Towards Fair Federated Learning with Zero-Shot Data Augmentation

Weituo Hao[1], Mostafa El-Khamy[2], Jungwon Lee[2], Jianyi Zhang[1],
Kevin J Liang[1], Changyou Chen[3], Lawrence Carin[1]
[1]Duke University [2]Samsung [3]State University of New York at Buffalo
weituo.hao@duke.edu

Abstract

Federated learning has emerged as an important distributed learning paradigm, where a server aggregates a global model from many client-trained models, while having no access to the client data. Although it is recognized that statistical heterogeneity of the client local data yields slower global model convergence, it is less commonly recognized that it also yields a biased federated global model with a high variance of accuracy across clients. In this work, we aim to provide federated learning schemes with improved fairness. To tackle this challenge, we propose a novel federated learning system that employs zero-shot data augmentation on under-represented data to mitigate statistical heterogeneity, and encourage more uniform accuracy performance across clients in federated networks. We study two variants of this scheme, Fed-ZDAC (federated learning with zero-shot data augmentation at the clients) and Fed-ZDAS (federated learning with zero-shot data augmentation at the server). Empirical results on a suite of datasets demonstrate the effectiveness of our methods on simultaneously improving the test accuracy and fairness.

1. Introduction

Major advances in deep learning over the last decade have in large part been possible due to the increasing availability of data. With the proliferation of personal computers, smart phones, and edge devices, data are being generated and collected at unprecedented rates, providing the large datasets needed to train the machine learning that power "intelligent" services that are becoming increasingly common in daily life. However, the rich content in these data that enables such smart behavior may also be revealing of personal information. Traditional learning methods pool the data into a central repository for training, which makes personal data vulnerable to breaches or interception.

Federated learning [26] has emerged as an alternative strategy, with an emphasis on user data privacy. In the federated learning paradigm, learning takes place on the client devices themselves, which means that the user's personal data never leaves the local device. In place of the data, the updated model itself is sent to a coordinating server, which then aggregates the updates and distributes the new model to the clients.

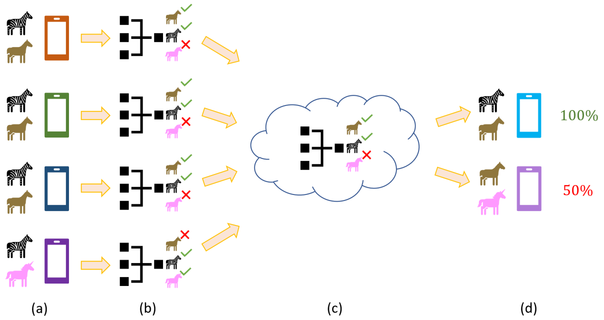


Figure 1: (a) Client data statistical heterogeneity. (b) Different model characteristics after local update. (c) Model aggregation at the server, which may drown out minority clients. (d) One-size-fits-all global model performs well in general, but poorly on minority clients.

While federated learning has demonstrated promise for user data privacy, a major challenge is statistical heterogeneity [19, 20, 5, 12]: data distributions between clients may exhibit significant differences. These differences may lead to variance in learned local models after training on each client's local data. Additionally, the formulation in [26] is fundamentally a one-size-fits-all solution, meaning the learned global model may perform worse for some clients, as shown in Figure 1. As a result of these factors, federated learning methods tend to perform poorer when the data are not independent and identically distributed (*i.i.d.*) among clients [26, 40].

What is more concerning, however, is that the accuracy loss due to statistical heterogeneity may be borne unequally among clients [21]. In populations with unequally sized subgroups, clients with less common classes tend to see worse performance [12]. This may be, in part, due to catastrophic forgetting [25, 28]: clients from outside a subpopulation have a tendency to forget features not found in their own data and, during aggregation, the less represented clients may have their learned features drowned out when the model weights are averaged. In the real world, these client characteristics may represent ethnicity [14], gender [4, 17], age [3], language [10], dialect, demographics, animal species, or disease trait. Therefore, the inability to cope with statistical heterogeneity my lead to potentially unfair algorithms, that provide in- accurate classifications based on certain characteristics of their input data. A popular and effective strategy for preventing forgetting is replay [23, 27]: storing a small buffer of samples for rehearsal. In federated learning, however, clients do not have access to data from parts of the distribution that are not well-represented in their own data. This is, in part, by design, as client data are kept private and local to the device.

In this work, we propose a federated learning system with zero-shot data augmentation (Fed-ZDA) to generate pseudo-exemplars of unseen classes, without having access to the private data. Such a strategy preserves the model's ability on previously sampled client data when learning the local client update. This makes the model less likely to lose representational ability for parts of the distribution that are rarer. We explore two strategies for using zero-shot data augmentation for federated learning, one in which synthetic samples are generated at the client (Fed-ZDAC), and another where they are generated at the server (Fed-ZDAS). Both methods are illustrated in Figure 2. Differential privacy analysis shows that our proposed approach satisfies $(0, \delta)$ differential privacy. Finally, experiments on MNIST [18], FMNIST [35], and CIFAR-10 [15] show that both Fed-ZDAC and Fed-ZDAS result in more equitable model performance.

2. Related Work

2.1. Federated Learning

Statistical Heterogeneity Statistical heterogeneity of the data distributions of client devices has long been recognized as a challenge for federated learning [40]. Despite acknowledging statistical heterogeneity, many federated learning algorithms still focus on learning a single global model [26]; such an approach often suffers from divergence of the model, as local models may vary significantly from each other. To address this challenge, a number of works break away from the single-global-model formulation. Several [29, 7] have cast federated learning as a multi-task learning problem, with each client treated as a separate task. FedProx [20] adds a proximal term to account for statistical heterogeneity by limiting the impact of local updates. In [40] performance degradation from skewed data is recognized, proposing global sharing of a small subset of data which, while effective, may compromise privacy.

Fairness There has been rising interest in developing fair methods for machine learning [37]. However, such concerns have been less addressed in federated learning. A commonly used fairness definition has been proposed in [38]. However, it forces the accuracy to be identical on each device across hundreds to millions of clients, given the significant variability of data in the network. Recent work [21] has taken a step towards addressing this by introducing uniformity to describe the fairness in federated learning, in which the goal is instead to ensure that the underfit groups are assigned more weight in the global learning objective. However, the proposed objective causes a performance drop in clients who could have better results under traditional federated average objective, which may reduce these clients' incentive to participate the federated learning process. The work in [12] proposed rank-one factorization on model parameters to ensure consistent model performance across clients, by leaving factors locally. However, this Bayesian approach usually costs more training time, and development of client-specific models is beyond the single-global-model focus of this paper.

2.2. Zero-Shot Data Augmentation

Deep learning performance is highly dependent on the quantity of data available [11, 30]. Data augmentation, which inflates the size of a dataset without necessitating further data collection, has proven effective in a wide range of settings [18, 16, 39, 22], improving machine learning model generalization. However, most data augmentations apply transformations to the existing data, thus making the implicit assumption that at least some data is available. These techniques are thus difficult to apply when no data is available. Consequently, *DeepInversion* [36] proposes a data-free knowledge transfer based on synthesizing data, effectively providing more teacher behavior for a student to learn. Also, [6] proposes a similar method for network quantization, by updating random input to match stored batch norm layer statistics. In our work, since the server has no access to the local data, synthesizing a reasonable amount of fake data for deficient classes would encourage a more fair global model. Also, unlike the work [13, 40] which violates the rule that clients should never share data to other clients or the server, zero-shot data augmentation synthesize data based on the model information only. Note that using synthesized samples for data augmentation differs from related works like [9], which take an approach similar to dataset distillation [32] to synthesize data for the purpose of compressing model updates for communication efficiency purposes.

2.3. Differentially Private Federated Learning

With the increasing awareness of data security and confidential user information, privacy has become an important topic for machine learning systems and algorithms. In order to solve this issue, differential privacy has been proposed to prevent revealing training data [1]. Even though federated learning enables local training without sharing the data to the server, it is still possible for an adversary to infer the private information to some extent, by analyzing the model parameters after local training [33, 24]. Therefore, combining differential privacy with federated learning has been studied in many previous works. To ensure federated learning approaches satisfy differential privacy, the work in [8] proposed a client level perspective by adding Gaussian noise to the model update, which can prevent the leakage of private information and achieve good privacy performance. In [31], a combination of differential privacy and secure multiparty computation was proposed to block differential attacks. However, previous approaches based on adding noise to model parameters struggle to capture the appropriate trade-off between the model performance and privacy budget. Our proposed zero-shot data augmentation can be interpreted as a new randomization mechanism different from adding Gaussian noise, satisfying differential privacy without hurting model performance.

3. Federated Learning with Zero-Shot Data Augmentation

We propose a federated learning method with zero-shot data augmentation (Fed-ZDA), for the purpose of improving the robustness and fairness of federated learning. To improve the fairness of the global model, Fed-ZDA introduces new synthetic data, generated either at the server or at the client nodes, to supplement training with underrepresented samples. Notably, these samples are generated without access to user data, but rather from shared models post-local update. We start by reviewing standard federated learning, which Fed-ZDA builds on. We then describe the zero-shot data-augmentation method we use for Fed-ZDA. We describe two deployments of Fed-ZDA, Fed-ZDAC and Fed-ZDAS, where the zero-shot data-augmentation is done at the client nodes and at the server node, respectively.

3.1. Federated Learning

In its most basic form, the federated learning objective is commonly expressed as the following:

$$\min_{w} f(w) = \sum_{i=1}^{Z} p_i F_i(w) \tag{1}$$

where $F_i(w) := \mathbb{E}_{\boldsymbol{x}_i \sim \mathcal{D}_i}[f_i(w; \boldsymbol{x}_i)]$ is the local objective function of the i^{th} client, Z is the number of devices or clients, and $p_i \geq 0$ is a weight assigned to the i^{th} client.

Standard federated learning aims to aggregate, at the centralized server, a federated global model from the client models, typically by averaging them. In this scenario, the clients only share their trained models with the server, and do not share the datasets on which their models have been trained. The server and the client communicate for T rounds to update the global model M. A single communication round contains three main steps:

1. The server randomly samples a subset of clients and distributes the model to the sampled clients.

2. Each sampled client updates the model by training it with their local training data.

3. Each client sends their updated model back to the server and the server aggregates the received client models into a new global model.

Typically, learning aggregates the models by federated averaging (FedAvg), in which the federated global model $M_{\mathcal{G},t}$ aggregated at the t^{th} communication round is simply a weighted average of all the client models received at this round. Let $M_{i,t}$ be the model trained by the i^{th} client C_i at the t^{th} communication round, and $\mathcal{S}_t$ is the set of indices of the sampled clients at the t^{th} round.

$$M_{\mathcal{G},t} = \sum_{i \in \mathcal{S}_t} w_i M_{i,t} \tag{2}$$

Different weights $0 \leq w_i \leq 1$ can be assigned to the clients depending on different factors, such as the amount of data they have been trained on, if such information is known at the server. Otherwise, a simple arithmetic mean is adopted.

Ideally, after sufficient communication rounds, the global model should converge to a solution that has learned using the data from all clients. However, heterogeneous data distributions across clients may cause inconsistent model performance. In particular, if the dataset distributions of the different clients are skewed towards a majority group of classes, FedAvg may result in a model with a large variance in accuracy across classes, resulting in a large variance in the global model accuracy on the data of different clients. Hence, standard federated learning suffers from the notion of unfairness towards the under-represented clients, providing poor accuracy on their data.

3.2. Zero-Shot Data Generation

Data augmentation has proven effective in many machine learning settings, such as when there is data scarcity or class imbalance. Commonly used techniques include performing transforms (*e.g.* rotations, flips, crops, added noise) based on the original true data, combinations in feature space, and synthesizing data by generative models. However, these techniques require access to training data or at least a few data sample seeds. In federated learning, these are not available, as data never leaves the individual clients,

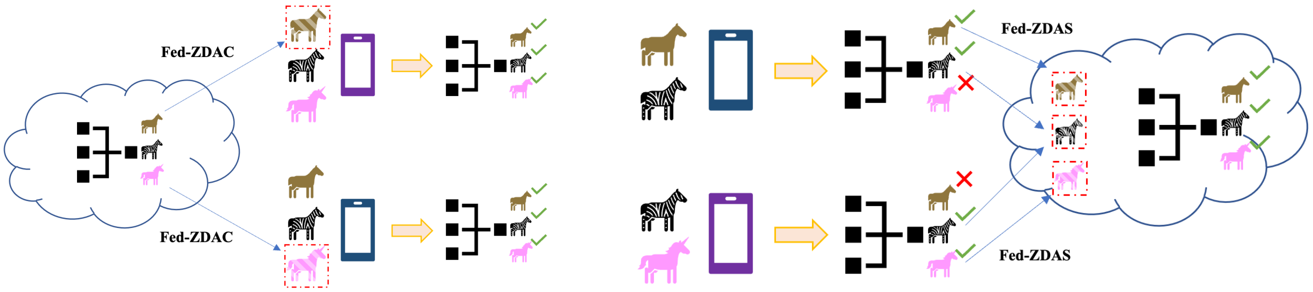


Figure 2: Illustration of Fed-ZDAC (left) and Fed-ZDAS (right). In Fed-ZDAC, clients train the model after data augmentation. In Fed-ZDAS, the server distributes the model after training on augmented fake data.

making conventional augmentation techniques challenging. In this work, we propose zero-shot data generation (ZSDG), to generate labeled synthetic data for data-augmentation at the clients, without having any access to any training data. This approach utilizes trained models (either the global model pre-update, or the local models post-update) to generate synthetic data of the desired classes without access to *any* non-local data.

One way to generate synthetic data whose statistics match those of the original training data is to find the data that results in similar statistics as those stored in the batch normalization (BN) layers of the pretrained model. However, without assigning class labels to this data, one cannot use this data in a data-augmentation regime for supervised training. For data augmentation with N possible classes, we generate data for each class $1 \leq n \leq N$, represented by its corresponding one-hot vector $\bar{y}(n)$, which has 1 at the n^{th} index and zero otherwise. Let model M be a neural network with L layers. For simplicity of notation, assume the model has L batch normalization (BN) layers and denote the activation before the ℓ^{th} BN layer to be z_ℓ. The ℓ^{th} BN layer is parameterized by a mean μ_ℓ and variance σ_ℓ calculated from the input feature maps when the model was being trained. During the forward propagation, z_ℓ is normalized with the parameters of the BN layer. Note that given a pretrained model M, batch norm statistics of all BN layers are stored and accessible. Given a target class $\bar{y}(n)$ the ZSDG reduces to the optimization problem that finds the input data $\bar{x}$ that result in the batch norm statistics matching those stored in the BN layers of the pretrained model, and are classified by the pretrained model as having label $\bar{y}(n)$. Given the pretrained model M, with BN statistics μ_ℓ and σ_ℓ stored in its layers $1 \leq \ell \leq L$, the ZSDG optimization problem to generate synthetic labeled data $(\bar{x}(n), \bar{y}(n))$ for $n \in \{1, 2, \cdots, N\}$ can be expressed as:

$$\begin{aligned}\bar{x}(n) = \arg\min_{\bar{x}} \sum_{\ell=1}^{L} \|\bar{\mu}_\ell - \mu_\ell\|_2^2 + \|\bar{\sigma}_\ell - \sigma_\ell\|_2^2 \\ + \mathcal{H}(M(\bar{x}), \bar{y}(n)),\end{aligned} \tag{3}$$

where $\bar{\mu}_\ell$, and $\bar{\sigma}_\ell$ are, respectively, the mean and standard deviation evaluated at layer ℓ with the generated input data, $M(\bar{x})$ denotes the model classification output when the input is $\bar{x}$, and $\mathcal{H}$ is the cross entropy loss function to learn the class labels. To solve Equation 3 for a selected class $\bar{y}(n)$, an input is initialized randomly from a normal distribution and, then, updated using gradient descent, while fixing the model parameters during back-propagation. The ZSDG is described in Algorithm 1.

Algorithm 1 Zero-Shot Data Generation (ZSDG)

```
Input: Model $M$ with $L$ batch normalization layers
Output: A batch of labeled fake data: $(\bar{x}, \bar{y})$
Get $\mu_\ell$, $\sigma_\ell$ from Batch Normalization layers of $M$, $\ell \in \{1, 2, \cdots, L\}$
for $n = 1, 2, \cdots, N$ do
    Generate $\bar{x}(n)$ randomly from a Gaussian distribution, assign it a label $\bar{y}(n)$
end for
for $j = 1, 2, \cdots$ do
    Forward propagate $M(\bar{x}(n))$ for all $n$
    Gather intermediate activations $\bar{z}_\ell$, $\ell \in \{1, 2, ..., L\}$
    Gather BN statistics: $\bar{\mu}_\ell$ and $\bar{\sigma}_\ell$ induced by intermediate activations $\bar{z}_\ell$, $\ell \in \{1, 2, ..., L\}$
    Compute the loss based on Equation 3
    Backward propagate and update the input $\bar{x}(n)$ only
end for
Return $(\bar{x}, \bar{y}) = \cup_{n \in \{1,2,\cdots,N\}} (\bar{x}(n), \bar{y}(n))$
```

3.3. Zero-Shot Data Augmentation at Clients

It is common to have statistical heterogeneity in the training data across clients. To address the deficiency of their training data in some classes, and promote the global model fairness, clients are instructed to augment their training data with fake data using ZSDG, before updating the received global model. Let the i^{th} client at the t^{th} communication round have the real local training data $(x_i, y_i)_t$ with input and label pairs. Let $(\bar{x}_i, \bar{y}_i)_t$ be the synthetic (fake)

Algorithm 2 Fed-ZDAC: Federated Learning with Zero-Shot Data Augmentation at Clients

```
Input: Communication rounds $T$, global model $M$
for $t = 1, \cdots, T$ do
    Server randomly selects subset $\mathcal{S}_t$ of clients
    Server sends $M_{\mathcal{G},t-1}$ to $\mathcal{S}_t$
    for Clients $C_i$, $i \in \mathcal{S}_t$ in parallel do
        Generate labeled fake data $(\bar{x}_i, \bar{y}_i)_t$ by ZSDG from the global model $M_{\mathcal{G},t-1}$
        Client $C_i$ produces the model $M_{i,t}$ by updating the model $M_{\mathcal{G},t-1}$ with the mix of real local data available at round $t$, and the fake ZSDG data: $\{(x_i, y_i)_t, (\bar{x}_i, \bar{y}_i)_t\}$
        Send the updated client model $M_{i,t}$ to the server.
    end for
    Server aggregates all client models $M_{i,t}$, $i \in \mathcal{S}_t$, e.g. by Equation 2, to obtain the updated $M_{\mathcal{G},t}$
end for
```

data generated using ZSDG over all classes from the received global model $M_{\mathcal{G},t-1}$. Then the procedure for federated learning with zero-shot data augmentation at the clients (Fed-ZDAC) is described by Algorithm 2.

3.4. Zero-Shot Data Augmentation at Server

In Section 3.3, we discussed federated learning with data augmentation at the client nodes. In practice, clients may be mobile computing devices that are limited in their computing resources and storage capacity, which may restrict their capacity for data augmentation. Clients may also not care about fairness of the global model towards other clients, and would like to train the best model for their classes of interest only. It is also in the best interest of the server to produce a fair and accurate model, that does not ignore data classes of the under-represented clients. In addition, if the global model is fair, and each client updates the global model from the same fair initialization, federated learning can convergence faster to a fair solution. Consequently, we propose federated learning with zero-shot data augmentation at the server (Fed-ZDAS). We use the same notation as described in Section 3.3. In more detail, the server distributes its global model to a subset of clients, Each of these clients update this global model with their local training data (x, y) and send it back to the server. In strive for fairness, the server will generate equal amount of fake data from each received client model, and combines all fake client data into a balanced synthetic dataset. The server aggregates all received client models into a single model, and then trains the single model by the combined synthetic dataset. To our knowledge, this is the first federated learning protocol which involves training at the server, since in general the server is assumed not to have any data. Fed-ZDAS is described in Algorithm 3.

Algorithm 3 Fed-ZDAS: Federated Learning with Zero-Shot Data Augmentation at the Server

```
Input: Communication rounds $T$, global model $M$
for $t = 1, \cdots, T$ do
    Server randomly selects subset $\mathcal{S}_t$ of clients
    Server sends $M_{\mathcal{G},t-1}$ to $\mathcal{S}_t$
    for Clients $C_i$, $i \in \mathcal{S}_t$ in parallel do
        Client $C_i$ produces the model $M_{i,t}$ by updating the model $M_{\mathcal{G},t-1}$ with its real local data available at round $t$ $(x_i, y_i)_t$ and sends the updated client model $M_{i,t}$ to the server.
    end for
    Server generates a class-balanced fake labeled data $(\bar{x}_i, \bar{y}_i)_t$ by ZSDG with each received client model $M_{i,t}$, $i \in \mathcal{S}_t$.
    Server combines the fake data generated with the different client models into a combined balanced dataset $(\bar{x}, \bar{y})_t = \cup_{i \in \mathcal{S}_t} (\bar{x}_i, \bar{y}_i)_t$.
    Server aggregates all client models $M_{i,t}$, $i \in \mathcal{S}_t$, e.g. by Equation 2, to obtain an interim global model $\tilde{M}_{\mathcal{G},t}$
    Server trains $\tilde{M}_{\mathcal{G},t}$ using the combined fake dataset $(\bar{x}, \bar{y})_t$ to produce the updated global model $M_{\mathcal{G},t}$
end for
```

Since the motivation of federated learning is protecting client data privacy, we also prove that our proposed method satisfies client-level differential privacy (DP), a local differential privacy adopted as [34]. Intuitively, before clients send updated model parameters back to the server, we seek for a randomized perturbation on these model parameters such that the server can not distinguish if certain client has been involved in the current communication round. A standard way to satisfy differential privacy is adding Gaussian noise to model parameters with trade-off between model's performance and privacy budget. [1, 34]. In contrast, our method can be considered as a kind of perturbation to model parameters with useful information as opposed to pure random noise. As a result, we show that our proposed method satisfies $(0, \delta)$ differential privacy. For more details about the proof, please check Appendix A.

4. Experiments

4.1. Datasets and Settings

Task and Datasets We conduct experiments on three standard datasets: MNIST [18], FMNIST [35], and CIFAR-10 [15]. Following [26] for the federated learning setting, the server selects a proportion $\gamma = 0.1$ of 100 clients during each communication round, with $T = 100$ total rounds for all methods. Each selected client trains their own model for $E = 5$ local epochs with mini-batch size $B = 10$. For the data partition, we focus on the non-*i.i.d.* setting, which is typically more challenging and realistic for federated learning. We divide the 60k images into a training set of 50k images and external test set of 10k images, then the training set is distributed to the clients, such that each client only has a subset Z of the classes, and divide their local data set as local training set and local testing set.

Following [12], we study two data splits, each representing different types of statistical heterogeneity. The first is unimodal non-*i.i.d.* which is identical to the data partition introduced by [26]. The second is multimodal non-*i.i.d.*, in which there exists subpopulations, with some being more prevalent than others. Each subpopulation group can be thought of as a mode of the overall distribution. In other words, the classes are imbalanced in the data set aggregating from all clients' data.

Model Architecture Our zero-shot data augmentation requires the model to contain batch normalization layers. For both MNIST and FMNIST, we use a convolutional network consisting of two 5×5 convolution layers with 16 and 32 output channels, respectively. Each convolution layer is followed by a batch normalization layer and a 2×2 max-pooling operation with ReLU activations. A fully connected layer with a softmax is added for the output. For CIFAR-10, we use a convolutional network consisting of two 3×3 convolution layers with 16 filters each. Each convolutional layer is followed by a batch normalization layer and a 2×2 max-pooling operation with ReLU activations. These two convolutions are followed by two fully-connected layers with hidden size 80 and 60, with a softmax applied for the final output probabilities. We utilize SGD as the optimizer and set the learning rate as 0.02 for all methods. We compare our methods with three baselines: FedAvg [26], FedProx [20] and q-FFL [21].

4.2. Local Test and Client-Level Fairness

Local test performance is a metric to evaluate the aggregated model on each client's local test set, that is usually class imbalanced. It is an important metric to demonstrate the personalization ability of the aggregated model. As with [21], the variance of local test performance across all clients is taken as the fairness metric. Lower variance means the learned model does not lean towards subpopulations who share prevalent data distributions, which is a more fair solution. This metric can be considered as fairness on clients level. We test all methods under both unimodal non-*i.i.d.* and multimodal non-*i.i.d.* The results are listed in Table 1. The mean accuracy is the average local test accuracy over all clients and the variance is the client level fairness metric. The standard deviation values are calculated based on the results of different trials by changing random seeds. For MNIST and FMNIST, the proposed method not only achieves the best mean accuracy, but also improves the fairness over all baselines. For CIFAR-10, our method achieves better accuracy than q-FFL and more fairness than FedAvg and FedProx.

4.3. Global Test and Class-Level Fairness

Global test performance is a metric to evaluate the aggregated model on an external test set, that is usually class balanced. This is an important criterion to justify the efficiency of the federated learning mechanism and the model's performance on newly coming clients. However, it is still a metric based on average which cannot fully capture whether the model is biased towards, if exists, any prevalent class distribution. We report the variance of accuracy across classes as an extra fairness metric on class level. In Table 2, the external accuracy is the accuracy of the federated model on the held out test set, and the variance is class level fairness metric. We observe better performance on MNIST and FMNIST and comparable results on CIFAR-10. Similarly, all the standard deviation values are calculated based on the results of different trials by changing random seeds.

4.4. The Analysis of Augmented Data

The augmented data are generated conditioned on the given label. To study the quality of the synthesized data, we separately trained three classifiers of the same architecture using the optimizer and learning rate described in Section 4.1, but in a centralized way for MNIST, FMNIST, and CIFAR-10. After training, each classifier achieves test accuracy 99.06%, 89.79% and 67.32%, respectively. These classifiers are taken as the standalone oracle to evaluate the augmented data. To obtain the synthesized data for test, we run ZSDG based on the model trained in federated learning under multimodal non-*i.i.d.* setting. For each class, we generate 64 images as the test data. The test results for augmented images are shown in Figure 3. We also list the trained model's ability to recognize each class as comparison. In general, the accuracy of synthesized data reflects the ability to 'fool' the oracle classifier, *i.e.* the ability to reduce the local data distribution divergence among clients. Since each client owns at most two classes in our experimental setting, the statistical heterogeneity can be mitigated, as long as deficient class of images is synthesized by ZSDG.

4.5. The Influence of Client Data Distribution

As mentioned in Section 4.4, the quality of synthetic data depends on the performance of the model we invert, and the model performance is highly affected by the client data distribution. To further study the influence of the client data distribution on data augmentation quality, we compare the models f_a, f_b, and f_c learned by three different algorithms:

		Unimodal		Multimodal	
Dataset	Method	Mean Accuracy ↑	Variance ↓	Mean Accuracy↑	Variance↓
MNIST	FedAvg	97.98±0.01	6.70±1.21	96.67±0.73	47±27
	FedProx	97.93±0.01	6.33±1.25	91.98±0.80	72±6
	q-FFL	95.84 ±0.45	17.00±9.20	94.81±7.55	78±20
	Fed-ZDAC	**98.23**±0.22	**3.54**±0.85	**97.07**±0.56	**27**±12
	Fed-ZDAS	97.34±0.61	6.22±0.33	95.49±0.99	49±22
FMNIST	FedAvg	85.30±2.67	368±222	83.43±2.28	245±41
	FedProx	85.64±2.19	360±215	83.37±2.04	237±38
	q-FFL	83.09±0.36	283±45	85.97±0.18	175±10
	Fed-ZDAC	84.65±2.81	280±112	**86.00**±0.07	161±40
	Fed-ZDAS	**86.23**±2.09	**188**±67	85.66±0.85	**135**±11
CIFAR-10	FedAvg	**50.30**±0.91	417±190	45.53±1.30	288±98
	FedProx	49.92±0.55	416±186	**45.88**±1.44	266±100
	q-FFL	41.72±3.00	**285**±115	38.25±1.12	**243**±49
	Fed-ZDAC	47.18±1.55	337±155	43.92±1.66	244±70
	Fed-ZDAS	47.78±1.02	325±145	42.18±0.81	**243**±64

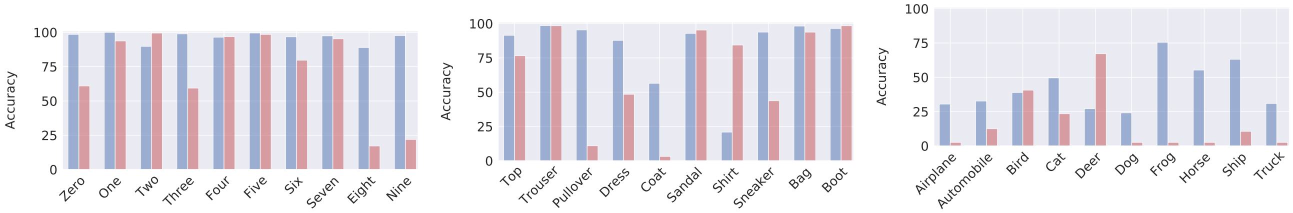
Table 1: Local test performance and client level fairness.

		Unimodal		Multimodal	
Dataset	Method	External Accuracy ↑	Variance ↓	External Accuracy↑	Variance↓
MNIST	FedAvg	98.02±0.14	3.69±0.55	93.54±2.38	78±48
	FedProx	98.05±0.15	3.69±0.60	93.62±2.38	75±58
	q-FFL	95.76 ±0.56	7.40±2.02	92.56±0.29	63±2
	Fed-ZDAC	**98.21**±0.08	**1.71**±0.21	**95.66**±0.72	**22**±7
	Fed-ZDAS	97.66±0.08	2.11±0.39	94.10±0.75	40±16
FMNIST	FedAvg	**85.03**±1.54	435±296	79.18±2.0	779±46
	FedProx	84.94±1.19	426±274	79.13±1.80	794±33
	q-FFL	80.99±1.23	558±192	81.24±0.43	673±12
	Fed-ZDAC	83.13±2.56	263±101	**83.41**±0.26	483±84
	Fed-ZDAS	83.90±1.56	**260**±76	83.27±0.25	**313**±68
CIFAR-10	FedAvg	48.89±1.04	473±195	**41.74**±4.30	361±154
	FedProx	48.83±0.89	258±13	37.06±0.62	480±50
	q-FFL	34.01±4.46	370 ±135	32.83 ±0.89	**218** ±38
	Fed-ZDAC	**49.50**±0.27	378 ±108	40.18±2.59	288±19
	Fed-ZDAS	48.26±1.02	**200**±69	39.07±1.85	295±98

Table 2: Global Test Performance and class level fairness.

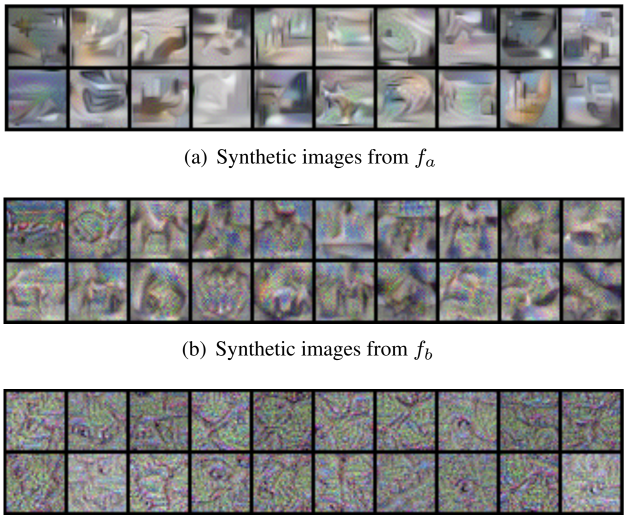
- f_a: the model trained by a regular machine learning process on aggregated dataset
- f_b : the model trained by federated learning framework on distributed dataset following an *i.i.d* setting
- f_c: the model trained by federated learning framework on distributed dataset following non-*i.i.d.* setting.Each client has at most 3 out of 10 classes of the images

We utilize the standard ResNet34 architecture and train it on CIFAR-10 dataset. The models' performance on test dataset for f_a,f_b, and f_c are 95.20%,73.96% and 58.10%, respectively. In other words, the model's performance decreases as more constraints put in the learning process, which is expected. Consequently, we invert the models and observe the quality of the synthetic images of the same target labels decreases as shown in Figure 4. This result not only validates that the quality of the synthetic data depends on the base model's performance but also suggests a burn-in stage before model inversion in the federated learning framework which is studied in Section 4.6.



(a) MNIST (b) FMNIST (c) CIFAR-10

Figure 3: The evaluation on augmented data of each class. The blue bars are the trained model's ability. The red bars are the accuracy of the augmented data that is queried from oracle classifiers.



(a) Synthetic images from f_a

(b) Synthetic images from f_b

(c) Synthetic images from f_c

Figure 4: The images recovered from models learned by three different learning algorithms.

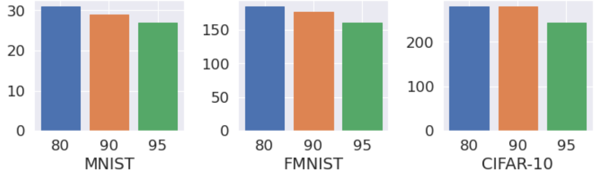


Figure 5: The influence of data augmentation starting point. The horizontal axis is the start global epoch and the vertical axis is the variance.

4.6. When to start data augmentation

It is important to choose the starting point for when the data augmentation is triggered, as the quality of reconstructed data highly depends on the model performance, which increases as the communication rounds between the server and clients climbs. High-quality augmented data help shrink the divergence of local data distribution among clients and improves privacy, therefore increasing the difficulty for the adversary to tell if certain clients have participated in training. Bad augmented data, such as random noise, can also help maintain privacy, but is likely to erase the useful information in learned model. We study the influence of starting epoch when data augmentation happens. We compare the Fed-ZDAC's fairness performance under the multimodal non-*i.i.d.* setting when the data augmentation starts from global epoch 80, 90, and 95, with the results shown in Figure 5. In federated learning, usually longer training epoch leads to solutions with better performance. As a result, the augmented data with higher quality make each client's local data distribution more similar, and contribute to reduce the variance more.

5. Conclusions and Future Work

To promote fairness and robustness in federated learning, we propose a federated learning system with zero-shot data augmentation, with possible deployments at the server (Fed-ZDAS), or at the clients (Fed-ZDAC). We provide a differential privacy analysis. We note that such methods only utilize the statistics of the shared models to generate fake data. Empirical results demonstrate our method achieves both better performance and fairness over commonly used federated learning baselines. For future research, we would like to investigate the combining of Fed-ZDAS and Fed-ZDAC in the same communication round, or at alternate rounds. Similarly, for clarity of the analysis in this paper, we assumed that Fed-ZDAC and Fed-ZDAS are deployed on top of the FedAvg with a simple arithmetic mean aggregation at the server. For future research, we would like to study the effect of deploying ZSDG on top of more complex aggregation schemes.

Acknowledgments

CC is partly supported by the Verizon Media FREP program.

References

[1] Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pages 308–318, 2016.

[2] Ganesh Ajjanagadde, Anuran Makur, Jason Klusowski, Sheng Xu, et al. Lecture notes on information theory. 2017.

[3] Vítor Albiero, Kevin Bowyer, Kushal Vangara, and Michael King. Does face recognition accuracy get better with age? deep face matchers say no. In *The IEEE Winter Conference on Applications of Computer Vision*, pages 261–269, 2020.

[4] Vítor Albiero, Krishnapriya KS, Kushal Vangara, Kai Zhang, Michael C King, and Kevin W Bowyer. Analysis of gender inequality in face recognition accuracy. In *Proceedings of the IEEE Winter Conference on Applications of Computer Vision Workshops*, pages 81–89, 2020.

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

[6] Yaohui Cai, Zhewei Yao, Zhen Dong, Amir Gholami, Michael W Mahoney, and Kurt Keutzer. Zeroq: A novel zero shot quantization framework. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13169–13178, 2020.

[7] Luca Corinzia and Joachim M Buhmann. Variational Federated Multi-Task Learning. *arXiv preprint arXiv:1906.06268*, 2019.

[8] Robin C Geyer, Tassilo Klein, and Moin Nabi. Differentially private federated learning: A client level perspective. *arXiv preprint arXiv:1712.07557*, 2017.

[9] Jack Goetz and Ambuj Tewari. Federated learning via synthetic data. *arXiv preprint arXiv:2008.04489*, 2020.

[10] Jiatao Gu, Hany Hassan, Jacob Devlin, and Victor OK Li. Universal neural machine translation for extremely low resource languages. *arXiv preprint arXiv:1802.05368*, 2018.

[11] Alon Halevy, Peter Norvig, and Fernando Pereira. The unreasonable effectiveness of data. *IEEE Intelligent Systems*, 2009.

[12] Weituo Hao, Nikhil Mehta, Kevin J Liang, Pengyu Cheng, Mostafa El-Khamy, and Lawrence Carin. WAFFLe: Weight Anonymized Factorization for Federated Learning. *arXiv preprint arXiv:2008.05687*, 2020.

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

[14] Brendan F Klare, Mark J Burge, Joshua C Klontz, Richard W Vorder Bruegge, and Anil K Jain. Face recognition performance: Role of demographic information. *IEEE Transactions on Information Forensics and Security*, 7(6):1789–1801, 2012.

[15] Alex Krizhevsky. Learning Multiple Layers of Features from Tiny Images. 2009.

[16] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Neural Information Processing Systems*, 2012.

[17] Susan Leavy. Gender bias in artificial intelligence: The need for diversity and gender theory in machine learning. In *Proceedings of the 1st international workshop on gender equality in software engineering*, pages 14–16, 2018.

[18] 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.

[19] Tian Li, Anit Kumar Sahu, Ameet Talwalkar, and Virginia Smith. Federated learning: Challenges, methods, and future directions. *IEEE Signal Processing Magazine*, 37(3):50–60, 2020.

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

[21] Tian Li, Maziar Sanjabi, Ahmad Beirami, and Virginia Smith. Fair resource allocation in federated learning. *arXiv preprint arXiv:1905.10497*, 2019.

[22] Kevin J Liang, Weituo Hao, Dinghan Shen, Yufan Zhou, Weizhu Chen, Changyou Chen, and Lawrence Carin. MixKD: Towards Efficient Distillation of Large-scale Language Models. *International Conference on Learning Representations*, 2021.

[23] David Lopez-Paz and Marc'Aurelio Ranzato. Gradient episodic memory for continual learning. *Neural Information Processing Systems*, 2017.

[24] Chuan Ma, Jun Li, Ming Ding, Howard H Yang, Feng Shu, Tony QS Quek, and H Vincent Poor. On safeguarding privacy and security in the framework of federated learning. *IEEE Network*, 2020.

[25] Michael McCloskey and Neal J Cohen. Catastrophic Interference in Connectionist Networks: The Sequential Learning Problem. *The Psychology of Learning and Motivation*, 1989.

[26] H Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, et al. Communication-efficient Learning of Deep Networks from Decentralized Data. *Artificial Intelligence and Statistics*, 2017.

[27] Hanul Shin, Jung Kwon Lee, Jaehong Kim, and Jiwon Kim. Continual Learning with Deep Generative Replay. *Neural Information Processing Systems*, 2017.

[28] Neta Shoham, Tomer Avidor, Aviv Keren, Nadav Israel, Daniel Benditkis, Liron Mor-Yosef, and Itai Zeitak. Overcoming forgetting in federated learning on non-iid data. *arXiv preprint arXiv:1910.07796*, 2019.

[29] Virginia Smith, Chao-Kai Chiang, Maziar Sanjabi, and Ameet S Talwalkar. Federated multi-task learning. In *Advances in Neural Information Processing Systems*, pages 4424–4434, 2017.

[30] Chen Sun, Abhinav Shrivastava, Saurabh Singh, and Abhinav Gupta. Revisiting unreasonable effectiveness of data in deep learning era. *International Conference on Computer Vision*, 2017.

[31] Stacey Truex, Nathalie Baracaldo, Ali Anwar, Thomas Steinke, Heiko Ludwig, Rui Zhang, and Yi Zhou. A hybrid approach to privacy-preserving federated learning. In *Proceedings of the 12th ACM Workshop on Artificial Intelligence and Security*, pages 1–11, 2019.

[32] Tongzhou Wang, Jun-Yan Zhu, Antonio Torralba, and Alexei A Efros. Dataset Distillation. *arXiv preprint arXiv:1811.10959*, 2018.

[33] Zhibo Wang, Mengkai Song, Zhifei Zhang, Yang Song, Qian Wang, and Hairong Qi. Beyond inferring class representatives: User-level privacy leakage from federated learning. In *IEEE INFOCOM 2019-IEEE Conference on Computer Communications*, pages 2512–2520. IEEE, 2019.

[34] Kang Wei, Jun Li, Ming Ding, Chuan Ma, Howard H Yang, Farhad Farokhi, Shi Jin, Tony QS Quek, and H Vincent Poor. Federated learning with differential privacy: Algorithms and performance analysis. *IEEE Transactions on Information Forensics and Security*, 2020.

[35] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

[36] Hongxu Yin, Pavlo Molchanov, Jose M Alvarez, Zhizhong Li, Arun Mallya, Derek Hoiem, Niraj K Jha, and Jan Kautz. Dreaming to distill: Data-free knowledge transfer via deep-inversion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8715–8724, 2020.

[37] Seyma Yucer, Samet Akçay, Noura Al-Moubayed, and Toby P Breckon. Exploring racial bias within face recognition via per-subject adversarially-enabled data augmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 18–19, 2020.

[38] Muhammad Bilal Zafar, Isabel Valera, Manuel Gomez Rogriguez, and Krishna P Gummadi. Fairness constraints: Mechanisms for fair classification. In *Artificial Intelligence and Statistics*, pages 962–970, 2017.

[39] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond Empirical Risk Minimization. *International Conference on Learning Representations*, 2018.

[40] 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.

A. Differential Privacy Analysis

We analyze the differential privacy of our proposed methods, adopting the same definition as [34] for differential privacy in randomized mechanisms. We show that our proposed method satisfies $(0, \delta)$ differential privacy or $(0, \delta)$-DP for short.

Definition A.1 ((ϵ, δ)-*DP*)**.** *A randomized mechanism $\mathcal{M} : \mathcal{X} \to \mathcal{R}$ with domain $\mathcal{X}$ and range $\mathcal{R}$ satisfies (ϵ, δ)-DP if for all measurable sets $\mathcal{S} \subset \mathcal{R}$ and for any two adjacent databases $\mathcal{C}$ and $\mathcal{C}' \in \mathcal{X}$,*

$$P(\mathcal{M}(\mathcal{C}) \in \mathcal{S}) \leq e^{\epsilon} P(\mathcal{M}(\mathcal{C}') \in \mathcal{S}) + \delta$$

Since we focus on the client level perspective, the databases $\mathcal{C}$ and $\mathcal{C}'$ here are the sets of clients, which differ on one client only, c and c', *i.e.*,

$$\begin{aligned} \mathcal{C} &= c \cup \mathcal{C}_0, \\ \mathcal{C}' &= c' \cup \mathcal{C}_0. \end{aligned} \tag{4}$$

Here, we denote the distributions of the datasets D and D' of the two client sets $\mathcal{C}$ and $\mathcal{C}'$ as $P_D(X)$ and $P_{D'}(X)$. Assume both clients start training their models, on their local datasets, starting from the same initial parameter W, e.g. the global model. If their datasets having different distributions, both clients will obtain two different models after local training, which have different parameter distributions. We denote the two parameter distributions as $P_{\mathcal{C}}(W)$ and $P_{\mathcal{C}'}(W)$. For simplicity, we assume the model training is a stochastic process estimating the following posterior distribution according to the Bayes' rule,

$$P(W|X) \propto P(X|W) P_0(W),$$

where $P_0(W)$ is the prior distribution of W. Since each client trains on the same model architecture, the likelihood model $P(W|X)$ will be the same for all clients. It is also reasonable to use the same prior distribution for every client.

Assumption A.1. *The total variation distance (TV) between the distributions of any two different augmented client datasets are less than δ: $TV(P_D(X), P_{D'}(X)) \leq \delta$.*

To verify the assumption A.1, we denote the distribution of generated data as G, and the i-th client's dataset is the union of the generated data and the raw data, and the distribution of this combined dataset is denoted as P_i. According to the definition of TV distance and its triangle inequality, given an arbitrary δ, we can always generate large enough samples such that $TV(G, P_i)$ is smaller than $\delta/2$. Thus for any two clients, we have $TV(P_j, P_i) \leq TV(P_j, G) + TV(P_i, G) \leq \delta/2 + \delta/2 = \delta$. As a result, the assumption A.1 is reasonable. With the above assumption, we use the data processing inequality stated in Lemma A.1 to derive the TV distance between $P_{\mathcal{C}}(W)$ and $P_{\mathcal{C}'}(W)$.

Lemma A.1. *(Theorem 6.2 in [2]) Consider a channel that produces Y given X based on the law $P_{Y|X}$ (illustrated in Figure 6). If P_Y is the distribution of Y when X is generated by P_X and Q_Y is the distribution of Y when X is generated by Q_X, then for any f-divergence $D_f(\cdot\|\cdot)$,*

$$D_f\left(P_Y \| Q_Y\right) \leq D_f\left(P_X \| Q_X\right)$$

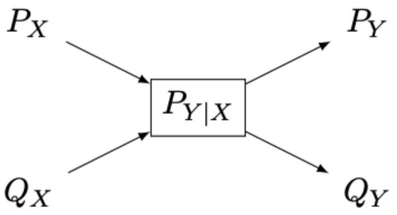


Figure 6: Data processing inequality

Theorem A.2. *Federated learning with zero-shot data augmentation satisfies the differential privacy $(0, \delta)$-DP.*

Proof. Since the total variation distance is an instance of f-divergence [2], applying Lemma A.1, we obtain

$$TV(P_{\mathcal{C}}(W), P_{\mathcal{C}'}(W)) \leq TV(P_D(X), P_{D'}(X)) \leq \delta.$$

In federated learning, we perform model aggregation, denoted as W_{agg}, as

$$W_{agg} = \frac{1}{n} W + \frac{n-1}{n} W_0$$

where W_0 is the parameter aggregated on the set of other clients $\mathcal{C}_0$ (as defined in Eq. 4) and n is the number of clients in $\mathcal{C}$. We denote the two different distributions of W_{agg} in the two models as $P_{\mathcal{C}}(W_{agg})$ and $P_{\mathcal{C}'}(W_{agg})$. Similarly, we can also use the Lemma A.1 to derive that,

$$TV(P_{\mathcal{C}}(W_{agg}), P_{\mathcal{C}'}(W_{agg})) \leq TV(P_{\mathcal{C}}(W), P_{\mathcal{C}'}(W)) \leq \delta$$

Based on the definition of total variation distance, we have

$$\sup_{S \subset R} |P_{\mathcal{C}}(W_{agg} \in S) - P_{\mathcal{C}'}(W_{agg} \in S)| \leq \delta$$

Define the stochastic mechanism M as the projection from the client set to any model parameter $W_{agg} \in \mathcal{R}$. Then the distribution of $M(\mathcal{C})$ and $M(\mathcal{C}')$ are the distributions of W_{agg} and W'_{agg}, respectively. Hence, for any $S \subset R$:

$$P(M(\mathcal{C}) \in S) \leq P(M(\mathcal{C}') \in S) + \delta \,,$$

which finishes the proof that Fed-ZDA satisfies $(0, \delta)$-DP. □