

APPENDIX

A TERMINOLOGIES

Global model vs. local model. Let us first clarify the concepts of “global” vs. “local” models: in each communication round, local models denote the ones updated by the clients after local training, and the global model denotes the model obtained by aggregating all local models at the server. Moreover, client models denote the models being trained during local training.

Vicious cycle vs. virtuous cycle. As shown in Figure 1 (a), the *vicious cycle* represents the phenomenon that inconsistent feature mappings of local models diverge the classifier updates, such that the diverged classifiers of different clients induce feature extractors to map to more inconsistent features across clients. As shown in Figure 1 (b), the *virtuous cycle* represents the phenomenon that consistent feature mappings of client local models make the classifier updates similar, such that the updated classifiers make feature extractors of clients map to more consistent features across clients.

Positive pair vs. negative pair. A *positive pair* denotes a pair of **samples** with the same label (i.e., the samples belong to the same class). A *negative pair* denotes a pair of **samples** with different labels (i.e., the samples do not belong to the same class).

Positive feature vs. negative feature. For the c -th proxy ϕ_c , the *positive features* denote the **features** of the c -th class, and the *negative features* denote the **features** of other classes except for the c -th class.

Positive proxy vs. negative proxy. For the feature of the c -th class, the *positive proxy* denotes the c -th **proxy** ϕ_c , and the *negative proxies* denote other **proxies** except for the c -th proxy ϕ_c .

Label distribution skew vs. feature distribution skew. Feature and label distribution skews are two representative data heterogeneity (Kairouz et al., 2021), both covered in this work. Suppose that the i -th client data distribution follows $P_i(x, y) = P_i(x|y)P_i(y) = P_i(y|x)P_i(x)$ where x and y denote the feature and label respectively. Following (Li et al., 2021a), the definition of feature distribution skew and label distribution skew can be given as:

- **Label distribution skew (prior probability):** The label marginal distribution $P_i(y)$ varies across clients while $P_i(x|y) = P_j(x|y)$ for all clients i and j .
- **Feature distribution skew (covariate shift):** The input feature marginal distribution $P_i(x)$ varies across clients while $P_i(y|x) = P_j(y|x)$ for all clients i and j .
- **Label & feature distribution skew:** At least one of the label distribution skew and feature distribution skew happens across clients. This means clients i and j still suffer from label marginal distribution skew $P_i(y)$ even if sharing the same $P_i(x)$, or clients i and j still suffer from input feature marginal distribution skew $P_i(x)$ even if sharing the same $P_i(y)$.
- **Local data distributions without skew:** Herein, for FMNIST, CIFAR-10, and CIFAR-100, we split them evenly into client-side local datasets based on an identical label distribution. For Mixed Digits, we first mix all the digit datasets as a global dataset and evenly split it into client-side local datasets based on an identical label distribution. Note that this case can not guarantee that local distributions share the same global distribution across clients. Still, it means local distributions are more homogeneous than the cases of label or feature distribution skew.

Center loss vs. feature anchor loss. Our feature anchor loss borrows the idea of the center loss (Wen et al., 2016), but the purposes of these two losses are different. Besides with a different updating method, the center loss aims to decrease the feature distance for intra-class samples in centralized training, rather than keeping feature mapping consistent across clients. Meanwhile, the feature anchors are not utilized to calibrate the classifier in (Wen et al., 2016), but our work does it to prevent the divergence of classifiers across clients in federated training. Therefore, we use the feature anchor loss to distinguish the center loss proposed by the research community on face recognition.

B RELATED WORKS

Federated learning is a fast-developing area, and we mainly introduce the methods close to ours (i.e., federated optimization-based methods) and briefly introduce other methods. Comprehensive field studies have appeared in (Kairouz et al., 2021; Wang et al., 2021; Tan et al., 2022a).

Tackle data heterogeneity on the client side. To avoid local models converging to their local minima instead of global minima, many works add a well-designed regularization term to penalize local models to make them not far away from the global model. For example, FedProx (Li et al., 2020) uses the Euclidean distance between local models and the global model as the regularization loss. FedDyn (Acar et al., 2021) modifies the local objective with a dynamic regularizer consisting of a linear term based on the first order condition and the Euclidean-distance term, such that the local minima are consistent with the global stationary point. MOON (Li et al., 2021b) utilizes the feature similarity between previous local models and the global model as model-contrastive regularization to correct the local training of each client. In place of the model-contrastive term in MOON, FedProc (Mu et al., 2021) introduces a prototype-contrastive term to regularize the features within each class with class prototypes (Snell et al., 2017). Besides, instead of implicit correction by regularization, a number of works reduce the bias explicitly in local updates by controlling variates or posterior sampling. Borrowing the variance-reduce technique in standard convex optimization, SCAFFOLD (Karimireddy et al., 2020) presents a control variate to correct the client updates, so they are much closer to the global update. Another way to reduce the bias is to run Markov Carlo, instead of stochastic gradient descent, which produces approximate local posterior samples like FedPA (Al-Shedivat et al., 2021). Similar to (Li et al., 2021a; Chen & Chao, 2022; Luo et al., 2021; He et al., 2021), our experiments in Section 5 show that these works may not provide stable better performance gains over FedAvg (McMahan et al., 2017) in classification tasks, which motivates us to analyze the relationship between classifier updates and feature mappings in local training.

Tackle data heterogeneity on the server side. In addition to improving on the client side, many works have developed alternative aggregation schemes on the server side to tackle data heterogeneity. For instance, (Wang et al., 2020b) finds an objective inconsistency problem caused by unbalanced data that induces a different number of local updates and propose FedNova to eliminate the inconsistency by normalizing the local updates before averaging. Besides, (Reddi et al., 2021) adopts adaptive momentum update on the server-side to mitigate oscillation of global model updates when the server activates the clients with a limited subset of labels. Beyond layer-weighted averaging, some works like FedMA (Wang et al., 2020a) and Fed² (Yu et al., 2021) introduce neuron-wise averaging because there may exist neuron mismatching from permutation invariance of neural networks in federated learning. These ideas complement our work and can be integrated into our method because our method only adds a regularizer on the client side.

Tackle data heterogeneity based on feature or classifier. Instead of considering from the federated-optimization view, some recent works such as (Li et al., 2021b; Mu et al., 2021; Li & Zhan, 2021; Luo et al., 2021; Zhang et al., 2022; Tang et al., 2022) pay more attention to feature space across clients. To improve feature consistency, MOON (Li et al., 2021b) and FedProc (Mu et al., 2021) introduce a feature-based local regularizer mentioned above. Meanwhile, (Tang et al., 2022) generates a shared virtual dataset for all clients before training, and calibrates features by minimizing the feature distribution distance between the virtual dataset and the real dataset. To improve classifier consistency, (Luo et al., 2021) observes that the classifier layer (i.e., the last layer of the model) suffers most from label distribution skew and proposes calibration of the classifier with virtual features after training. Moreover, (Li & Zhan, 2021; Zhang et al., 2022) introduce a restricted loss cross-entropy and a fine-grained calibrated cross-entropy loss, respectively. The key idea of the two methods is to prevent the overfitting of missing classes (Li & Zhan, 2021) and minority classes (Zhang et al., 2022) (i.e., both under the label distribution skew) with an improved cross-entropy loss. However, compared with our method, these methods only consider the label distribution skew setting by improving the performance degeneration based on feature calibration or classifier calibration, and neglect *vicious cycle* between different classifier updates and inconsistent features, which hurts their performance.

Other methods. The data-centric method is one recent new direction, which shares common datasets with all clients like the public dataset in (Zhao et al., 2018). To avoid violating the privacy requirement, some works focus on sharing synthesized data like (Luo et al., 2021; Li et al., 2022; Tang et al., 2022) and coded data (Sun et al., 2022; Shao et al., 2022) with privacy protection to

construct a more homogeneous dataset for federated learning. Moreover, another line of research aims to train a personalized model for each client, rather than a global model (Tan et al., 2022a). Since there is still no standard approach to personalized federated learning, many researchers achieve it by personalized regularization (T Dinh et al., 2020), meta learning (Fallah et al., 2020), prototype learning (Tan et al., 2022b) and personalized layers (Chen & Chao, 2022), etc.

Our work aims at the typical federated learning (McMahan et al., 2017) and tries to improve the local optimization by feature alignment in federated optimization. There are two existing works similar to ours, i.e., MOON (Li et al., 2021b) which introduces a model-contrastive loss to maximize the agreement of the features extracted by the local model and that by the global model, and FedProc (Mu et al., 2021) which proposes a prototype-contrastive loss to correct features by class prototypes. However, compared with our method, although considering the feature mapping inconsistency across local models, MOON and FedProc neglect the *vicious cycle* between different classifiers' updates and inconsistent features, which hurts their performance.

C PROOF OF PROPERTY 1

Proof 1 (Property 1) Let $A = \sum_{j_i=1, y_{j_i}=c}^{n_i} (1 - p_{j_i,c}^{(i)}) \mathbf{h}_{i,y_{j_i}} - \sum_{j_i=1, y_{j_i} \neq c}^{n_i} p_{j_i,c}^{(i)} \mathbf{h}_{i,y_{j_i}}$, and $B = \sum_{j_v=1, y_{j_v}=c}^{n_v} (1 - p_{j_v,c}^{(v)}) \mathbf{h}_{v,y_{j_v}} - \sum_{j_v=1, y_{j_v} \neq c}^{n_v} p_{j_v,c}^{(v)} \mathbf{h}_{v,y_{j_v}}$.

We compute the similarity of classifier updates between client i and client v ,

$$\cos(\Delta\phi_{i,c}, \Delta\phi_{v,c}) = \frac{\langle \Delta\phi_{i,c}, \Delta\phi_{v,c} \rangle}{\|\Delta\phi_{i,c}\| \|\Delta\phi_{v,c}\|} = \frac{\eta^2 \langle A, B \rangle}{n_i n_v \|A\| \|B\|}. \quad (10)$$

Let $A_1 = \sum_{j_i=1, y_{j_i}=c}^{n_i} (1 - p_{j_i,c}^{(i)}) \mathbf{h}_{i,y_{j_i}}$, $A_2 = \sum_{j_i=1, y_{j_i} \neq c}^{n_i} p_{j_i,c}^{(i)} \mathbf{h}_{i,y_{j_i}}$, $B_1 = \sum_{j_v=1, y_{j_v}=c}^{n_v} (1 - p_{j_v,c}^{(v)}) \mathbf{h}_{v,y_{j_v}}$ and $B_2 = \sum_{j_v=1, y_{j_v} \neq c}^{n_v} p_{j_v,c}^{(v)} \mathbf{h}_{v,y_{j_v}}$.

Herein, we represent $\langle A, B \rangle$ as:

$$\begin{aligned} \langle A, B \rangle &= \langle A_1 - A_2, B_1 - B_2 \rangle \\ &= \langle A_1, B_1 \rangle + \langle A_2, B_2 \rangle - \langle A_1, B_2 \rangle - \langle A_2, B_1 \rangle. \end{aligned} \quad (11)$$

According to Assumption 1, when $\mathbf{h}_c \in \mathcal{H}_c$ and $\mathbf{h}_q \in \mathcal{H}_q$ and $c \neq q$, the inner product $\langle \mathbf{h}_c, \mathbf{h}_q \rangle$ is less than or equal to 0. Thus, we have:

$$\begin{aligned} \langle A_1, B_2 \rangle &= \left\langle \sum_{j_i=1, y_{j_i}=c}^{n_i} (1 - p_{j_i,c}^{(i)}) \mathbf{h}_{i,y_{j_i}}, \sum_{j_v=1, y_{j_v} \neq c}^{n_v} p_{j_v,c}^{(v)} \mathbf{h}_{v,y_{j_v}} \right\rangle \\ &= \sum_{j_v=1, y_{j_v} \neq c}^{n_v} \sum_{j_i=1, y_{j_i}=c}^{n_i} (1 - p_{j_i,c}^{(i)}) p_{j_v,c}^{(v)} \langle \mathbf{h}_{i,y_{j_i}}, \mathbf{h}_{v,y_{j_v}} \rangle \\ &\leq 0 \end{aligned} \quad (12)$$

where the inequality holds because $y_{j_i} = c$ but $y_{j_v} \neq c$ and thus $\langle \mathbf{h}_{i,y_{j_i}}, \mathbf{h}_{v,y_{j_v}} \rangle \leq 0$.

Similarly, we have:

$$\begin{aligned} \langle A_2, B_1 \rangle &= \left\langle \sum_{j_i=1, y_{j_i} \neq c}^{n_i} p_{j_i,c}^{(i)} \mathbf{h}_{i,y_{j_i}}, \sum_{j_v=1, y_{j_v}=c}^{n_v} (1 - p_{j_v,c}^{(v)}) \mathbf{h}_{v,y_{j_v}} \right\rangle \\ &\leq 0. \end{aligned} \quad (13)$$

Also, according to Assumption 1, when $\|\mathbf{a}_c\| > \sqrt{2}\delta$, for any $\mathbf{h}_{i,c} \in \mathcal{H}_c$ and $\mathbf{h}_{v,c} \in \mathcal{H}_c$, we obtain that the inner product $\langle \mathbf{h}_{i,c}, \mathbf{h}_{v,c} \rangle$ is larger than 0 since the arccosine of largest angle between

$\mathbf{h}_{i,c}$ and $\mathbf{h}_{v,c}$ is $2\arccos(\|\mathbf{a}_c\|/\sqrt{2}\delta)$ (i.e., the largest angle is smaller than $\pi/2$).

$$\begin{aligned}
\langle A_1, B_1 \rangle &= \langle \sum_{j_i=1, y_{j_i}=c}^{n_i} (1 - p_{j_i,c}^{(i)}) \mathbf{h}_{i,y_{j_i}}, \sum_{j_v=1, y_{j_v}=c}^{n_v} (1 - p_{j_v,c}^{(v)}) \mathbf{h}_{v,y_{j_v}} \rangle \\
&= \sum_{j_v=1, y_{j_v}=c}^{n_v} \sum_{j_i=1, y_{j_i}=c}^{n_i} (1 - p_{j_i,c}^{(i)}) (1 - p_{j_v,c}^{(v)}) \langle \mathbf{h}_{i,y_{j_i}}, \mathbf{h}_{v,y_{j_v}} \rangle \\
&> 0.
\end{aligned} \tag{14}$$

Similarly, we have:

$$\begin{aligned}
\langle A_2, B_2 \rangle &= \langle \sum_{j_i=1, y_{j_i} \neq c}^{n_i} p_{j_i,c}^{(i)} \mathbf{h}_{i,y_{j_i}}, \sum_{j_v=1, y_{j_v} \neq c}^{n_v} p_{j_v,c}^{(v)} \mathbf{h}_{v,y_{j_v}} \rangle \\
&> 0.
\end{aligned} \tag{15}$$

Combining (11) to (15), we obtain that $\langle A, B \rangle$ is larger than 0, and thus $\cos(\Delta\phi_{i,c}, \Delta\phi_{v,c}) > 0$.

D PSEUDOCODE OF FEDFA

Herein, we get the inspiration from Batch Normalization (Ioffe & Szegedy, 2015) and perform moving average to estimate $\bar{\mathbf{h}}_{c,i}^{(t,k)}$ based on the c -th class samples $\mathcal{B}_{i,c}$ of one mini batch \mathcal{B}_i when performing prediction in (8), as presented as following:

$$\bar{\mathbf{h}}_{c,i}^{(t,k)} = \frac{1}{2} \sum_{\tau}^B (\bar{\mathbf{h}}_{c,i}^{(t,k_{\tau-1})} + \bar{\mathbf{h}}_{c,i}^{(t,k_{\tau})})$$

where $\bar{\mathbf{h}}_{c,i}^{(t,k_{\tau})} = \frac{1}{|\mathcal{B}_{i,c}^{(t,k_{\tau})}|} \sum_{(\mathbf{x},c) \in \mathcal{B}_{i,c}^{(t,k_{\tau})}} f_{\theta_i}(\mathbf{x})$, $\bar{\mathbf{h}}_{c,i}^{(t,k_0)} = \bar{\mathbf{h}}_{c,i}^{(t,k-1)}$, and $\tau \in [1, B]$ denotes the τ -th mini batch of the total number B at the k -th epoch.

The pseudocode of FedFA is shown as the following Algorithm 1. Compared with FedAvg, FedFA adds a feature anchor loss and calibrates the classifier locally.

E DETAILS OF EXPERIMENT SETUP

E.1 SPECIFIC MODELS

Our validation and test experiments, including label distribution skew, feature distribution skew and label & feature distribution skews, use the models according to Table 4. Herein, to ablate the effect of BN layers, we follow (Hsieh et al., 2020) to replace the BN layer with the GroupNorm layer in all experiments except for the test under Mixed-digit datasets. For a fair comparison, our models follow those reported in the baselines' works. Specifically, following (Acar et al., 2021), we use a CNN model for EMNIST, FMNIST, and CIFAR-10, consisting of two 5x5 convolution layers followed by 2x2 max pooling and two fully-connected layers with ReLU activation. Following (Li et al., 2021b) and (Li et al., 2021c), we utilize the ResNet-18 (He et al., 2016) with a linear projector for CIFAR-100 and a CNN model with three 5x5 convolution layers followed by five batch normalization (BN) layers for the Mixed Digits dataset.

E.2 VALIDATION EXPERIMENT SETUP

The total number of training samples per client is 1000 in this case. We separately sample a subset from test sets of FMNIST and Mixed Digit to visualize the normalized feature mappings of the local models based on t-SNE visualization (Van der Maaten & Hinton, 2008). In Figure 2, although we input the same Validation samples into all clients' local modes, we only show their features mappings for which clients have the corresponding class (i.e., if client 1 only holds class 1 and class 2 samples,

Algorithm 1 FedFA (Proposed Framework): Federated Learning with Feature Alignment

Input: initial model $\mathbf{w} = \{\theta, \phi\}$, initial feature anchors $\{\mathbf{a}_c\}_{c=1}^C$, learning rate η , local epoch K , client number N , class number C

for each round $t = 1, \dots, R$ **do**

 Server samples clients $\mathcal{S} \subseteq \{1, \dots, N\}$

 Server communicates $\mathbf{w}^{(t-1)}$ and $\{\mathbf{a}_c^{(t-1)}\}_{c=1}^C$ to all clients $i \in \mathcal{S}$

on client $i \in \mathcal{S}$ **in parallel do**

 Initialize the local model $\mathbf{w}_i \leftarrow \mathbf{w}^{(t-1)}$, the local feature anchor $\mathbf{a}_{c,i} \leftarrow \mathbf{a}_c^{(t-1)}$

for local epoch $k = 1, \dots, K$ **do**

 Calculate the local loss $l_i \leftarrow l_{\text{sup}_i} + \mu l_{\text{fa}_i}$ according to (6)

 Compute mini-batch gradient $g_i(\mathbf{w}_i) \leftarrow \nabla_{\mathbf{w}_i} l_i$

 Update local model $\mathbf{w}_i \leftarrow \mathbf{w}_i - \eta g_i(\mathbf{w}_i)$

 Calibrate classifiers with feature anchors according to (7)

 Accumulate dynamic momentum $\mathbf{m}_i^{(t,k)}$ according to (8)

end for

 Update local feature anchors $\mathbf{a}_{c,i}^{(t)} \leftarrow \sum_{k=1}^K \mathbf{m}_{c,i}^{(t,k)}$

 Communicate $\mathbf{w}_i^{(t)}$ and $\{\mathbf{a}_{c,i}^{(t)}\}_{c=1}^C$ back to the server

end on client

 Server aggregates the global model $\mathbf{w}^{(t)} \leftarrow \frac{1}{|\mathcal{S}|} \sum_{i \in \mathcal{S}} \mathbf{w}_i^{(t)}$, and the feature anchors $\mathbf{a}_c^{(t)} \leftarrow \frac{1}{|\mathcal{S}|} \sum_{i \in \mathcal{S}} \mathbf{a}_{c,i}^{(t)}$

end for

Table 4: The specific parameters settings for all the models used in our experiments.

Layer	Validation Experiment		Test Experiment			
	Label Skew	Feature Skew	Label Skew		Feature Skew	
	FMNIST	Mixed-digit dataset	FMNIST/EMNIST	CIFAR-10	CIFAR-100	Mixed-digit dataset
1	Conv2d(1, 32, 5) ReLU, MaxPool2D(2,2)	Conv2d(3, 64, 5) ReLU, MaxPool2D(2,2)	Conv2d(1, 32, 5) ReLU, MaxPool2D(2,2)	Conv2d(3, 64, 5) ReLU, MaxPool2D(2,2)	Basicbone of Resnet18 with GroupNorm	Conv2d(3, 64, 5, 1, 2) BN(64), ReLU, MaxPool2D(2,2)
2	Conv2d(32, 32, 5) ReLU, MaxPool2D(2,2)	Conv2d(64, 64, 5) ReLU, MaxPool2D(2,2)	Conv2d(32, 32, 5) ReLU, MaxPool2D(2,2)	Conv2d(64, 64, 5) ReLU, MaxPool2D(2,2)	FC(512, 512) ReLU	Conv2d(64, 64, 5, 1, 2) BN(64), ReLU, MaxPool2D(2,2)
3	FC(992, 384) ReLU	FC(1024, 384) ReLU	FC(992, 384) ReLU	FC(1600, 384) ReLU	FC(512, 256)	Conv2d(3, 128, 5, 1, 2) BN(128), ReLU
4	FC(384, 100)	FC(384, 100)	FC(384, 192) ReLU	FC(384, 192) ReLU	FC(256, 100)	FC(6272, 2048) BN(2048), ReLU
5	FC(100, 10)	FC(100, 10)	FC(192, 10)	FC(192, 10)		FC(2048, 512) BN(512), ReLU
6						FC(512, 10)
Source			model from (Acar et al., 2021)	model from (Acar et al., 2021)	model from (Li et al., 2021b)	model from (Li et al., 2021c)

we only offer the feature maps of the client 1 model for these two classes, as it would be unfair to ask the local model of client 1 to map the feature of classes on which it did not learn.). We visualize the feature mappings of client models according to the labels (digit dataset) owned by the corresponding client for label (distribution) distribution skew. The specific setup is described as:

- **Label Distribution Skew:** The experiment has 10 clients where each client has 2 classes with 500 samples per class from FMNIST, and utilizes the SGD optimizer with a 0.01 learning rate and without momentum. The federated setting involves 10 local epoch numbers, 15 communication rounds, and a 100% client sample rate. The top-1 accuracy of global model of all method at the targeted communication round is that FedAvg without skew: 80.32%; FedAvg:52.66%; FedProx:51.43%; FedDyn:51.90%; MOON:45.67%; FedProc:49.87%; FedFA(our): 67.54%.
- **Feature Distribution Skew:** The experiment has 10 clients where each client has 10 classes with 100 samples per class from one of the digit datasets in Mixed Digit (i.e., MNIST, SVHN, USPS, SynthDigits, and MNIST-M), and utilizes the SGD optimizer with a 0.01 learning rate and without momentum. The federated setting involves 10 local epoch numbers, 15 communication rounds, and a 100% client sample rate. The Mean top-1 accuracy of the global models of all methods at the targeted communication round is that FedAvg without

skew: 81.66%; FedAvg:79.56%; FedProx:78.76%; FedDyn:79.60%; MOON: 79.58%; FedProc:79.30%; FedFA(our): 80.44%.

E.3 TEST EXPERIMENT SETUP

Baselines. Federated learning (McMahan et al., 2017) aims to train a global model parameterized by \mathbf{w} by collaborating a total of N clients with a central server to solve the following optimization problem:

$$\min_{\mathbf{w} \in \mathbb{R}^d} \mathcal{L}(\mathbf{w}) := \mathbb{E}_i[\mathcal{L}_i(\mathbf{w})] = \sum_i^N \frac{n_i}{n} \mathcal{L}_i(\mathbf{w})$$

where $n = \sum_i n_i$ represents the total sample size with n_i being the sample size of the i -th client, and $\mathcal{L}_i(\mathbf{w}) := \mathbb{E}_{\xi \in \mathcal{D}_i} [l_i(\mathbf{w}; \xi)]$ is the local objective function in local dataset \mathcal{D}_i of the i -th client.

Many methods have been proposed to solve this optimization problem and alleviate the negative impact of data heterogeneity across clients. Herein, from the view of local-optimization methods, we compare FedFA with the common federated learning algorithms, including FedAvg (McMahan et al., 2017), FedProx (Li et al., 2020) and the state-of-the-art methods based on well-designed local regularization including FedDyn (Acar et al., 2021), MOON (Li et al., 2021b) and FedProc (Mu et al., 2021). The specific description of these methods can be denoted as:

- **FedAvg:** As a canonical method to solve (1) proposed by (McMahan et al., 2017), in each communication round, FedAvg firstly selects a subset of clients and initiates client models as \mathbf{w} and then updates the local models \mathbf{w}_i by minimizing $\mathcal{L}_i(\mathbf{w})$, and finally aggregates the local models \mathbf{w}_i as the new global model \mathbf{w} until $\mathcal{L}(\mathbf{w})$ arrives at a stationary point.
- **FedProx:** FedProx (Li et al., 2020) adds the Euclidean regularization loss between local models and the global model in the local optimization problem, which can be described as:

$$\mathcal{L}_i(\mathbf{w}) = \min_{\mathbf{w}_i} \mathbb{E}_{(\mathbf{x}, y) \in \mathcal{D}_i} [l_i(\mathbf{w}_i; \mathbf{w}^{(t-1)}) + \frac{\mu}{2} \|\mathbf{w}_i - \mathbf{w}^{(t-1)}\|^2]. \quad (16)$$

- **FedDyn:** FedDyn (Acar et al., 2021) modifies the local objective with a dynamic regularization consisting of a linear term based on the first order condition and an above Euclidean-distance term, such that the local minima are consistent with the global stationary point, which can be described as:

$$\mathcal{L}_i(\mathbf{w}) = \min_{\mathbf{w}_i} \mathbb{E}_{(\mathbf{x}, y) \in \mathcal{D}_i} [l_i(\mathbf{w}_i; \mathbf{w}^{(t-1)}) - \langle \nabla \mathcal{L}_i(\mathbf{w}^{(t-1)}), \mathbf{w}_i \rangle + \frac{\mu}{2} \|\mathbf{w}_i - \mathbf{w}^{(t-1)}\|^2]. \quad (17)$$

- **MOON:** MOON (Li et al., 2021b) utilizes the feature similarity of the client model with previous-round local models and with the global model as model-contrastive regularization to correct the local training of each client, which can be described as:

$$\mathcal{L}_i(\mathbf{w}) = \min_{\mathbf{w}_i} \mathbb{E}_{(\mathbf{x}, y) \in \mathcal{D}_i} [l_i(\mathbf{w}_i; \mathbf{w}^{(t-1)}) - \mu \log \frac{\exp(\text{sim}(\mathbf{h}_i, \mathbf{h}_{\text{global}})/\tau)}{\exp(\text{sim}(\mathbf{h}_i, \mathbf{h}_{\text{global}})/\tau) + \exp(\text{sim}(\mathbf{h}_i, \mathbf{h}_{\text{pre}})/\tau)}] \quad (18)$$

where $\mathbf{h}_i, \mathbf{h}_{\text{global}}, \mathbf{h}_{\text{pre}}$ denote the feature mappings of the local model \mathbf{w}_i , the global model \mathbf{w} , and the local model at previous round \mathbf{w}_i^{t-1} given the same input \mathbf{x} , respectively; τ is the hyperparameter to control the effect of cosine similarity in model-contrastive loss.

- **FedProc:** Instead of the model-contrastive term in MOON, FedProc (Mu et al., 2021) introduces a prototype-contrastive term to regularize the features within each class with class prototypes (Snell et al., 2017), which can be described as:

$$\mathcal{L}_i(\mathbf{w}) = \min_{\mathbf{w}_i} \mathbb{E}_{(\mathbf{x}, y) \in \mathcal{D}_i} [\frac{t}{T} l_i(\mathbf{w}_i; \mathbf{w}^{(t-1)}) + (1 - \frac{t}{T}) \log \frac{\exp(\text{sim}(\mathbf{h}_i, \mathbf{p}_c)/\tau)}{\sum_{c=1}^{c=C} \exp(\text{sim}(\mathbf{h}_i, \mathbf{p}_c)/\tau)}] \quad (19)$$

where T is the targeted communication round, and \mathbf{p}_c is the prototype of class c . In FedProc, \mathbf{p}_c is updated by the whole local dataset at the end of one communication round (i.e., $\mathbf{p}_{c,i}^{(t,k)} = \frac{1}{|\mathcal{D}_{i,c}|} \sum_{(\mathbf{x}, c) \in \mathcal{D}_{i,c}} \mathbf{h}_{i,c}$). However, we need to denote that if \mathbf{p}_c is updated like this, rather than the momentum update as (8), and we found that FedProc would suffer from

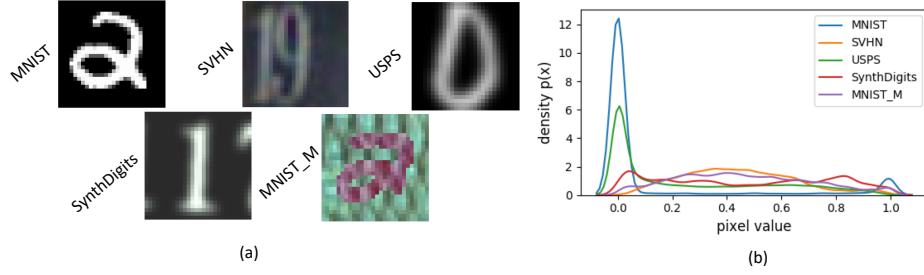


Figure 4: Data visualization. (a) Examples from each dataset (client) in Mixed Digit. (b) feature distributions skew across the datasets (over random 100 samples for each dataset).

the divergence because the update of \mathbf{p}_c is too drastic in our experiments². Therefore, we improve FedProc with momentum update (8).

Datasets. This work aims at image classification tasks under label distribution skew, feature distribution skew and label & feature distribution skew, and uses benchmark datasets with the same data heterogeneity setting as (McMahan et al., 2017; Yurochkin et al., 2019; Li et al., 2021a), including EMNIST (Cohen et al., 2017), FMNIST (Xiao et al., 2017), CIFAR-10, CIFAR-100 (Krizhevsky et al., 2009), and Mixed Digits dataset (Li et al., 2021c). Specifically, for label distribution skew, we consider two settings:

- **Same size of local dataset:** Following (McMahan et al., 2017), we split data samples based on classes to clients (e.g., $\#C = 2$ denotes each client holds two class samples), where each client holds 250 samples per class;
- **Different sizes of local dataset:** Following (Yurochkin et al., 2019), we first sample p_i from Dirichlet distribution $Dir(\alpha)$ and then assign $p_{i,c}$ proportion of the samples of class c to client i , where we set α as 0.1 and 0.5 to measure the level of data heterogeneity in our experiments. Moreover, when $\alpha = 0.1$, the label distributions across clients are so skewed that the quantity of clients’ local dataset is also skewed. That is, the experiment cases related to $\alpha = 0.1$ would involve label distribution skew and quantity distribution skew, which denotes the unbalanced data size of the local dataset across clients.

For feature distribution skew, we consider two settings:

- **Real-world feature imbalance:** We use a subset of the real-world dataset with natural feature imbalance, EMNIST (Cohen et al., 2017), including 10 classes and 341873 samples (about 34000 samples per class) totally;
- **Artificial feature imbalance:** We use a mixed-digit dataset from (Li et al., 2021c) consisting of five benchmark digit datasets: MNIST (LeCun et al., 1998), SVHN (Netzer et al., 2011), USPS (Hull, 1994), SynthDigits and MNIST-M (Ganin et al., 2015), including 7430 samples for one digit dataset and 743 sample per class. The data visualization is shown as Figure 4³.

Note that for the experiments on Mixed Digits, we report the average top-1 accuracy on five benchmark digit datasets in Table 2 and Table 3, and show the top-1 accuracy on each digit dataset during the training in Figure 13 to Figure 19. For the experiments on other datasets except for Mixed Digits, we test the top-1 accuracy on all datasets based on the global model and report them during the training as shown in Figure 9 to Figure 12.

Federated Simulation Setup. All experiments are performed based on PyTorch Paszke et al. (2019) and one node of the High-Performance Computing platform with 4 NVIDIA A30 Tensor Core GPUs with 24GB. We use an existing dataset-split tool FedLab (Zeng et al., 2021) to generate federated local datasets for all clients. There are in total 100 clients, and 10 clients participating in

²The codes of FedProc are not open source, and thus our reproduction settings cannot be completely consistent to the original setting, but we fine-tune the hyperparameter of FedProc carefully and report the best results.

³Figure comes from (Li et al., 2021c)

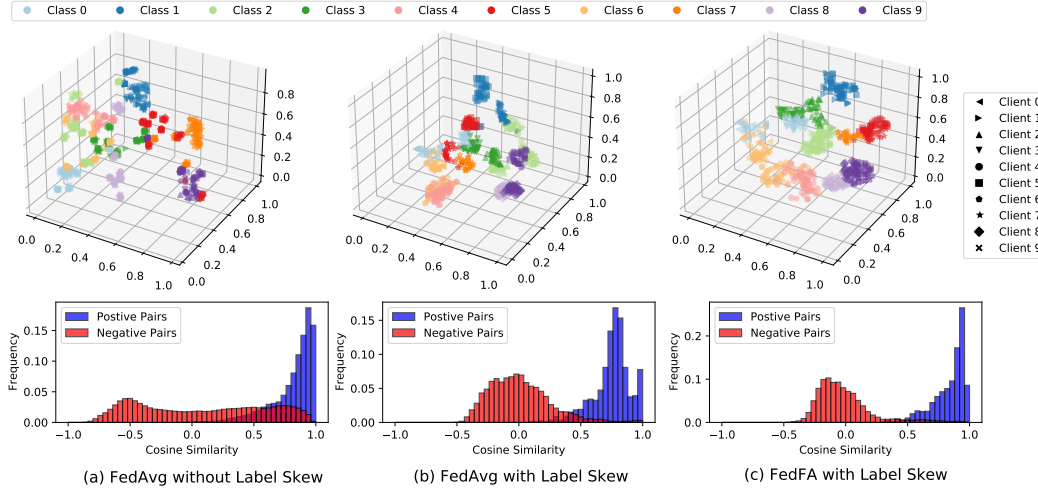


Figure 5: The t-SNE visualization and the histogram of cosine similarity of feature mappings for FedAvg under data homogeneity and for FedAvg and FedFA under label distribution skew with 10 clients.

federated training at each communication round. We use the SGD optimizer with a 0.01 learning rate and 0.001 weight decay for all experiments except for the CIFAR-100 experiment, which uses 0.9 momentum additionally. The local batch size is 64, the number of local epochs is set to 5, and the number of communication rounds is set to 200. Moreover, we carefully select the coefficient of local regularization from $\{1, 0.1, 0.01\}$ (i.e., $\mu/2 = 0.05$ for FedProx and FedDyn, $\mu = 0.1$ for MOON), set the temperature hyperparameter $\tau = 0.5$ for MOON and FedProc, and report their best results in our experiments.

FedFA setup. We set the anchor momentum coefficient $\lambda = 0.5$ in (8) and local loss coefficient $\mu = 0.1$ in (6) like our baselines, and according to Property 1, we initiate the pairwise orthogonal feature anchors \mathbf{a}_c by sampling column vector from an identity matrix whose dimension is the same as the size of the feature mappings. Other settings of FedFA are the same as baselines in all experiments, such as the same random seed (seed: 2021, 2022, 2023) and the same training and test dataset.

F ADDITIONAL EXPERIMENT RESULTS

F.1 ADDITIONAL VALIDATION EXPERIMENT RESULTS

F.1.1 FEATURE VISUALIZATION AND SIMILARITY HISTOGRAM FOR ALL METHODS UNDER LABEL DISTRIBUTION SKEW

Figures 5 and 6 show the t-SNE visualization and the histogram of cosine similarity of feature mappings for label distribution skew for all methods. We observe that all baselines under label skew exist feature mapping inconsistency across clients. Still, our method FedFA alleviates it significantly, such as class 1 (i.e., dark blue), class 5 (i.e., dark red) and class 9 (i.e., dark purple) in Figures 5 and 6. Besides, similar to the analysis of Figure 2, the histograms also show that label distribution skew could induce the lower similarity for *positive pairs*, which means feature inconsistency. Moreover, there exists a low frequency of *positive pairs* and a small gap between *positive pairs* and *negative pairs*, which indicates inconsistent polymerization and discrimination (i.e., sizeable intra-class feature distance and small inter-class feature distance) across clients in classification tasks. These results of label distribution skew reveal that all client models are trained in inconsistent feature spaces by our baselines, which hurts their performance.

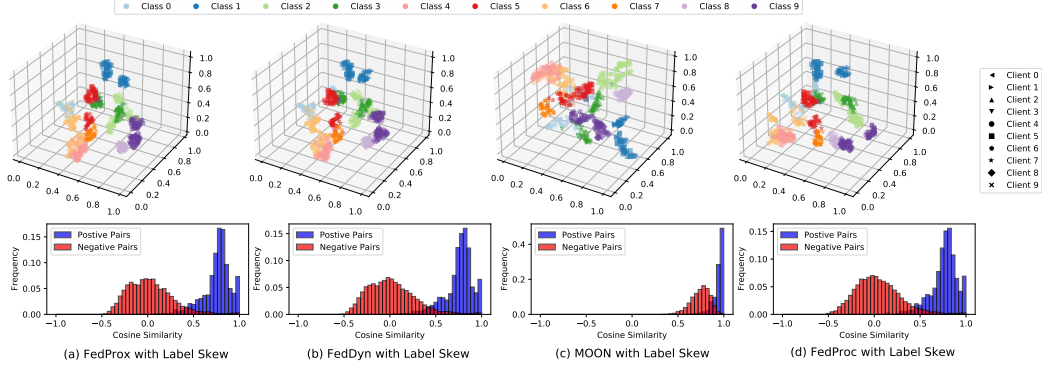


Figure 6: The t-SNE visualization and the histogram of cosine similarity of feature mappings for all baselines under label distribution skew with 10 clients.

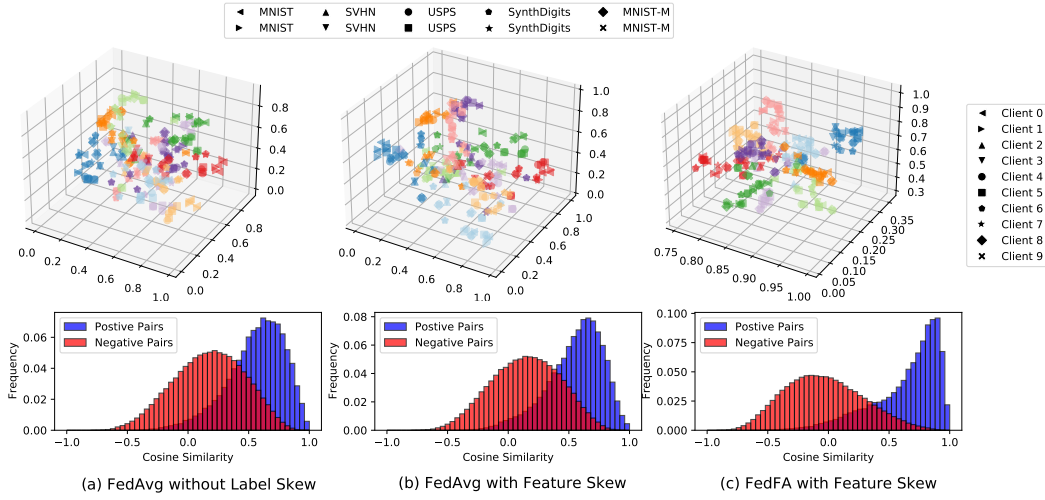


Figure 7: The t-SNE visualization and the histogram of cosine similarity of feature mappings for FedAvg under data homogeneity and for FedAvg and FedFA under feature distribution skew with 10 clients.

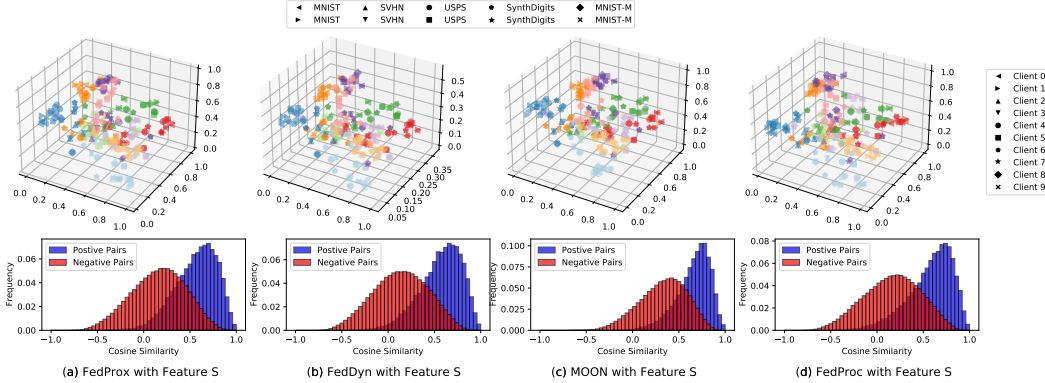


Figure 8: The t-SNE visualization and the histogram of cosine similarity of feature mappings for all baselines under feature distribution skew with 10 clients.

F.1.2 FEATURE VISUALIZATION AND SIMILARITY HISTOGRAM FOR ALL METHODS UNDER FEATURE DISTRIBUTION SKEW

Similar to label distribution skew, Figures 7 and 8 show the t-SNE visualization and the histogram of cosine similarity of feature mappings for feature distribution skew for all methods. We also observe that all baselines under feature skew still suffer from feature mapping inconsistency across clients, but our method does not. Moreover, without the feature alignment, all baselines present the weak feature polymerization and feature discrimination of clients' local models, which would make the classifier updates divergent as denoted in (3). Therefore, these results of feature distribution skew reveal that all client models are trained in inconsistent feature spaces by our baselines, which hurts their performance.

F.2 ADDITIONAL TEST EXPERIMENT RESULTS

F.2.1 COMMUNICATION EFFECTIVENESS

To compare the communication effectiveness, we perform the experiments under label and feature skew to test the communication round number when the global model trained by all methods can reach the target accuracy. As shown in Tables 5 and 6, FedFA realizes better communication efficiency than our baselines with data heterogeneity or not on all datasets except for CIFAR-100. Meanwhile, the training accuracy shown in the following figures also presents similar results.

However, FedFA achieves better generalization than baselines on CIFAR-100 with ResNet but takes more communication rounds to converge. Moreover, as the number of model layers increases, the performance advantages of FedFA diminish. This is probably because the feature anchor loss only regularizes the feature maps of the penultimate layer and is less effective on the shallow layers, which we will explore in future work.

Table 5: The top-1 accuracy (round number when the accuracy reaches the target accuracy) the of all methods under feature distribution skew on the test dataset.

Method (lr = 0.01)	Label Distribution Skew								
	FMNIST $\alpha = 0.1$			CIFAR-10 $\alpha = 0.1$			CIFAR-100 $\alpha = 0.1$		
	#C = 2	$\alpha = 0.1$	$\alpha = 0.5$	#C = 2	$\alpha = 0.1$	$\alpha = 0.5$	#C = 20	$\alpha = 0.1$	$\alpha = 0.5$
Targeted Accuracy	59.682666	55.85066	66.24000	28.85600	28.1600	38.92533	18.09866	17.4346	21.22399
FedAvg w/o skew	62.99(4)	57.31(3)	67.38(6)	29.56(9)	28.34(8)	39.24(23)	18.52(21)	17.69(20)	21.25(28)
FedFA w/o skew	62.62(2)	58.89(1)	67.40(3)	30.06(3)	30.06(3)	40.57(8)	18.17(35)	18.17(35)	21.48(45)
FedAvg	60.22(25)	57.43(34)	66.47(12)	29.64(38)	28.19(43)	39.04(47)	18.34(52)	17.56(73)	21.39(48)
FedProx	59.88(25)	58.55(34)	66.42(13)	29.90(38)	28.62(42)	39.21(42)	18.13(51)	17.76(72)	21.41(52)
FedDyn	62.45(28)	61.99(34)	66.88(12)	30.55(32)	30.19(40)	40.68(36)	0	0	0
Moon	60.54(28)	57.49(34)	66.77(13)	29.09(38)	30.14(48)	39.46(43)	18.35(59)	17.46(81)	21.23(48)
FedProc	61.83(29)	56.44(32)	67.62(22)	30.79(42)	30.10(48)	39.85(49)	18.41(51)	17.52(81)	21.23(52)
FedFA (Our)	62.14(10)	59.46(7)	67.63(5)	30.67(15)	29.94(11)	41.72(14)	18.76(75)	17.70(106)	22.99(87)

Table 6: The top-1 accuracy (round number when the accuracy reaches the target accuracy) of all methods under feature distribution skew on the test dataset.

Method	Feature Distribution Skew			Label & Feature Distribution Skew					
	EMNIST	Mixed Digits Local BN	Mixed Digits Federated BN	Mixed Digits with Local BN #C = 2	$\alpha = 0.1$	$\alpha = 0.5$	Mixed Digits with Federated BN #C = 2	$\alpha = 0.1$	$\alpha = 0.5$
	78.800286	63.07	63.767359	43.58	48.48	60.14	43.94	50.05	62.64
FedAvg w/o skew	80.34(6)	63.44(65)	66.61(72)	44.13(34)	49.57(42)	60.53(61)	44.26(35)	50.63(45)	62.66(64)
FedFA w/o skew	87.85(2)	63.25(45)	66.53(49)	44.74(24)	48.81(28)	60.95(42)	44.61(24)	50.55(30)	63.00(43)
FedAvg	81.12(6)	63.31(72)	66.13(70)	43.75(111)	48.61(84)	60.80(76)	45.87(112)	50.15(85)	63.39(68)
FedProx	81.43(7)	63.91(73)	66.47(72)	44.31(112)	48.54(90)	60.30(76)	45.20(112)	50.49(90)	62.70(68)
FedDyn	79.28(5)	64.42(63)	66.47(63)	43.90(105)	49.82(78)	61.45(64)	44.13(111)	51.24(78)	63.21(63)
Moon	80.99(6)	63.79(74)	67.20(79)	43.63(105)	49.70(84)	60.81(68)	44.13(111)	51.24(94)	63.02(68)
FedProc	82.62(8)	63.21(83)	66.20(79)	44.18(72)	48.82(61)	60.98(60)	44.87(69)	51.43(61)	62.83(58)
FedFA	87.99(2)	64.21(14)	67.80(14)	46.03(24)	48.65(18)	60.89(17)	45.20(20)	50.47(18)	64.85(17)

F.2.2 TRAINING ACCURACY UNDER LABEL DISTRIBUTION SKEW

The training accuracy under label skew is shown as Figure 9 to Figure 11, which illustrates that the performance of FedFA is better than all baselines on FMNIST, CIFAR-10, and CIFAR-100. FedFA achieves better generalization than baselines on CIFAR-100 with ResNet but takes more communication rounds to converge. This observation is reasonable because regularizing only the penultimate layer by feature anchor loss takes more time to align the feature maps of the shallow layers, which we will explore in future work.

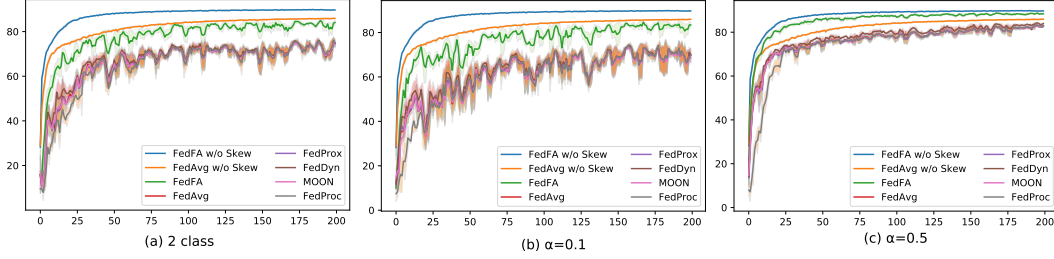


Figure 9: FMNIST with label skew.

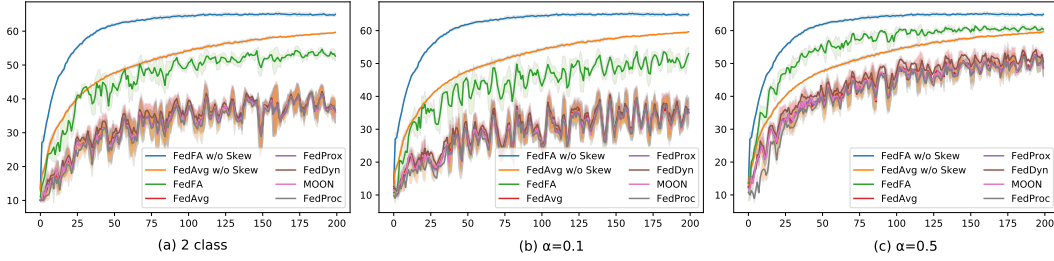


Figure 10: CIFAR-10 with label skew.

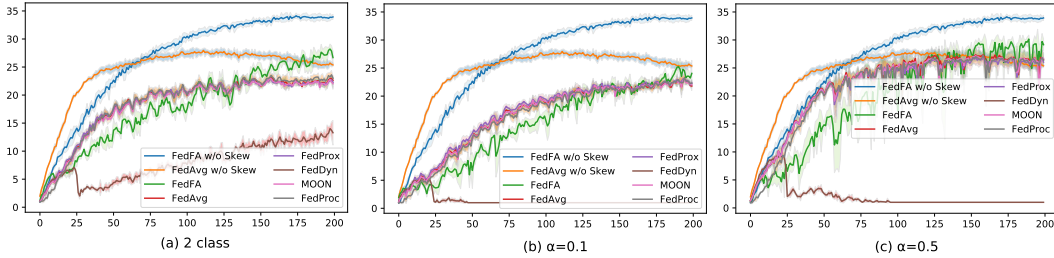


Figure 11: CIFAR-100 with label skew.

F.2.3 TRAINING ACCURACY UNDER FEATURE DISTRIBUTION SKEW

Figures 12 to 13 show the better performance of FedFA over all baselines under feature distribution skew on EMNIST and Mixed Digit.

F.2.4 TRAINING ACCURACY UNDER LABEL & FEATURE DISTRIBUTION SKEW

Figures 14 to 20 show the better performance of FedFA over all baselines under label & feature skew on EMNIST and Mixed Digit.

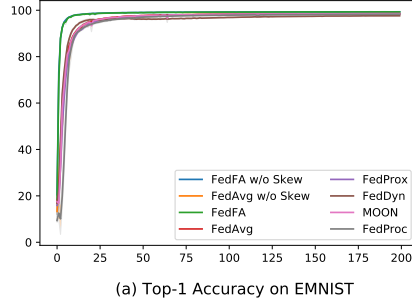


Figure 12: EMNIST with feature skew.

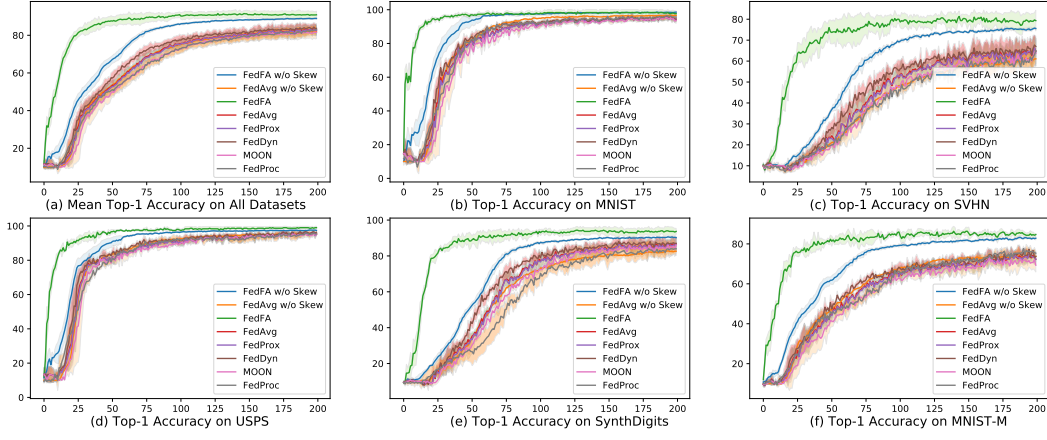
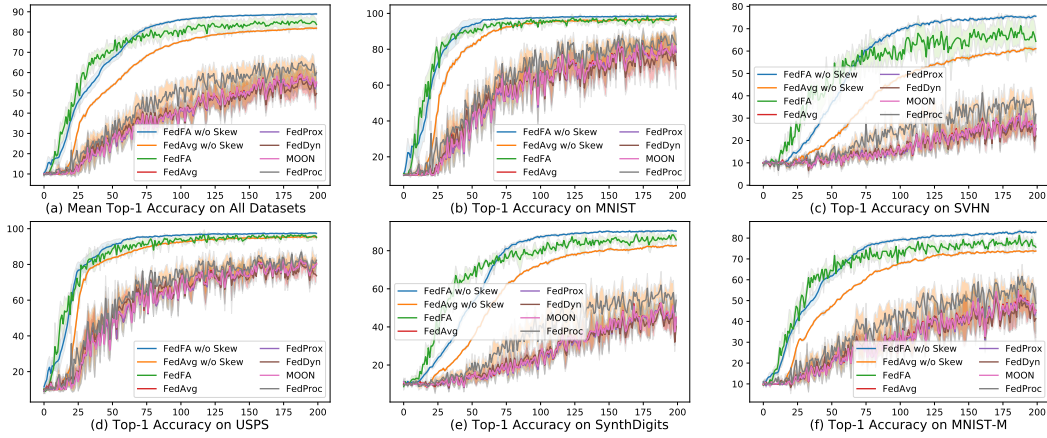


Figure 13: Mixed Digit without label skew.

Figure 14: Mixed Digit with label skew $\#C = 2$ and federated BN.

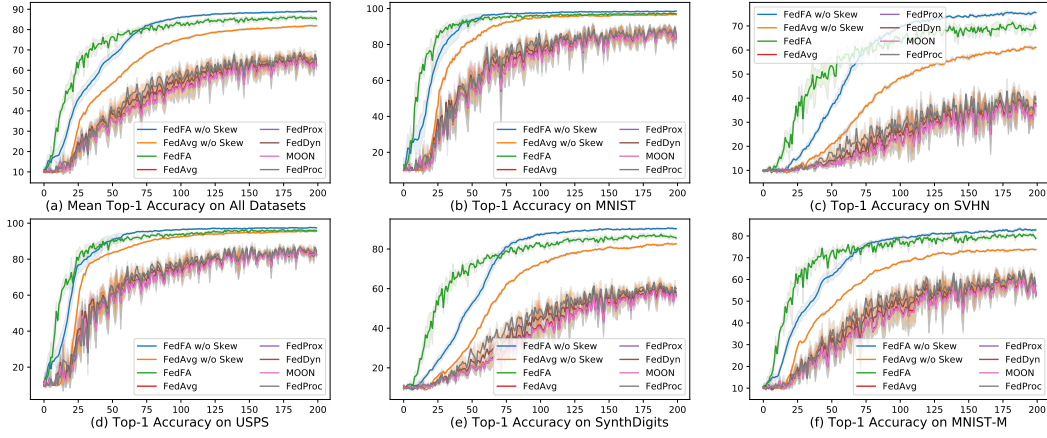
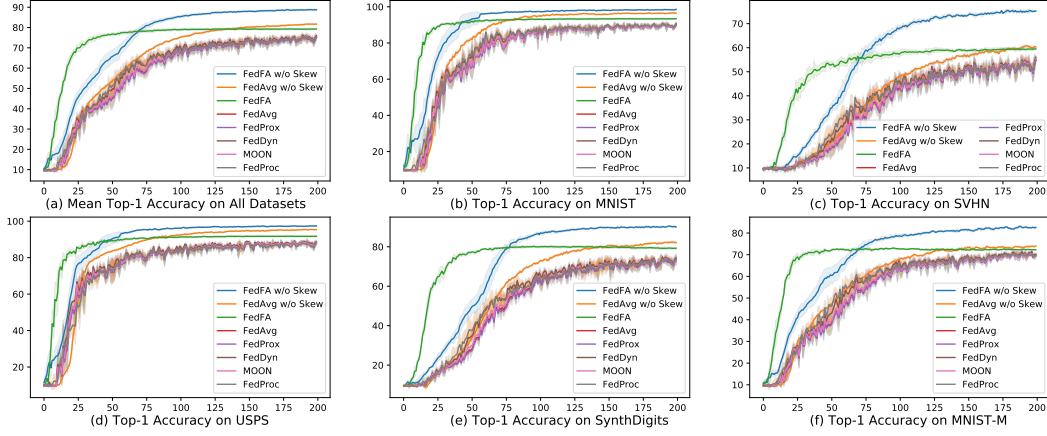
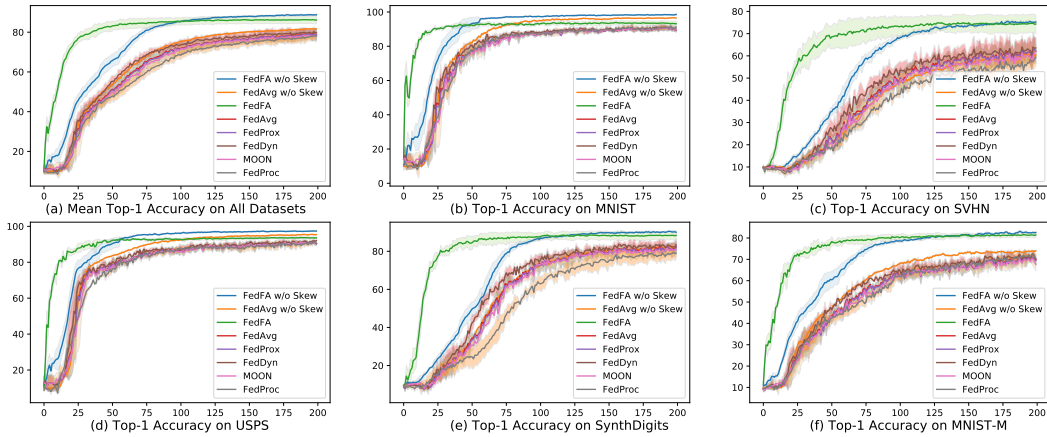
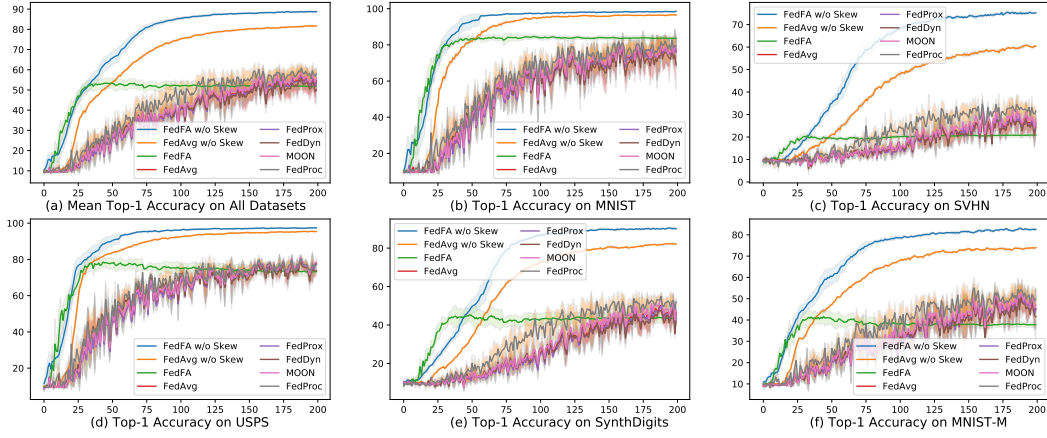
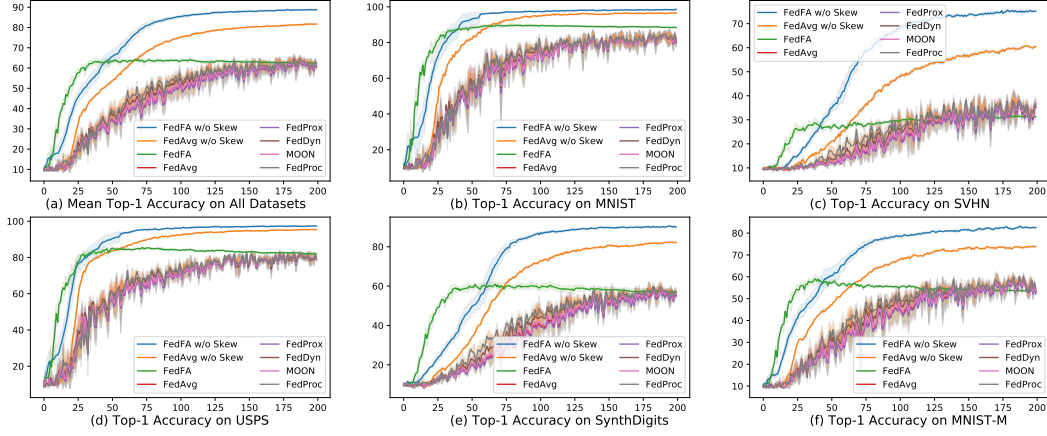
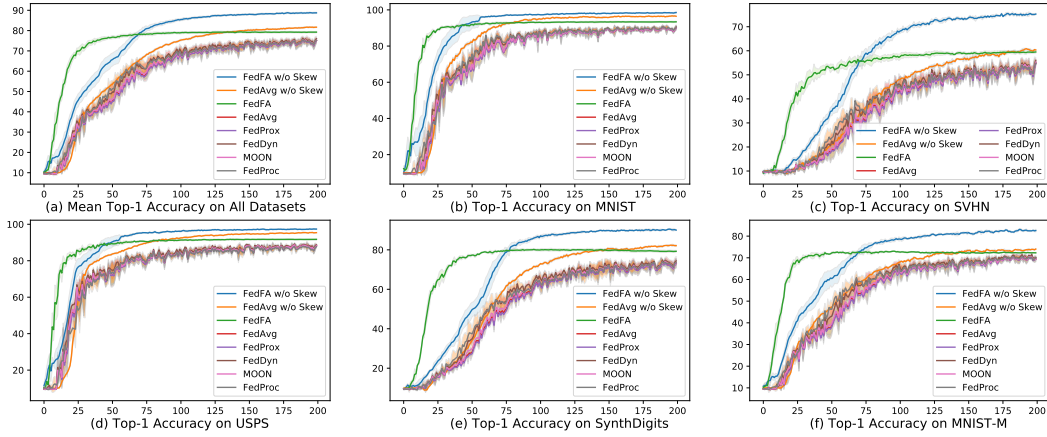
Figure 15: Mixed Digit with label skew $\alpha = 0.1$ and federated BN.Figure 16: Mixed Digit with label skew $\alpha = 0.5$ and federated BN.

Figure 17: Mixed Digit without label skew and with local BN.

Figure 18: Mixed Digit with label skew $\#C = 2$ and with local BN.Figure 19: Mixed Digit with label skew $\alpha = 0.1$ and with local BN.Figure 20: Mixed Digit with label skew $\alpha = 0.5$ and with local BN.