# Federated Learning with Classifier Shift for Class Imbalance

**Anonymous Author(s)**
Affiliation
Address
email

## Abstract

Federated learning aims to learn a global model collaboratively while the training data belongs to different clients and is not allowed to be exchanged. However, the statistical heterogeneity challenge on non-IID data, such as class imbalance in classification, will cause client drift and significantly reduce the performance of the global model. This paper proposes a simple and effective approach named FedShift which adds the shift on the classifier output during the local training phase to alleviate the negative impact of class imbalance. We theoretically prove that the classifier shift in FedShift can make the local optimum consistent with the global optimum and ensure the convergence of the algorithm. Moreover, our experiments indicate that FedShift significantly outperforms the other state-of-the-art federated learning approaches on various datasets regarding accuracy and communication efficiency.

## 1 Introduction

There are already numerous edge devices such as smartphones and IoT devices that can collect valuable raw data, and ones expect to use these data to complete some intelligent tasks such as image recognition or text generation. However, deep learning, the most effective algorithm for accomplishing these tasks, requires huge data to train the model, making it challenging to learn a good enough model from the data owned by a single edge device. Besides, due to data privacy, data protection regulations (Voigt and Von dem Bussche, 2017), and the massive overhead of data transmission, it is unrealistic to aggregate data from different clients (edge devices) in a server for training. Therefore, federated learning (FL) (Kairouz et al., 2019) has emerged to solve the problem of jointly learning a global model without sharing the private data.

Although federated learning has shown good performance in many applications (Kaissis et al., 2020; Liu et al., 2020), there are still several important challenges that require researchers to pay attention to, namely privacy, communication cost, and statistical heterogeneity (Ji et al., 2021). Statistical heterogeneity means that client data is non-IID (independent and identically distributed). Zhao et al. (2018) show that the accuracy of the federated learning algorithm has decreased significantly in the case of non-IID data. There are many methods proposed to address the challenge of statistical heterogeneity. FedProx (Li et al., 2020) introduces a proximal term to constrain the update of the local model, and SCAFFOLD (Karimireddy et al., 2020) corrects the gradient of each local update to reduce the variance. However, these methods do not bring significant improvement because they only implicitly deal with the fundamental dilemma caused by statistical heterogeneity, that is, the optimal objective of local update is inconsistent with the optimal objective of global update.

In this work, we propose an approach FedShift to explicitly solve the above fundamental dilemma in the statistical heterogeneity challenge. FedShift is a simple and effective approach which adds the shift on the classifier output calculated by the client category distribution and makes the local optimal

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

models satisfy the global optimum. We also prove the convergence results of FedShift in the strongly convex and non-convex cases and compare with FedAvg, which does not have the classifier shift. Numerous experiments are conducted to evaluate the effectiveness of FedShift, which demonstrate that FedShift outperforms the other state-of-the-art federated learning algorithms in test accuracy and communication efficiency on various datasets, including Cifar10, Cinic10 and Tiny-Imagenet.

# 2 Related Works

FedAvg (McMahan et al., 2017) is the benchmark method in federated learning, which has demonstrated reliability in image classification and language modeling tasks. Each round of FedAvg mainly contains two phases, client update and server aggregation. First, each client that is selected to participate in training downloads the latest global model from the server, and updates the model locally using stochastic gradient descent. Then, the server collects the updated models from each client and aggregates them to obtain a new global model by averaging the model weights.

Unlike the privacy and communication challenges, the statistical heterogeneity challenge is a unique and popular issue in the federated learning paradigm (Kairouz et al., 2019). According to the two stages of federated learning mentioned above, the contributions of these studies can be roughly divided into local update improvements and aggregation improvements. Our work is an improvement in local update phase, so it can be combined with existing aggregation improvements without any conflict.

As for the aggregation improvements on non-IID data, there are a series of related studies. PFNM (Yurochkin et al., 2019) and FedMA (Wang et al., 2020a) apply the Bayesian non-parametric mechanism to study the permutation invariance of the neural network, and match the neurons of client neural networks to the global neurons. Moreover, methods such as adaptive weights (Yeganeh et al., 2020), attention mechanisms (Ji et al., 2019), and normalization (Wang et al., 2020b) are also used to improve the aggregation effect for statistical heterogeneity.

There are also many studies to alleviate the negative effects of statistical heterogeneity in the local update phase. Li et al. (2020) propose the FedProx algorithm, which adds a regular term to the loss function of the client. This proximal term uses the $\ell_2$ norm to explicitly constrain the local model to be close to the latest global model, limiting the local update of non-IID clients to be not too far apart. However, this explicit constraint could inhibit FedProx from quickly finding a better model in the early stage. Similarly, based on multi-task learning, FedCurv (Shoham et al., 2019) adds penalty items for changes in important parameters related to other clients during local training. In addition, SCAFFOLD (Karimireddy et al., 2020) introduces control variables to correct the gradient of each local update and make it the same as the global update direction. In each round, the control variables are updated as the estimations of the difference between the update of the server model and the local model. However, these methods only implicitly reduce the impact of the inconsistency of objective, which is the fundamental dilemma of statistical heterogeneity, rather than eliminating it. This is exactly our motivation for proposing FedShift.

In detail, we formulate the problem and propose our method FedShift in Section 3, which also includes the convergence analysis of FedShift under different assumptions and its superiority compared with FedAvg. In Section 4, we report our experimental results, we compare the accuracy and communication efficiency of our algorithm with other algorithms, and study the influence of different settings on the algorithm, such as degree of heterogeneity, the local epoch number and clients number. Finally, Section 5 concludes our paper.

# 3 FedShift: Federated Learning with Classifier Shift

## 3.1 Problem Formulation

In federated learning, the global objective is to solve the following optimization problem:

$$\min_{\boldsymbol{w}} \left[ L(\boldsymbol{w}) \triangleq \sum_{i=1}^{N} \frac{|\mathcal{D}_i|}{|\mathcal{D}|} L_i(\boldsymbol{w}) \right], \tag{1}$$

where $L_i(\boldsymbol{w}) = \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}_i}\left[\ell_i(f(\boldsymbol{w};\boldsymbol{x}),y)\right]$ is the empirical loss of the $i$-th client that owns the local dataset $\mathcal{D}_i$, and $\mathcal{D} \triangleq \bigcup_{i=1}^{N} \mathcal{D}_i$ is a virtual entire dataset that includes all client's local data. $f(\boldsymbol{w};\boldsymbol{x})$ is the output of the model $\boldsymbol{w}$ when the input $\boldsymbol{x}$ is given, and $\ell_i$ denotes the loss function of the $i$-th client. Here, FL expects to learn a global model $\boldsymbol{w}$ that can perform well on the entire dataset $\mathcal{D}$.

However, due to the inability to communicate local data, each client usually learns a local model $\boldsymbol{w}_i$ on its local dataset by minimizing the experience loss $L_i(\boldsymbol{w}_i)$. Then the server aggregates multiple local models to obtain a global model $\bar{\boldsymbol{w}}$. In FedAvg algorithm (McMahan et al., 2017), each client adopts stochastic gradient descent (SGD) to update the local model $\boldsymbol{w}_i^{(t,\tau)}$ starting from $\boldsymbol{w}_i^{(t,0)} \triangleq \bar{\boldsymbol{w}}^{t-1}$ which is the latest global model. The local update process can be formulated as follows:

$$\boldsymbol{w}_i^{(t,\tau)} = \boldsymbol{w}_i^{(t,\tau-1)} - \eta \nabla_{\boldsymbol{w}} \ell_i(\boldsymbol{w}_i^{(t,\tau-1)}, \mathcal{B}_i^{(t,\tau)}) \tag{2}$$

where $\eta$ is the client learning rate and $\boldsymbol{w}_i^{(t,\tau)}$ denotes the local model of client $i$ after the $\tau$-th local update in the $t$-th communication round. Also, $\ell_i(\boldsymbol{w}_i^{(t,\tau-1)}, \mathcal{B}_i^{(t,\tau)}) \triangleq \sum_{(\boldsymbol{x},y)\sim\mathcal{B}_i^{(t,\tau)}} \frac{1}{|\mathcal{B}_i^{(t,\tau}|}\left[\ell_i(f(\boldsymbol{w}_i^{(t,\tau-1)};\boldsymbol{x}),y)\right]$ where $\mathcal{B}_i^{(t,\tau)}$ represents the $\tau$-th mini-batch samples of the local dataset $\mathcal{D}_i$ in the $t$-th communication round.

And then the server updates the global model by averaging the local model updates of all clients at the end of each communication round as:

$$\bar{\boldsymbol{w}}^t = \bar{\boldsymbol{w}}^{t-1} + \sum_{i=1}^{N} \frac{|\mathcal{D}_i|}{|\mathcal{D}|}(\boldsymbol{w}_i^{(t,\tau_i)} - \boldsymbol{w}_i^{(t,0)}) \tag{3}$$

where $\tau_i$ is the local iterations completed by client $i$ in the SGD optimizer with a fixed batch-size.

**Client drift** As mentioned in (Karimireddy et al., 2020; Zhao et al., 2018), the problem of client drift will occur during the federated learning process due to the statistical heterogeneity ($P_i(\boldsymbol{x},y) \neq P(\boldsymbol{x},y)$) where $P_i$ denotes the probability distribution for $(\boldsymbol{x},y)$ in local client and $P$ denotes the probability distribution for global data.

Let $\boldsymbol{w}^*$ be the global optimum of $L(\boldsymbol{w})$ and $\boldsymbol{w}_i^*$ be the optimum of each client's empirical loss $L_i(\boldsymbol{w})$. Actually, we have $\boldsymbol{w}_i^* \neq \boldsymbol{w}^*$ and $\sum_{i=1}^{N} \frac{|\mathcal{D}_i|}{|\mathcal{D}|}\boldsymbol{w}_i^* \neq \boldsymbol{w}^*$ due to the heterogeneous data distribution and our equation 1. Therefore, the direction of each local update of clients will deviate from the global update. This deviation is accumulated in multiple iterations of SGD, which will eventually lead to a drift between $\boldsymbol{w}^t$ (the true global update) and $\bar{\boldsymbol{w}}^t$ (the average of the client update aggregated by the server).

**Class Imbalance** For classification tasks, the statistical heterogeneity of federated learning is usually caused by class imbalance. Suppose the label space $Y = [1, 2, \ldots, K]$, and $P(y)$ is the probability distribution of each class. The distribution of training data can be expanded as $P(\boldsymbol{x},y) = P(\boldsymbol{x}|y)P(y)$ and $P_i(\boldsymbol{x},y) = P_i(\boldsymbol{x}|y)P_i(y)$, where $P(\boldsymbol{x}|y)$ is the conditional probability distribution of class $y$. The subscript $i$ represents the data distribution and probability of client $i$. In many real-world application scenarios of federated learning, data collected by different clients (such as IoT cameras) usually has approximately the same conditional probability distribution of each class, which implies $P_i(\boldsymbol{x}|y) \approx P(\boldsymbol{x}|y)$. Therefore, the statistical heterogeneity of federated learning often appears as class imbalance, that is, $P_i(y) \neq P(y)$.

### 3.2 Method

In order to alleviate the degradation of model performance due to class imbalance, we propose FedShift which is shown in Algorithm 1. We first start from some intuitions of our proposed method.

Note that $\boldsymbol{w}_i^* \triangleq \arg\min_{\boldsymbol{w}} \mathbb{E}_{(\boldsymbol{x},y)\sim P_i}\left[\ell_i(f(\boldsymbol{w};\boldsymbol{x}),y)\right]$ is not the optimum of the global optimization problem ($\min_{\boldsymbol{w}} \mathbb{E}_{(\boldsymbol{x},y)\sim P}\left[\ell(f(\boldsymbol{w};\boldsymbol{x}),y)\right]$), because of the statistical heterogeneity ($P \neq P_i$), although the global evaluation function is consistent with the local evaluation function ($\ell_i = \ell$). An intuitive idea can be inspired, that is, resampling or reweighting the client data to make $P = P_i$. However, since there are few or no samples in some categories, this method does not achieve good

**Algorithm 1:** FedShift

**Input:** number of communication rounds T, number of clients N, the fraction of clients C, number of local epochs E, batch size B, learning rate $\eta$, the global label distribution $P(y)$
**Output:** the global model $\boldsymbol{w}^T$

initialize $\boldsymbol{w}^0$
$m \leftarrow \max(\lfloor C * N \rfloor, 1)$
**for** *communicate round* $t = 0, 1, 2, \ldots, T-1$ **do**
  $M_t \leftarrow$ randomly select a subset containing $m$ clients
  **foreach** *client* $i \in M_t$ **do**
    $\boldsymbol{w}_i^t = \boldsymbol{w}^t$
    $\boldsymbol{w}_i^{t+1} \leftarrow$ **LocalUpdate**($\boldsymbol{w}_i^t$)
  **end**
  $\boldsymbol{w}^{t+1} = \boldsymbol{w}^t + \sum_{i \in M_t} \frac{|\mathcal{D}_i|}{|\mathcal{D}|}(\boldsymbol{w}_i^{t+1} - \boldsymbol{w}_i^t)$
**end**

**LocalUpdate** ($\boldsymbol{w}_i^t$)**:**
**for** *epoch* $e = 1, 2, \ldots, E$ **do**
  **foreach** *batch* $\mathcal{B}_i^t = (\boldsymbol{x}, y) \in \mathcal{D}_i$ **do**
    $\tilde{\ell}_i(\boldsymbol{w}_i^t, \mathcal{B}_i^t) = \sum_{(\boldsymbol{x},y)} \frac{1}{|\mathcal{B}_i^{(t,k)}|} [\ell_i(f(\boldsymbol{w}_i^t; \boldsymbol{x}) + \boldsymbol{s}_i, y)]$ **;** // $s_i$ follows Eq.5
    $\boldsymbol{w}_i^t = \boldsymbol{w}_i^t - \eta \nabla \tilde{\ell}_i(\boldsymbol{w}_i^t, \mathcal{B}_i^t)$
  **end**
**end**
**return** $\boldsymbol{w}_i^t$

results in practice. Some empirical results in Section 4 show that reweighting is not effective and even bring a severe drop in accuracy compared to FedAvg.

Different from reweighting, FedShift modifies the local optimization objective of each client to satisfy that $\tilde{\boldsymbol{w}}_i^* = \arg\min_{\boldsymbol{w}} \mathbb{E}_{(\boldsymbol{x},y)\sim P_i}\left[\tilde{\ell}_i(\boldsymbol{f}(\boldsymbol{w}; \boldsymbol{x}), y)\right]$ also is the optimum of the global optimization problem $\min_{\boldsymbol{w}} \mathbb{E}_{(\boldsymbol{x},y)\sim P}\left[\ell(\boldsymbol{f}(\boldsymbol{w}; \boldsymbol{x}), y)\right]$. Let $\tilde{\ell}_i$ denote the modified local optimization objective of client $i$. In FedShift, we add the shift $s_i$ on the classifier output of the model to modify the local optimization objective of client $i$, shown as:

$$\tilde{\ell}_i = \ell_i(\tilde{\boldsymbol{f}}(\boldsymbol{w}_i^t; \boldsymbol{x}), y) = \ell_i\left(\boldsymbol{f}(\boldsymbol{w}_i^t; \boldsymbol{x}) + \boldsymbol{s}_i, y\right) \tag{4}$$

The shift $\boldsymbol{s}_i = [s_{i,1}, s_{i,2}, \ldots, s_{i,K}]$ is calculated by the local category probability to the classifier at the end of network, as follows:

$$\boldsymbol{s}_{i,k} = \ln\left(\frac{P_i(y=k)}{P(y=k)}\right) \qquad k = 1, 2, \ldots, K \tag{5}$$

where $P(y=k) = \sum_{i=1}^{N} \frac{|\mathcal{D}_i|}{|\mathcal{D}|} P_i(y=k)$. [1] Then, we propose our following Theorem 1 to show the advantages of our FedShift theoretically.

**Theorem 1.** *For FedShift, by add shift $s_i$ in the output of model, the local optimum $\boldsymbol{w}_i^*$ satisfies the global optimum of* $\min_{\boldsymbol{w}} \mathbb{E}_{(\boldsymbol{x},y)\sim P}\left[\ell(f(\boldsymbol{w}; \boldsymbol{x}), y)\right]$.

**Proof:** For classification, cross entropy loss is the most commonly used loss function, and the output of the neural network model usually passes through a softmax function to get the predicted category

[1] The probability can be calculated using the secure aggregation algorithm (Bonawitz et al., 2016) without leaking any client information at the beginning of the entire learning process. More specifically, we use Laplace smoothing on each client to approximate the probability by frequency to guarantee secure computation of the class probability.

probability. Therefore, by definition, we have

$$\boldsymbol{w}_i^* = \arg\min_{\boldsymbol{w}} \mathbb{E}_{(\boldsymbol{x},y)\sim P_i}\left[\ell_i(\tilde{f}(\boldsymbol{w};\boldsymbol{x}),y)\right] = \arg\min_{\boldsymbol{w}} \mathbb{E}_{(\boldsymbol{x},y)\sim P_i}\left[-\sum_{k=1}^{K}\mathbb{1}_{y=k}\log\frac{e^{\tilde{f}_k(\boldsymbol{w};\boldsymbol{x})}}{\sum_{j=1}^{K}e^{\tilde{f}_j(\boldsymbol{w};\boldsymbol{x})}}\right] \tag{6}$$

Let $q_i(y=k|\boldsymbol{x};\boldsymbol{w}) \triangleq \frac{e^{\tilde{f}_k(\boldsymbol{w};\boldsymbol{x})}}{\sum_{j=1}^{K}e^{\tilde{f}_j(\boldsymbol{w};\boldsymbol{x})}}$, we can derive that the optimal model $\boldsymbol{w}_i^*$ should satisfy $q_i(y=k|\boldsymbol{x};\boldsymbol{w}_i^*) = P_i(y=k|\boldsymbol{x})$. And according to Bayes' theorem, we have $p_i(y=k|\boldsymbol{x}) = \frac{p_i(\boldsymbol{x}|y=k)p_i(y=k)}{\sum_{j=1}^{K}p_i(\boldsymbol{x}|y=j)p_i(y=j)}$, then we have

$$\tilde{f}_k(\boldsymbol{w}_i^*;\boldsymbol{x}) = \ln(p_i(\boldsymbol{x}|y=k)p_i(y=k)) + const, k=1,2,\ldots,K \tag{7}$$

Then, we consider the origin output which is added the classify shift $s_i$ in client $i$ by equation 4, we have

$$\begin{aligned} f_k(\boldsymbol{w}_i^*;\boldsymbol{x}) &= \tilde{f}_k(\boldsymbol{w}_i^*;\boldsymbol{x}) - s_{i.k} \\ &= \ln(p_i(\boldsymbol{x}|y=k)p_i(y=k)) - \ln(\frac{p_i(y=k)}{p(y=k)}) + const \\ &= \ln(p(\boldsymbol{x}|y=k)p(y=k)) + const \end{aligned} \tag{8}$$

$$q(y=k|\boldsymbol{x};\boldsymbol{w}_i^*) = \frac{e^{f_k(\boldsymbol{w}_i^*;\boldsymbol{x})}}{\sum_{j=1}^{K}e^{f_j(\boldsymbol{w}_i^*;\boldsymbol{x})}} = p(y=k|\boldsymbol{x}), k=1,2,\ldots,K \tag{9}$$

which means that $\boldsymbol{w}_i^*$ satisfies $\min_{\boldsymbol{w}} \mathbb{E}_{(\boldsymbol{x},y)\sim P}\left[\ell(f(\boldsymbol{w};\boldsymbol{x}),y)\right]$, which is the global optimum. $\square$

Note that FedProx can also be considered to have made such a modification,that is $\tilde{\ell}_i(\tilde{\boldsymbol{f}}(\boldsymbol{w};\boldsymbol{x}),y) = \ell_i(\boldsymbol{f}(\boldsymbol{w};\boldsymbol{x}),y) + \lambda\|\boldsymbol{w}-\bar{\boldsymbol{w}}\|_2^2$. but it does not guarantee that this modification has the properties shown in Theorem 1.

FedShift is designed as a simple and effective approach based on FedAvg, only introducing lightweight but novel modifications in the local training phase. Benefiting from the lightweight modifications in local training, FedShift will not damage the data privacy and add any communication cost, which potentially can be combined with other aggregation optimization approaches.

### 3.3 Convergence Analysis

Properties outlined in Theorem 1 motivate our FedShift convergence analysis. We will present theoretical results for strongly convex and non-convex functions. We first give some common assumptions about the function $L_i$ and $\nabla\ell_i(\boldsymbol{w},\mathcal{B}_i)$, which is the unbiased stochastic gradient of $L_i$.

**Assumption 1.** *For all i, $L_i$ has the properties of $\mu$-strong convexity and $\beta$-smooth:*

$$\mu\text{-strongly convex: } L_i(\boldsymbol{v}) \geq L_i(\boldsymbol{w}) + \langle(\boldsymbol{v}-\boldsymbol{w}),\nabla L_i(\boldsymbol{w})\rangle + \frac{\mu}{2}||\boldsymbol{v}-\boldsymbol{w}||_2^2$$

$$\beta\text{-smooth: } L_i(\boldsymbol{v}) \leq L_i(\boldsymbol{w}) + \langle(\boldsymbol{v}-\boldsymbol{w}),\nabla L_i(\boldsymbol{w})\rangle + \frac{\beta}{2}||\boldsymbol{v}-\boldsymbol{w}||_2^2$$

**Assumption 2.** *Bounded variances and second moments: There exits constants $\sigma > 0$ and $G > 0$ such that*

$$\mathbb{E}_{\mathcal{B}_i\sim\mathcal{D}_i}\|\nabla\ell_i(\boldsymbol{w};\mathcal{B}_i) - \nabla L_i(\boldsymbol{w})\|_2^2 \leq \sigma^2, \forall\boldsymbol{w},\forall i$$
$$\mathbb{E}_{\mathcal{B}_i\sim\mathcal{D}_i}\left[\|\nabla\ell_i(\boldsymbol{w},\mathcal{B}_i)\|_2^2\right] \leq G^2, \forall\boldsymbol{w},\forall i$$

Then, we give a lemma about the gap between the local model and the local optimum as follows, where the detailed proof is in Appendix A.

**Lemma 1.** *Under Assumption 1 and 2, we have $\mathbb{E}(||\boldsymbol{w}_i^{t+1}-\boldsymbol{w}_i^*||_2^2) \leq (1-\eta\mu)^{I+1}||\bar{\boldsymbol{w}}^t-\boldsymbol{w}_i^*||_2^2+\frac{\eta}{\mu}G^2$, where $I$ denotes the iterations of SGD for each client in each rounds.*

**Theorem 2.** *Under Assumption 1 and 2, in FedShift, we have $\mathbb{E}(||\bar{\boldsymbol{w}}^{t+1}-\boldsymbol{w}^*||_2^2) \leq (1-\eta\mu)^{(I+1)t}||\bar{\boldsymbol{w}}^0-\boldsymbol{w}^*||_2^2 + \frac{\eta\left[1-(1-\eta\mu)^{(I+1)(t+1)}\right]}{\mu\left[1-(1-\eta\mu)^{I+1}\right]}G^2$, where $\bar{\boldsymbol{w}}^{t+1} \triangleq \sum_{i=1}^{N}\frac{\boldsymbol{w}_i^{t+1}}{N}$ and $\bar{\boldsymbol{w}}^0$ is the initial global model.*

**Proof:** Following Theorem 1 and the strongly convex of $\mathcal{L}$, we can derive that $\boldsymbol{w}_i^* = \boldsymbol{w}^*$. Then, we have

$$
\begin{aligned}
\mathbb{E}(\|\bar{\boldsymbol{w}}^{t+1} - \boldsymbol{w}^*\|_2^2) &= \mathbb{E}(\|\sum_{i=1}^{N} \frac{\boldsymbol{w}_i^{t+1}}{N} - \boldsymbol{w}^*\|_2^2) = \mathbb{E}(\|\frac{1}{N}\sum_{i=1}^{N}(\boldsymbol{w}_i^{t+1} - \boldsymbol{w}^*)\|_2^2) \\
&\overset{(a)}{\leq} \frac{1}{N}\sum_{i=1}^{N}\mathbb{E}(\|(\boldsymbol{w}_i^{t+1} - \boldsymbol{w}_i^*)\|_2^2) \overset{(b)}{\leq} (1-\eta\mu)^{I+1}\|\bar{\boldsymbol{w}}^t - \boldsymbol{w}_i^*\|_2^2 + \frac{\eta}{\mu}G^2 \\
&\overset{(recurrence)}{\leq} (1-\eta\mu)^{(I+1)t}||\bar{\boldsymbol{w}}^0 - \boldsymbol{w}^*||_2^2 + \frac{\eta\left[1-(1-\eta\mu)^{(I+1)(t+1)}\right]}{\mu\left[1-(1-\eta\mu)^{I+1}\right]}G^2
\end{aligned}
\tag{10}
$$

where (a) follows from the Jensen's Inequality and $\boldsymbol{w}_i^* = \boldsymbol{w}^*$, (b) follows from Lemma 1. □

Theorem 2 shows us that under the strongly convex assumption of the function, benefiting from the classifier shift in FedShift, the global model can converge to the global optimum when there are enough iterations and communication rounds and a decayed learning rate.

However, since Theorem 1 does not hold on FedAvg, FedAvg does not have such good properties. We can get a lower bound of the gap between the global model and the global optimum in FedAvg, expressed as Theorem 3.

**Theorem 3.** *For FedAvg, in the case of non-IID client data, there is a gap between the local optimal and the global optimal. Mark* $\bar{\boldsymbol{w}}^* \triangleq \sum_{i=1}^{N}\frac{\boldsymbol{w}_i^*}{N}$*. If we assume that* $||\bar{\boldsymbol{w}}^* - \boldsymbol{w}^*||_2 = \delta > 0$*,* $||\boldsymbol{w}_i^* - \boldsymbol{w}^*||_2 = \zeta > 0$ *and* $||\bar{\boldsymbol{w}}^0 - \boldsymbol{w}^*||_2 = \gamma > 0$*, then under Assumption 1 and 2, we have* $\mathbb{E}(||\bar{\boldsymbol{w}}^{t+1} - \boldsymbol{w}^*||_2^2) \geq \frac{\delta^2}{2}$ *when I satisfies that* $I \geq \max\left\{\frac{\ln(\frac{\frac{\delta^2}{16} - \frac{\eta}{\mu}G^2}{(\frac{5\delta}{4}+\zeta)^2})}{\ln(1-\eta\mu)} - 1, \frac{\ln(\frac{\frac{\delta^2}{16} - \frac{\eta}{\mu}G^2}{(\zeta+\gamma)^2})}{\ln(1-\eta\mu)} - 1\right\}$.

**Proof:** Considering the first local update, from Lemma 1, we can get that $\mathbb{E}(\|\boldsymbol{w}_i^1 - \boldsymbol{w}_i^*\|_2^2) \leq (1-\eta\mu)^{I+1}\|\bar{\boldsymbol{w}}^0 - \boldsymbol{w}_i^*\|_2^2 + \frac{\eta}{\mu}G^2 \leq (1-\eta\mu)^{I+1}\left(\|\bar{\boldsymbol{w}}^0 - \boldsymbol{w}^*\|_2 + \|\boldsymbol{w}^* - \boldsymbol{w}_i^*\|_2\right)^2 + \frac{\eta}{\mu}G^2 \leq (1-\eta\mu)^{I+1}\left(\gamma+\zeta\right)^2 + \frac{\eta}{\mu}G^2 \leq \frac{\delta^2}{16}$. Then, we do a mathematical induction proof for $\mathbb{E}(\|\boldsymbol{w}_i^t - \boldsymbol{w}_i^*\|_2^2) \leq \frac{\delta^2}{16}$, which holds at round t. Then, we can derive

$$
\begin{aligned}
\mathbb{E}(\|\boldsymbol{w}_i^{t+1} - \boldsymbol{w}_i^*\|_2^2) &\leq (1-\eta\mu)^{I+1}\|\bar{\boldsymbol{w}}^t - \boldsymbol{w}_i^*\|_2^2 + \frac{\eta}{\mu}G^2 \\
&\leq (1-\eta\mu)^{I+1}\left(\|\bar{\boldsymbol{w}}^t - \boldsymbol{w}^*\|_2 + \|\boldsymbol{w}^* - \boldsymbol{w}_i^*\|_2\right)^2 + \frac{\eta}{\mu}G^2 \\
&\leq (1-\eta\mu)^{I+1}\left(\|\bar{\boldsymbol{w}}^t - \bar{\boldsymbol{w}}^*\|_2 + \|\bar{\boldsymbol{w}}^* - \boldsymbol{w}^*\|_2 + \zeta\right)^2 + \frac{\eta}{\mu}G^2 \\
&\leq (1-\eta\mu)^{I+1}\left(\frac{1}{N}\sum_{i=1}^{N}\|\boldsymbol{w}_i^t - \boldsymbol{w}_i^*\|_2 + \delta + \zeta\right)^2 + \frac{\eta}{\mu}G^2 \\
&\leq (1-\eta\mu)^{I+1}\left(\frac{\delta}{4} + \delta + \zeta\right)^2 + \frac{\eta}{\mu}G^2 \leq \frac{\delta^2}{16}
\end{aligned}
\tag{11}
$$

Then, we have

$$
\begin{aligned}
\mathbb{E}(\|\bar{\boldsymbol{w}}^{t+1} - \boldsymbol{w}^*\|_2^2) &= \mathbb{E}(\|\bar{\boldsymbol{w}}^{t+1} - \bar{\boldsymbol{w}}^* + \bar{\boldsymbol{w}}^* - \boldsymbol{w}^*\|_2^2) \\
&\overset{(c)}{\geq} \mathbb{E}(\|\bar{\boldsymbol{w}}^{t+1} - \bar{\boldsymbol{w}}^*\|_2 - \|\bar{\boldsymbol{w}}^* - \boldsymbol{w}^*\|_2)^2 \\
&= \mathbb{E}(\|\bar{\boldsymbol{w}}^{t+1} - \bar{\boldsymbol{w}}^*\|_2^2) + \mathbb{E}(\|\bar{\boldsymbol{w}}^* - \boldsymbol{w}^*\|_2^2) - 2\delta\mathbb{E}(\|\bar{\boldsymbol{w}}^{t+1} - \bar{\boldsymbol{w}}^*\|_2) \\
&\geq 0 + \delta^2 - 2\delta\mathbb{E}(\|\bar{\boldsymbol{w}}^{t+1} - \bar{\boldsymbol{w}}^*\|_2) \\
&\overset{(d)}{\geq} \delta^2 - \frac{2\delta}{N}\sum_{i=1}^{N}\mathbb{E}(\|\boldsymbol{w}_i^{t+1} - \boldsymbol{w}_i^*\|_2) \\
&\overset{(e)}{\geq} \delta^2 - 2\delta\sqrt{\frac{\delta^2}{16}} = \frac{\delta^2}{2} > 0
\end{aligned}
\tag{12}
$$

where (c,d,e) follow the Triangle Inequality, the Jensen's Inequality and equation 11 respectively. □

Furthermore, we consider the convergence of FedShift in the non-convex case, expressed as Theorem 4. The detailed proof is in Appendix A.

**Theorem 4.** *Under assumption 1 and 2, and removing the $\mu$-strongly convex assumption, we have* $\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}(||\nabla_{\boldsymbol{w}}L(\bar{\boldsymbol{w}}^{t-1})||_2^2) \le \frac{2}{\eta T}(L(\bar{\boldsymbol{w}}^0) - L(\bar{\boldsymbol{w}}^*)) + 4\eta^2 I^2 G^2 \beta^2 + \frac{\beta}{N}\eta\sigma^2$.

# 4 Experiments

## 4.1 Experimental Setup

**Datasets and Models** We conduct experiments on three public datasets including Cifar10 (60,000 images with 10 classes) (Krizhevsky, Hinton, et al., 2009), Cinic10 (270,000 images with 10 classes) (Darlow et al., 2018), and Tiny-Imagenet (100,000 images with 200 classes) (Le and Yang, 2015). We follow the setting in (Wang et al., 2019; Yurochkin et al., 2019) to generate the non-IID data partition by using Dirichlet distribution. Specifically, for class $c$, we sample $p_c \sim Dir_N(\alpha)$, where $p_{c,i}$ represents the proportion of data with category $k$ allocated to client $i$. The smaller $\alpha$ means the heavier statistical heterogeneity. For all experiments unless there are special instructions, we set $\alpha = 0.1$ and the number of clients $N = 10$ by default. In order to show that the algorithm is feasible on the actual deep learning model, we use ResNet18 (He et al., 2016) as our network architecture for Cifar10 and Cinic10. For Tiny-Imagenet, we use ResNet50 (He et al., 2016) to deal with more complex data.

**Baselines** We compare FedShift with three state-of-the-art approaches which are the most relevant to us, including FedAvg (McMahan et al., 2017), FedProx (Li et al., 2020), and SCAFFOLD (Karimireddy et al., 2020) on all three datasets.

**Implementation** We use PyTorch (Paszke et al., 2019) to implement FedShift and the other baselines. We use the SGD with momentum as our optimizer for all experiments, where the SGD weight decay is set to $0.0001$ and the momentum to $0.9$. We adjust batchsize $B = 40$ and the learning rate $\eta = 0.01$ with a decay rate $0.95$ for every $10$ communication rounds. We take the best performance of each method for comparison.

## 4.2 Accuracy Comparison

For each dataset, we tune the number of local epochs $E$ from $\{1, 5, 10, 20\}$ based on FedAvg, and choose the best $E$ as the hyperparameter of other algorithms. The best $E$ for Cifar10, Cinic10, and Tiny-Imagenet are $5$, $1$, and $1$, respectively. Besides, for FedProx, we tune $\lambda$ from $\{0.001, 0.01, 0.1\}$, which is a hyperparameter to control the weight of its proximal term. The best $\lambda$ of FedProx for Cifar10, Cinic10, and Tiny-Imagenet are $0.01$, $0.001$, and $0.01$, respectively. Unless explicitly specified, we use $E$ and $\lambda$ for all the remaining experiments. The number of communication rounds is set to $100$ for Cifar10, $150$ for Cinic10 and $50$ for Tiny-Imagenet, where all federated learning approaches have little or no accuracy gain with more communications.

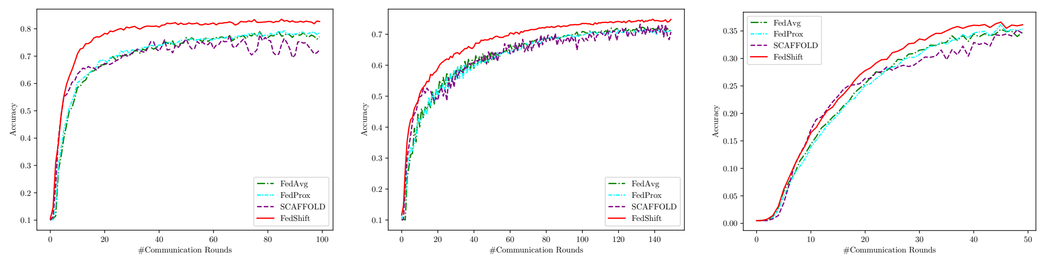
Table 1: The accuracy of Reweight, FedShift and three baselines (FedAvg, FedProx and SCAFFOLD) on three test datasets (Cifar10, Cinic10 and Tiny-Imagenet).

| Methods | Cifar10 | Cinic10 | Tiny-Imagenet |
|---|---|---|---|
| FedAvg | 78.94% | 72.26% | 35.14% |
| FedProx | 79.33% | 71.57% | 36.16% |
| SCAFFOLD | 77.75% | 73.22% | 35.18% |
| FedShift(ours) | **83.52%** | **74.86%** | **36.61%** |
| Reweighting | 63.15% | 30.06% | 13.25% |

Table 1 shows the test accuracy of all approaches with the above settings. Comparing different federated learning approaches, we can observe that FedShift is the best approach among all tasks, which even can outperform FedAvg by $4.58\%$ accuracy on Cifar10. For FedProx and SCAFFOLD, they are only superior to FedAvg in specific datasets and do not have a significant improvement. Reweighting has much worse accuracy than other methods as we mentioned in Section 3.

### 4.3 Discussion of Communication Efficiency

Figure 1 shows the accuracy in each communication round during training. As we can see, FedShift obviously has a faster convergence speed and higher accuracy compared with the other three methods. Moreover, unlike the other three methods, FedShift has a more stable upward curve due to the same optimization objective in all clients.




(a) Cifar10 (b) Cinic10 (c) Tiny-Imagenet

Figure 1: The test accuracy in each communication round during training.

In order to verify the communication efficiency of different approaches, we show the number of communication rounds to achieve the same accuracy for FedAvg in Table 2. We can observe that the number of communication rounds is significantly reduced in FedShift. FedShift only needs less than a quarter of the number of rounds to reach the accuracy of FedAvg on the Cifar10. The speedup of FedShift is also significant on Cinic10 or Tiny-Imagenet. Therefore, we can consider FedShift is much more communication efficient than the other approaches.

Table 2: The number of rounds of FedShift and three baselines (FedAvg,FedProx and SCAFFOLD) to achieve a consistent accuracy on three test datasets (Cifar10, Cinic10 and Tiny-Imagenet) respectively.

| Method | Cifar10 | | Cinic10 | | Tiny-Imagenet | |
|---|---|---|---|---|---|---|
| | rounds | speedup | rounds | speedup | rounds | speedup |
| FedAvg | 100 | $1\times$ | 150 | $1\times$ | 50 | $1\times$ |
| FedProx | 85 | $1.18\times$ | \ | $< 1\times$ | 46 | $1.09\times$ |
| SCAFFOLD | \ | $< 1\times$ | 133 | $1.13\times$ | 49 | $1.02\times$ |
| FedShift (ours) | **23** | $\mathbf{4.34}\times$ | **81** | $\mathbf{1.85}\times$ | **37** | $\mathbf{1.35}\times$ |

### 4.4 Impact of Local Epochs

Table 3: The top-1 accuracy of FedShift and three baselines (FedAvg,FedProx and SCAFFOLD) on Cifar10 dataset with different number of local epochs.

| Methods | E=1 | E=5 | E=10 | E=20 |
|---|---|---|---|---|
| FedAvg | 75.25% | 78.94% | 76.47% | 72.28% |
| FedProx | 75.31% | 79.33% | 77.27% | 76.19% |
| SCAFFOLD | 70.60% | 77.75% | 77.95% | 78.04% |
| FedShift (ours) | **80.61%** | **83.52%** | **82.67%** | **81.63%** |

We next focus on the effect of the number of local epochs on Cifar10. The results are shown in Table 3. When the number of local epochs is 1, the local update is tiny, which leads to lower accuracy than more local epochs' results. However, FedShift still has the best accuracy. When the number of local epochs becomes too large, the accuracy of all approaches drops unless SCAFFOLD, which is due to the overfitting of local updates. Moreover, benefiting from the proximal term, FedProx performs better than FedAvg in all settings. Note that SCAFFOLD is far inferior to other algorithms when $E = 1$, and has higher accuracy as the number of local epochs increases. Because the control variables in SCAFFOLD can be estimated more accurately with more local updates, SCAFFOLD has a higher tolerance for the number of local epochs. Nevertheless, FedShift clearly outperforms the other approaches. This further verifies that FedShift can effectively mitigate the negative effects of the accumulative client drift.

### 4.5 Impact of Data Heterogeneity

Table 4: The top-1 accuracy of FedShift and three baselines (FedAvg,FedProx and SCAFFOLD) on Cifar10 dataset with different parameter $\alpha$ of dirichlet distribution.

| Methods | $\alpha$=0.1 | $\alpha$=0.15 | $\alpha$=0.2 | $\alpha$=0.5 |
|---|---|---|---|---|
| FedAvg | 78.94% | 80.11% | 89.18% | 91.27% |
| FedProx | 75.95% | 82.10% | 89.43% | 91.21% |
| SCAFFOLD | 77.75% | 83.11% | **90.99%** | **92.22%** |
| FedShift (ours) | **83.52%** | **86.21%** | 90.76% | 91.26% |

Data heterogeneity is changed in this numerical study by varying the concentration parameter $\alpha$ of Dirichlet distribution on Cifar10. The results are shown in Table 4. For a smaller $\alpha$, the partition will be more unbalanced, we can significantly see the effectiveness of FedShift. When the unbalanced level decreases (i.e., $\alpha = 0.5$), all approaches have similar accuracy, and the control variable in SCAFFOLD actually degenerates into more considerable momentum to obtain higher accuracy.

### 4.6 Impact of the Number of Clients

Table 5: The top-1 accuracy of FedShift and three baselines (FedAvg,FedProx and SCAFFOLD) on Cifar10 dataset with different number of clients.

| Method | N=10,C=1.0 | N=20,C=0.5 | N=50,C=0.2 |
|---|---|---|---|
| FedAvg | 78.94% | 79.49% | 55.22% |
| FedProx | 79.33% | 79.57% | 56.24% |
| SCAFFOLD | 77.75% | 78.79% | **64.92%** |
| FedShift (ours) | **83.52%** | **83.13%** | 64.69% |

To show the scalability of FedShift, we try more number of clients on Cifar10, including two settings: 20 clients and 50 clients. For better comparison, we adjust the proportion of clients participating in training in each round so that there are exactly 10 clients each time, where $C = 0.5/0.2$ for $N = 20/50$. The communication round remains the same as the previous experiment, which is 100 rounds. The results are shown in Table 5. FedShift achieves higher accuracy than FedAvg and FedProx in the different number of clients. Moreover, SCAFFOLD even outperforms FedShift with 50 clients and the fraction $C = 0.2$ because of its estimation of the global gradient even if the gradients of some clients participating in training are small.

## 5 Conclusion

Focusing on the class imbalance in the statistical heterogeneity of federated learning, we propose FedShift in this paper, which is a simple and effective method that adds the shift on the classifier output based on the client class distribution in the local training phase. Then, we theoretically prove that the classifier shift in FedShift make the local optimal model satisfies the global optimum. Additionally, we prove the convergence of the FedShift algorithm and compare with FedAvg. We also conduct numerical studies, and the experimental results show that FedShift significantly outperforms the popular state-of-the-art algorithms on various datasets. Finally, as a future prospect, Fedshift has the potential to combine with the research of feature representation to deal with the inconsistency of the category conditional probabilities in each client, which is relaxation of our assumptions in our work.

# References


Keith Bonawitz, Vladimir Ivanov, Ben Kreuter, Antonio Marcedone, H Brendan McMahan, Sarvar Patel, Daniel Ramage, Aaron Segal, and Karn Seth. Practical secure aggregation for federated learning on user-held data. *arXiv preprint arXiv:1611.04482*, 2016.

Luke N Darlow, Elliot J Crowley, Antreas Antoniou, and Amos J Storkey. Cinic-10 is not imagenet or cifar-10. *arXiv preprint arXiv:1810.03505*, 2018.

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

Shaoxiong Ji, Shirui Pan, Guodong Long, Xue Li, Jing Jiang, and Zi Huang. Learning private neural language modeling with attentive aggregation. In *2019 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2019.

Shaoxiong Ji, Teemu Saravirta, Shirui Pan, Guodong Long, and Anwar Walid. Emerging trends in federated learning: From model fusion to federated x learning. *arXiv preprint arXiv:2102.12920*, 2021.

Peter Kairouz, H Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Keith Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, et al. Advances and open problems in federated learning. *arXiv preprint arXiv:1912.04977*, 2019.

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

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

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

Ya Le and X. Yang. Tiny imagenet visual recognition challenge. 2015.

Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. In I. Dhillon, D. Papailiopoulos, and V. Sze, editors, *Proceedings of Machine Learning and Systems*, volume 2, pages 429–450, 2020. URL `https://proceedings.mlsys.org/paper/2020/file/38af86134b65d0f10fe33d30dd76442e-Paper.pdf`.

Yang Liu, Anbu Huang, Yun Luo, He Huang, Youzhi Liu, Yuanyuan Chen, Lican Feng, Tianjian Chen, Han Yu, and Qiang Yang. Fedvision: An online visual object detection platform powered by federated learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 13172–13179, 2020.

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

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in Neural Information Processing Systems*, 32: 8026–8037, 2019.

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.

Paul Voigt and Axel Von dem Bussche. The eu general data protection regulation (gdpr). *A Practical Guide, 1st Ed., Cham: Springer International Publishing*, 10:3152676, 2017.

Hongyi Wang, Mikhail Yurochkin, Yuekai Sun, Dimitris Papailiopoulos, and Yasaman Khazaeni. Federated learning with matched averaging. In *International Conference on Learning Representations*, 2019.

Hongyi Wang, Mikhail Yurochkin, Yuekai Sun, Dimitris Papailiopoulos, and Yasaman Khazaeni. Federated learning with matched averaging. In *International Conference on Learning Representations*, 2020a.

Jianyu Wang, Qinghua Liu, Hao Liang, Gauri Joshi, and H Vincent Poor. Tackling the objective inconsistency problem in heterogeneous federated optimization. *Advances in Neural Information Processing Systems*, 33, 2020b.

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

Mikhail Yurochkin, Mayank Agarwal, Soumya Ghosh, Kristjan Greenewald, Nghia Hoang, and Yasaman Khazaeni. Bayesian nonparametric federated learning of neural networks. In *International Conference on Machine Learning*, pages 7252–7261. PMLR, 2019.

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.

# Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes] See Section3.2 and 4.
   (b) Did you describe the limitations of your work? [Yes] See 5.
   (c) Did you discuss any potential negative societal impacts of your work? [No]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes] See Section 3.3.
   (b) Did you include complete proofs of all theoretical results? [Yes] See Section 3.3 and appendix.
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] All datasets we used are public datasets, and the code is in supplementary material.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See Section 4.1.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] See appendix.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] We use 8 GPUs with GTX 1080Ti (12GB).
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] See Section 4.1.
   (b) Did you mention the license of the assets? [Yes] See Section 4.1.
   (c) Did you include any new assets either in the supplemental material or as a URL? [No] We don't use any new assets.
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [No]

(e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [No]

5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [No]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [No]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [No]