FedAA: Using Non-sensitive Modality to Solve Non-IID Puzzles in Federated Learning

Anonymous authors

ABSTRACT

Federated Learning (FL) aims to train a better global model without sharing the sensitive training samples of local clients. Since the sample distributions in local clients tend to be different from each other (i.e., non-IID), one of the major challenges for FL is to prevent model degradation when training with clients of non-IID data. The degradation can be attributed to the weight divergence that quantifies the difference of local models from different training processes with the same weight initialization. Furthermore, non-IID also results in feature space heterogeneity during local training, making neurons of local models in the same location have different functions and further exacerbating weight divergence. In this paper, we demonstrate that the problem can be solved by sharing a very small portion of information from the non-sensitive modality (e.g., metadata, non-sensitive descriptions, etc.) while keeping the sensitive information of raw training samples protected. In particular, we propose Federated Learning with Adversarial Example and Adversarial Identifier (FedAA) that trains adversarial examples based on the shared non-sensitive modality to fine-tune local models before global aggregation. The training of local models is enhanced by client identifiers that discriminate the source of inputs to force different local models to get similar outputs and be more homogeneous during the local training. Experiments show that FedAA significantly outperforms the state-of-the-art non-IID federated learning algorithm, by only sharing about 0.2% and 0.1% data from the non-sensitive modality in the classification experiments and the image caption experiments, respectively.

CCS CONCEPTS

• **Security and privacy** → *Privacy protections*; • **Computing methodologies** → **Distributed artificial intelligence**.

KEYWORDS

federated learning, non-IID, non-sensitive modality, sensitive modality, adversarial learning

ACM Reference Format:

Anonymous authors. 2022. FedAA: Using Non-sensitive Modality to Solve Non-IID Puzzles in Federated Learning. In *Proceedings of ACM Conference (Conference'17).* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/1122445.1122456

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-1-4503-XXXX-X/18/06…$15.00
https://doi.org/10.1145/1122445.1122456

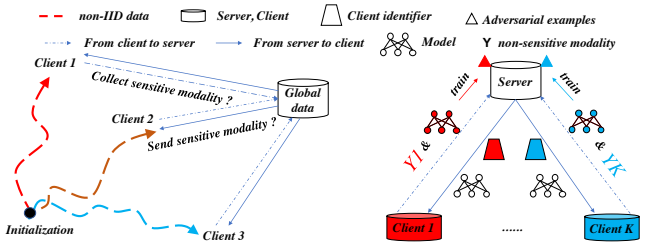


Figure 1: *Left:* The feasible way for the server to hold a global dataset. As the server cannot decide which client takes part in federated learning, especially for online training, the most feasible way is to collect raw data containing sensitive information from each client, which violates the federated setting. *Right:* We propose sharing the information from the non-sensitive modality and training client identifiers to alleviate the non-IID issue.

1 INTRODUCTION

Recent years have seen a surge of interest in federated learning, which provides a method to train a global model across a collection of distributed clients in the absence of mutual trust.

The most widely used federated learning algorithm, FedAvg [21], suffers when the assumption of Independent and Identically Distributed (IID) samples across local clients does not hold. In this case, the weight divergence that quantifies the difference of local models from different training processes with the same weight initialization is much larger than that trained on IID data, which will significantly degrade the performance of the aggregated model. As illustrated in the left part of Figure 1, a popular federated strategy [37] to alleviate the non-IID issue by creating a small dataset D_α that contains samples from each client, which can be regarded as a set of IID samples from the global distribution D of all clients. Therefore, D_α can be used to align the training data distributions across local clients, thus alleviating weight divergence and preventing model degradation. However, D_α is often unrealistic to obtain since clients should strictly protect the training samples. Even if there is a way for the central server to obtain D_α without violating privacy, it is still very hard to examine when the accumulated D_α can approximate the distribution of D. In specific scenarios (e.g., online learning), D is constantly changing, which makes D_α hard to maintain.

Following prior works [17, 18, 23, 37], for tasks such as image classification and image caption, we regard the information conveyed by raw images as private information that easily exposes personal privacy, such as portrait and residential address. Therefore, we regard the vision modality as the sensitive modality and keep raw images locally. In contrast, task-specific information like label names or non-sensitive descriptions is the non-sensitive modality. They carry no private information but can be used to identify

samples from the sensitive modality. For example, in the image classification task, the label name is a non-sensitive modality, which cannot show any details of raw images, even the categories (note that the label name we use can be abstract encoding numbers like "00", "01", "10", "11" instead of actual category names like "cat", "dog", etc.). Furthermore, in some image caption tasks, the caption is a non-sensitive modality (general description information) with less private information. As shown in the example in Figure 2, we cannot know the details of the man in the picture according to the caption "A man riding a wave on a surfboard in the ocean".

We propose to share a tiny amount of information from the non-sensitive modality D_β instead of the sensitive training samples D_α. Since D_β and D_α are correlated, D_β can also help the system to align the data distributions across local clients. With non-sensitive modality and the corresponding local models, we can train adversarial examples [8] that provide each client with distribution information of unseen classes. Then, we can fine-tune local models and alleviate the weight divergence issue with the trained adversarial examples. Moreover, different input and output distributions also result in local models' feature space heterogeneity, which makes neurons of local models in the same location have different functions and further exacerbates weight divergence. To align the feature space during local training, we propose client identifiers trained by adversarial examples on the server to discriminate the source of inputs. More specifically, local models try to learn a similar feature space to mislead the client identifier so that it cannot distinguish the source of the input data, thus achieving feature homogeneity across local clients. This paper proposes Federated Learning with Adversarial Example and Adversarial Identifier (FedAA). As illustrated in the right part of Figure 1, in FedAA, each client sends a local model and the non-sensitive modality to the server to train adversarial examples, then the server will send a new global model and a client identifier back. During local training, local models will be trained with client identifiers to align the feature space. For privacy concerns, we theoretically and experimentally show that sharing the non-sensitive modality will not expose information of the sensitive modality. Furthermore, we show the trained adversarial examples that are incomprehensible to humans. Hence, the trained adversarial examples are also privacy-preserving.

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

- We investigate the issues of the popular federated algorithm for non-IID data [37], which requests to share a global dataset of raw data. In contrast, we propose to share information from the non-sensitive modality and develop an effective framework named FedAA.
- We propose to quantify data privacy by distance correlation and theoretically prove that sharing information from the non-sensitive modality is privacy-preserving under our setting.
- The proposed method yields promising results on CIFAR-10, CIFAR-100 and MS COCO. More specifically, by sharing only a small amount of non-sensitive information (about 0.2% and 0.1%), FedAA outperforms the popular non-IID method, FedBN, by up to 21.26%, 12.52%, 5.90% relatively on different datasets.

2 RELATED WORK

2.1 Federated learning on non-IID data

The most widely used federated learning algorithm FedAvg cannot accurately capture the diversity of non-IID data splits [7], which is a challenging problem in federated learning. Lots of work has been proposed to deal with such issues. FedBN [17] trains a federated model with local batch normalization, but this method has difficulty in dealing with text data due to batch normalization's limitation. FedProx [14] adds a proximal term to the local cost functions, which forces local models to keep close to the global model. Moon [13] alleviates the non-IID issue with model-contrastive loss to improve the representation of local models. [37] proposed a strategy that the server holds a subset of data that contains examples from each client and globally shared between all the clients. However, the server cannot hold such a dataset in practice. Especially for online training, the server even has no idea of the number of clients. However, if clients send raw data containing sensitive information to the server to construct the global dataset, it will violate the federated setting. Therefore, we propose FedAA that is suitable for various types of data. Besides, FedAA only shares a tiny amount of information from the non-sensitive modality (such as label names in classification tasks and captions in some image caption tasks) that is not critical to users instead of sharing raw data.

2.2 Adversarial Example

Adversarial examples[20, 36] is one of the key applications of Adversarial training. They are inputs to machine learning models that an attacker has intentionally designed to cause the model to make a mistake. For example, in image classification tasks, using adversarial samples to perturb a specific pixel of the image to be classified can lead to a high-confidence misjudgment[22, 28]. Related work mainly focuses on the generation and defense of adversarial samples for specific downstream tasks[4, 32]. According to [8], adversarial examples are features, and adversarial vulnerability is a direct result of sensitivity to well-generalizing features in the data. Hence, we introduce the adversarial example to our work, and train the client identifier by the trained adversarial examples. Moreover, the trained adversarial examples are very different from the raw data, as the shared information contains no sensitive information.

3 METHODOLOGY

The most widely used federated learning algorithm, FedAvg [21] keeps data and computation locally on the clients. However, in practice, sharing some information from non-sensitive modalities will not threaten the users' privacy. Therefore, we propose Federated Learning with Adversarial Example and Adversarial Identifier (FedAA), which alleviates the non-IID issue by sharing a tiny amount of information from the non-sensitive modality.

3.1 Federated Learning

Assume there are K clients, each with a fixed local dataset. A random fraction C of clients will be selected in each global round when performing federated learning. The selected clients use the current global model's parameters as initialization to train local models.

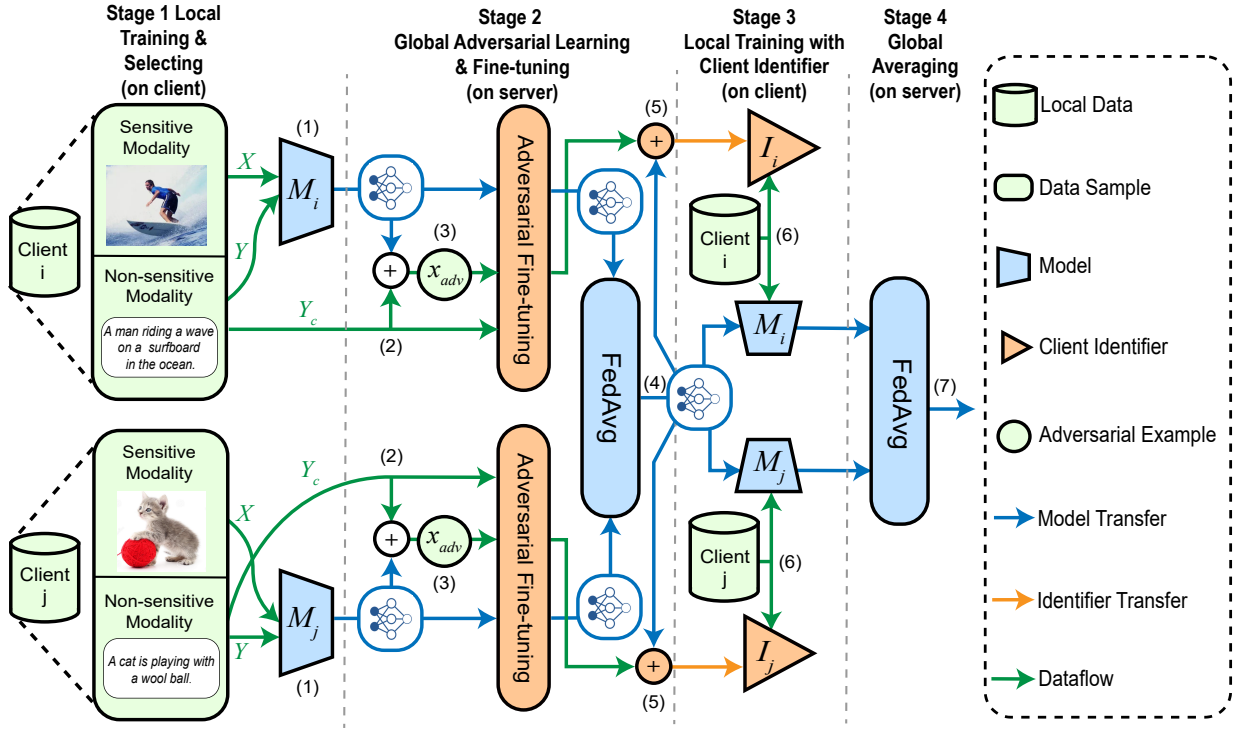


Figure 2: Illustration of FedAA. There are four stages and seven steps in the figure, where stages 1, 3 run on clients, and stages 2, 4 run on the server. In step (1), we train local models with local data. In step (2), we select a non-sensitive modality without private information and send a tiny amount of information from the non-sensitive modality to the server. In step (3), we train adversarial examples by the local models and the corresponding non-sensitive modality. In step (4), after the adversarial fine-tuning, we aggregate the local models to get an averaged model and send it to all clients. In step (5), we train a client identifier based on the adversarial examples and the output of the averaged model for each client, then send the client identifier to the corresponding client. In step (6), we train local models with client identifiers to alleviate weight divergence. In step (7), the local models are sent to the server and aggregated again.

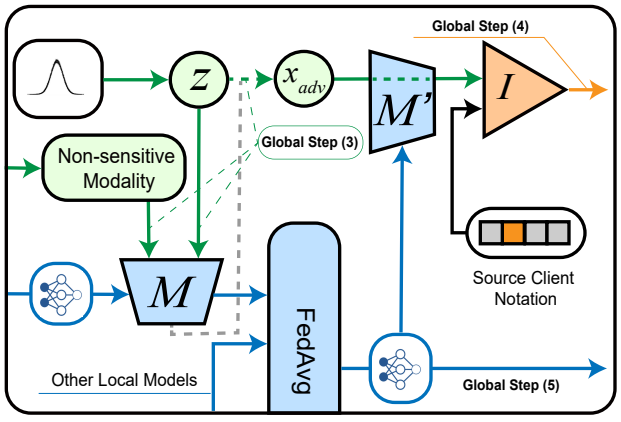


Figure 3: The details of training adversarial examples module, adversarial fine-tuning module, and client identifier training module.

After the fixed local training epoch le, local models will be sent to the server and aggregated.

For a machine learning problem, we typically divide the model into a feature extractor module $F(\cdot)$ and a downstream task module $T(\cdot)$. If the downstream task is classification, $T(\cdot)$ is a classifier. On the other hand, if the downstream task is image caption, $T(\cdot)$ is a decoder. The formal representation of federated learning can be:

$$\min_{F,T} L = \sum_{k=1}^{m} L_k, \quad \text{where} \quad L_k = \min_{F_k,T_k} \sum_{i=1}^{n_k} \mathrm{J}(T_k(F_k(x_i)), y_i) \quad (1)$$

where J is the loss function corresponding to the downstream task, x is a raw image (sensitive modality), y is a label name or caption (non-sensitive modality), m is the number of selected clients, n_k is the number of samples of client k.

For client k, the local model can be updated as follow:

$$\begin{aligned} F_k &= F_k - \eta_1 \nabla L_k(F_k; x) \\ T_k &= T_k - \eta_2 \nabla L_k(T_k; F_k(x)) \end{aligned} \quad (2)$$

where η_1, η_2 are learning rates. The Local Training is summarized in Algorithm 1.

Algorithm 1 LocalTraining. The K clients are indexed by k; B is the local minibatch size; E is the number of local epochs; η_1, η_2 are learning rates.

LocalTraining(k, F, T)**:** *//Run on client k*
$\mathcal{B} \leftarrow$(split local data into batches of size B)
for each local epoch i from 1 to E **do**
  for batch $b \in \mathcal{B}$ **do**
    $F \leftarrow F - \eta_1 \nabla L(F; b)$
    $T \leftarrow T - \eta_2 \nabla L(T; F(b))$
  end for
end for

3.2 Non-sensitive Modality

In this paper, we regard the vision modality as the sensitive modality. As for the non-sensitive modality, it depends on the downstream task. This paper takes the classification task and image caption task as examples. For the classification task, the label name is a non-sensitive modality, as it will not reveal the details of the image content. For instance, one client has images of dogs and flowers, while the other one has images of cats and trees. Now we train a cat, dog, flower, and tree classifier. In this case, sharing the corresponding label names ("00", "01", "10", "11") of different clients will not expose the private information of images. As for the image caption task, we select caption as the non-sensitive modality. For example, in Figure 2, "A man riding a wave on a surfboard in the ocean", we cannot deduce relevant information about this man from this general description, but the picture can show the appearance and other information of this man. Hence, the caption is the non-sensitive modality, while the image is the sensitive modality.

For the classification task, we send clients' label names to the server. As for the image caption task, we select the captions that fit the trained local model best, that is

$$y_c^k = \arg\min_{(x,y)} \mathrm{J}(T_k(F_k(x)), y) \tag{3}$$

where x is a raw image, and y is the corresponding caption, y_c^k is the selected caption of client k. Then, we get the global non-sensitive modality set $Y_c = \{y_c^1, y_c^2, ..., y_c^m\}$ from all selected clients.

3.3 Train Adversarial Examples from Gaussian Noise and Adversarially Fine-tune

According to [8], adversarial examples are features that are a direct result of sensitivity to well-generalizing features in the data. With the trained local models and non-sensitive modality, we can train the adversarial examples of client k as follows:

$$x_{adv}^k = \arg\min_{x_{sample}} \mathrm{J}(T_k(F_k(x_{sample})), y_c^k) \tag{4}$$

where $x_{sample} \sim \mathrm{N}(0, \sigma^2)$. Specifically, when training adversarial examples, we first randomly initialize the inputs x_{sample} of F. Then, we fix F and D to train x_{sample} by L-BFGS [38], which makes the output of D as close as possible to y_c.

Note that the trained adversarial examples from Equation 4 will not leak users' privacy. For example, in the image caption task, we train an adversarial example according to the caption *"A man is walking"*, which cannot be used to reproduce the man's details in the image. As for an image classification task, if there are two label names, 0 and 1. Each label name corresponds to lots of images. Hence, it is impossible to leak the details of any image through the shared label name.

After the adversarial examples and the non-sensitive modality are available, we can adjust local models by Equation 5 before aggregating.

$$\begin{aligned} T_k', F_k' &= \arg\min_{F,T} \mathrm{J}(T_k(F_k(X_{adv}^R)), Y_c^R), \\ \text{where} \quad X_{adv}^R &= X_{adv} - x_{adv}^k, Y_c^R = Y_c - y_c^k \end{aligned} \tag{5}$$

where X_{adv}^R is the adversarial examples that do not relate to client k, and Y_c^R is the non-sensitive modality information from all clients except k.

3.4 Client Identifier Update

Algorithm 2 IdentifierUpdate. η is the learning rate; X_{adv} is the trained adversarial examples; I is the client identifier.

IdentifierUpdate(k, F_{avg}, I, X_{adv})**:** *//Run on client k*
$\mathcal{B} \leftarrow$(split X_{adv} into batches of size B)
for each local epoch i from 1 to E **do**
  for batch $b_{adv} \in \mathcal{B}$ **do**
    $V_{embedding} \leftarrow F_{avg}(b_{adv})$
    $I \leftarrow I - \eta \nabla J_{adv}(V_{embedding})$
  end for
end for

When data is non-IID, due to the distance between the data distribution, the divergence between different local feature space will be large and accumulate very fast [37]. To address this issue, we propose to train a client identifier I for each client by the adversarial examples X_{adv}. At first, we aggregate selected local models to get an averaged extractor F_{avg}. Then, for client k, the loss function of the client identifier is:

$$\begin{aligned} \mathrm{J}_{adv}^k = &- \mathbb{E}_{x_i \sim X_{adv}^k} \left[\log I_k\left(F_{avg}(x_i)\right)\right] \\ &- \mathbb{E}_{x_j \sim X_{adv}^R} \left[\log\left(1 - I_k\left(F_{avg}(x_j)\right)\right)\right], \\ \text{where} \quad X_{adv}^R &= X_{adv} - X_{adv}^k, \quad F_{avg} = \frac{1}{m}\sum_{k=i}^{m} F_k' \end{aligned} \tag{6}$$

In Equation 6, X_{adv}^k is the adversarial examples trained by the local model and non-sensitive modality from client k. Due to the distance between the data distributions, the features extracted by F_{avg} are different. In this case, the object of client identifier I_k is to determine whether the input adversarial examples (processed by the extractor F_{avg}) come from client k, if yes, output 1, otherwise output 0. Furthermore, every client has a corresponding client identifier. The training process of I is summarized in Algorithm 2.

3.5 Local Training with Client Identifier

As the client identifier can distinguish different clients' features, we send the averaged model and the corresponding client identifier to the corresponding client, then train the local model with the client

Algorithm 3 LocalTrainingwithI. η_1, η_2, η_3 are learning rates.

LocalTrainingwithI(k, F, T, I)**:** *//Run on client k*
$\mathcal{B} \leftarrow$(split local data into batches of size B)
for each local epoch i from 1 to E **do**
  for batch $b \in \mathcal{B}$ **do**
    $F \leftarrow F - \eta_1 \nabla L(F; b) - \eta_3 \nabla L_{adv}(F; b)$
    $T \leftarrow T - \eta_2 \nabla L(T; F(b))$
  end for
end for

identifier to force different clients to be more similar. And the new objective function is

$$\begin{aligned} L_{new} &= L_k + L_{adv} \\ \text{where} \quad L_{adv} &= -\mathbb{E}_{x_i \sim X_k} \left[\log\left(1 - I_k\left(F_k(x_i)\right)\right)\right] \end{aligned} \tag{7}$$

where $L_k = \sum_{i=1}^{n_k} \mathrm{J_k}(T_k(F_k(x_i)), y_i)$. Note that the I_k remains unchanged when we update the T_k and F_k. After the local training with the client identifier, new local models will be sent to the server and aggregated again to get a global model. The local training with client identifier is summarized in Algorithm 3. The whole proposed algorithm is summarized in Algorithm 4.

3.6 Theoretical Analysis of Privacy

Assumption 3.1. *For any two variables, the lower the correlation, the more difficult it is to infer one variable's value from the other variable's value.*

We propose to use distance correlation $\mathcal{R}_n^2(\mathbf{X}, \mathbf{Y})$ between sensitive modality $\mathbf{X}$ and non-sensitive modality $\mathbf{Y}$ to quantify data privacy that may leak, which belongs to $[0, 1]$. The closer $\mathcal{R}_n^2$ is to 0, the lower the correlation between $\mathbf{X}$ and $\mathbf{Y}$. When $\mathcal{R}_n^2 = 0$, $\mathbf{X}$ and $\mathbf{Y}$ are independent. Suppose there are N different one-hot label names (classes) and each with the same number of data. We get,

$$\mathcal{R}_n^2(\mathbf{X}, \mathbf{Y}) = \frac{(1 - \frac{1}{N})\frac{1}{N}(\sum B_1 - \sum B_2)}{\sqrt{(1 - \frac{1}{N})\frac{1}{N} \sum B_3}} \tag{8}$$

where B_1, B_2, B_3 are coefficients of sensitive modality data, N is the number of classes. According to Equation 8, we can get,

$$\begin{cases} \lim\limits_{N \to 1, \infty} \mathcal{R}_n^2(\mathbf{X}, \mathbf{Y}) = 0, \\ \max \mathcal{R}_n^2(\mathbf{X}, \mathbf{Y}) \leq 0.5, \end{cases} \tag{9}$$

We defer the proof to the *Appendix A*. According to Equation 9, the non-sensitive data always gets a low correlation with sensitive modality data, especially when there is only one value in the non-sensitive modality, $\mathcal{R}_n^2(\mathbf{X}, \mathbf{Y}) = 0$, which shows that non-sensitive data and raw data are independent (i.e., N=1 that corresponds to our experiments). Therefore, sharing non-sensitive modality is privacy-preserving. Moreover, when $2 \leq N$, the shared modality is a random arrangement (i.e., not paired with samples) that makes the relevant terms of B in the numerator of Equation 8 appear random sign, which further leads to the decrease of $\mathcal{R}$.

4 EXPERIMENTS

In our experiments, we aim to (1) validate the effectiveness of FedAA for two different tasks: image classification and image caption. (2) validate the effectiveness of FedAA in two different scenarios: data islands and mobile terminals, (3) validate that FedAA can effectively alleviate weight divergence, (4) validate whether FedAA is privacy-preserving. Note that the settings of our experiments are more challenging, as the data heterogeneity is more obvious, which makes the reported results lower than that of prior works. Moreover, we train the adversarial examples by L-BFGS [38] for all experiments.

Algorithm 4 FedAA. The K clients are indexed by k; y_c is the data from the non-sensitive modality; x_{adv} is the trained adversarial examples.

Server executes:
Initialize global model parameters F_0, T_0
for each round t=1,2,... **do**
  $m \leftarrow \max(C \cdot K, 1)$
  $S_t \leftarrow$ (random set of m clients)
  for each client $k \in S_t$ in parallel **do**
    $T_{t+1}^k, F_{t+1}^k \leftarrow$ LocalTraining(k, F_t, T_t)
    For image caption, select non-sensitive modality data y_c by Equation 3. For classification, y_c is label names.
  end for
  $Y_c \leftarrow \{y_c^1, y_c^2, ..., y_c^m\}$
  for each client $k \in S_t$ (on server) **do**
    Sample noise $X_{sample} \sim \mathrm{N}(0, \sigma^2)$
    $x_{adv}^k \leftarrow \arg\min_{X_{sample}} \mathrm{J}(T_{t+1}^k(F_{t+1}^k(X_{sample})), y_c^k))$
  end for
  $X_{adv} \leftarrow \{x_{adv}^1, x_{adv}^2, ..., x_{adv}^m\}$
  for each client $k \in S_t$ (on server) **do**
    $X_{adv}^R = X_{adv} - x_{adv}^k$, $Y_c^R = Y_c - y_c^k$
    $\tilde{F}_{t+1}^k, \tilde{T}_{t+1}^k \leftarrow \arg\min_{F_{t+1}, T_{t+1}} \mathrm{J}(T_k(F_k(X_{adv}^R)), Y_c^R)$
  end for
  $F_{t+1} \leftarrow \sum_{k=1}^m \frac{n_k}{n} \tilde{F}_{t+1}^k$
  $T_{t+1} \leftarrow \sum_{k=1}^m \frac{n_k}{n} \tilde{T}_{t+1}^k$
  for each client $k \in S_t$ (on server) **do**
    Initialize client identifier I_k
    $I_k \leftarrow IdentifierUpdate(k, F_{t+1}, I_k, X_{adv})$
  end for
  for each client $k \in S_t$ in parallel **do**
    $\widehat{T}_{t+1}^k, \widehat{F}_{t+1}^k \leftarrow$ LocalTrainingwithI$(k, F_{t+1}, T_{t+1}, I_k)$
  end for
  $F_{t+1} \leftarrow \sum_{k=1}^m \frac{n_k}{n} \widehat{F}_{t+1}^k$
  $T_{t+1} \leftarrow \sum_{k=1}^m \frac{n_k}{n} \widehat{T}_{t+1}^k$
end for

4.1 Federated Classification with Data Islands

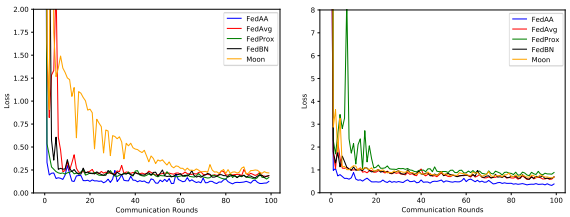
We run classification of data islands on CIFAR-10 [12] (50,000 samples) and CIFAR-100 [12] (5,000 samples). There are 10 clients, each with one class (for CIFAR-100, we randomly sample 10 classes and assign them to 10 clients). In this experiment, we use a network with two convolutional layers with 64 and 256 filters, respectively, followed by two fully connected layers, denoted as F. As we focus on the classification task in this section, the downstream task module T consists of a fully connected layer and a softmax layer. For the client identifier I, we use three fully connected layers, and the

Table 1: Data islands results of 10 clients on non-IID CIFAR-10 and non-IID CIFAR-100. Each experiment runs 100 rounds, and we report the 95% confidence interval over the highest results.

Dataset	C	FedAvg [21]	FedProx [15]	FedBN[17]	Moon[13]	FedAA
CIFAR-10	0.5	17.05 ± 7.54%	21.32 ± 12.58%	19.9 ± 11.54%	21.65 ± 12.37%	**22.95 ± 3.06%**
	0.8	34.46 ± 4.53%	37.17 ± 5.24%	33.35 ± 5.44%	35.93 ± 5.31%	**40.44 ± 4.23%**
	1.0	39.09 ± 12.71%	**43.94 ± 8.91%**	40.93 ± 9.02%	43.04 ± 7.37%	43.69 ± 3.42%
CIFAR-100	0.5	28.30 ± 14.63%	38.33 ± 15.20%	37.80 ± 7.97%	37.57 ± 14.63%	**41.93 ± 14.31%**
	0.8	50.73 ± 4.02%	50.40 ± 8.38%	45.53 ± 4.1%	48.33 ± 4.93%	**51.23 ± 9.22%**
	1.0	56.03 ± 3.87%	49.07 ± 22.06%	53.00 ± 5.47%	51.00 ± 11.21%	**57.00 ± 3.48%**

hidden layer has 512 neurons. When updating F and T, we use the SGD with a fixed learning rate of 0.0025, batch size 256. Moreover, D is updated by Adam with learning rate of 0.0004, $beta_1 = 0.5$, $beta_2 = 0.9$. Furthermore, we fix the number of the first local epochs to be 10 and the number of the second local epochs to be 1 in every global round, and the number of global epochs is 100. All experiments are run on GPU 2080. The number of shared samples from the non-sensitive modality depends on the number of clients. For instance, if there are 10 clients and each with 1 class, we will share 10 label names from the non-sensitive modality that are 10/50000(0.02%) on CIFAR-10 and 10/5000(0.2%) on CIFAR-100.

As shown in Table 1, FedAA consistently outperforms other methods, except when $C = 1.0$ on CIFAR-10, the results of FedAA are slightly lower than that of FedProx. On the other hand, FedAA performs much better on CIFAR-100 compared to other methods. Especially, FedAA outperforms FedProx by 16% relatively when $C = 1.0$. These results show that algorithms such as FedProx are not suitable for cases with a small amount of data, as each client has 5000 and 500 samples on CIFAR-10 and CIFAR-100, respectively. In contrast, FedAA can work well in both cases. We further show the convergence by loss in Figure 4, which illustrates that FedAA gets a lower loss compared to other methods.

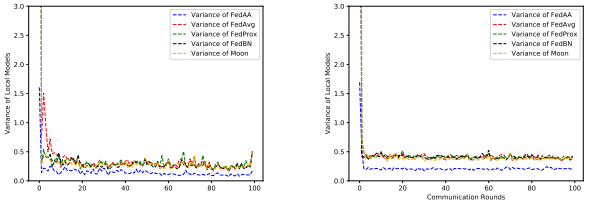


(a) CIFAR-10. (b) CIFAR-100.

Figure 4: Averaged loss of 10 Clients when C=0.8.

4.2 Federated Classification with Mobile Terminals

As for the scenario of many mobile terminals in the federated setting, we run experiments with 100 clients. For the non-IID setting, the training data is sorted by class and divided into 100 partitions. Then these partitions are randomly distributed to 100 clients (for CIFAR-10, one in ten clients will have the same class). The hyperparameters of this experiment are the same as that of the previous



(a) CIFAR-10, 100 clients, C=0.1 (b) CIFAR-100, 100 clients, C=0.1

Figure 5: Variance of local models.

experiment. We only use a small fraction of clients in this experiment due to the limited bandwidth.

Results in Table 2 show that our approach consistently outperforms other baselines, the trend of which is more significant with the low fraction C, as the adversarial examples of FedAA provides more information of unseen classes in each round. Its distinguished performance from FedAvg verifies the efficacy of sharing the non-sensitive modality under this challenging scenario.

We further show the ablation of FedAA in Table 3. Non-I shows the results of FedAA without the client identifier module. Non-F shows the results of FedAA without the adversarially fine-tune module. It can be seen that both identifier and fine-tuning play a vital role in FedAA, as almost all results are higher than that of FedAvg. In addition, these results also show that fine-tuning offers a more accurate identifier, as all results are lower than that of FedAA, especially when $C = 0.3$.

To validate that FedAA can force different local models to be more homogeneous (alleviate weight divergence). We show the variance of local models before aggregating in Figure 10. It can be seen that the blue curves of FedAA are much lower than that of baselines. More results of variance are presented in Appendix C.

4.3 Federated Image Caption on Multimedia Dataset

In practice, most federated multimedia tasks such as image caption, image-text matching, etc., suffer non-IID issues, as clients usually have different data categories in multiple modalities. For instance, different specialized hospitals usually hold different categories of X-ray images and corresponding diagnostic results. A popular multimedia dataset MS COCO [19] contains 123,603 images, and each is annotated with five sentences using Amazon Mechanical Turk.

Table 2: Mobile terminals results of 100 clients on non-IID CIFAR-10 and non-IID CIFAR-100. Each experiment runs 100 rounds, and we report the 95% confidence interval over the highest results.

Dataset	C	FedAvg [21]	FedProx [15]	FedBN[17]	Moon[13]	FedAA
CIFAR-10	0.1	35.73 ± 2.47%	36.87 ± 6.12%	34.41 ± 13.25%	34.78 ± 8.23%	**37.91 ± 3.24%**
	0.2	37.65 ± 7.21%	38.53 ± 5.39%	38.20 ± 6.89%	40.32 ± 1.37%	**40.69 ± 1.23%**
	0.3	42.39 ± 2.64%	42.59 ± 4.00%	43.05 ± 3.57%	41.1 ± 5.48%	**43.14 ± 0.31%**
CIFAR-100	0.1	3.97 ± 0.30%	3.83 ± 0.34%	3.88 ± 0.70%	3.29 ± 2.18%	**4.85 ± 1.10%**
	0.2	6.31 ± 0.50%	6.98 ± 1.65%	6.75 ± 3.36%	6.15 ± 1.86%	**7.42 ± 0.40%**
	0.3	8.59 ± 0.93%	9.86 ± 1.30%	9.79 ± 2.77%	8.20 ± 1.00%	**10.11 ± 0.69%**

Table 3: The ablation results of FedAA on CIFAR-100, 100 clients.

C	Non-I	Non-F	FedAvg
0.1	4.21 ± 0.95%	4.12 ± 0.8%	3.97 ± 0.30%
0.2	7.06 ± 0.52%	7.08 ± 1.07%	6.31 ± 0.50%
0.3	9.35 ± 0.57%	8.64 ± 1.91%	8.59 ± 0.93%

To evaluate our method on the non-IID multimedia dataset, we propose non-IID MS COCO. The training set is grouped according to the objects in the images. Besides, we drop images that contain two or more objects. Then there are 25,211 images with 126,055 captions divided into 80 groups. Moreover, we use 5,000 global images for validation and 5,000 global images for testing. We set the number of clients to 20 and randomly assigned 80 groups to these clients, e.g., each with four groups. Besides, each selected client shares only one caption every global round. As for the shared non-sensitive modality, take the fraction of clients $C = 0.1$ as an example. Each client has to select five times on average in 50 global training rounds. In other words, each client should share at most five captions (the same caption may be selected multiple times), and there are at most 100 shared captions from 126,055 captions (about 0.079%).

For the image caption task, the feature extractor module F is the encoder, and the downstream module T is the decoder. In our experiments, we use the resnet50 [5] as encoder to get a (2048, 4, 4) embedding vector. As for decoder, we use a LSTM [6] with attention [29]. F and T are updated by Adam with batch size 32. The learning rate of the encoder and decoder are 0.0001 and 0.0004, respectively. The client identifier I and its related hyperparameters are the same as that in section 4.1. Furthermore, we fix all the number of local epochs to be 1 in every global round for all methods, and the number of global epochs is 50.

As shown in Table 4, the results of FedAA on BLEU-4 surpass that of other methods, which indicates the effectiveness of FedAA on the image caption task. However, FedProx, which works well on the classification task, severely degrades the image caption task. This may result from the proximal term of FedProx cannot capture the effective features in the more complicated task.

We further show the results of FedAA, FedAA without adversarial fine-tuning, and FedAA without client identifier in Figure 7. It can be seen that FedAA and FedAA without adversarial fine-tuning both outperform FedAA without client identifiers, which confirms the effectiveness of the client identifier introduced in FedAA. Moreover, the adversarial fine-tune module can make the proposed algorithm more stable during training, as there is less vibration in the red curve.

Table 4: The BLEU-4 score ($\times 10^{-1}$) of FedAA and baselines on non-IID MS COCO. Since there is only a slight difference between the results of each experiment, we only report the average of three experiments.

Algorithm	$C = 0.1$	$C = 0.2$	$C = 0.3$	$C = 0.4$	$C = 0.5$
FedAvg	0.652	0.695	0.703	0.708	0.703
FedBN	0.746	0.781	0.779	0.776	0.790
FedProx	0.569	0.603	0.599	0.595	0.594
Moon	0.721	0.744	0.744	0.758	0.764
FedAA	**0.790**	**0.797**	**0.795**	**0.806**	**0.801**

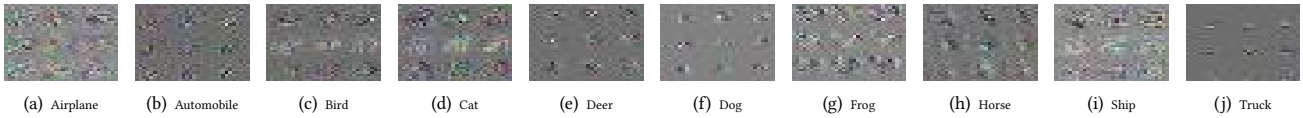
4.4 Privacy Analysis

We have theoretically analyzed that FedAA is privacy-preserving in section 3.6. This section empirically shows that FedAA is privacy-preserving on CIFAR-10, CIFAR-100 and MS COCO.

We propose to quantify the leakage of data privacy through distance correlation, as it measures dependence between two paired random vectors of arbitrary, not necessarily equal, dimension. The correlations between six distributions are illustrated in Figure 8, it can be seen that the correlations between *raw data* and *1 shared name*, *all shared name* are much lower than that between *raw data* and *random Gaussian distributions*. It implies that the information of raw data exposed by the shared non-sensitive modality is even far less than that of the generated data according to random Gaussian distributions. These results verify the correctness of the conclusions in section 3.6.

We further show the trained adversarial examples and the corresponding category on CIFAR-10 in Figure 6. The trained adversarial examples are very different from raw samples, as they are well-generalizing features of raw data. Therefore, adversarial examples will not leak private information on classification.

As image caption is an instance-level task, we randomly present two trained adversarial examples and the corresponding captions in Figure 9. In this case, adversarial examples do not show any private information, which validate the caption is a non-sensitive modality.



(a) Airplane (b) Automobile (c) Bird (d) Cat (e) Deer (f) Dog (g) Frog (h) Horse (i) Ship (j) Truck

Figure 6: The trained adversarial examples of classification task. Each image is corresponding to a label of CIFAR-10.

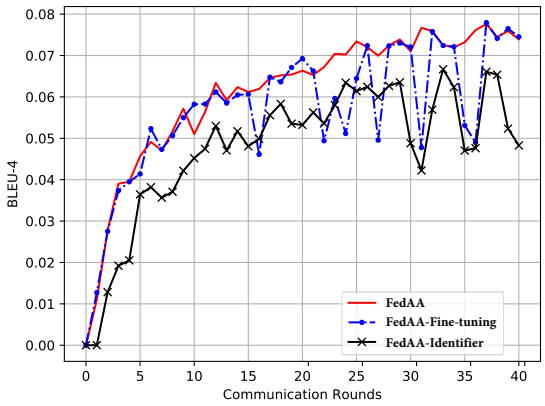
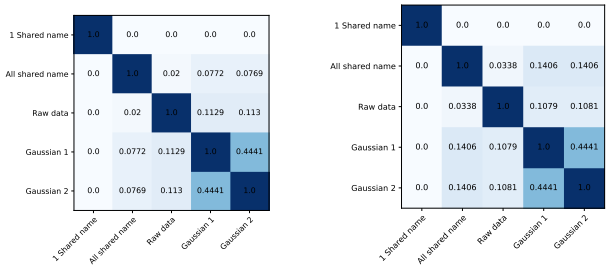


Figure 7: Ablation study. The FedAA and FedAA without adversarial fine-tuning get better results compared to FedAA without the client identifier. However, the adversarial fine-tuning module makes the algorithm to be more stable.



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

Figure 8: Heat map of distance correlation. *1 shared name* denotes each client holds 1 class. *all shared name* denotes each client holds all classes and the shared label names are randomly shuffled. *Gaussian 1* and *Gaussian 2* denote different Gaussian distributions.

Specifically, (a) in Figure 9, there is no clue for the model to generate detailed information about the man in the image according to the caption. All these results show that FedAA is privacy-preserving.

(a) **caption:** a man riding a wave on a surfboard in the ocean.

(b) **caption:** a close up of a kite flying in the sky.

Figure 9: The trained adversarial examples and the corresponding captions in non-IID MS COCO.

4.5 Communication Cost Analysis

In FedAA, two extra pieces of information need to be sent: 1. the non-sensitive modality 2. the client identifier model. As mentioned in previous sections, the amount of sent non-sensitive modality is tiny. For classification, each client only sends the label ids (such as 0, 1). As for image caption, each client only sends 1 caption each global round. Furthermore, the identifier is a small model with three fully connected layers. Hence, there is only a slight increase in the communication cost compared to FedAvg.

5 CONCLUSION

Federated learning suffers when trained on non-IID data because of the weight divergence issue. Moreover, the setting of the popular strategy for non-IID data to create a global dataset is flawed, as it is unrealistic for the server to hold a dataset containing all clients' distributions. Furthermore, we propose that the non-sensitive modality will not leak users' privacy while improving federated learning, as it can help train adversarial examples related to well-generalizing features. This paper proposes FedAA, which alleviates weight divergence by fine-tuning and training with client identifiers. Moreover, the proposed algorithm outperforms the popular non-IID baselines on image classification and image caption tasks over CIFAR-10, CIDAR-100 and MS COCO by sharing a tiny amount of information from the non-sensitive modality. The lower variance of FedAA also verifies that the proposed method can effectively alleviate the weight divergence.

REFERENCES

[1] Keith Bonawitz, Vladimir Ivanov, Ben Kreuter, Antonio Marcedone, H. Brendan McMahan, Sarvar Patel, Daniel Ramage, Aaron Segal, and Karn Seth. 2017. Practical secure aggregation for privacy-preserving machine learning. In *proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*. 1175–1191.

[2] Sebastian Caldas, Jakub Konečny, H. Brendan McMahan, and Ameet Talwalkar. 2018. Expanding the reach of federated learning by reducing client resource requirements. *arXiv preprint arXiv:1812.07210* (2018).

[3] Moustapha Cisse, Piotr Bojanowski, Edouard Grave, Yann Dauphin, and Nicolas Usunier. 2017. Parseval networks: Improving robustness to adversarial examples. In *International Conference on Machine Learning*. PMLR, 854–863.

[4] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. 2014. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572* (2014).

[5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Identity mappings in deep residual networks. In *European conference on computer vision*. Springer, 630–645.

[6] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997), 1735–1780.

[7] Kevin Hsieh, Amar Phanishayee, Onur Mutlu, and Phillip Gibbons. 2020. The non-iid data quagmire of decentralized machine learning. In *International Conference on Machine Learning*. PMLR, 4387–4398.

[8] Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Logan Engstrom, Brandon Tran, and Aleksander Madry. 2019. Adversarial examples are not bugs, they are features. *arXiv preprint arXiv:1905.02175* (2019).

[9] Eunjeong Jeong, Seungeun Oh, Hyesung Kim, Jihong Park, Mehdi Bennis, and Seong-Lyun Kim. 2018. Communication-efficient on-device machine learning: Federated distillation and augmentation under non-iid private data. *arXiv preprint arXiv:1811.11479* (2018).

[10] Jakub Konecnuy, H. Brendan McMahan, Felix X. Yu, Peter Richtárik, Ananda Theertha Suresh, and Dave Bacon. 2016. Federated learning: Strategies for improving communication efficiency. *arXiv preprint arXiv:1610.05492* (2016).
[11] Jakub Konecny, H. Brendan McMahan, Daniel Ramage, and Peter Richtárik. 2016. Federated optimization: Distributed machine learning for on-device intelligence. *arXiv preprint arXiv:1610.02527* (2016).
[12] Alex Krizhevsky, Geoffrey Hinton, et al. 2009. Learning multiple layers of features from tiny images. (2009).
[13] Qinbin Li, Bingsheng He, and Dawn Song. 2021. Model-Contrastive Federated Learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 10713–10722.
[14] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. 2018. Federated optimization in heterogeneous networks. *arXiv preprint arXiv:1812.06127* (2018).
[15] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. 2018. Federated optimization in heterogeneous networks. *arXiv preprint arXiv:1812.06127* (2018).
[16] Tian Li, Maziar Sanjabi, Ahmad Beirami, and Virginia Smith. 2019. Fair resource allocation in federated learning. *arXiv preprint arXiv:1905.10497* (2019).
[17] Xiaoxiao Li, Meirui Jiang, Xiaofei Zhang, Michael Kamp, and Qi Dou. 2021. FedBN: Federated Learning on Non-IID Features via Local Batch Normalization. arXiv:cs.LG/2102.07623
[18] Tao Lin, Lingjing Kong, Sebastian U Stich, and Martin Jaggi. 2020. Ensemble distillation for robust model fusion in federated learning. *Advances in Neural Information Processing Systems* 33 (2020), 2351–2363.
[19] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. 2014. Microsoft coco: Common objects in context. In *European conference on computer vision*. Springer, 740–755.
[20] Alireza Makhzani, Jonathon Shlens, Navdeep Jaitly, Ian Goodfellow, and Brendan Frey. 2016. Adversarial Autoencoders. arXiv:cs.LG/1511.05644
[21] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. 2017. Communication-efficient learning of deep networks from decentralized data. In *Artificial Intelligence and Statistics*. PMLR, 1273–1282.
[22] Anh Nguyen, Jason Yosinski, and Jeff Clune. 2015. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 427–436.
[23] Xingchao Peng, Zijun Huang, Yizhe Zhu, and Kate Saenko. 2019. Federated adversarial domain adaptation. *arXiv preprint arXiv:1911.02054* (2019).
[24] Xingchao Peng, Zijun Huang, Yizhe Zhu, and Kate Saenko. 2019. Federated adversarial domain adaptation. *arXiv preprint arXiv:1911.02054* (2019).
[25] Daniel Rothchild, Ashwinee Panda, Enayat Ullah, Nikita Ivkin, Ion Stoica, Vladimir Braverman, Joseph Gonzalez, and Raman Arora. 2020. Fetchsgd: Communication-efficient federated learning with sketching. In *International Conference on Machine Learning*. PMLR, 8253–8265.
[26] Virginia Smith, Chao-Kai Chiang, Maziar Sanjabi, and Ameet Talwalkar. 2017. Federated multi-task learning. *arXiv preprint arXiv:1705.10467* (2017).
[27] Dawn Song, Kevin Eykholt, Ivan Evtimov, Earlence Fernandes, Bo Li, Amir Rahmati, Florian Tramer, Atul Prakash, and Tadayoshi Kohno. 2018. Physical adversarial examples for object detectors. In *12th {USENIX} Workshop on Offensive Technologies ({WOOT} 18)*.
[28] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. 2013. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199* (2013).
[29] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *arXiv preprint arXiv:1706.03762* (2017).
[30] Riccardo Volpi, Hongseok Namkoong, Ozan Sener, John Duchi, Vittorio Murino, and Silvio Savarese. 2018. Generalizing to unseen domains via adversarial data augmentation. *arXiv preprint arXiv:1805.12018* (2018).
[31] Hongyi Wang, Mikhail Yurochkin, Yuekai Sun, Dimitris Papailiopoulos, and Yasaman Khazaeni. 2020. Federated learning with matched averaging. *arXiv preprint arXiv:2002.06440* (2020).
[32] Chaowei Xiao, Bo Li, Jun-Yan Zhu, Warren He, Mingyan Liu, and Dawn Song. 2018. Generating adversarial examples with adversarial networks. *arXiv preprint arXiv:1801.02610* (2018).
[33] Cihang Xie, Mingxing Tan, Boqing Gong, Jiang Wang, Alan L Yuille, and Quoc V Le. 2020. Adversarial examples improve image recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 819–828.
[34] Qiang Yang, Yang Liu, Tianjian Chen, and Yongxin Tong. 2019. Federated machine learning: Concept and applications. *ACM Transactions on Intelligent Systems and Technology (TIST)* 10, 2 (2019), 1–19.
[35] Felix Yu, Ankit Singh Rawat, Aditya Menon, and Sanjiv Kumar. 2020. Federated learning with only positive labels. In *International Conference on Machine Learning*. PMLR, 10946–10956.
[36] Jiliang Zhang and Chen Li. 2019. Adversarial examples: Opportunities and challenges. *IEEE transactions on neural networks and learning systems* 31, 7 (2019), 2578–2593.
[37] Yue Zhao, Meng Li, Liangzhen Lai, Naveen Suda, Damon Civin, and Vikas Chandra. 2018. Federated learning with non-iid data. *arXiv preprint arXiv:1806.00582* (2018).
[38] Ciyou Zhu, Richard H Byrd, Peihuang Lu, and Jorge Nocedal. 1997. Algorithm 778: L-BFGS-B: Fortran subroutines for large-scale bound-constrained optimization. *ACM Transactions on mathematical software (TOMS)* 23, 4 (1997), 550–560.

A PROOF OF PRIVACY-PRESERVING FOR SHARING NON-SENSITIVE MODALITY

Defination. Non-sensitive modality vector and raw data vector are defined as $\mathbf{Y}$ and $\mathbf{X}$, respectively. For observed samples $\{x_i, y_i\}$ from distribution of vectors $\mathbf{X}$ and $\mathbf{Y}$, respectively, compute the Euclidean distance matrices $a_{ij} = (\mid y_i - y_j \mid)$ and $b_{ij} = (\mid x_i - x_j \mid)$, where $\mid \cdot \mid$ denotes the Euclidean norm. Define

$$A_{ij} = a_{ij} - \bar{a}_{i\cdot} - \bar{a}_{\cdot j} + \bar{a}_{\cdot\cdot}, \quad i, j = 1, \ldots, n \tag{10}$$

where

$$\bar{a}_{i\cdot} = \frac{1}{n}\sum_{q=1}^{n} a_{iq}, \quad \bar{a}_{\cdot j} = \frac{1}{n}\sum_{k=1}^{n} a_{kj}, \quad \bar{a}_{\cdot\cdot} = \frac{1}{n^2}\sum_{k,q=1}^{n} a_{kq} \tag{11}$$

Similarly, define $B_{ij} = b_{ij} - \bar{b}_{i\cdot} - \bar{b}_{\cdot j} + \bar{b}_{\cdot\cdot}$, for $i, j = 1, \ldots, n$. And n is the number of observed samples.

The sample distance covariance $\mathrm{dCov}_n^2(\mathbf{X}, \mathbf{Y})$ and distance correlation $\mathcal{R}_n(\mathbf{X}, \mathbf{Y})$ are defined by

$$\mathrm{dCov}_n^2(\mathbf{X}, \mathbf{Y}) = \frac{1}{n^2}\sum_{i,j}^{n} A_{i,j} B_{i,j} \tag{12}$$

and

$$\mathcal{R}_n^2(\mathbf{X}, \mathbf{Y}) = \frac{\mathrm{dCov}_n^2(\mathbf{X}, \mathbf{Y})}{\sqrt{\mathrm{dCov}_n^2(\mathbf{X}, \mathbf{X})\,\mathrm{dCov}_n^2(\mathbf{Y}, \mathbf{Y})}} \tag{13}$$

where $0 \le \mathcal{R}_n^2(\mathbf{X}, \mathbf{Y}) \le 1$, and the larger the $\mathcal{R}$, the higher the correlation between $\mathbf{X}$ and $\mathbf{Y}$.

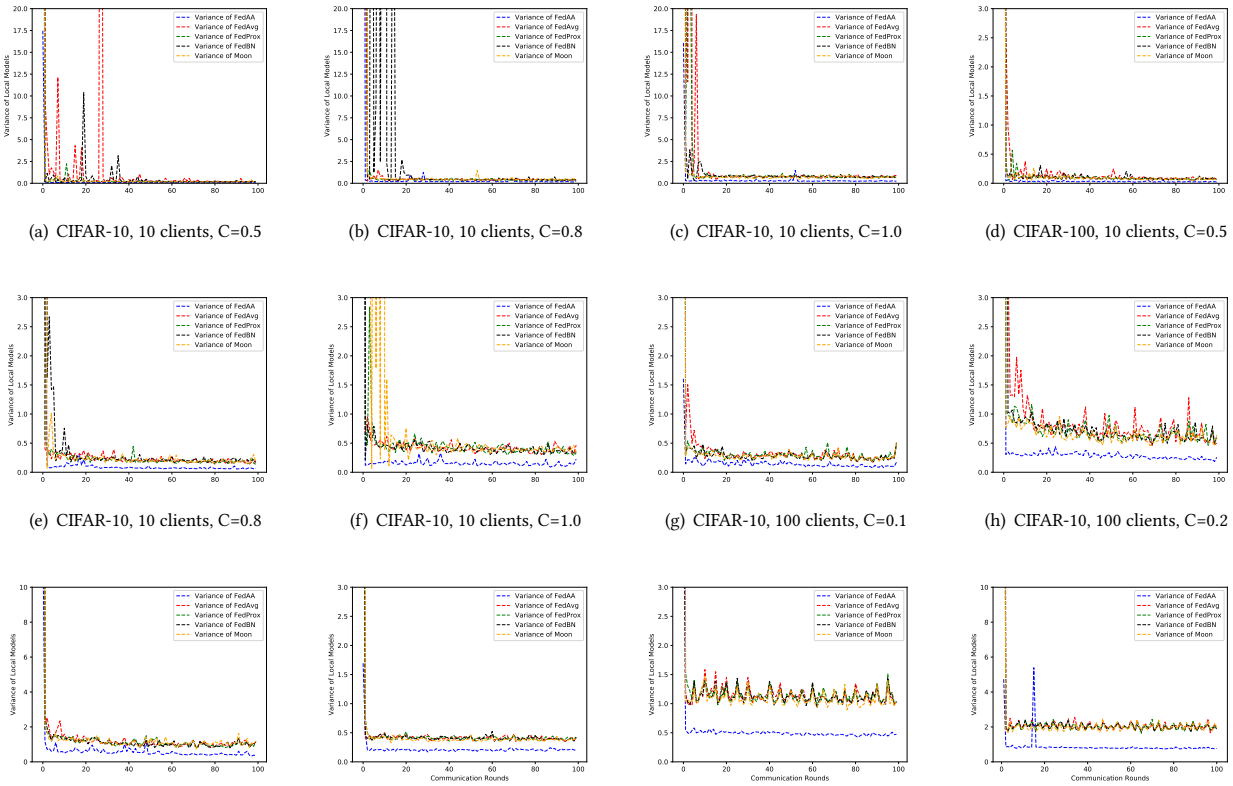
Suppose there are N classes (i.e. N different values of the non-sensitive modality), and each class with the same number of data. For convenience, let all different observed samples that belong to different classes of non-sensitive modality hold the same difference C, that is

$$\begin{cases} \mid y_i - y_j \mid = C, & \text{when} \quad y_i \neq y_j, p = 1 - \dfrac{1}{N}, \\ \mid y_i - y_j \mid = 0, & \text{when} \quad y_i = y_j, p' = \dfrac{1}{N}. \end{cases} \tag{14}$$

where p is the probability that sample i and sample j have the same value in the non-sensitive modality, p' is the probability that sample i and sample j have different values in the non-sensitive modality. Note that, if the non-sensitive modality is one-hot label name, C will be 1 (e.g. $y_i = [0, 01]$, $y_j = [0, 1, 0]$, and $\mid y_i - y_j \mid = 1$).

Substitute Equation 11, into Equation 10,

$$\begin{aligned} A_{ij} = \mid y_i - y_j \mid &- \frac{1}{n}\sum_{k=1}^{n} \mid y_k - y_j \mid - \frac{1}{n}\sum_{q=1}^{n} \mid y_i - y_q \mid \\ &+ \frac{1}{n^2}\sum_{k,q=1}^{n} \mid y_k - y_q \mid \end{aligned} \tag{15}$$



(a) CIFAR-10, 10 clients, C=0.5 (b) CIFAR-10, 10 clients, C=0.8 (c) CIFAR-10, 10 clients, C=1.0 (d) CIFAR-100, 10 clients, C=0.5

(e) CIFAR-10, 10 clients, C=0.8 (f) CIFAR-10, 10 clients, C=1.0 (g) CIFAR-10, 100 clients, C=0.1 (h) CIFAR-10, 100 clients, C=0.2

(i) CIFAR-10, 100 clients, C=0.3 (j) CIFAR-100, 100 clients, C=0.1 (k) CIFAR-100, 100 clients, C=0.2 (l) CIFAR-100, 100 clients, C=0.3

Figure 10: Variance of local models.

There are two situations for the results of Equation 15 based on $| y_i - y_j |$ in Equation 11. And compute the expections of Equation 15,

$$\mathbb{E}(A_{ij}) = \begin{cases} (1 - \frac{1}{N})C, \\ \frac{1}{N}C. \end{cases} \tag{16}$$

Substitute the Equation 16 into Equation 12,

$$\begin{aligned} \mathrm{dCov}_n^2(\mathbf{X}, \mathbf{Y}) &= (1 - \frac{1}{N})\frac{C}{N}\sum B_1 + \frac{1}{N}(\frac{1}{N} - 1)C\sum B_2 \\ &= (1 - \frac{1}{N})\frac{C}{N}(\sum B_1 - \sum B_2) \end{aligned} \tag{17}$$

Similarly, compute

$$\mathrm{dCov}_n^2(\mathbf{X}, \mathbf{X}) = (1 - \frac{1}{N})\frac{C^2}{N^2} + \frac{1}{N}(\frac{C}{N} - C)^2 = (\frac{1}{N} - \frac{1}{N^2})C^2 \tag{18}$$

Substitute the results of Equation 17 and Equation 18 into the Equation 13,

$$\mathcal{R}_n^2(\mathbf{X}, \mathbf{Y}) = \frac{(1 - \frac{1}{N})\frac{1}{N}(\sum B_1 - \sum B_2)}{\sqrt{(1 - \frac{1}{N})\frac{1}{N}\sum B_3}} \tag{19}$$

where B_1, B_2, B_3 are dCov_n^2 of raw data. According to Equation 19, we can get,

$$\begin{cases} \lim\limits_{N \to 1,\infty} \mathcal{R}_n^2(\mathbf{X}, \mathbf{Y}) = 0, \\ \max \mathcal{R}_n^2(\mathbf{X}, \mathbf{Y}) \leq 0.5, \quad \text{when } N = 2 \end{cases} \tag{20}$$

According to Equation 20, the non-sensitive data always gets a low correlation with raw data, especially when there is only one value in non-sensitive modality, $\mathcal{R}_n^2(\mathbf{X}, \mathbf{Y}) = 0$, which shows that non-sensitive data and raw data are independent (i.e., N=1 that corresponds to our experiemts). Therefore, sharing non-sensitive modality (label name) is privacy-preserving. Moreover, when $2 \leq N$, the shared modality is a random arrangement (i.e., not paired with samples) that makes the relevant terms of B in the numerator of Equation 19 appear random sign, which further leads to the decrease of $\mathcal{R}$.

B ALLEVIATE WEIGHT DIVERGENCE

We show the variance of local models that correspond to weight divergence in Figure 10. It can be seen that the variance of local models learned by FedAA is lower than all other methods, the trend of which is more significant given more clients.