ell(z,h)]\leq\mathbb{E}_{z\in D^*}\left[\ell_{\lambda_n}^{d_z^*}(z,h)\right]-\mathbb{E}_{z\in\hat{D}}\left[\ell_{\lambda_n}^{d_z^*}(z,h)\right]+\frac{LB}{\sqrt{\epsilon}}, \tag{26}$$

as well as

$$\sup_{D:W\left(D,\hat{D}\right)\leq\epsilon}\mathbb{E}_{z\in D}[\ell(z,h)]-\sup_{D:W^*(D,D^*)\leq\epsilon}\mathbb{E}_{z\in D}[\ell(z,h)]\leq\mathbb{E}_{z\in\hat{D}}\left[\ell_{\lambda_*}^{d_z^*}(z,h)\right]-\mathbb{E}_{z\in D^*}\left[\ell_{\lambda_*}^{d_z^*}(z,h)\right]+\frac{LB}{\sqrt{\epsilon}}. \tag{27}$$

Combining Eq.(26) and Eq.(27), we have the following bound

$$\left|\sup_{D:W_*(D,D^*)\leq\epsilon}\mathbb{E}_{z\in D^*}[\ell(z,h)]-\sup_{D:W\left(D,\hat{D}\right)\leq\epsilon}\mathbb{E}_{z\in\hat{D}}[\ell(z,h)]\right|\leq\left|\mathbb{E}_{z\in\hat{D}}\left[\ell_{\lambda_*}^{d_z^*}(z,h)\right]-\mathbb{E}_{z\in D^*}\left[\ell_{\lambda_*}^{d_z^*}(z,h)\right]\right|+\frac{LB}{\sqrt{\epsilon}}. \tag{28}$$

Algorithm 1 obtain the projection T by gradient descent

```
Input: initial $\lambda_0$, learning rate $\alpha_\lambda$,
while in the $t$ iteration;
for each data $z_p \in \hat{D}^k$ do
    $T_\lambda(z_p) = \arg\max_{z \in Z} \ \ell(z,h) - \lambda_t \cdot d_z(z, z_p)$;
end for
update $\lambda_{t+1} =$
$\max\left\{0, \lambda_t - \alpha_\lambda \cdot (\epsilon - \frac{1}{n^k}\sum_{j=1}^{n^k} d_z(z_p, T_\lambda(z_p)))\right\}$;
until converged.
```

From Assumption 2, $\ell_\lambda^{d_z^*}$ is also bounded because

$$0 \le \ell(z_1,h) - \lambda_* d_z^*(z_1,z_1) \le \sup_{z \in Z} \ell(z,h) - \lambda_* d_z^*(z,z_1) \le \sup_{z \in Z} \ell(z,h) \le M. \tag{29}$$

Suppose $\mathcal{L}^{d_z^*} = \left\{\ell_\lambda^{d_z^*}(\cdot,h) : \lambda \in \left[0, \frac{L}{\sqrt{\epsilon}}\right], h \in \mathcal{H}\right\}$,

$$\left|\mathbb{E}_{z\in\hat{D}}\left[\ell_{\lambda_*}^{d_z^*}(z,h)\right] - \mathbb{E}_{z\in D^*}\left[\ell_{\lambda_*}^{d_z^*}(z,h)\right]\right| \le \max\left\{\sup_{f\in\mathcal{L}^{d_z^*}} \mathbb{E}_{z\in\hat{D}} f(z) - \mathbb{E}_{z\in D^*} f(z), \sup_{f\in\mathcal{L}^{d_z^*}} \mathbb{E}_{z\in D^*} f(z) - \mathbb{E}_{z\in\hat{D}} f(z)\right\}. \tag{30}$$

From the Theorem 1 in (Lee & Raginsky, 2017), we have

$$\max\left\{\sup_{f\in\mathcal{L}^{d_z^*}} \mathbb{E}_{z\in\hat{D}} f(z) - \mathbb{E}_{z\in D^*} f(z), \sup_{f\in\mathcal{L}^{d_z^*}} \mathbb{E}_{z\in D^*} f(z) - \mathbb{E}_{z\in\hat{D}} f(z)\right\} \le \frac{48}{\sqrt{n}}\mathfrak{C}(\mathcal{H}) + M\left(\frac{\log\frac{2}{t}}{2n}\right)^{\frac{1}{2}}, \tag{31}$$

with probability at least $1-t$, where $\mathfrak{C}(\mathcal{H})$ denotes the entropy integral complexity of the model space $\mathcal{H}$.

Combining Eq.(28) and Eq.(31), we have

$$\left|\sup_{D:W_*(D,D^*)\le\epsilon} \mathbb{E}_{z\in D^*}[\ell(z,h)] - \sup_{D:W\left(D,\hat{D}\right)\le\epsilon} \mathbb{E}_{z\in\hat{D}}[\ell(z,h)]\right| \le \frac{48\mathfrak{C}(\mathcal{L})}{\sqrt{n}} + \frac{LB}{\sqrt{\epsilon}} + M\left(\frac{\log\frac{2}{t}}{2n}\right)^{\frac{1}{2}}, \tag{32}$$

with probability at least $1-t$.

B More Discussion about our Decentralized Framework Fed-SPS

We articulate the whole pipeline of our framework in Algorithm 2. Firstly, we parallelly generate the adversarial distribution $\widetilde{D}^i$ for all clients. This process corresponds to Lines 1–7. Then we optimize the models collaboratively according to Eq.(13) in the maintext. In particular, each client sends its model to other clients to obtain the overlapped adversarial data, which are used to approximate the ODR defined in Eq.(12) in the maintext. Then all clients update their models using local data (including the original and the generated adversarial data) and the approximated ODR. This process corresponds to Lines 9–13.

B.1 Optimization Efficiency

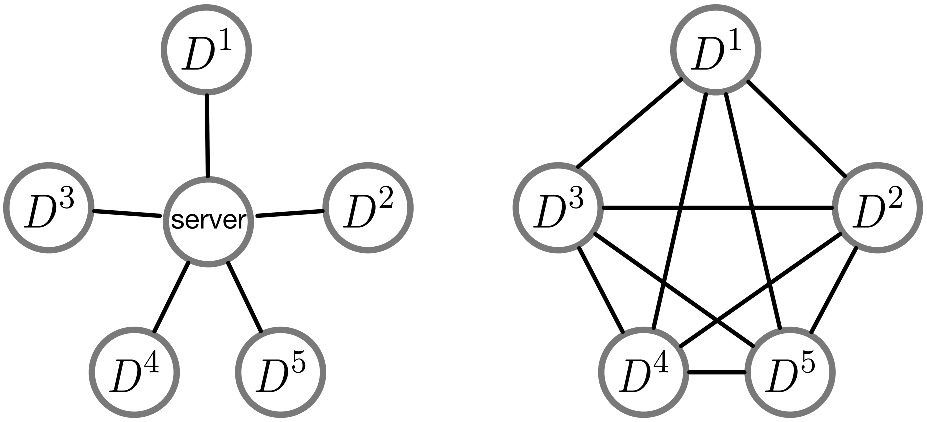
Compared with a centralized method, our proposed decentralized framework makes full use of local computing ability, which could learn personalized models in a parallel way. Moreover, we optimize the computation of ODR, which achieves a $O(K)$ time complexity.

The original objective has a $O(K^2)$ time complexity for an entire pairwise collaboration. In our approach, we approximate the ODR using Eq.(12) in the maintext, which reuses the adversarial

Algorithm 2 Decentralized framework `Fed-SPS`

Input: epoch T, batch size Bs, initial λ_0, initial models $\{h^1, ..., h^K\}$, weights α and β, learning rate α_λ, α_h for updating the model, tolerance ϵ;

```
for $t = 0, ..., T-1$ do
    randomly select a subset of clients $S_t$
    *obtain the perturbed samples for each client*
    for client $D^i \in S_t$ in parallel do
        draw mini-batch $(x^i_{t_1}, y^i_{t_1}), ..., (x^i_{t_B}, y^i_{t_B}) \sim D^i$
        obtain the projected sample $T(z^i_{t_b}, h^i), b \in [Bs]$ according to Algorighm 1
    end for
    *optimize each model by minimizing the objective in Eq.(13) in the maintext*
    for client $D^i \in S_t$ in parallel do
        send the model $h^i$ of the client $D^i$ to every other client $D^j \in S_t$;
        identify the perturbed samples $T(z^j, h^j)$ that fool the model $h^i$ and $h^j$ simultaneously to
            calculate the approximated ODR according to Eq.(12) in the maintext;
        update every personalized model $h^i$ by minimizing the objective in Eq.(13) in the maintext;
    end for
    fine-tune the local models $h^i \in S_t$ on their own data.
end for
Output: the learned personalized models $\{h^1, ..., h^K\}$.
```



(a) centralized collaboration

(b) decentralized collaboration

Figure 3: (a). Illustration of centralized collaboration, where each client sends its model (or gradient) to the server for information aggregation; (b). Illustration of decentralized collaboration. Since there is no main server, each client collaborates in a peer-to-peer way.

distribution $\widetilde{D}(\epsilon, h^j, \hat{D}^j)$ in Eq.(4) in the maintext and has a $O(K)$ time complexity as stated in Algorithm 2.

We also provide the comparisons of run-time consumption with baselines in Table 6. The experiments are on eICU dataset with 100 rounds. All methods are evaluated on the same device NVIDIA GeForce RTX 3090.

From Table 6, as all methods share a $O(K)$ time complexity, our method has a comparable time consumption as baselines. FedAvg averages the weights of all local models which costs less time. FedPer and LG-FedAve have a global-local structure, and they have a similar time consumption. Compared with other methods, Fed-MTL requires computing the correlation coefficient of the parameters of all local models, which could cause much time when the model structure is complex with many parameters.

Table 6: Run-time consumption comparisons

Methods	Run-time consumption
local training	1 min 30 s
local training + α	5 min 20 s
FedAvg	7 min 40 s
FedPer	10 min 32 s
Fed-MTL	251 min 41 s
LG-FedAvg	12 min 20 s
FedRep	10 min 20 s
Fed-SPS	18 min 45 s

B.2 Data Confidentiality and Privacy Preserving

Data confidentiality refers to the protection of transmitted data from passive attacks, such as eavesdropping (Lou et al., 2004). Privacy-Preserving means that there is no leakage of sensitive information (e.g., gender, age, income, etc.) in the data when learning models (McMahan et al., 2017).

Federated learning maintains data confidentiality when learning models. For centralized collaboration shown in Figure 3(a), the models are updated locally and then returned to the main server for aggregation. For decentralized collaboration shown in Figure 3(b), as there is no main server, the collaborative model learning relies on the model transfer between local clients. Recently, to further keep data confidentiality, there are researchers proposing to use blockchain techniques when broadcasting models between clients (Warnat-Herresthal et al., 2021).

Federated learning may need further exploration to maintain data privacy. Some researchers claim that there is no compromise of privacy when sharing models with other clients (Roy et al., 2019). Meanwhile, recent studies reveal that there is information leakage when sharing models or gradients (Zhu et al., 2019). For the centralized federated method shown in Figure 3(a), it requires the model (or gradient) sharing between local clients and the main server, which could cause privacy leakage on the server during the transmission process. For our decentralized federated method shown in Figure 3(b), the learning of personalized models needs model sharing between local clients, which could also cause privacy leakage on other clients.

C Ablation Studies

C.1 Discussions about Existing Distributional Robustness Methods

Our method proposes to boost the utilization of the overlaps between local clients in a pairwise way. There is research aiming to defend against adversarial shifts by strengthening the distributional robustness of the learned global model (Deng et al., 2020b; Reisizadeh et al., 2020). We also conduct experiments to compare our method with these methods. The results on the synthetic data (Adult), the benchmark data (FEMNIST) and the real data (eICU) are in Table 7. Our method achieves better performance, because

Table 7: comparisons with the methods proposed by Deng *et al.* and Diamandis *et al.*

Dataset	Deng(%)	Diamandis(%)	FedAvg	ours(%)
Adult	$81.1_{\pm.1}$	$82.3_{\pm.1}$	$82.4_{\pm.1}$	$\mathbf{82.9}_{\pm.3}$
FEMNIST	$61.6_{\pm.6}$	$60.4_{\pm1.2}$	$60.7_{\pm1.8}$	$\mathbf{67.2}_{\pm.3}$
eICU	$83.1_{\pm.5}$	$83.7_{\pm1.1}$	$84.4_{\pm.9}$	$\mathbf{86.6}_{\pm.4}$

1). **different goals:** the referred algorithms mainly learn robust models to defend against adversarial shifts, which could even lead to performance degradation. For example, FedAvg achieves a comparable or better performance compared with baselines. We explore the shared underlying structures and provide a partial and pairwise collaboration among clients to avoid potential negative transfer to improve the performance.

2). **different learning paradigms:** the above two algorithms learn a global model. Our method learns a personalized model for each client by pairwise colla boration, which provides more inherent benefits to fit local distributions.

C.2 Discussions about the Objective of Fed-SPS

From Eq.(13) in the main text, the objective of our method consists of three terms. 1). local training ($\alpha = 0, \beta = 0$), 2). FedAvg ($\alpha = 0, \beta = 1, \epsilon \to 0$) 3). local training with distributional robustness ($\alpha \neq 0, \beta = 0$) are the three special cases of our method. We compare our method with the three methods on several datasets, including the synthetic dataset Adult, benchmark image datasets CIFAR10 ($S = 2$) and CIFAR100 ($S = 20$) and FEMNIST, and real-world dataset eICU. The results are shown in Table 8.

Table 8: Comparisons with local training, FedAvg and local training with distributional robustness.

Dataset	Local training	FedAvg	Local training + α	ours
Adult	$82.3_{\pm.0}$	$82.3_{\pm.1}$	$82.3_{\pm.1}$	$82.9_{\pm.3}$
CIFAR10($K = 100$)	$87.3_{\pm.6}$	$44.3_{\pm.4}$	$85.3_{\pm.2}$	$87.8_{\pm.2}$
CIFAR10($K = 1000$)	$72.2_{\pm.3}$	$41.1_{\pm.6}$	$71.3_{\pm.1}$	$77.8_{\pm.3}$
CIFAR100($K = 100$)	$32.0_{\pm.7}$	$15.7_{\pm.8}$	$34.3_{\pm.3}$	$39.5_{\pm.5}$
CIFAR100($K = 500$)	$21.1_{\pm.6}$	$19.4_{\pm.7}$	$21.4_{\pm.1}$	$24.1_{\pm.4}$
FEMNIST	$29.1_{\pm.9}$	$60.7_{\pm1.8}$	$31.3_{\pm.3}$	$67.2_{\pm.3}$
eICU	$82.9_{\pm1.2}$	$84.4_{\pm.9}$	$83.1_{\pm.7}$	$86.6_{\pm.4}$

For the dataset Adult, we use a linear model following baselines, which has relatively few parameters. Since there are more than 40000 samples, learning a local or global model achieves a similar performance (82.3%). For CIFAR10, there are total 10 classes with 50000 samples. When $K = 100$, each client has 500 samples and local training achieves 87.3 accuracy. In this case, learning local models with distributional robustness cause performance degradation (85.3) because of the trade-off between performance and robustness. This corresponds to the analysis in Section C.1. For CIFAR100 dataset, there are total 100 classes with 50000 samples. The prediction task on CIFAR100 is more difficult than on CIFAR10. In this case, when $K = 100$, local training achieves a relatively low accuracy 32.0. As we stated in the maintext, local limited data is hard to describe the true implied distribution and distributional robustness may improve the generalization. Distributional robust training generates other samples and improves the model learning (34.3 accuracy). A similar phenomenon could also be found in the experiments on FEMNIST and eICU dataset.

C.3 Discussions about Hyperparameters

As we stated in above, we propose Fed-SPS which has three terms. To explore the effect of distributional robustness on the generalization, we did experiments on the benchmark dataset (CIFAR10) and real-world dataset (eICU).

Since ϵ controls the scale of the uncertainty about the data distribution, we experimentally evaluate the effect of ϵ when learning models on CIFAR10 with $\alpha = 0.2, \beta = 0.0$.

Table 9: the effect of ϵ on benchmark dataset

β	0.0	0.01	0.03	0.05	0.1
ACC	$67.8_{\pm.5}$	$68.8_{\pm.4}$	$70.0_{\pm.4}$	$69.2_{\pm.2}$	$69.1_{\pm.4}$

From Table 9, when $\epsilon \leq 0.03$, the performance of the learned models increases. However, when $\epsilon \leq 0.03$, the crafted adversarial samples could be significantly different from the natural data, which degrades the performance of the learned models.

To quantify the effect of α, we conduct experiments on CIFAR10 with $S = 5, K = 100$, with $\epsilon = 0.03$ and $\beta = 0.0$. The results are shown in the following Table 10.

Table 10: the effect of α on the benchmark dataset

α	0.0	0.01	0.05	0.1	0.2	0.5	1.0	2.0	5.0
ACC	$67.8_{\pm.5}$	$67.7_{\pm.5}$	$68.2_{\pm.5}$	$68.5_{\pm.4}$	$70.0_{\pm.4}$	$69.0_{\pm.5}$	$67.8_{\pm.6}$	$68.0_{\pm.3}$	$66.1_{\pm.4}$

From Table 10, when $0 \leq \alpha \leq 0.2$, the performance increases. As each client has 500 samples with 5 classes, the limited data is hard to cover all cases for this classification task. Improving distributional robustness stabilizes the model learning. When $\alpha \geq 0.2$, the performance decreases. In particular, as $\alpha = 5.0$, the performance(66.1) is lower than the performance of the model learned when $\alpha = 0.0$ (67.8), which indicates that the distributional robustness costs more modal capacity and hurts the model learning.

We evaluate the effect of β by conducting experiments on eICU dataset, in which we set an optimal $\alpha = 0.1$. The results are shown in Table 11.

Table 11: the effect of β on real-data

β	0.0	0.1	0.5	1.0	2.0	5.0
AUC	82.6$_{\pm.4}$	84.7$_{\pm.7}$	86.9$_{\pm.6}$	86.1$_{\pm.3}$	85.5$_{\pm.7}$	85.4$_{\pm.2}$

From Table 11, as $0 \leq \beta \leq 0.5$, the performance of the learned model increases. In this case, enforcing ODR facilitates the utilization of the data from the overlapped data. When $\beta \geq 0.5$, the performance of the learned model decreases. It means that focusing too much on the robustness could harm the generalization of the learned model.

D Experiments and Implementation Details

D.1 Datasets and Implementations

In our synthetic datasets, we use the public data set Adult from UCI. We follow the dataset preprocessing procedure in (Mohri et al., 2019) and split it into two clients. All features are binary. We run all experiments five times and report the mean accuracy in the main text. For the benchmark datasets, CIFAR10, CIFAR100, and FEMNIST are public datasets. We follow the setting in the work of (Collins et al., 2021). All results of baselines are from (Collins et al., 2021). We conduct the experiments on the real-world clinical dataset eICU (Pollard et al., 2018), for which permission is required. We follow the procedure on the website https://eicu-crd.mit.edu and got the approval for the dataset. In this experiment, we follow the data preprocessing as in (Sheikhalishahi et al., 2019) and select 14 hospitals as introduced in the main text. All hospitals own patient data samples $500 \leq n \leq 1300$. We implement all baselines using the source codes from (Collins et al., 2021). The source codes are made publicly available at https://github.com/fedips/Fed-SPS.

D.2 Devices

We run our experiments on a local Linux server that has two physical CPU chips (Intel(R) Xeon(R) CPU E5-2640 v4 @ 2.40GHz) and 32 logical kernels. We use Pytorch framework to implement our model and train the models on GeForce RTX 2080 Ti GPUs and GeForce RTX 3090 GPUs.