OPTIMIZING SERVER-SIDE AGGREGATION FOR ROBUST FEDERATED LEARNING VIA SUBSPACE TRAINING

Anonymous authors
Paper under double-blind review

ABSTRACT

Non-IID data distribution across clients and poisoning attacks are two main challenges in real-world federated learning systems. While both of them have attracted great research interest with specific strategies developed, no known solution manages to address them in a unified framework. To overcome both challenges, we propose SmartFL, a generic approach that optimizes the server-side aggregation process with a small amount of on-server proxy data (e.g., around one hundred samples for CIFAR-10) via a subspace training technique. Specifically, the aggregation weight of each participating client at each round is optimized using the server-side proxy data, which is essentially the optimization of the global model in the convex hull spanned by client models. Since at each round, the number of tunable parameters optimized on the server side equals the number of participating clients (thus independent of the model size), we are able to train a global model with massive parameters using only a small amount of server-side proxy data. We provide theoretical analyses of the convergence and generalization capacity for SmartFL. Empirically, SmartFL achieves state-of-the-art performance on both federated learning with non-IID data distribution and federated learning with malicious clients.

1 INTRODUCTION

Data security and privacy have raised increasing interest in machine learning research, especially in privacy-sensitive areas such as health care (Rieke et al., 2020). Federated Learning (FL) emerges as an effective privacy-preserving machine learning approach to jointly optimize a global model over decentralized data (Konečnỳ et al., 2016; Yang et al., 2019). Typically, generic FL involves multiple rounds of clients' local training followed by server-side aggregation. The **server-side aggregation** plays an essential role that aggregates the client models into a global model, which is then used to initialize the clients in the next training round. The standard aggregation strategy Federated Averaging (FedAVG) (McMahan et al., 2017), which takes the sample number weighted average over clients' weights, is shown to converge to an ideal model as centralized training and works well in IID data distribution without poisoning attacks (Zinkevich et al., 2010; McMahan et al., 2017; Zhou & Cong, 2017).

However, in real-world scenarios, the non-IID distribution of data across clients and the potential presence of malicious clients severely compromise the effectiveness of standard FL aggregation (Konečnỳ et al., 2016; Yang et al., 2019; Yin et al., 2018). Plenty of advanced server-side aggregation strategies have been proposed to address these two problems separately due to their seemingly different objectives. Specifically, to tackle the non-IID distribution of data, prior studies propose to reweight the updates based on statistics of local updates (Wang et al., 2020; Yeganeh et al., 2020; Xiao et al., 2021) or perform further tuning with proxy data on the global model (Lin et al., 2020; Chen & Chao, 2021a) in every communication round to alleviate the influence of large deviation of clients' local models. To improve robustness against attacks, Byzantine-robust aggregations (Yin et al., 2018; Blanchard et al., 2017; Guerraoui et al., 2018) are introduced to exclude statistically suspicious outliers of updates; proxy data are utilized to provide additional clues for enhanced performance (Park et al., 2021; Cao et al., 2021).

Considering the practical scenario that the server has some reasonable knowledge of the task (i.e., a small amount of server proxy data), can we leverage such knowledge to optimize the aggrega-

tion process to handle challenges from malicious and heterogeneous clients? A straightforward data-driven optimization strategy using proxy data would be further finetuning the global model aggregated with FedAVG in every communication round, which is shown to be effective in tackling non-IID distribution of data (Lin et al., 2020; Chen & Chao, 2021a) using ensemble knowledge distillation or ground truth labels. We term those approaches *full-space training* since they optimize the global model in the entire parameter space. However, to tune the global model with massive parameters, a large amount of proxy data is required as the carrier of knowledge. Otherwise, severe overfitting may occur, which is verified in the experimental section (see Section 5.2). Unfortunately, in real-world scenarios, federated learning is generally applied in privacy-sensitive areas, such as health care, where collecting lots of on-server proxy data becomes almost impractical. What's more, with limited proxy data, the full-space training approaches are unlikely to mitigate the negative effects of malicious clients, which have been aggregated into the global model with FedAVG as initialization. This is verified in the experimental section (see Section 5.3). Also, full-space training leads to low aggregation efficiency and long latency because of the large dataset used and huge amounts of parameters to optimize. Finally, it remains unclear whether the full-space training-based FL systems can be theoretically guaranteed to converge to the optimum.

In light of the above-mentioned issues, we propose SmartFL with a generic and powerful aggregation strategy that optimizes the aggregated global model via *subspace training* within the convex hull spanned by the client models' parameters. To be precise, each time after local training, SmartFL updates the global model to be the optimal convex combination of the received client models' parameters by fitting the on-server labelled proxy data. This extracted subspace is mainly inspired by two facts. On the one hand, prior studies on mode connectivity (Garipov et al., 2018; Draxler et al., 2018) show that low-cost solutions found by two networks can be connected by simple (e.g., piecewise linear) paths with constant error or loss. Some mathematical explanations (Kuditipudi et al., 2019) for this phenomenon have been provided recently. On the other hand, this subspace naturally contains the aggregation solutions of several prior efforts for heterogeneous FL and attack-robust FL (Yeganeh et al., 2020; Xiao et al., 2021; Wang et al., 2020; Park et al., 2021). These facts suggest that the extracted subspace has the potential to contain the desirable global model. Through constructing the subspace, we can significantly reduce the degree of freedom for training. This makes SmartFL enjoy a much lower demand for on-server proxy data, better generalization, and higher aggregation efficiency. What's more, the negative effects of malicious clients can be readily alleviated when the weights for malicious clients are optimized to low values. We also establish theoretical guarantees on the convergence and generalization of SmartFL.

It is worth mentioning that our setup is practical, which assumes the server collects a small clean labelled proxy dataset (around a hundred samples by default). Actually, on-server labelled proxy data is widely utilized in the advanced aggregation methods for heterogeneous FL (Xiao et al., 2021) and attack-robust FL (Cao et al., 2021; Park et al., 2021). The number of required proxy data for SmartFL is among the lowest ones. To further empower practical usage for extreme conditions and compare comprehensively with the state-of-the-art aggregation strategies for heterogeneous FL using unlabelled proxy data (Lin et al., 2020; Chen & Chao, 2021a), we also extend to the usage of a small amount of unlabelled data (SmartFL-U) for heterogeneous FL. Specifically, we optimize the combination coefficients for labelled data with ground truth labels and unlabelled data with pseudo-labels generated by the ensemble of clients. The corresponding full-space training counterparts for our methods are regular Finetuning and FedDF (Lin et al., 2020) for labelled and unlabelled data, respectively.

We conduct extensive experiments on CIFAR-10/100, MNIST, and 20Newsgroups. The results demonstrate that SmartFL can boost the performance of FL with heterogeneous data distribution with very few proxy data samples. For instance, with only 128 samples of server proxy data for CIFAR-10, we can attain a significant performance improvement compared with state-of-the-art methods for heterogeneous FL (Lin et al., 2020; Chen & Chao, 2021a; Xiao et al., 2021; Li et al., 2020; Karimireddy et al., 2020) and full-space training counterparts (Lin et al., 2020; Chen & Chao, 2021a). Also, when malicious clients exist, our solution manages to defend against the attacks even in the condition of a large portion of attacks and highly-non-IID distribution of data, yielding state-of-the-art performance compared with existing attack-robust methods with proxy data (Cao et al., 2021; Park et al., 2021) and statistical filtering-based methods (Yin et al., 2018; Blanchard et al., 2017). Our contributions can be summarized as follows:

- We propose SmartFL, which effectively optimizes server-side aggregation with a small amount of proxy data via subspace training.
- As far as we know, SmartFL is the first FL framework that simultaneously handles two major challenges in FL systems (i.e., non-IID distribution of data and poisoning attacks) in a unified framework.
- We provide theoretical analysis for convergence and generalization capacity for SmartFL.
- Extensive experiments on multiple datasets with non-IID data distribution and poisoning attacks demonstrate the superiority of our method.

2 Related Work

2.1 Federated Learning with non-IID Data Distribution

Increasing research efforts are devoted to improving the FL performance with heterogeneous data distribution. They can be classified into modifying local training and modifying server-side aggregation. In this section, we focus on the latter one, which is more closely related to our work. More related works on improving local training are discussed in Appendix A. Several prior studies propose to reweight the model updates with some statistical property. FedNova (Wang et al., 2020) proposes to normalize the aggregation weights according to the local training steps. IDA (Yeganeh et al., 2020) sets the weights according to the inverse distance of the client models to the global model. FedAvgM (Hsu et al., 2019) further goes beyond the weighted average and adopts server-side momentum to improve the aggregation. Recently, solutions leveraging server unlabelled/labelled data to further tune the aggregated global model in every communication round have drawn much research attention with promising performance. Specifically, FedDF (Lin et al., 2020) leverages ensemble knowledge with average logits of clients' predictions on the server unlabelled data to finetune the global model. FedBE (Chen & Chao, 2021a) further proposes generating pseudo labels with the Bayesian ensemble technique and validates that the ground truth labels (referred as Finetuning in our paper) lead to the best finetuning performance if available. FedET (Cho et al., 2022) and FedAUX (Gu et al., 2022) include more carefully logit ensembling strategies. However, these solutions have a high demand for data on the server to tune the global model, which is not always realistic for FL systems, even for unlabelled data. ABAVG (Xiao et al., 2021) uses proxy data accuracy on the server labelled dataset to determine the aggregation weight of clients to enable quality-aware aggregation. However, this solution heuristically assumes the coefficients should be proportional to the proxy data accuracy, which does not fully utilize the ground truth knowledge and does not get pleasant gain.

2.2 Attack-resistant Aggregation

It is well known that FL is vulnerable to poisoning attacks due to a vast number of uncontrolled clients, some of which may be malicious (Shejwalkar et al., 2022). Plenty of attack-resistant aggregation methods are proposed to tackle the problem. Blanchard et al. (2017) propose a vector-wise filtering technique named Krum and raises attention to Byzantine-robust aggregation techniques. Afterward, dimension-wise filtering techniques are introduced, such as Median (Yin et al., 2018), Trimmed Mean (Yin et al., 2018), and signSGD based on majority voting (Bernstein et al., 2019). Also, advanced vector-wise filtering methods include Multi-Krum (Blanchard et al., 2017), Bulyan (Guerraoui et al., 2018), RFA (Pillutla et al., 2019), RSA (Li et al., 2019), DnC (Shejwalkar & Houmansadr, 2021), residual-based reweighting (Fu et al., 2021), and attack-adaptive aggregation (Wan & Chen, 2021). Most of these solutions can guarantee the success of defense under certain assumptions, such as IID distribution of data or the constrained portion of malicious clients. However, such assumptions do not always hold in real scenarios. The recent work (Karimireddy et al., 2022) provides CClip, which combines bucketing with existing solutions to provide a convergence-guaranteed defense under non-IID scenarios, yet still cannot deal with a large portion of attackers. Leveraging proxy data provides the possibility to further leverage server knowledge to help defend against attacks. FLTrust (Cao et al., 2021) maintains a server model and utilizes the statistical properties of the client model and server model to reweight the client updates. Sageflow (Park et al., 2021) combines entropy-based filtering and loss-based reweighting with the proxy data. Both methods leverage proxy data to perform some statistical analysis to heuristically reweight the client updates, while our method directly uses server proxy data to optimize the aggregation and leads to stabler defense performance, faster convergence, and functionality beyond solely tackling attacks such as improving FL with heterogeneous data distribution without attacks.

3 BACKGROUND

Generic FL. Suppose we have M clients with local private dataset $\mathcal{D}_m = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{|\mathcal{D}_m|}$ drawn from the heterogeneous local distributions, and $\mathcal{D} = \cup_{m=1}^{M} \mathcal{D}_m$ denotes all data from all clients, which can be viewed as sampled from the global distribution. Then the generic federated learning optimization problem can be formulated as

$$\min_{\boldsymbol{w}} \mathcal{L}(\boldsymbol{w}, \mathcal{D}) = \sum_{m=1}^{M} \alpha_m \mathcal{L}_m(\boldsymbol{w}, \mathcal{D}_m), \tag{1}$$

where $\boldsymbol{w} \in \mathbb{R}^d$ is the model parameter, $\alpha_m = \frac{|\mathcal{D}_m|}{|\mathcal{D}|}$, and $\mathcal{L}_m(\boldsymbol{w}, \mathcal{D}_m) = \frac{1}{|\mathcal{D}_m|} \sum_{\xi \in \mathcal{D}_m} \ell(\boldsymbol{w}, \xi)$ is the empirical risk for client m with $\ell(\cdot, \cdot)$ being the loss function. We denote the optimal solution of (1) as $\boldsymbol{w}^*$.

FedAVG. Since the data is retained on local clients, the optimization problem cannot be directly solved. To approximately approach the problem, a standard solution is FedAVG (McMahan et al., 2017), which aggregates the locally trained models to a global shared model on the server. The global model $\boldsymbol{w}^{t+1}$ is aggregated as follows at the end of t-th communication round:

$$\boldsymbol{w}^{t+1} = \frac{1}{C^t} \sum_{m \in \mathcal{M}^t} \alpha_m \boldsymbol{w}_m^t = \boldsymbol{w}^t + \frac{1}{C^t} \sum_{m=1}^{M} \alpha_m \Delta_m^t, \tag{2}$$

where $\mathcal{M}^t \subset [M] = \{1, 2, \ldots, M\}$ is the set of clients sampled in the t-th round, $C^t = \sum_{m \in \mathcal{M}^t} \alpha_m$, $\boldsymbol{w}_m^t$ denotes the client m's local model trained with the local dataset $\mathcal{D}_m$ at the end of t-th communication round, and $\Delta_m^t = \boldsymbol{w}_m^t - \boldsymbol{w}^t$ denotes the cumulative local updates of client m in round t.

4 SMARTFL

4.1 METHOD

In this section, we introduce SmartFL with a generic and powerful server-side aggregation strategy to smartly aggregate an optimized global model from clients' updated models using a small amount of proxy data. Through optimizing the aggregation process in every communication round, SmartFL simultaneously tackles various challenging conditions (i.e., non-IID data distribution and poisoning attacks) and enables a stable and robust aggregation. We first introduce the **formulation of server-side optimization problem** and the straightforward regular training scheme. Then, we demonstrate the key component of SmartFL, i.e., **the subspace training technique**, to overcome the drawbacks of regular data-driven optimization on the entire model parameters. Afterward, we show the strategy for **the extension to unlabelled proxy data**. Finally, we provide the **implementation and overall process** (algorithm 1).

Server-side Optimization. We aim to leverage server proxy data to optimize the global model based on the clients' local models. Note that the server-side optimization is performed on the global model $\boldsymbol{w}^{t+1}$, for $t = 0, 1, 2 \ldots, T$, in the server-side aggregation process at the end of every communication round. For simplicity, we denote the global model as $\boldsymbol{w}$ and demonstrate the on-server optimization for the aggregation at the end of t-th communication round as follows. Assuming the server holds a small amount of unbiased labelled proxy data $\mathcal{D}_s$ sampled from the global distribution $\mathcal{D}$, we can optimize the global model $\boldsymbol{w}$ with the empirical risk on the proxy data, denoted as $\mathcal{L}_s(\boldsymbol{w}, \mathcal{D}_s) = \frac{1}{|\mathcal{D}_s|} \sum_{\xi \in \mathcal{D}_s} \ell(\boldsymbol{w}, \xi)$.

A straightforward data-driven optimization strategy is further finetuning the global model initialized with FedAVG, which is validated to be effective in dealing with non-IID data distribution in the prior study (Chen & Chao, 2021a) if plenty of proxy data is available. The optimization process is as follows:

$$\textbf{Initialization} : \boldsymbol{w} \leftarrow \frac{1}{C^t} \sum_{m \in \mathcal{M}^t} \alpha_m \boldsymbol{w}_m^t; \qquad \textbf{Full-space Trainig} : \min_{\boldsymbol{w}} \mathcal{L}_s(\boldsymbol{w}, \mathcal{D}_s). \tag{3}$$

However, this strategy suffers from severe overfitting in the practical scenario, where the on-server proxy dataset is not likely to be impractically large. Also, this method can not effectively eliminate the effects of poisoning attacks.

Subspace Training for Server-side Optimization. Inspired by prior studies on mode connectivity and the success of reweighting-based methods for heterogeneous/attack-robust FL, as we discussed

in the introduction section, we constrain the optimization process in the promising subspace, i.e., the convex hull spanned by the clients' models. Instead of training the global model in the entire parameter space, we optimize the model in the reduced subspace with a significantly lower dimension. The subspace optimization problem at the end of communication round t can be formulated as

$$\min_{\boldsymbol{w}} \mathcal{L}_s(\boldsymbol{w}, \mathcal{D}_s), \qquad s.t.\ \ \boldsymbol{w} = \sum_{m \in \mathcal{M}^t} p_m \boldsymbol{w}_m^t,\ \ and\ \ \boldsymbol{p} \in \Lambda, \tag{4}$$

where p_m is the aggregation coefficient for client m, and $\mathcal{L}_s(\boldsymbol{w}, \mathcal{D}_s)$ is the empirical risk, Λ is defined as

$$\Lambda = \{\boldsymbol{p} \in \mathbb{R}^M : p_m \geq 0 \text{ for } m \in \mathcal{M}^t,\ \ p_m \equiv 0 \text{ for } m \in [M] \setminus \mathcal{M}^t,\ \ and \sum_{m \in \mathcal{M}^t} p_m = 1\}.$$

Note that in solving problem (4), we optimize $\boldsymbol{w}$ over its coefficients $\boldsymbol{p}$ with fixed $\boldsymbol{w}_m^t$. We denote $\mathcal{L}_s(\boldsymbol{w}, \mathcal{D}_s)$, with a slight abuse of notation, as $\mathcal{L}_s(\boldsymbol{p}, \mathcal{D}_s)$ and the problem can then be rewritten as

$$\min_{\boldsymbol{p} \in \Lambda} \mathcal{L}_s(\boldsymbol{p}, \mathcal{D}_s). \tag{5}$$

We would like to point out that all the elements in $[M] \setminus \mathcal{M}^t$ are fixed to be 0, and Λ is essentially a $|\mathcal{M}^t|$ dimensional set. Thus, we only need to optimize $|\mathcal{M}^t|$ parameters, i.e., p_m with $m \in \mathcal{M}^t$, instead of the entire neural network parameter space. This aggregation process can find the optimal model fusion, i.e., a convex combination of client models trained on non-IID datasets, by learning on the labelled proxy data. Benefiting from such a small optimization space, the generalization ability of our approach can be significantly reinforced so that it can work well even with a small amount of proxy data. This will be further discussed in our theoretical analysis. Moreover, when malicious clients exist, our aggregation can mitigate their negative effects by optimizing corresponding p_m to small values.

Extension to Unlabelled Samples. Note that for the labelled proxy data, the loss function $\ell(\cdot, \cdot)$ of the empirical risk $\mathcal{L}_s(\boldsymbol{w}, \mathcal{D}_s)$ for server-side optimization is the same as the global optimization in (1), which is cross entropy loss in practice. To further facilitate the practical usage for different conditions and fairly compare with the full-space training solutions using unlabelled proxy data for heterogeneous FL (Lin et al., 2020; Chen & Chao, 2021a), we provide an extension to unlabelled samples (**SmartFL-U**). Specifically, we utilize the exact strategy in the prior work (Lin et al., 2020) to generate pseudo labels with clients' ensemble logits and use Kullback-Leibler divergence loss to drive the global model to mimic the prediction of the ensemble of client models. The only difference is that we train the model in the reduced subspace instead of full space in Lin et al. (2020). Since there is no quality guarantee for the pseudo labels generated from client predictions, SmartFL-U is only applied for the empirical study of handling heterogeneous data distribution. Our theoretical analysis and studies on FL with poisoning attacks focus on SmartFL with labelled proxy data.

Algorithm 1: SmartFL.

for *each communication round* $t = 0, \ldots, T$ **do**
  $\mathcal{M}_t \leftarrow$ selected subset of the M clients
  for *each client* $m \in \mathcal{M}_t$ ***in parallel*** **do**
    $\boldsymbol{w}_m^t \leftarrow$ Client-LocalUpdate$(m, \boldsymbol{w}^t)$
  end
  initialize the aggregation coefficients with local samples as
  $\boldsymbol{p}^{t,0} \leftarrow \tilde{\boldsymbol{\alpha}}$ with $\tilde{\alpha}_m = \alpha_m / C^t$ for all $m \in \mathcal{M}^t$ and otherwise $\tilde{\alpha}_m = 0$
  for *j in* $\{1, \ldots, E_s\}$ **do**
    update in mini-batches $\boldsymbol{p}^{t,j} \leftarrow \text{proj}_\Lambda\left(\boldsymbol{p}^{t,j-1} - \eta_s \nabla \mathcal{L}_s(\boldsymbol{p}, \mathcal{D}_s)\right)$
  end
  $\boldsymbol{p}^t \leftarrow \boldsymbol{p}^{t,E_s}$
  $\boldsymbol{w}^{t+1} \leftarrow \sum_{m \in \mathcal{M}_t} p_m^t \boldsymbol{w}_m^t$
end
return $\boldsymbol{w}_{T+1}$

Implementation and Overall Process. Algorithm 1 demonstrates the overall process of SmartFL. The optimization process in (5) can be solved by general projected stochastic gradient descent algorithms (Zhou et al., 2021). To stabilize the training process, we can optionally introduce some regularization terms into our aggregation process. Specifically, when dealing with non-IID data

distribution, we can adopt the L2-norm based regularization as follows:

$$\min_{\boldsymbol{p}\in\Lambda} \mathcal{J}(\boldsymbol{p}) = \mathcal{L}_s(\boldsymbol{p}, \mathcal{D}_s) + \lambda \cdot \frac{1}{2}\|\boldsymbol{p} - \tilde{\boldsymbol{\alpha}}\|_2^2, \tag{6}$$

where $\tilde{\alpha}_m = \alpha_m / C^t$ for all $m \in \mathcal{M}^t$ and otherwise $\tilde{\alpha}_m = 0$. This regularization encourages SmartFL to find a good aggregation near the averaged model.

4.2 Theoretical Analysis

In this section, we provide a convergence property of SmartFL under poisoning attacks. Then, we show the advantages of SmartFL over FedAVG and full-space training regarding generalization capacity. Detailed description and derivations are deferred to Appendix B.

Property 1 (**Convergence**). [informal] *Assume in each server-side aggregation, there exists at least one honest client among the M sampled clients. With other assumptions specified in the appendix, the expected error of SmartFL, i.e., $\mathbb{E}\left[\mathcal{L}(\boldsymbol{w}^T, \mathcal{D}) - \mathcal{L}(\boldsymbol{w}^*, \mathcal{D}_s)\right]$, can converge linearly as $T \to \infty$.*

Remark. *The above result shows that $SmartFL$ can converge to the optimum $\boldsymbol{w}^*$ in the global optimization problem (1) efficiently even when a large number of malicious clients exist, which is consistent with our empirical results (see Section 5.3). Note that in Property 1, we allow the data on the clients to be non-IID. Therefore, this result holds naturally for the cases of non-IID data distribution without poisoning attacks.*

Property 2 (**Generalization in Aggregation**). *Assume Λ contains $|\Lambda|$ discrete choices. Denote the dataset $\mathcal{D}_s^{-1}$ generated by replacing one sample in $\mathcal{D}_s$ with another arbitrary sample. We assume there exists $\kappa > 0$, such that $|\mathcal{L}_s(\boldsymbol{w}, \mathcal{D}_s) - \mathcal{L}_s(\boldsymbol{w}, \mathcal{D}_s^{-1})| \leq \kappa / |\mathcal{D}_s|$ for all $\boldsymbol{w}$. Given the received client models $\boldsymbol{w}_m^t$, $m \in \mathcal{M}^t$ in round t, with the probability at least $1 - \delta$, the server-side aggregations $\boldsymbol{w}_{Smart}$ and $\boldsymbol{w}_{AVG}$ of SmartFL and FedAVG satisfy*
(i) *the generalization upper bound:*

$$\mathbb{E}_{\mathcal{D}_s}\mathcal{L}_s(\boldsymbol{w}_{Smart}, \mathcal{D}_s) \leq \mathcal{L}_s(\boldsymbol{w}_{Smart}, \mathcal{D}_s) + \kappa\sqrt{\frac{\ln(2|\Lambda|/\delta)}{2|\mathcal{D}_s|}}, \tag{7}$$

(ii) *when the L2-norm is adopted, SmartFL has the generalization gap with FedAVG:*

$$\mathbb{E}_{\mathcal{D}_s}\mathcal{L}_s(\boldsymbol{w}_{Smart}, \mathcal{D}_s) \leq \mathbb{E}_{\mathcal{D}_s}\mathcal{L}_s(\boldsymbol{w}_{AVG}, \mathcal{D}_s) - \frac{\lambda}{2}\|\boldsymbol{p}^* - \tilde{\boldsymbol{\alpha}}\|^2 + \kappa\sqrt{\frac{\ln(4|\Lambda|/\delta)}{2|\mathcal{D}_s|}} + \kappa\sqrt{\frac{\ln(4/\delta)}{2|\mathcal{D}_s|}}, \tag{8}$$

where $\boldsymbol{p}^*$ is the optimum of problem (5), $\tilde{\boldsymbol{\alpha}}$ is defined in Eqn.(6), and the expectation is taken over the global data distribution since $\mathcal{D}_s$ is sampled from the global distribution $\mathcal{D}$.

Remark. *The bound in Eqn.(7) demonstrates that, in each aggregation, SmartFL can generalize well because of the extremely small set Λ, which is essentially a $|\mathcal{M}^t|$-dimension space. We can also see that this upper bound is independent of the model size. In contrast, the generalization bound of the full-space training approaches corresponds to Eqn.(7) would be $\mathcal{L}_s(\boldsymbol{w}, D_s) + \kappa\sqrt{\ln(2|\mathcal{W}|/\delta)/2|D_s|}$ with $|\mathcal{W}|$ being the number of discrete choices in the entire parameter space, which would be larger than $|\Lambda|$ by lots of orders of magnitude due to the high dimension. This verifies the superiority of SmartFL in generalization over full-space training approaches. The result in Eqn.(8) shows that SmartFL can generalize better than FedAVG by learning the weights for each client model, since $\frac{\lambda}{2}\|\boldsymbol{p}^* - \boldsymbol{\alpha}\|^2$ can dominate the last two items in Eqn.(8). These properties are consistent with our empirical results in Section 5.2.*

5 Experiments

5.1 Setup

Datasets, models, and settings. We consider four datasets: CIFAR-10/100 (Krizhevsky et al., 2009), MNIST (Deng, 2012), and 20 Newsgroup (Lang, 1995) for both computer vision and natural language processing tasks. Detailed dataset descriptions are illustrated in Appendix C. We evaluate different FL methods on the architectures of logistic regression, 2-layer ConvNet (LeCun et al., 1998), MobileNet(Howard et al., 2017), ResNet-8 (He et al., 2016) and ShuffleNet (Ma et al., 2018). For the methods involving on-server data, we randomly sample 128 training samples as unlabelled/labelled proxy data on the server by default, and the others are distributed to the clients. For other models, all the training data are distributed to clients. We evaluate the FL methods with the

Table 1: Comparison of maximum top-1 test accuracy achieved by different FL methods with ResNet-8 on CIFAR-10 in $T = 200$ communication rounds with different degrees of data heterogeneity α and participation rates C. *Methods assume the availability of unlabelled proxy data. †Methods assume the availability of labelled proxy data.

	$\alpha = 0.01$		$\alpha = 0.04$		$\alpha = 0.16$	
Method	$C = 40\%$	$C = 20\%$	$C = 40\%$	$C = 20\%$	$C = 40\%$	$C = 20\%$
FedAVG	35.77±2.82	27.00±4.09	59.29±2.43	57.68±2.65	69.02±0.83	71.57±0.20
FedProx	39.40±2.42	37.79±5.37	60.89±1.71	59.84±1.65	68.37±0.50	71.58±1.62
Scaffold	39.63±2.53	30.35±3.98	59.75±1.98	57.73±1.43	68.73±1.06	71.76±0.45
FedDF*	37.51±0.95	25.59±2.33	59.63±1.57	59.28±0.80	68.72±1.36	70.60±1.32
FedBE*	36.27±2.31	27.25±3.68	58.86±1.91	59.40±5.11	69.06±0.70	70.47±1.34
SmartFL-U*	43.37±2.06	32.88±4.00	60.94±1.18	61.18±2.65	70.38±0.58	72.10±0.36
ABAVG†	38.36±5.83	31.59±7.75	61.89±2.13	61.18±2.84	69.77±0.97	71.20±1.20
Finetuning†	46.52±3.81	34.54±6.56	60.50±0.40	60.66±1.20	69.01±0.30	71.52±0.53
SmartFL†	**53.65±2.30**	**50.13±1.66**	**63.09±0.97**	**64.73±0.46**	**70.57±0.49**	**72.12±0.15**

official test set with the global model. We give the results over three times of experiments and report mean ± standard deviation.

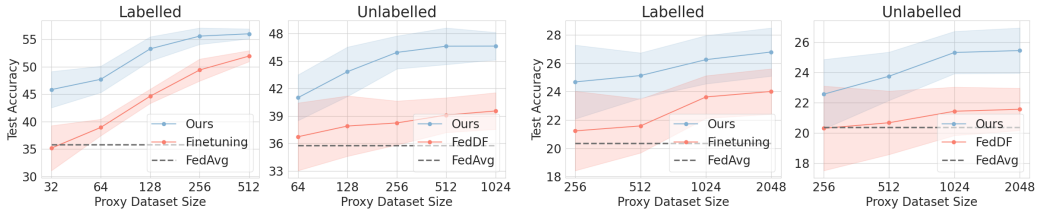
Federated learning environment. Similar to the prior studies (Gu et al., 2022; Chen & Chao, 2021b), we consider FL system with a practical number $n = 80$ clients with partial participation rate $C \in \{20\%, 40\% 60\%\}$. To simulate **non-IID data distributions** across clients, we follow prior studies (Lin et al., 2020; Chen & Chao, 2021a) to use the Dirichlet distribution to create non-IID distribution of client training data. (Hsu et al., 2019) The parameter α controls the degree of non-IIDness. The smaller the value of α, the partition is closer to that one client only holds samples from a single class. Overall, we consider various non-IID degrees with $\alpha \in \{0.01, 0.04, 0.1, 0.16, 0.32, 0.64, 1\}$. For the studies involving **poisoning attacks**, we consider two kinds of attacks, including Label Flip Attack (Fung et al., 2018) and Omniscient Attack (Blanchard et al., 2017), which represent the data poisoning attack and model poisoning attack for FL, respectively. Specifically, Label Flip Attack switches the label to be the next class of the ground truth, while Omniscient Attack negates the original benign gradients.

Baselines. We consider both state-of-the-art solutions against non-IID data distribution and poisoning attacks. For the studies on **robustness against non-IID distribution without poisoning attacks**, we include 1) without proxy data: FedAVG (McMahan et al., 2017), FedProx (Li et al., 2020), Scaffold (Karimireddy et al., 2020), 2) leveraging unlabelled proxy data (i.e., FedDF (Lin et al., 2020) and FedBE (Chen & Chao, 2021a)), 3) leveraging labelled proxy data(i.e., ABAVG (Xiao et al., 2021)) and full-space Finetuning with labelled proxy data. For the studies on **robustness against poisoning attacks under different scenarios**, besides the applicable ones of the mentioned solutions, we further include Median (Yin et al., 2018), Krum (Blanchard et al., 2017), and Trimmed Mean (Yin et al., 2018), and the state-of-the-art defense with the availability of labelled proxy data, i.e., Sageflow (Park et al., 2021) and FLTrust (Cao et al., 2021). Detailed descriptions and settings of baselines are shown in Appendix C.

Detailed local training setting and server aggregation setting. For the local training of all the models, we use the fixed learning rate of $\eta = 10^{-3}$ and the batch size of 32 with Adam optimizer (Kingma & Ba, 2014). Local training epoch E is set to 1, and the total round is 200. For the on-server optimization of our method, we use the batch size of 32 and Adam optimizer, fix $E_s = 20$, and $\eta_s = 1e-2$ for **SmartFL** with labelled data, $\eta_s = 5e-4$ for **SmartFL-U** with unlabelled data by default. More detailed hyperparameter settings are demonstrated in Appendix C.

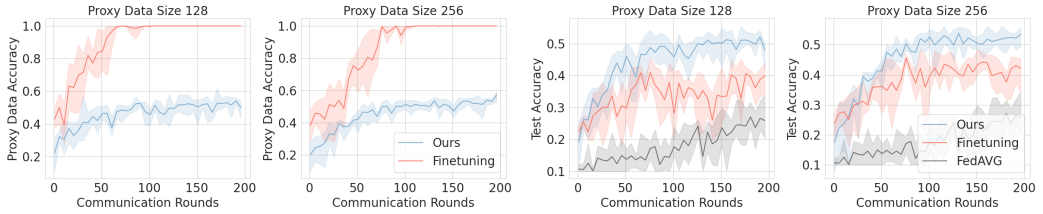
5.2 Robustness against non-IID distribution of data

Performance overview for different scenarios. We evaluate the performance of SmartFL on a widely used benchmark of image classification on CIFAR-10 under various scenarios. Table 1 summarizes the results. Our observations are as follows: First, FedAVG suffers from significant performance degradation when the data distribution is highly non-IID, and the advanced training technique FedProx and Scaffold can alleviate the problem to some extent. Second, leveraging a practical amount of server proxy data with advanced aggregation strategies can further improve the performance in most cases, indicating the potential of improving aggregation with reasonable



(a) CIFAR-10 (b) CIFAR-100

Figure 1: Effects of **the amount of server data**. We compare our method with Finetuning and FedDF with labelled and unlabelled server data, respectively. The horizontal axis is the size of the server data, while the vertical axis is the test accuracy. We can see that our method shows superior performances given different amounts of server data in both scenarios.



(a) Proxy Data Accuracy (b) Test Accuracy

Figure 2: **Generalization ability with a small amount of server data**. We compare SmartFL with Finetuning, which directly finetune the model parameters with the proxy data in every communication round. We can observe that Finetuning quickly reaches 100% accuracy on the server data, while our method prevents overfitting and consistently demonstrate better test performance.

server knowledge. Third, for both data availability settings of labelled and unlabelled data, SmartFL and SmartFL-U consistently outperform the full-space training tuning counterpart, i.e., Finetuning and FedDF, as well as the advanced ensemble solution FedBE and heuristic reweighting solution ABAVG by a noticeable margin under various non-IIDness and participation rate settings. Moreover, we further demonstrate in Appendix D.1.1 that SmartFL greatly accelerates convergence and requires much fewer communication rounds to achieve the target accuracy. More experiment on **20newsgroup** are shown in Appendix D.1.2. Overall, the results indicate that SmartFL effectively improves the robustness of server-side aggregation against non-IID data distribution.

In-depth Analysis. We investigate the **effect of the amount of server data** on CIFAR-10/100 with ResNet-8, under the high level of heterogeneity with $\alpha = 0.01$ and $C = 40\%$. For unlabelled/labelled data, we compare SmartFL/SmartFL-U with FedAVG and full-space training counterpart FedDF/Finetuning. As shown in Figure 1, with a reasonable amount of proxy labelled/unlabelled data, all the optimization strategies outperform FedAVG and benefit from the increase of available data. Our solution consistently outperforms the full-space training counterparts in two datasets for both labelled and unlabelled data. This aligns with our intuition that a limited amount of proxy data can not well supervise the learning of a deep learning model with massive parameters, while our extracted subspace effectively solves the problem and enables taking advantage of even a small amount of data.

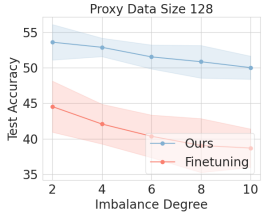
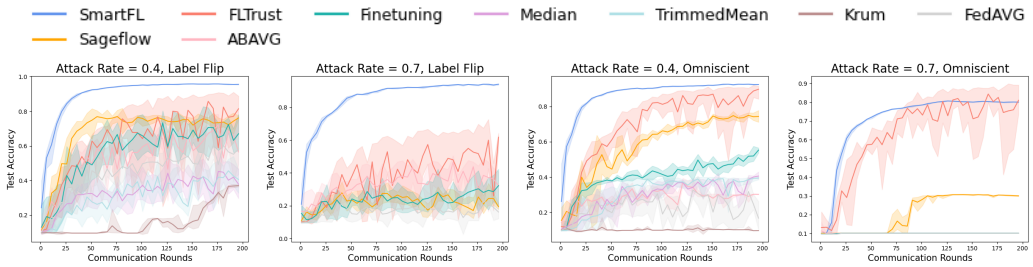
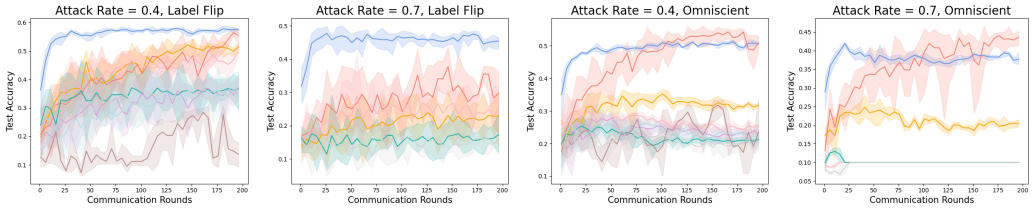


Figure 3: Comparison of effects of **biased server proxy data** on SmartFL and Finetuning.

We then empirically demonstrate the **generalization ability with a small amount of server data** by comparing SmartFL with the full-space training counterpart, i.e., Finetuning, on CIFAR-10 with ResNet-8 and $\alpha = 0.01, C = 40\%$. As shown in Figure 2, for the Finetuning approach, even though we only finetune the aggregated model for one epoch on the server at each round to try to eliminate overfitting, the accuracy calculated over the proxy data still converges to 100% after multiple rounds, while the test accuracy does not boost significantly. On the other hand, though our method does not achieve perfect proxy data accuracy, its test performance consistently surpasses the Finetuning, which verifies its robustness against overfitting. Moreover, Figure 3 shows the performance with various **class imbalance degree of server proxy data** under the same setting (the imbalance degree is calculated with the maximum class sample number divided by the minimum class sample



(a) MNIST



(b) CIFAR-10

Figure 4: Defence against attacks on MNIST/CIFAR-10 with the degree of data heterogeneity $\alpha = 0.01$ and $\alpha = 0.1$, respectively, under different types of attacks (Label Flip and Omniscient Attack) and different attack rates $AR = 0.4/0.7$.

number). SmartFL demonstrates strong robustness on highly-imbalanced proxy data. Overall, the results verify that by optimizing the coefficient $\boldsymbol{p}$, which has a much lower dimension than the model parameters $\boldsymbol{w}$, our method is less prone to overfitting the proxy data and more robust to imbalanced proxy data.

5.3 Robustness against attacks

We demonstrate the robustness of our solution against Label Flip Attack and Omniscient Attack in various scenarios. We experiment on CIFAR 10 with $\alpha \in \{0.1, 1\}, C = 60\%$ and model ResNet-8, and MNIST with $\alpha \in \{0.01, 1\}, C = 60\%$ and model 2-layer ConvNet, and consider attack rate $AR \in \{0.2, 0.3, 0.4, 0.7\}$. Figure 4 demonstrates the server test accuracy in the federated learning process with attack rate $AR = 0.4/0.7$ and high data heterogeneity. More results in all the scenarios are shown in Appendix D.2. For Attack Rate $= 0.7$, the classical defenses against attack are not applicable because their assumption that less than half of the clients are malicious does not hold. We have the following observations. First, weight-based statistical solutions generally cannot perform well when the data distribution is highly non-IID, which is in line with prior studies (Karimireddy et al., 2022), indicating the potential to leverage additional server knowledge to further improve the robustness. Second, on-server full-space training after performing FedAVG, i.e., Finetuning, is hard to dilute the influence of poisoned models with a small amount of server data. Third, the state-of-the-art methods using labelled proxy data (i.e., Sageflow and FLTrust) show a relatively good performance defending against both attacks but still suffer from unstable learning and some failure cases. Finally, SmartFL yields stable and good performance against various attacks in different scenarios, indicating the effectiveness of mitigating negative effects from malicious clients through subspace training.

6 Conclusion and Discussions

Data heterogeneity across clients and poisoning attacks are among the main bottlenecks for robust server-side aggregation. In this work, we propose SmartFL, which optimizes the aggregation to overcome both challenges by subspace training. We extract a reduced subspace spanned by the clients' models to achieve effective and efficient optimization of the global model in every communication round with a small amount of proxy data. We provide theoretical analysis for SmartFL on convergence and generalization ability. Extensive experiments demonstrate the state-of-the-art performance of SmartFL for both FL with non-IID data distribution and FL with poisoning attacks. We involve more discussions in Appendix E.

7 REPRODUCIBILITY STATEMENT

We use the framework of the prior work (Gu et al., 2022) (https://github.com/fedl-repo/fedaux). All the experiments in the paper are based on public datasets. The hyperparameters used to reproduce our methods and baselines are provided in Section 5.1 and Appendix C. We provide the source code with a demo script in the supplementary material.

REFERENCES

Durmus Alp Emre Acar, Yue Zhao, Ramon Matas Navarro, Matthew Mattina, Paul N. Whatmough, and Venkatesh Saligrama. Federated learning based on dynamic regularization. In *ICLR*, 2021.

Jeremy Bernstein, Jiawei Zhao, Kamyar Azizzadenesheli, and Anima Anandkumar. signsgd with majority vote is communication efficient and fault tolerant. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019.

Peva Blanchard, El Mahdi El Mhamdi, Rachid Guerraoui, and Julien Stainer. Machine learning with adversaries: Byzantine tolerant gradient descent. *Advances in Neural Information Processing Systems*, 30, 2017.

Xiaoyu Cao, Minghong Fang, Jia Liu, and Neil Zhenqiang Gong. Fltrust: Byzantine-robust federated learning via trust bootstrapping. In *28th Annual Network and Distributed System Security Symposium, NDSS 2021, virtually, February 21-25, 2021*. The Internet Society, 2021.

Hong-You Chen and Wei-Lun Chao. Fedbe: Making bayesian model ensemble applicable to federated learning. In *ICLR*, 2021a.

Hong-You Chen and Wei-Lun Chao. On bridging generic and personalized federated learning for image classification. In *International Conference on Learning Representations*, 2021b.

Yae Jee Cho, Andre Manoel, Gauri Joshi, Robert Sim, and Dimitrios Dimitriadis. Heterogeneous ensemble knowledge transfer for training large models in federated learning. In *IJCAI*, 2022.

Li Deng. The mnist database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012.

Felix Draxler, Kambis Veschgini, Manfred Salmhofer, and Fred Hamprecht. Essentially no barriers in neural network energy landscape. In *International conference on machine learning*, pp. 1309–1318. PMLR, 2018.

Shuhao Fu, Chulin Xie, Bo Li, and Qifeng Chen. Attack-resistant federated learning with residual-based reweighting. In *AAAI Workshops*, 2021.

Clement Fung, Chris JM Yoon, and Ivan Beschastnikh. Mitigating sybils in federated learning poisoning. *arXiv preprint arXiv:1808.04866*, 2018.

Timur Garipov, Pavel Izmailov, Dmitrii Podoprikhin, Dmitry P Vetrov, and Andrew G Wilson. Loss surfaces, mode connectivity, and fast ensembling of dnns. *Advances in neural information processing systems*, 31, 2018.

Frithjof Gressmann, Zach Eaton-Rosen, and Carlo Luschi. Improving neural network training in low dimensional random bases. *Advances in Neural Information Processing Systems*, 33:12140–12150, 2020.

Hang Gu, Bin Guo, Jiangtao Wang, Wen Sun, Jiaqi Liu, Sicong Liu, and Zhiwen Yu. Fedaux: An efficient framework for hybrid federated learning. In *IEEE International Conference on Communications, ICC 2022, Seoul, Korea, May 16-20, 2022*, pp. 195–200. IEEE, 2022. doi: 10.1109/ICC45855.2022.9839129.

Rachid Guerraoui, Sébastien Rouault, et al. The hidden vulnerability of distributed learning in byzantium. In *International Conference on Machine Learning*, pp. 3521–3530. PMLR, 2018.

Guy Gur-Ari, Daniel A Roberts, and Ethan Dyer. Gradient descent happens in a tiny subspace. *arXiv preprint arXiv:1812.04754*, 2018.

Filip Hanzely, Slavomír Hanzely, Samuel Horváth, and Peter Richtárik. Lower bounds and optimal algorithms for personalized federated learning. *Advances in Neural Information Processing Systems*, 33:2304–2315, 2020.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

Tzu-Ming Harry Hsu, Hang Qi, and Matthew Brown. Measuring the effects of non-identical data distribution for federated visual classification. *arXiv preprint arXiv:1909.06335*, 2019.

Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank J Reddi, Sebastian U Stich, and Ananda Theertha Suresh. Scaffold: Stochastic controlled averaging for on-device federated learning. In *ICML*, 2020.

Sai Praneeth Karimireddy, Lie He, and Martin Jaggi. Byzantine-robust learning on heterogeneous datasets via bucketing. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022.

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

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.

Rohith Kuditipudi, Xiang Wang, Holden Lee, Yi Zhang, Zhiyuan Li, Wei Hu, Rong Ge, and Sanjeev Arora. Explaining landscape connectivity of low-cost solutions for multilayer nets. *Advances in neural information processing systems*, 32, 2019.

Viraj Kulkarni, Milind Kulkarni, and Aniruddha Pant. Survey of personalization techniques for federated learning. In *2020 Fourth World Conference on Smart Trends in Systems, Security and Sustainability (WorldS4)*, pp. 794–797. IEEE, 2020.

Ken Lang. Newsweeder: Learning to filter netnews. In *Machine Learning Proceedings 1995*, pp. 331–339. Elsevier, 1995.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Chunyuan Li, Heerad Farkhoor, Rosanne Liu, and Jason Yosinski. Measuring the intrinsic dimension of objective landscapes. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018.

Daliang Li and Junpu Wang. Fedmd: Heterogenous federated learning via model distillation. *arXiv preprint arXiv:1910.03581*, 2019.

Liping Li, Wei Xu, Tianyi Chen, Georgios B Giannakis, and Qing Ling. Rsa: Byzantine-robust stochastic aggregation methods for distributed learning from heterogeneous datasets. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pp. 1544–1551, 2019.

Qinbin Li, Bingsheng He, and Dawn Song. Model-contrastive federated learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10713–10722, 2021a.

Tao Li, Lei Tan, Zhehao Huang, Qinghua Tao, Yipeng Liu, and Xiaolin Huang. Low dimensional trajectory hypothesis is true: Dnns can be trained in tiny subspaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Tao Li, Yingwen Wu, Sizhe Chen, Kun Fang, and Xiaolin Huang. Subspace adversarial training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13409–13418, 2022.

Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. In *MLSys*, 2020.

Tian Li, Shengyuan Hu, Ahmad Beirami, and Virginia Smith. Ditto: Fair and robust federated learning through personalization. In *International Conference on Machine Learning*, pp. 6357–6368. PMLR, 2021b.

Tao Lin, Lingjing Kong, Sebastian U Stich, and Martin Jaggi. Ensemble distillation for robust model fusion in federated learning. *Advances in Neural Information Processing Systems*, 33:2351–2363, 2020.

Ningning Ma, Xiangyu Zhang, Hai-Tao Zheng, and Jian Sun. Shufflenet v2: Practical guidelines for efficient cnn architecture design. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 116–131, 2018.

H Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, et al. Communication-efficient learning of deep networks from decentralized data. In *AISTATS*, 2017.

Seungeun Oh, Jihong Park, Eunjeong Jeong, Hyesung Kim, Mehdi Bennis, and Seong-Lyun Kim. Mix2fld: Downlink federated learning after uplink federated distillation with two-way mixup. *IEEE Communications Letters*, 24(10):2211–2215, 2020.

Jungwuk Park, Dong-Jun Han, Minseok Choi, and Jaekyun Moon. Sageflow: Robust federated learning against both stragglers and adversaries. In Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pp. 840–851, 2021.

Krishna Pillutla, Sham M Kakade, and Zaid Harchaoui. Robust aggregation for federated learning. *arXiv preprint arXiv:1912.13445*, 2019.

Boris T Polyak. Gradient methods for solving equations and inequalities. *USSR Computational Mathematics and Mathematical Physics*, 4(6):17–32, 1964.

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.

Virat Shejwalkar and Amir Houmansadr. Manipulating the byzantine: Optimizing model poisoning attacks and defenses for federated learning. In *NDSS*, 2021.

Virat Shejwalkar, Amir Houmansadr, Peter Kairouz, and Daniel Ramage. Back to the drawing board: A critical evaluation of poisoning attacks on production federated learning. In *2022 IEEE Symposium on Security and Privacy (SP)*, pp. 1354–1371, 2022. doi: 10.1109/SP46214.2022.9833647.

MyungJae Shin, Chihoon Hwang, Joongheon Kim, Jihong Park, Mehdi Bennis, and Seong-Lyun Kim. Xor mixup: Privacy-preserving data augmentation for one-shot federated learning. *arXiv preprint arXiv:2006.05148*, 2020.

Canh T Dinh, Nguyen Tran, and Josh Nguyen. Personalized federated learning with moreau envelopes. *Advances in Neural Information Processing Systems*, 33:21394–21405, 2020.

Oriol Vinyals and Daniel Povey. Krylov subspace descent for deep learning. In *Artificial intelligence and statistics*, pp. 1261–1268. PMLR, 2012.

Martin J Wainwright. *High-dimensional statistics: A non-asymptotic viewpoint*, volume 48. Cambridge University Press, 2019.

Ching Pui Wan and Qifeng Chen. Robust federated learning with attack-adaptive aggregation. In *IJCAI Workshops*, 2021.

Jianyu Wang, Qinghua Liu, Hao Liang, Gauri Joshi, and H Vincent Poor. Tackling the objective inconsistency problem in heterogeneous federated optimization. *Advances in neural information processing systems*, 33:7611–7623, 2020.

Jianhang Xiao, Chunhui Du, Zijing Duan, and Wei Guo. A novel server-side aggregation strategy for federated learning in non-iid situations. In *2021 20th International Symposium on Parallel and Distributed Computing (ISPDC)*, pp. 17–24. IEEE, 2021.

Cong Xie, Sanmi Koyejo, and Indranil Gupta. Zeno: Distributed stochastic gradient descent with suspicion-based fault-tolerance. In *International Conference on Machine Learning*, pp. 6893–6901. PMLR, 2019.

Cong Xie, Sanmi Koyejo, and Indranil Gupta. Zeno++: Robust fully asynchronous sgd. In *International Conference on Machine Learning*, pp. 10495–10503. PMLR, 2020.

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.

Yousef Yeganeh, Azade Farshad, Nassir Navab, and Shadi Albarqouni. Inverse distance aggregation for federated learning with non-iid data. In *Domain Adaptation and Representation Transfer, and Distributed and Collaborative Learning*, pp. 150–159. Springer, 2020.

Dong Yin, Yudong Chen, Ramchandran Kannan, and Peter Bartlett. Byzantine-robust distributed learning: Towards optimal statistical rates. In *International Conference on Machine Learning*, pp. 5650–5659. PMLR, 2018.

Tehrim Yoon, Sumin Shin, Sung Ju Hwang, and Eunho Yang. Fedmix: Approximation of mixup under mean augmented federated learning. In *ICLR*, 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.

Fan Zhou and Guojing Cong. On the convergence properties of a k-step averaging stochastic gradient descent algorithm for nonconvex optimization. *arXiv preprint arXiv:1708.01012*, 2017.

Xiao Zhou, Weizhong Zhang, Zonghao Chen, Shizhe Diao, and Tong Zhang. Efficient neural network training via forward and backward propagation sparsification. *Advances in Neural Information Processing Systems*, 34:15216–15229, 2021.

Martin Zinkevich, Markus Weimer, Lihong Li, and Alex J Smola. Parallelized stochastic gradient descent. In *NIPS*, 2010.

SUPPLEMENTARY MATERIAL

- Appendix A: additional related work (cf. section 2 of the main paper).
- Appendix B: proof and additional analysis (cf. subsection 4.2 of the main paper).
- Appendix C: additional details of experimental setups (cf. subsection 5.1 of the main paper).
- Appendix D: additional experimental results and analysis (cf. subsection 5.2 and subsection 5.3 of the main paper).
- Appendix E: additional discussions (cf. section 6 of the main paper).

A ADDITIONAL RELATED WORK

A.1 TRAINING IN SUBSPACE

Several prior studies (Li et al., 2018; Gur-Ari et al., 2018; Vinyals & Povey, 2012) uncover the low-dimensionality essence in training neural networks, laying the foundation for the research on training in subspace. Li et al. (2018) first proposes to train networks in a smaller, randomly oriented subspace and demonstrate that the required dimension is much lower than the original dimension of parameters to obtain a relatively good performance. Afterward, Gressmann et al. (2020) proposes re-drawing the random subspace during training to improve the performance. Recently, Li et al. improves the random-oriented subspace by analyzing the optimization trajectory, and verifies that a carefully-extracted 40-dimensional space is enough to achieve comparable performance to regular training. The following study (Li et al., 2022) applies subspace training in adversarial training problems to prevent overfitting. In our work, we take advantage of the efficiency and generalization of subspace training to optimize server-side aggregation. We leverage prior knowledge on aggregation for FL to construct the subspace as the convex hull spanned by client models.

A.2 FEDERATED LEARNING WITH NON-I.I.D. DATA DISTRIBUTION

In this section, we supplement the other line of solutions discussed in the main paper for heterogeneous FL, i.e., **modifying local training and inference.** Multiple branches of solutions are proposed to solve non-i.i.d data distribution through modifying local training and inference process. Several solutions propose to mitigate client drift through *regularing local training*. FedPROX (Li et al., 2020) and FedDYN (Acar et al., 2021) propose to regularize the drift of local model with global model. MOON (Li et al., 2021a) introduces a contrastive loss and SCAFFOLD (Karimireddy et al., 2020) introduces control variates to correct local gradients. *Data sharing or augmentation* based solutions (Shin et al., 2020; Oh et al., 2020; Yoon et al., 2021; Zhao et al., 2018) approach the problem from the data perspective and add to some shared/augmented data in local training to alleviate data heterogeneity. *Personalized FL* (Kulkarni et al., 2020; T Dinh et al., 2020; Hanzely et al., 2020; Li et al., 2021b) is also a branch of solutions that modify the local inference process. Instead of training a global model, these approaches seek to find the best local model, and the evaluation is performed locally. Recently, a work (Chen & Chao, 2021b) proposes to bridge generic FL and personalized FL to improve performance.

B PROOF AND ADDITIONAL ANALYSIS

B.1 PROOF OF PROPERTY 1

We prove the property 1 with the following definitions and assumptions, which are widely adopted in the existing related studies (Xie et al., 2019; 2020; Park et al., 2021).

Definition 1 (L-smoothness)**.** *We say a differentiable $f(\boldsymbol{w})$ L-smoothness if there exists $L > 0$ such that*

$$f(\boldsymbol{v}) - f(\boldsymbol{w}) \leq \langle \nabla f(\boldsymbol{w}), \boldsymbol{v} - \boldsymbol{w} \rangle + \frac{L}{2} \|\boldsymbol{v} - \boldsymbol{w}\|^2, \forall \boldsymbol{w}, \boldsymbol{v}.$$

Definition 2 (Polyak-Łojasiewicz (PL) Inequality(Polyak, 1964))**.** *A function $f(\boldsymbol{w})$ satisfies the Polyak-Łojasiewicz (PL) inequality if there exists a constant $\mu > 0$, such that*

$$f(\boldsymbol{w}) - f(\boldsymbol{w}^*) \leq \frac{1}{2\mu}\|\nabla f(\boldsymbol{w})\|^2, \forall \boldsymbol{w},$$

where $\boldsymbol{w}^*$ is the minimum of $f(\boldsymbol{w})$.

Assumption 1. *We assume in each iteration t, there exists at least one honest client i_t among the M clients, who return the local models, in a sense that*

$$\langle \nabla \mathcal{L}_s(\boldsymbol{w}^t, \mathcal{D}_s), \Delta_{i_t}^t \rangle + \gamma \|\nabla \mathcal{L}_s(\boldsymbol{w}^t, \mathcal{D}_s)\|^2 \leq \epsilon,$$

where $\gamma > 0$ and $\epsilon > 0$ are two constants.

Remark. *Assumption 1 is practical and it is adopted in attack-robust studies (Xie et al., 2019; 2020). It means that $\mathcal{L}_s(\boldsymbol{w}^t, \mathcal{D}_s)$ can be reduced a little by involving $\Delta_{i_t}^t$ into $\boldsymbol{w}^t$. If it is not satisfied in some extreme round, we can skip it and wait for the next communication round.*

Assumption 2. *Given the client models $\boldsymbol{w}_1, \ldots, \boldsymbol{w}_M$, we assume $|\mathcal{L}(\boldsymbol{p}, \mathcal{D}) - \mathcal{L}_s(\boldsymbol{p}, \mathcal{D}_s)| < \delta/2$ holds for a small constant $\delta > 0$.*

Remark. *Note that $\mathcal{L}(\boldsymbol{p}, \mathcal{D}) = \frac{1}{|\mathcal{D}|}\sum_{\xi \in \mathcal{D}} \ell(\boldsymbol{w}, \xi)$, and $\mathcal{L}_s(\boldsymbol{p}, \mathcal{D}_s) = \frac{1}{|\mathcal{D}_s|}\sum_{\xi \in \mathcal{D}_s} \ell(\boldsymbol{w}, \xi)$, where $\boldsymbol{w}$ is the weighted average of the given client models with coeffecient $\boldsymbol{p}$. Also, as mentioned in Section 4, we assume $\mathcal{D}_s$ is sampled from global distribution $\mathcal{D}$. As our $\boldsymbol{p}$ has a low dimension, $\mathcal{L}(\boldsymbol{p}, \mathcal{D})$ can be approximated by $\mathcal{L}_s(\boldsymbol{p}, \mathcal{D}_s)$ with a small subset $\mathcal{D}_s$.*

Then, we would like to rephrase Property 1 into a more formal form below:

Property 3. *Besides Assumptions 1 and 2, we assume the losses $\mathcal{L}(\boldsymbol{w}^t, \mathcal{D})$ and $\mathcal{L}_s(\boldsymbol{w}^t, \mathcal{D}_s)$ are L-smoothness and satisfy the PL inequality (potentially non-convex). Further, for the true and stochastic gradients, we assume that $\|\nabla \mathcal{L}(\boldsymbol{w}^t, \mathcal{D})\|^2 \leq V_1, V_2 \leq \|\nabla \mathcal{L}_s(\boldsymbol{w}^t, \mathcal{D}_s)\|^2 \leq V_1$ and $\|\nabla \mathcal{L}_s(\boldsymbol{w}^t, \mathcal{D}_s) - \nabla \mathcal{L}(\boldsymbol{w}, \mathcal{D})\|^2 \leq V_3$ with $V_1 \geq V_2 > 0$ and $V_3 > 0$. Then, for our SmartFL, we have*

$$\begin{aligned}\mathbb{E}\left[\mathcal{L}(\boldsymbol{w}^T, \mathcal{D}) - \mathcal{L}(\boldsymbol{w}^*, \mathcal{D})\right] \leq & (1 - 2\mu\gamma\frac{V_2}{V_1})^T \mathbb{E}\left[\mathcal{L}(\boldsymbol{w}^0, \mathcal{D}) - \mathcal{L}(\boldsymbol{w}^*, \mathcal{D})\right] \\ & + \frac{V_1}{2\mu\gamma V_2}\left[\eta\left(\frac{1}{2}V_3 + \frac{L+1}{2}V_1\right) + \epsilon + \delta\right],\end{aligned}$$

where $\eta < \min(1, 1/L)$.

Proof. of Property 3:
Denote the honest client in iteration t to be i_t and from Assumption 1, we have

$$\langle \nabla \mathcal{L}_s(\boldsymbol{w}^t, \mathcal{D}_s), \Delta_{i_t}^t \rangle \leq -\gamma \|\nabla \mathcal{L}_s(\boldsymbol{w}^t, \mathcal{D}_s)\|^2 + \epsilon.$$

Thus, we can have

$$\begin{aligned}&\langle \nabla \mathcal{L}(\boldsymbol{w}^t, \mathcal{D}), \Delta_{i_t}^t \rangle \\ &\leq \langle \nabla \mathcal{L}(\boldsymbol{w}^t, \mathcal{D}) - \nabla \mathcal{L}_s(\boldsymbol{w}^t, \mathcal{D}_s), \Delta_{i_t}^t \rangle - \gamma \|\nabla \mathcal{L}_s(\boldsymbol{w}^t, \mathcal{D}_s)\|^2 + \epsilon \\ &\leq \langle \nabla \mathcal{L}(\boldsymbol{w}^t, \mathcal{D}) - \nabla \mathcal{L}_s(\boldsymbol{w}^t, \mathcal{D}_s), \Delta_{i_t}^t \rangle - \gamma \frac{V_2}{V_1} \|\nabla \mathcal{L}(\boldsymbol{w}^t, \mathcal{D})\|^2 + \epsilon \\ &\leq \frac{\eta_{i_t}}{2}\|\nabla \mathcal{L}(\boldsymbol{w}^t, \mathcal{D}) - \nabla \mathcal{L}_s(\boldsymbol{w}^t, \mathcal{D}_s)\|^2 + \frac{\eta_{i_t}}{2}\|\nabla \mathcal{L}_s(\boldsymbol{w}^t, \mathcal{D}_s)\|^2 - \gamma \frac{V_2}{V_1} \|\nabla \mathcal{L}(\boldsymbol{w}^t, \mathcal{D})\|^2 + \epsilon \\ &\leq -\gamma \frac{V_2}{V_1} \|\nabla \mathcal{L}(\boldsymbol{w}^t, \mathcal{D})\|^2 + \frac{\eta_{i_t}}{2} V_1 + \frac{\eta_{i_t}}{2} V_3 + \epsilon.\end{aligned}$$

where $\eta_{i_t} = \|\Delta_{i_t}^t\| / \|\nabla \mathcal{L}_s(\boldsymbol{w}^t, \mathcal{D}_s)\|$, which can be controlled by tuning the learning rate and length of local training. Therefore, we can assume $\eta_{i_t} \leq \eta < \min(1, 1/L)$.

Notice that in our server-side aggregation, we search the model fusion in the convex hull spanned by the received client models, which contains these models. Therefore, we have

$$\mathcal{L}_s(\boldsymbol{w}^{t+1}, \mathcal{D}_s) \leq \mathcal{L}_s(\boldsymbol{w}_{i_t}^t, \mathcal{D}_s).$$

From Assumption 2, we can get

$$\mathcal{L}(\boldsymbol{w}^{t+1}, \mathcal{D}) \leq \mathcal{L}_s(\boldsymbol{w}^{t+1}, \mathcal{D}_s) + \delta/2 \leq \mathcal{L}_s(\boldsymbol{w}_{i_t}^t, \mathcal{D}_s) + \delta/2 \leq \mathcal{L}(\boldsymbol{w}_{i_t}^t, \mathcal{D}) + \delta.$$

According to the smoothness of $\mathcal{L}(\boldsymbol{w}, \mathcal{D})$, we can get

$$\begin{aligned}
&\mathbb{E}\left[\mathcal{L}(\boldsymbol{w}^{t+1}, \mathcal{D}) - \mathcal{L}(\boldsymbol{w}^*, \mathcal{D})\right] \\
&\leq \mathbb{E}\left[\mathcal{L}(\boldsymbol{w}_{i_t}^t, \mathcal{D}) - \mathcal{L}(\boldsymbol{w}^*, \mathcal{D})\right] + \delta \\
&\leq \mathbb{E}\left[\mathcal{L}(\boldsymbol{w}^t, \mathcal{D}) - \mathcal{L}(\boldsymbol{w}^*, \mathcal{D}) + \langle \nabla \mathcal{L}(\boldsymbol{w}^t, \mathcal{D}), \Delta_{i_t}^t \rangle + \frac{L}{2}\|\Delta_{i_t}^t\|^2\right] + \delta \\
&\leq \mathbb{E}\left[\mathcal{L}(\boldsymbol{w}^t, \mathcal{D}) - \mathcal{L}(\boldsymbol{w}^*, \mathcal{D}) - \gamma \frac{V_2}{V_1}\|\nabla \mathcal{L}(\boldsymbol{w}^t, \mathcal{D})\|^2\right] + \frac{\eta}{2} V_3 + \frac{L+1}{2}\eta V_1 + \eta\epsilon + \delta \\
&\leq (1 - 2\mu\gamma\frac{V_2}{V_1})\mathbb{E}\left[\mathcal{L}_s(\boldsymbol{w}^t, \mathcal{D}_s) - \mathcal{L}_s(\boldsymbol{w}^*, \mathcal{D}_s)\right] + \eta\left(\frac{1}{2}V_3 + \frac{L+1}{2}V_1\right) + \epsilon + \delta.
\end{aligned}$$

Hence, for the model after T aggregations, we can have

$$\begin{aligned}
\mathbb{E}\left[\mathcal{L}(\boldsymbol{w}^T, \mathcal{D}) - \mathcal{L}(\boldsymbol{w}^*, \mathcal{D})\right] \leq &(1 - 2\mu\gamma\frac{V_2}{V_1})^T \mathbb{E}\left[\mathcal{L}(\boldsymbol{w}^0, \mathcal{D}) - \mathcal{L}(\boldsymbol{w}^*, \mathcal{D})\right] \\
&+ \frac{V_1}{2\mu\gamma V_2}\left[\eta\left(\frac{1}{2}V_3 + \frac{L+1}{2}V_1\right) + \epsilon + \delta\right]
\end{aligned}$$

By choosing an appropriate γ satisfying $0 < 1 - 2\mu\gamma\frac{V_2}{V_1} < 1$, the expected error can converge linearly. □

B.2 Proof of Property 2

Proof. (i) Eqn.(7) can be obtained immediately from the bounded difference inequality (Corollary 2.21 of (Wainwright, 2019)).

(ii) As the L2 regularization is adopted, from the server-side aggregation of SmartFL and note that $\tilde{\boldsymbol{\alpha}} \in \Lambda$, we have

$$\mathcal{J}(\boldsymbol{p}^*) \leq \mathcal{J}(\tilde{\boldsymbol{\alpha}}).$$

That is

$$\mathcal{L}_s(\boldsymbol{p}^*, \mathcal{D}_s) + \lambda \cdot \frac{1}{2}\|\boldsymbol{p}^* - \tilde{\boldsymbol{\alpha}}\|_2^2 \leq \mathcal{L}_s(\tilde{\boldsymbol{\alpha}}, \mathcal{D}_s),$$

which is equivalent to

$$\mathcal{L}_s(\boldsymbol{w}_{Smart}, \mathcal{D}_s) + \lambda \cdot \frac{1}{2}\|\boldsymbol{p}^* - \tilde{\boldsymbol{\alpha}}\|_2^2 \leq \mathcal{L}_s(\boldsymbol{w}_{AVG}, \mathcal{D}_s),$$

Then, similarly with (i), we have, with probability at least $1 - \delta/2$,

$$\begin{aligned}
&\mathbb{E}_{\mathcal{D}_s}\mathcal{L}_s(\boldsymbol{w}_{Smart}, \mathcal{D}_s) \\
&\leq \mathcal{L}_s(\boldsymbol{w}_{Smart}, \mathcal{D}_s) + \kappa\sqrt{\frac{\ln(4|\Lambda|/\delta)}{2|\mathcal{D}_s|}} \\
&\leq \mathcal{L}_s(\boldsymbol{w}_{AVG}, \mathcal{D}_s) - \frac{1}{2}\|\boldsymbol{p}^* - \tilde{\boldsymbol{\alpha}}\|_2^2 + \kappa\sqrt{\frac{\ln(4|\Lambda|/\delta)}{2|\mathcal{D}_s|}}.
\end{aligned} \tag{9}$$

For $\boldsymbol{w}_{AVG}$, using bounded difference inequality, we have, with probability at least $1 - \delta/2$,

$$\mathcal{L}_s(\boldsymbol{w}_{AVG}, \mathcal{D}_s) \leq \mathbb{E}_{\mathcal{D}_s}\mathcal{L}_s(\boldsymbol{w}_{AVG}, \mathcal{D}_s) + \kappa\sqrt{\frac{\ln(4/\delta)}{2|\mathcal{D}_s|}}. \tag{10}$$

Combine the equations (9) and (10), we can get, with probability at least $1 - \delta$,

$$\mathbb{E}_{\mathcal{D}_s}\mathcal{L}_s(\boldsymbol{w}_{Smart}, \mathcal{D}_s) \leq \mathbb{E}_{\mathcal{D}_s}\mathcal{L}_s(\boldsymbol{w}_{AVG}, \mathcal{D}_s) - \frac{\lambda}{2}\|\boldsymbol{p}^* - \tilde{\boldsymbol{\alpha}}\|^2 + \kappa\sqrt{\frac{\ln(4|\Lambda|/\delta)}{2|\mathcal{D}_s|}} + \kappa\sqrt{\frac{\ln(4/\delta)}{2|\mathcal{D}_s|}}. \tag{11}$$

□

C Detailed Experiments Setups

C.1 Dataset

CIFAR-10/100 (Krizhevsky et al., 2009) contain 50K training and 10K testing images for 10/100 class. MNIST (Deng, 2012) includes 60K training and 10K testing samples of written digits. The 20newsgroups (Lang, 1995) text dataset comprises around 20K news documents belonging to 20 categories, and it is split into 18K documents for training and 2000 documents for testing.

C.2 Baselines

- FedAVG (McMahan et al., 2017): The standard communication-efficient aggregation strategy for federated learning.
- FedPROX (Li et al., 2020): An advanced method for heterogeneous federated learning technique that regularizes the drift of local model with the global model.
- Scaffold (Karimireddy et al., 2020): An advanced method for heterogeneous federated learning technique that introduces control variates to current local gradients.
- FedDF (Lin et al., 2020): An advanced aggregation strategy for heterogeneous federated learning using knowledge distillation with **unlabelled proxy data**.
- FedBE (Chen & Chao, 2021a): An advanced aggregation strategy for heterogeneous federated learning using bayesian ensemble-based knowledge distillation with **unlabelled proxy data**.
- ABAVG (Xiao et al., 2021): An advanced aggregation strategy for heterogeneous federated learning using validation accuracy to reweight the clients with **labelled proxy data**.
- Finetuning: An advanced aggregation strategy for heterogeneous federated learning using **labelled proxy data** to finetune the aggregated model in every communication round, mentioned in Chen & Chao (2021a).
- Median (Yin et al., 2018): A Byzantine-robust aggregation strategy that calculates dimension-wise median for client updates.
- Krum (Blanchard et al., 2017): A Byzantine-robust aggregation strategy that vector-wisely selects an update.
- Trimmed Mean (Yin et al., 2018): A Byzantine-robust aggregation strategy that dimension-wisely removes a certain portion of the largest and smallest updates and calculates the mean of remaining values.
- Sageflow (Park et al., 2021): A state-of-the-art attack-resistant aggregation strategy that combines entropy-based filtering and loss-based reweighting with **labelled proxy data.**
- FLTrust (Cao et al., 2021): A state-of-the-art attack-resistant aggregation strategy that maintains a server model, trains the server model with **labelled proxy data**, and reweights the client updates with the server update.

C.3 Detailed Hyperparameter Setting

Baseline. Generally, we follow the settings of the original papers without otherwise mentioning them. For the local training of FedPROX, we always tune the parameter according to the suggestion of the original paper to obtain the best performance for various conditions. For baseline models involving on-sever optimization with unlabelled/labelled data, the learning rate η_s is tuned from $[5e-5, 1e-2]$, and the epochs is tuned from $E_s = \{1, 5, 10, 20\}$. Same as ours, the batch size is 32, and Adam Optimizer is used for on-server optimization. It is worth mentioning that the full-space training methods, i.e., Finetuning, FedDF, and FedBE, generally have the best performance with small epoch numbers 1, 5, and 5, respectively. This phenomenon aligns with our analysis for overfitting with a few proxy samples using full-space training. For FedBE, the sampling number for models is set to 10, according to the original paper.

SmartFL & SmartFL-U. The default setting is mentioned in Section 5.1. Since the optimal balance parameter λ depends on multiple factors (e.g., participation rates, and data heterogeneity), we always

tune $\lambda \in \{1, 5, 15\}$ for the experiment for non-IID data distribution (Table 1). It is worth mentioning that our method is not highly sensitive to λ and the optimization without regularization can already get a significant performance gain. Notice that for the in-depth analysis (Figure 1,Figure 2,Figure 3), we do not include the regularization term, namely, set $\lambda = 0$ to fairly compare with full-space training technique. For the experiments involving poisoning attacks(Figure 4,Figure 6,Figure 7), the regularization term is also not involved. Also, we enlarge the server training epoch E_s to be 50 for the experiment with attacks on CIFAR-10 since the server-side optimization requires more steps to converge under poisoning attacks.

D ADDITIONAL EXPERIMENTS

D.1 ROBUSTNESS AGAINST DATA HETEROGENEITY

In this section, we include additional experiments on robustness against data heterogeneity, including **convergence speed** and the **extension to the NLP task**.

D.1.1 CONVERGENCE SPEED

Table 2: Comparison of **the number of communication rounds** to reach target accuracy. We evaluate different FL methods with ResNet-8 on CIFAR-10 with different degrees of data heterogeneity α and participation rates $C = 0.4$. *Methods assume the availability of unlabelled proxy data. †Methods assume the availability of labelled proxy data.

Method	$\alpha = 0.01$ $target = 0.35$	$\alpha = 0.04$ $target = 0.57$	$\alpha = 0.16$ $target = 0.68$	$\alpha = 0.32$ $target = 0.72$	$\alpha = 0.64$ $target = 0.735$
FedAVG	196.3±35.9	136.7±37.5	150.3±18.1	165.0±21.0	151.7±37.0
FedPROX	101.0±5.0	133.7±36.0	157.7±15.0	150.0±20.7	111.0±26.5
Scaffold	137.7±10.6	125.0±11.4	128.0±38.5	135.0±18.7	113.0±25.6
FedDF*	160.0±15.1	127.0±25.5	168.7±38.1	164.3±10.0	162.3±39.6
FedBE*	182.3±15.9	132.0±31.2	143.7±32.9	177.0±10.0	146.7±49.2
SmartFL-U*	135.3±22.0	117.7±11.2	91.0±11.1	153.0±33.0	124.0±46.5
ABAVG†	176.0±39.8	165.0±25.4	115.7±26.1	149.3±15.0	129.7±14.6
Finetuning†	72.3±6.0	96.0±12.2	97.3±3.8	197.0±25.1	177.3±24.6
SmartFL†	**34.7±6.1**	**48.3±2.1**	**58.3±1.5**	**121.3±22.1**	**98.0±17.6**

Highly non-i.i.d. distribution of data also severely influences the convergence speed of standard aggregation strategies. Table 2 shows the number of communication rounds for the different methods to reach the target accuracy with ResNet-8 on CIFAR-10. Advanced aggregation strategies for heterogenous FL also accelerate convergence compared with FedAVG. SmartFL always requires much fewer communication rounds to achieve target performance in all conditions, indicating the efficiency and effectiveness of optimizing the aggregation via subspace training.

D.1.2 EXTENSION TO NLP TASK

To verify the effectiveness of our method beyond the computer vision domain, we also evaluate our method using logistic regression on 20newsgroup (Lang, 1995), a popular NLP benchmark for news classification. As shown in Table 3, SmartFL and SmartFL-U outperform the full-space training counterpart and FedAVG by a large margin across different α with both labelled and unlabelled proxy data.

Table 3: Comparison of maximum test accuracy achieved by different methods with Logistic Regression on 20newsgroup with $C = 40\%$.

Methods	$\alpha = 0.01$	$\alpha = 0.04$	$\alpha = 0.16$
FedAVG	30.64±3.2	38.58±2.3	59.76±1.9
FedDF*	36.10±2.6	38.87±3.1	59.90±1.5
SmartFL-U*	39.53±1.9	43.10±1.8	60.32±1.0
Finetune†	37.10±3.5	37.22±2.5	59.93±1.1
SmartFL†	**44.51±1.1**	**47.33±1.3**	**60.77±0.7**

D.2 ROBUSTNESS AGAINST ATTACKS

This section includes more results and comprehensive analysis under different scenarios for the MNIST and CIFAR-10 datasets in the setting mentioned in Section 5.3. Figure 6 and Figure 7 show a comparison of various aggregation strategies on MNIST and CIFAR-10 with high and normal data heterogeneity under Label Flip and Omniscient Attack. We have the following observation classified by the methods:

First, statistical filtering-based Byzantine-robust methods such as Krum, Trimmed Mean, and Median can successfully defend against attacks in most cases when the attack rate is small and non-i.i.d. degree is not high, which is in line with the prior studies (Yin et al., 2018; Blanchard et al., 2017) However, they are not applicable when the attack rate get higher than half. Also, their performance is largely degraded when the data distribution is highly non-IID.

Second, the full-space training counterpart (i.e., Finetuning) performs relatively well among the methods on MNIST when the attack is not high but worse on CIFAR-10. This is because, for the simpler dataset, even overfitting on proxy data can to some extent help robust aggregation, while it does not work for the harder dataset. The results verify our intuition that finetuning massive parameters on a small amount of data can not dilute the negative effect brought by malicious clients.

Third, the methods leveraging server proxy data get the most competitive performance among all the solutions, suggesting the potential to improve the robustness of the server-side aggregation against attacks with reasonable server knowledge.

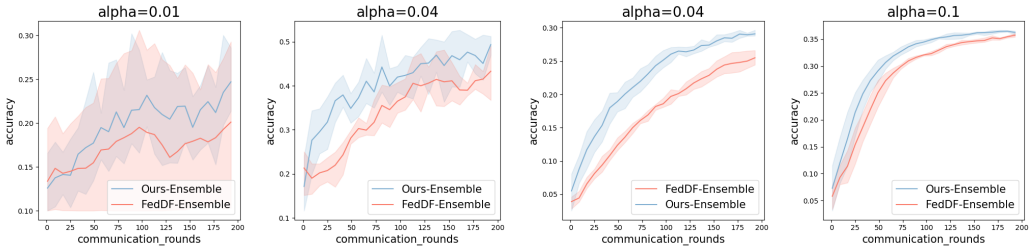
- ABAVG (Xiao et al., 2021), which uses the validation accuracy on proxy data to reweight the clients, performs relatively well in defending against Label Flip Attack but fails to defend against Omniscient attack. This is because, with a Label Flip attack, the attacker models are trained to predict a wrong label, and therefore the weight can be adjusted to a small value according to their low validation performance. However, for the model poisoning attack, the validation performance is not necessarily low enough.
- Sageflow (Park et al., 2021), which combines entropy-based filtering and loss-based reweighting, can get competitive performance under both types of attacks when the attack rate is not high and the distribution is not highly non-IID. However, it still fails in other conditions, especially with the Omniscient attack in that when the distribution is highly non-IID, the entropy of benign and malicious clients is not well separated.
- FLTrust (Cao et al., 2021) is the most competitive baseline that maintains a server model with proxy data and reweights the client updates according to the similarity with server model updates. We can observe that such a strategy enables robustness against attacks in almost all scenarios, especially model poisoning attacks, in that it can successfully capture and exclude the updates in an inverse direction of the server model. However, we still observe the instability of such a method during training since the stochastic gradient of the server model can not stably ensure "good" aggregation in all communication rounds. This can be a severe problem and sometimes leads to failure, as shown in Figure 7c.

Finally, different from the above solutions that heuristically leverage server proxy data, we aggregate a global model with optimized combination coefficients for client models with proxy data in every communication round and stably mitigate the negative effects brought by malicious clients.

E DISCUSSIONS

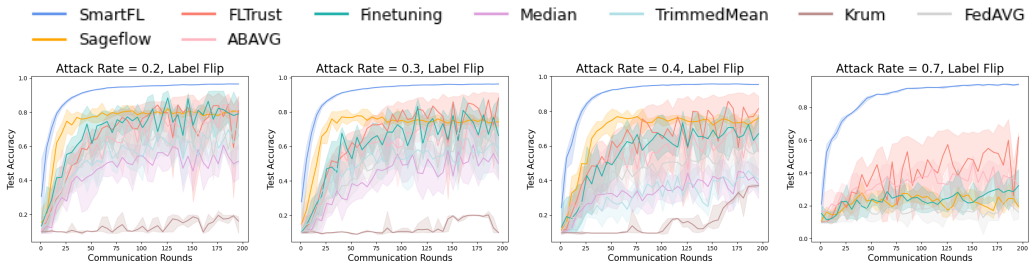
This section discusses the limitation and possible solutions. Since we still optimize the combination weights for the local clients, one limitation of SmartFL is that the aggregated client model should be the same architecture and can not be directly applied on **heterogeneous model architectures**. This can be alleviated by using multiple groups of model architectures. As illustrated in FedDF (Lin et al., 2020), knowledge distillation on unlabelled data using ensemble logits can allow information flow across models of different groups of architectures, and the server can use the ensemble of aggregated global models to make the final prediction. Here we show that our solution for unlabelled data (SmartFL-U) shares the merits of regular knowledge distillation (Lin et al., 2020) in allowing information flow across heterogeneous neural architectures (Li & Wang, 2019) by using the ensemble logits of all clients to supervise the combination with groups. Figure 5 visualizes the test

accuracy in every communication round of ensemble performance of SmartFL and the state-of-the-art FedDF for heterogeneous model architectures (ResNet-8, MobileNet, and ShuffleNet) with 128 unlabelled data on CIFAR-10, and 512 unlabelled data on CIFAR-100. SmartFL consistently dominates FedDF, demonstrating the effectiveness of breaking the knowledge barrier of heterogeneous models by leveraging averaged logits to optimize the global models in the subspace. We leave the possible improvement through leveraging both ground truth labels and ensemble client knowledge as future work.

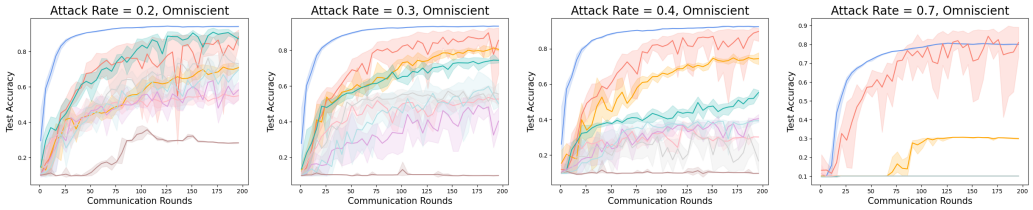


(a) CIFAR-10 (b) CIFAR-100

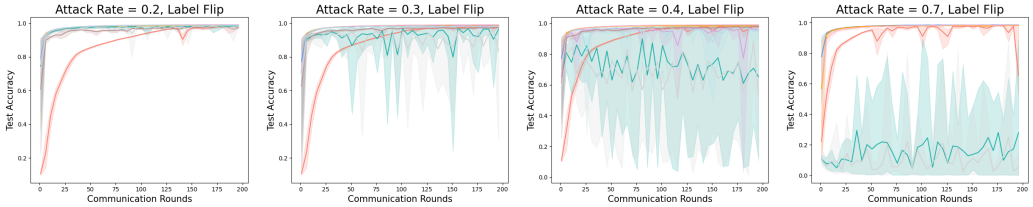
Figure 5: Studies on Heterogeneous Model Architectures (ResNet- 8, MobileNet, and ShuffleNet). We compare our method with FedDF with unlabelled proxy data on CIFAR-10/100. We show the test accuracy of server ensemble model in every communication round.



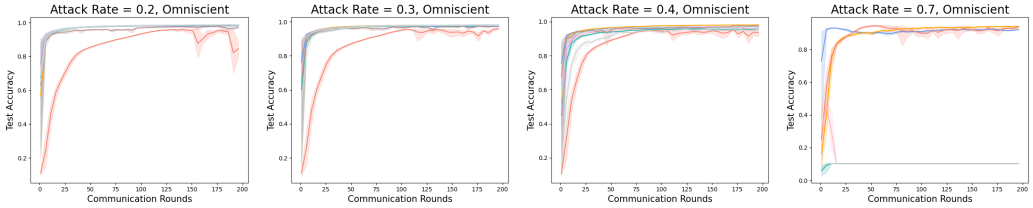
(a) MNIST, $\alpha = 0.01$, Label Flip



(b) MNIST, $\alpha = 0.01$, Omniscient

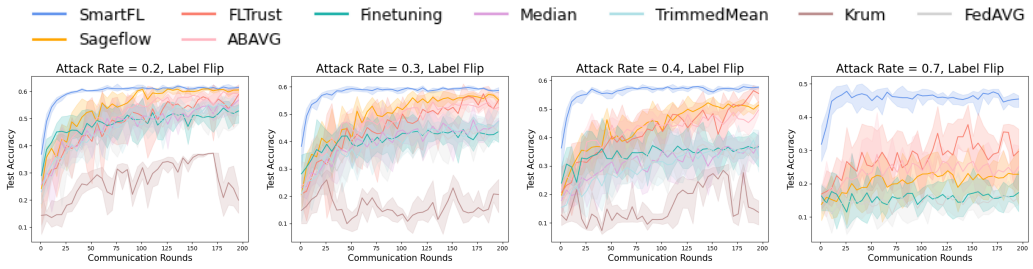


(c) MNIST, $\alpha = 1$, Label Flip

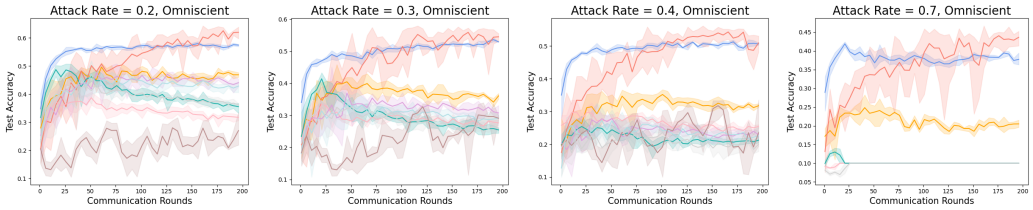


(d) MNIST, $\alpha = 1$, Omniscient

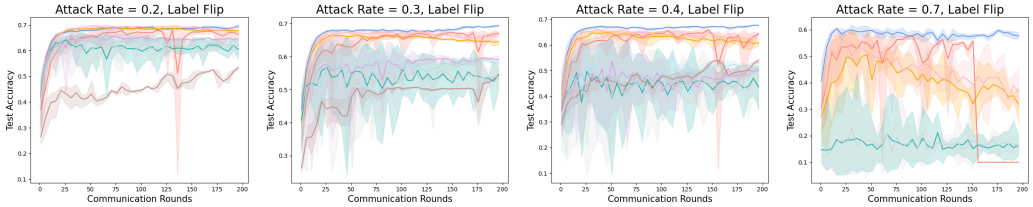
Figure 6: Defence against Attacks on MNIST with the degree of data heterogeneity $\alpha = 0.01$ and $\alpha = 1$, under different types of attacks (Label Flip and Omniscient Attack) and different attack rates $AR \in \{0.2, 0.3, 0.4, 0.7\}$.



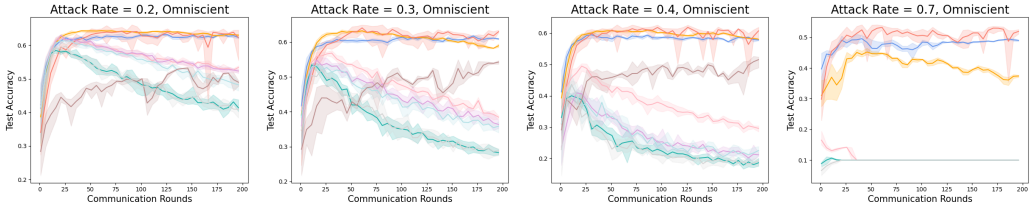
(a) CIFAR-10, $\alpha = 0.1$, Label Flip



(b) CIFAR-10, $\alpha = 0.1$, Omniscient



(c) CIFAR-10, $\alpha = 1$, Label Flip



(d) CIFAR-10, $\alpha = 1$, Omniscient

Figure 7: Defence against Attacks on CIFAR-10 with the degree of data heterogeneity $\alpha = 0.1$ and $\alpha = 1$, under different types of attacks (Label Flip and Omniscient Attack) and different attack rates $AR \in \{0.2, 0.3, 0.4, 0.7\}$.