Roadmap of Appendix The Appendix is organized as follows. We list the notations table in Section A. We provide theoretical proof of convergence in Section B. The algorithm of FedBN is described in Section C. The details of experimental setting are in Section D and additional results on benchmark datasets are in Section E. We show experiment on synthetic data in Section F. We demonstrate the ability of generalizing FedBN to test on a new client in Section G.

A NOTATION TABLE

Notations	Description
$\mathbf{x}$	features, $\mathbf{x} \in \mathbb{R}^d$
d	dimension of $\mathbf{x}$
y	labels, $y \in \mathbb{R}$
$P(\cdot)$	probability distribution
N	total number of clients
T	total number of epochs in training
E	number of local iteration in FL
M	number of training samples in each client
$[N]$	set of numbers, $[N] = \{1, \ldots, N\}$
i	indicator for client, $i \in [N]$
j	indicator for sample in each client, $j \in [M]$
$(\mathbf{x}_j^i, y_j^i)$	the j-th training sample in client i
m	number of neurons in the first layer
k	indicator for neuron, $k \in [m]$
$\mathbf{v}_k$	parameters for the k-th neuron in the first layer
$\parallel \mathbf{v} \parallel_{\mathbf{S}}$	vector norm, $\parallel \mathbf{v} \parallel_{\mathbf{S}} \triangleq \sqrt{\mathbf{v}^\top \mathbf{S} \mathbf{v}}$, given a matrix $\mathbf{S}$
$\mathbf{S}_i$	covariance matrix for features in client i, $\mathbf{S}_i = \mathbb{E}\mathbf{x}^i \mathbf{x}^{i\top}$
p, q	indicator for sample, $p, q \in [NM]$
f	two layer ReLU neural network with BN
f^*	two layer ReLU neural network with BN with client-specified BN parameters
$\mathbf{V}$	parameters of the first phase neurons, $\mathbf{V} \in \mathbb{R}^{m \times d}$
$\boldsymbol{\gamma}$	the scaling parameter of BN
$\mathbf{c}$	top layer parameters of the network
$\sigma(\cdot)$	ReLU activation function, $\sigma(\cdot) = \max\{\cdot, 0\}$
$N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$	Gaussian with mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$
$U[-1, 1]$	Rademacher distribution
α	variance of $\mathbf{v}_k$ at initialization
$L(f)$	empirical risk with square loss for network f
$\boldsymbol{\Lambda}(t)$	evolution dynamic for FedAvg at epoch t
$\mathbf{V}(t)$	evolution dynamic with respect to $\mathbf{V}$ for FedAvg at epoch t
$\mathbf{G}(t)$	evolution dynamic with respect to $\boldsymbol{\gamma}$ for FedAvg at epoch t
$\boldsymbol{\Lambda}^*(t)$	evolution dynamic for FedBN at epoch t
$\mathbf{V}^*(t)$	evolution dynamic with respect to $\mathbf{V}$ for FedBN at epoch t
$\mathbf{G}^*(t)$	evolution dynamic with respect to $\boldsymbol{\gamma}$ for FedBN at epoch t
$\lambda_{min}(A)$	the minimal eigenvalue of matrix A
$\mathbf{G}^\infty$	expectation of $\mathbf{G}(t)$
$\mathbf{G}^{*\infty}$	expectation of $\mathbf{G}^*(t)$

Table 2: Notations occurred in the paper.

B CONVERGENCE PROOF

B.1 EVOLUTION DYNAMICS

In this section, we calculate the evolution dynamics $\mathbf{\Lambda}(t)$ for training with function f and $\mathbf{\Lambda}^*(t)$ for training with f^*. Since the parameters are updated using gradient descent, the optimization dynamics of parameters are

$$\frac{d\mathbf{v}_k}{dt} = -\frac{\partial L}{\partial \mathbf{v}_k}, \quad \frac{d\gamma_k}{dt} = -\frac{\partial L}{\partial \gamma_k}.$$

Let $f_p = f(\mathbf{x}_p^{i_p})$. Then, the dynamics of the prediction of the p-th data point in site i_p is

$$\frac{\partial f_p}{\partial t} = \sum_{k=1}^{m} \frac{\partial f_p}{\partial \mathbf{v}_k}\frac{d\mathbf{v}_k}{dt} + \frac{\partial f_p}{\partial \gamma_k}\frac{d\gamma_k}{dt} = \underbrace{-\sum_{k=1}^{m} \frac{\partial f_p}{\partial \mathbf{v}_k}\frac{\partial L}{\partial \mathbf{v}_k}}_{T_{\mathbf{v}}^p} \underbrace{-\sum_{k=1}^{m} \frac{\partial f_p}{\partial \gamma_k}\frac{\partial L}{\partial \gamma_k}}_{T_{\gamma}^p}.$$

The gradients of f_p and L with respect to $\mathbf{v}_k$ and γ_k are computed as

$$\begin{aligned}
\frac{\partial f_p}{\partial \mathbf{v}_k}(t) &= \frac{1}{\sqrt{m}} \frac{c_k \cdot \gamma_k(t)}{\|\mathbf{v}_k(t)\|_{\mathbf{S}_{i_p}}} \cdot \mathbf{x}_p^{\mathbf{v}_k^{i_p}(t)^\perp} \mathbb{1}_{pk}(t), \\
\frac{\partial L}{\partial \mathbf{v}_k}(t) &= \frac{1}{\sqrt{m}} \sum_{q=1}^{NM} (f_q(t) - y_q) \frac{c_k \cdot \gamma_k(t)}{\|\mathbf{v}_k(t)\|_{\mathbf{S}_{i_q}}} \mathbf{x}_q^{\mathbf{v}_k^{i_q}(t)^\perp} \mathbb{1}_{qk}(t), \\
\frac{\partial f_p}{\partial \gamma_k}(t) &= \frac{1}{\sqrt{m}} \frac{c_k}{\|\mathbf{v}_k(t)\|_{\mathbf{S}_{i_p}}} \sigma\left(\mathbf{v}_k(t)^\top \mathbf{x}_p\right), \\
\frac{\partial L}{\partial \gamma_k}(t) &= \frac{1}{\sqrt{m}} \sum_{q=1}^{NM} (f_q(t) - y_q) \frac{c_k}{\|\mathbf{v}_k(t)\|_{\mathbf{S}_{i_q}}} \sigma\left(\mathbf{v}_k(t)^\top \mathbf{x}_q\right),
\end{aligned}$$

where $f_p = f(\mathbf{x}_p^{i_p})$, $\mathbf{x}_p^{\mathbf{v}_k^{i_p}(t)^\perp} \triangleq (\mathbf{I} - \frac{\mathbf{S}_{i_p}\mathbf{u}\mathbf{u}^\top}{\|\mathbf{u}\|_{\mathbf{S}_{i_p}}^2})\mathbf{x}$, and $\mathbb{1}_{pk}(t) \triangleq \mathbb{1}_{\{\mathbf{v}_k(t)^\top \mathbf{x}_p \geq 0\}}$.

We define Gram matrix $\mathbf{V}(t)$ and $\mathbf{G}(t)$ as

$$\mathbf{V}_{pq}(t) = \frac{1}{m}\sum_{k=1}^{m} (\alpha c_k \cdot \gamma_k(t))^2 \|\mathbf{v}_k(t)\|_{\mathbf{S}_{i_p}}^{-1} \|\mathbf{v}_k(t)\|_{\mathbf{S}_{i_q}}^{-1} \left\langle \mathbf{x}_p^{\mathbf{v}_k^{i_p}(t)^\perp}, \mathbf{x}_q^{\mathbf{v}_k^{i_q}(t)^\perp} \right\rangle \mathbb{1}_{pk}(t)\mathbb{1}_{qk}(t), \tag{6}$$

$$\mathbf{G}_{pq}(t) = \frac{1}{m}\sum_{k=1}^{m} c_k^2 \|\mathbf{v}_k(t)\|_{\mathbf{S}_{i_p}}^{-1} \|\mathbf{v}_k(t)\|_{\mathbf{S}_{i_q}}^{-1} \sigma\left(\mathbf{v}_k(t)^\top \mathbf{x}_p\right) \sigma\left(\mathbf{v}_k(t)^\top \mathbf{x}_q\right). \tag{7}$$

It follows that

$$T_{\mathbf{v}}^p(t) = \sum_{q=1}^{NM} \frac{\mathbf{V}_{pq}(t)}{\alpha^2} (f_q(t) - y_q), \quad T_{\gamma}^p(t) = \sum_{q=1}^{NM} \mathbf{G}_{pq}(t) (f_q(t) - y_q).$$

Let $\mathbf{f} = (f_1, \ldots, f_n)^\top = (f(\mathbf{x}_1), \ldots, f(\mathbf{x}_{NM}))^\top$. The full evolution dynamics are given by

$$\frac{d\mathbf{f}}{dt} = -\mathbf{\Lambda}(t)(\mathbf{f}(t) - \mathbf{y}), \quad \text{where} \quad \mathbf{\Lambda}(t) := \frac{\mathbf{V}(t)}{\alpha^2} + \mathbf{G}(t).$$

We compute the NTK for FedBN with f^* analogously. We define the Gram matrix $\mathbf{V}^*(t)$ and $\mathbf{G}^*(t)$ as

$$\mathbf{V}_{pq}^*(t) = \frac{1}{m}\sum_{k=1}^{m} (\alpha c_k)^2 \gamma_{k,i_p}(t)\gamma_{k,i_q}(t) \|\mathbf{v}_k(t)\|_{\mathbf{S}_{i_p}}^{-1} \|\mathbf{v}_k(t)\|_{\mathbf{S}_{i_q}}^{-1} \left\langle \mathbf{x}_p^{\mathbf{v}_k^{i_p}(t)^\perp}, \mathbf{x}_q^{\mathbf{v}_k^{i_q}(t)^\perp} \right\rangle \mathbb{1}_{pk}(t)\mathbb{1}_{qk}(t), \tag{8}$$

$$\mathbf{G}_{pq}^*(t) = \frac{1}{m}\sum_{k=1}^{m} c_k^2 \|\mathbf{v}_k(t)\|_{\mathbf{S}_{i_p}}^{-1} \|\mathbf{v}_k(t)\|_{\mathbf{S}_{i_q}}^{-1} \sigma\left(\mathbf{v}_k(t)^\top \mathbf{x}_p\right) \sigma\left(\mathbf{v}_k(t)^\top \mathbf{x}_q\right) \mathbb{1}\{i_p = i_q\}. \tag{9}$$

The full evolution dynamics are then

$$\frac{d\mathbf{f}^*}{dt} = -\mathbf{\Lambda}^*(t)(\mathbf{f}^*(t) - \mathbf{y}), \quad \text{where} \quad \mathbf{\Lambda}^*(t) := \frac{\mathbf{V}^*(t)}{\alpha^2} + \mathbf{G}^*(t).$$

B.2 PROOF OF LEMMA 4.3

Dukler et al. (2020) proved that the matrix $\mathbf{G}^\infty$ is strictly positive definite. In their proof, $\mathbf{G}^\infty$ is the covariance matrix of the functionals ϕ_p define as

$$\phi_p(\mathbf{v}) := \sigma\left(\mathbf{v}^\top \mathbf{x}_p\right)$$

over the Hilbert space $\mathcal{V}$ of $L^2\left(N\left(0, \alpha^2\mathbf{I}\right)\right)$. $\mathbf{G}^{*\infty}$ is strictly positive definite by showing that $\phi_1, \cdots, \phi_{NM}$ are linearly independent, which is equivalent to that

$$c_1\phi_1 + c_2\phi_2 + \cdots + c_{NM}\phi_{NM} = 0 \text{ in } \mathcal{V} \tag{10}$$

holds only for $c_p = 0$ for all p.

Let $\mathbf{G}_i^\infty$ be the i-th $M \times M$ block matrices on the diagonal of $\mathbf{G}^\infty$. Then $\mathbf{G}^{*\infty} = diag(\mathbf{G}_1^\infty, \cdots, \mathbf{G}_N^\infty)$. To prove that $\mathbf{G}^{*\infty}$ is strictly positive definite, we will show that $\mathbf{G}_i^\infty$ is positive definite. Define

$$\phi_{j,i}^*(\mathbf{v}) := \sigma\left(\mathbf{v}^\top \mathbf{x}_j\right) \mathbb{1}\{j \in \text{ site } i\}, \quad j = 1, \cdots, M.$$

Then, we are going to show that

$$c_1\phi_{1,i}^* + c_2\phi_{2,i}^* + \cdots + c_M\phi_{M,i}^* = 0 \tag{11}$$

holds only for $c_j = 0$ for all $j \in [M]$. Suppose there exist $c_1, \cdots, c_M$ that are not identically 0, satisfying (11). Let the coefficients for client i c be the same as $c_1, \cdots, c_M$ and let the coefficients for other client be 0. Then, we have s sequence of coefficients satisfying (10), which is a contradiction of that $\mathbf{G}^\infty$ is strictly positive definite. This implies $\mathbf{G}_i^\infty$ is strictly positive definite, which means the eigenvalues are positive. Since the eigenvalues of $\mathbf{G}^{*\infty}$ are exactly the eigenvalues of $\mathbf{G}_i^\infty$, $\lambda_{min}(\mathbf{G}^{*\infty})$ is positive and thus, $\mathbf{G}^{*\infty}$ is strictly positive definite.

B.3 PROOF OF COROLLARY 4.6

To compare the convergence rates of FedAvg and FedBN when $E = 1$, we compare the exponential factor in the convergence rates, which are $(1 - \eta\mu_0/2)$ and $(1 - \eta\mu_0^*/2)$ for FedAvg and FedBN, respectively. Then, it reduces to comparing $\mu_0 = \lambda_{\min}(\mathbf{G}^\infty)$ and $\mu_0^* = \lambda_{\min}(\mathbf{G}^{*\infty})$. Comparing equation (7) and (9), $\mathbf{G}^{*\infty}$ takes the $M \times M$ block matrices on the diagonal of $\mathbf{G}^\infty$:

$$\mathbf{G}^\infty = \begin{bmatrix} \mathbf{G}_1^\infty & \mathbf{G}_{1,2}^\infty & \cdots & \mathbf{G}_{1,N}^\infty \\ \mathbf{G}_{1,2}^\infty & \mathbf{G}_2^\infty & \cdots & \mathbf{G}_{2,N}^\infty \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{G}_{1,N}^\infty & \mathbf{G}_{2,N}^\infty & \cdots & \mathbf{G}_N^\infty \end{bmatrix}, \quad \mathbf{G}^{*\infty} = \begin{bmatrix} \mathbf{G}_1^\infty & 0 & \cdots & 0 \\ 0 & \mathbf{G}_2^\infty & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \mathbf{G}_N^\infty \end{bmatrix},$$

where $\mathbf{G}_i^\infty$ is the i-th $M \times M$ block matrices on the diagonal of $\mathbf{G}^\infty$. By linear algebra,

$$\lambda_{\min}(\mathbf{G}_i^\infty) \geq \lambda_{\min}(\mathbf{G}^\infty), \quad \forall i \in [N].$$

Since the eigenvalues of $\mathbf{G}^{*\infty}$ are exactly the union of eigenvalues of $\mathbf{G}_i^\infty$, we have

$$\begin{aligned} \lambda_{\min}(\mathbf{G}^{*\infty}) &= \min_{i \in [N]}\{\lambda_{\min}(\mathbf{G}_i^\infty)\}, \\ &\geq \lambda_{\min}(\mathbf{G}^\infty). \end{aligned}$$

Thus, $(1 - \eta\mu_0/2) \geq (1 - \eta\mu_0^*/2)$ and we can conclude that the convergence rate of FedBN is faster than the convergence of FedAvg.

C FEDBN ALGORITHM

We describe the details algorithm of our proposed FedBN as following Algorithm 1:

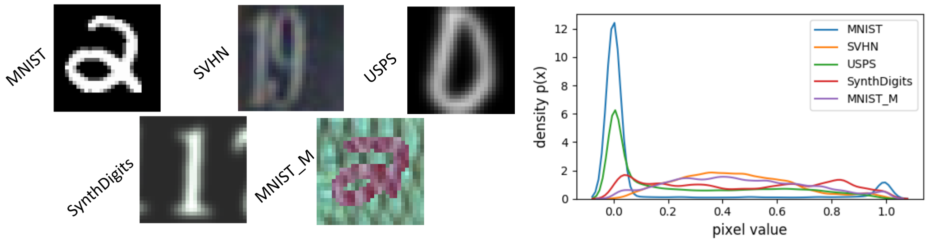
Algorithm 1 Federated Learning using FedBN

Notations: The user indexed by k, neural network layer indexed by l, initialized model parameters: $w_{0,k}^{(l)}$, local update pace: E, and total optimization round T.

```
for each round $t = 1, 2, \dots, T$ do
    for each user $k$ and each layer $l$ do
        $w_{t+1,k}^{(l)} \leftarrow SGD(w_{t,k}^{(l)})$
    end for
    if mod$(t, E) = 0$ then
        for each user $k$ and each layer $l$ do
            if layer $l$ is not BatchNorm then
                $w_{t+1,k}^{(l)} \leftarrow \frac{1}{K}\sum_{k=1}^{K} w_{t+1,k}^{(l)}$
            end if
        end for
    end if
end for
```

D EXPERIMENTAL DETAILS

D.1 VISUALIZATION OF BENCHMARK DATASETS



(a)

(b)

Figure 6: Data visualization. (a) Examples from each dataset (client). (b) Non-iid feature distributions across the datasets (over random 100 samples for each dataset).

D.2 MODEL ARCHITECTURE AND TRAINING DETAILS ON BENCHMARK

We illustrate our model architecture and training details of the digits classification experiments in this section.

Model Architecture. For our benchmark experiment, we use a six-layer Convolutional Neural Network (CNN) and its details are listed in Table 3.

Layer	Details

1	Conv2D(3, 64, 5, 1, 2)
	BN(64), ReLU, MaxPool2D(2, 2)

2	Conv2D(64, 64, 5, 1, 2)
	BN(64), ReLU, MaxPool2D(2, 2)

3	Conv2D(64, 128, 5, 1, 2) BN(128), ReLU
3	Conv2D(64, 128, 5, 1, 2) BN(128), ReLU
4	FC(6272, 2048) BN(2048), ReLU
5	FC(2048, 512) BN(512), ReLU
6	FC(512, 10)

Table 3: Model architecture of the benchmark experiment. For convolutional layer (Conv2D), we list parameters with sequence of input and output dimension, kernal size, stride and padding. For max pooling layer (MaxPool2D), we list kernal and stride. For fully connected layer (FC), we list input and output dimension. For BatchNormalization layer (BN), we list the channel dimension.

Training Details. We give detailed settings for the experiments conducted in 5.1: (1) convergence rate (Table 4), (2) analysis of local update epochs (Table 5), (3) analysis of local dataset size (Table 6), (4) effects of statistical heterogeneity (Table 7) and (5) comparison with state-of-the-art (Table 8). Each table describes the number of clients, samples and the local update epochs.

During training process, we use SGD optimizer with learning rate 10^{-2} and cross-entropy loss, we set batch size to 32 and training epochs to 300. For hyper-parameter μ, we use the best value $\mu = 10^{-2}$ founded by grid search from the the default settings in FedProx Li et al. (2020b).

Datasets	SVHN	USPS	SynthDigits	MNIST-M	MNIST
Number of clients	1	1	1	1	1
Number of samples	743	743	743	743	743
Local update epochs	1	1	1	1	1

Table 4: Settings for convergence rate. Each dataset has 1 client with 743 samples, local update epoch is set to 1.

Datasets	SVHN	USPS	SynthDigits	MNIST-M	MNIST
Number of clients	1	1	1	1	1
Number of samples	743	743	743	743	743
Local update epochs	1,4,8,16	1,4,8,16	1,4,8,16	1,4,8,16	1,4,8,16

Table 5: Settings for local update epochs. Each dataset has 1 client with 743 samples, local update epoch for all datasets is set to 1, 4, 8, 16 successively.

Datasets	SVHN	USPS	SynthDigits	MNIST-M	MNIST
Number of clients	1	1	1	1	1
Number of samples	ω	ω	ω	ω	ω
Local update epochs	1	1	1	1	1

Table 6: Settings for local dataset size, we set local update epochs to 1 and each dataset has 1 client. The number of samples $\omega \in \{74, 371, 743, 1487, 2975, 4462, 7438\}$.

Datasets	SVHN	USPS	SynthDigits	MNIST-M	MNIST
Number of clients	[1, 10]	[1, 10]	[1, 10]	[1, 10]	[1, 10]
Number of samples	[1, 10]×743	[1, 10]×743	[1, 10]×743	[1, 10]×743	[1, 10]×743
Local update epochs	1	1	1	1	1

Table 7: Settings for statistical heterogeneity, [1, 10] for the range from 1 to 10. We increase number of clients step by step and number of samples will increase accordingly.

Datasets	SVHN	USPS	SynthDigits	MNIST-M	MNIST
Number of clients	1	1	1	1	1
Number of samples	743	743	743	743	743
Local update epochs	1	1	1	1	1

Table 8: Settings for comparison with SOTA, we use 1 client with 743 samples and 1 local update epoch for comparison experiment.

D.3 MODEL ARCHITECTURE AND TRANING DETAILS OF IMAGE CLASSIFICATION TASK ON OFFICE-CALTECH10 AND DOMAINNET

In this section, we provide the details of our model and training process on both Office-Caltech10 Gong et al. (2012) and DomainNet Peng et al. (2019) dataset.

Model Architecture. For the image classification tasks on these two real-worlds datasets Office-Caltech10 and DomainNet data, we use adapted AlexNet added with BN layer after each convolutional layer and fully-connected layer (except the last layer), architecture is shown in Table 9.

Layer	Details

1	Conv2D(3, 64, 11, 4, 2)
	BN(64), ReLU, MaxPool2D(3, 2)

2	Conv2D(64, 192, 5, 1, 2)
	BN(192), ReLU, MaxPool2D(3, 2)

3	Conv2D(64, 128, 5, 1, 2) BN(128), ReLU
3	Conv2D(192, 384, 3, 1, 1) BN(384), ReLU
4	Conv2D(384, 256, 3, 1, 1) BN(256), ReLU

5	Conv2D(256, 256, 3, 1, 1)
	BN(256), ReLU, MaxPoll2D(3, 2)

6	AdaptiveAvgPool2D(6, 6)
7	FC(9216, 4096) BN(4096), ReLU
8	FC(4096, 4096) BN(4096), ReLU
9	FC(4096, 10)

Table 9: Model architecture for Office-Caltech10 and DomainNet experiment. For convolutional layer (Conv2D), we list parameters with sequence of input and output dimension, kernal size, stride and padding. For max pooling layer (MaxPool2D), we list kernal and stride. For fully connected layer (FC), we list input and output dimension. For BatchNormalization layer (BN), we list the channel dimension.

Training Details. Office-Caltech10 selects 10 common objects in Office-31 Saenko et al. (2010) and Caltech-256 datasets Griffin et al. (2007). There are four different data sources, one from Caltech-256 and three from Office-31, namely Amazon(images collected from online shopping website), DSLR and Webcam(images captured in office environment using Digital SLR camera and web camera).

We first reshape input images in the two dataset into $256 \times 256 \times 3$, then for training process, we use cross-entropy loss and SGD optimizer with learning rate of 10^{-2}, batch size is set to 32 and training epochs is 300. When comparing with FedProx, we set μ to 10^{-2} which is tuned from the default settings. The data sample number are kept into the same size according to the smallest dataset, i.e. Office-Caltech10 uses 62 training samples and DomainNet uses 105 training samples on each dataset. In addition, for simplicity, we choose top-10 class based on data amount from DomainNet containing images over 345 categories.

D.4 ABIDE DATASET AND TRAINING DETAILS

Here we describe the real-world medical datasets, the preprocessing and training details.

Dataset: The study was carried out using resting-state fMRI (rs-fMRI) data from the Autism Brain Imaging Data Exchange dataset (ABIDE I preprocessed, (Di Martino et al., 2014)). ABIDE is a consortium that provides preciously collected rs-fMRI ASD and matched controls data for the purpose of data sharing in the scientific community. We downloaded Regions of Interests (ROIs) fMRI series of the top four largest sites (UM, NYU, USM, UCLA viewed as clients) from the preprocessed ABIDE dataset with Configurable Pipeline for the Analysis of Connectomes (CPAC) and parcellated by Harvard-Oxford (HO) atlas. Skipping subjects lacking filename, resulting in 88, 167, 52, 63 subjects for UM, NYU, USM, UCLA separately. Due to a lack of sufficient data, we used sliding windows (with window size 32 and stride 1) to truncate raw time sequences of fMRI. The compositions of four sites were shown in Table 10. The number of overlapping truncate is the dataset size in a client.

	NYU	**UM1**	**USM**	**UCLA1**
Total Subject	167	88	52	63
ASD Subject	73	43	33	37
HC Subject	94	45	19	26
ASD Percentage	44%	49%	63%	59%
fMRI Frames	176	296	236	116
Overlapping Trunc	145	265	205	85

Table 10: Data summary of the dataset used in our study.

Training Process : For all the strategies, we set batch size as 100. The total training local epoch is 50 with learning rate 10^{-2} with SGD optimizer. Local update epoch for each client is $E = 1$. We selected the best parameters $\mu = 0.2$ in FedProx through grid search.

E MORE EXPERIMENTAL RESULTS ON BENCHMARK DATASETS

E.1 CONVERGENCE COMPARISON OVER FEDAVG AND FEDBN

In this section we conduct an additional convergence analysis experiment over different local update epochs settings: $E = 1, 4, 8, 16$. As shown in Fig. 7, FedBN converges faster than FedAvg under different values of E, which is supportive to our theoretical analysis in section 4 and experimental results in section 5.

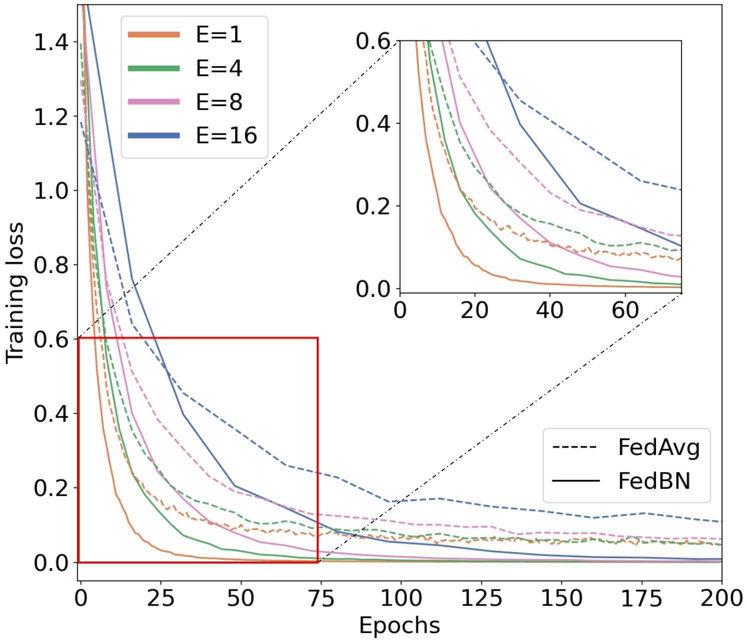


Figure 7: Training loss over epochs with different local update frequency.

E.2 DETAILED STATISTICS OF FIGURE 5

Methods	SVHN	USPS	Synth	MNIST-M	MNIST
Single	65.82 (0.76)	95.44 (0.20)	80.69 (0.52)	77.99 (0.45)	94.46 (0.14)
FedAvg	70.59 (0.51)	96.91 (0.11)	86.66 (0.21)	82.44 (0.41)	97.38 (0.05)
FedProx	71.55 (0.75)	96.98 (0.19)	86.60 (0.18)	82.67 (0.75)	97.30 (0.17)
FedBN	**76.93 (0.25)**	**97.69 (0.10)**	**87.46 (0.20)**	**83.57 (0.38)**	**97.55 (0.11)**

Table 11: The detailed statistics reported with format mean (std) of accuracy presented on Fig. 5 .

E.3 COMPARE FEDBN WITH CENTRALIZED TRAINING

To better understand the significance of the numbers reported in our main context, we compare FedBN with centralized training, that pools all training data in to a center. We present the testing accuracy on each digit dataset in Table 12. FedBN, federated learning with data-specific BN layers, even performs better than vanilla centralized training strategy.

	SVHN	USPS	SynthDigits	MNIST-M	MNIST
Centralized	74.18 (0.44)	96.46 (0.30)	84.57 (0.38)	79.65 (0.24)	96.53 (0.19)
FedBN	**76.93 (0.25)**	**97.69 (0.10)**	**87.46 (0.20)**	**83.57 (0.38)**	**97.55 (0.11)**

Table 12: Testing accuracy on each testing sets with format mean(std) from 5-trial run.

E.4 DIFFERENT COMBINATIONS OF E AND B

In this section, we show different combinations of local update epochs E and batch size B. Specifically, $E \in \{1, 4, 16\}$ and $B \in \{10, 50, \infty\}$, ∞ denotes full batch learning. Following the setting in original FedAvg paper McMahan et al. (2017), we present the comparisons between FedBN and FedAvg on each combination of E and B in Table 13. The results are in good agreement that FedBN can consistently outperform FedAvg and robust to batch size selection. Further, we depicts the test sets accuracy vs. communication epochs under different combination of E and B in Figure 8. In conformity with the observation in McMahan et al. (2017), increasing local update frequency by decreasing E can produce a striking decrease on the communication round to meet a given accuracy target (e.g., the gray line in Figure 8). B does not have a clear trend of affecting communication cost in Figure 8, but smaller B achieved better final accuracy as shown in Table 13.

Setting		SVHN	USPS	SynthDigits	MNIST-M	MNIST
B=10, E=1	FedAvg	72.26	97.20	88.57	81.66	97.46
	FedBN	78.23	97.74	89.64	86.82	97.78
B=10, E=4	FedAvg	75.18	97.47	89.37	84.88	97.47
	FedBN	78.69	97.96	90.19	87.68	97.89
B=10, E=16	FedAvg	73.87	96.77	88.43	83.53	97.22
	FedBN	79.35	98.12	90.34	87.00	97.70
B=50, E=1	FedAvg	69.11	96.40	85.01	80.26	96.96
	FedBN	75.89	97.69	86.30	82.26	97.34
B=50, E=4	FedAvg	71.34	96.67	85.81	80.91	97.05
	FedBN	76.13	97.69	86.25	82.64	97.09
B=50, E=16	FedAvg	70.86	96.45	85.67	81.16	97.09
	FedBN	76.43	97.26	86.11	82.05	97.03
B=∞, E=1	FedAvg	68.73	96.77	84.70	79.30	96.71
	FedBN	73.30	97.96	85.00	81.39	97.41
B=∞, E=4	FedAvg	71.87	97.10	85.38	80.27	96.91
	FedBN	76.02	97.58	85.40	82.49	97.17
B=∞, E=16	FedAvg	71.56	96.77	84.24	79.99	96.56
	FedBN	74.83	97.04	83.76	81.44	97.25

Table 13: Test sets accuracy using different combinations of batch size B and local update epoch E on benchmark experiment with the default non-iid setting.

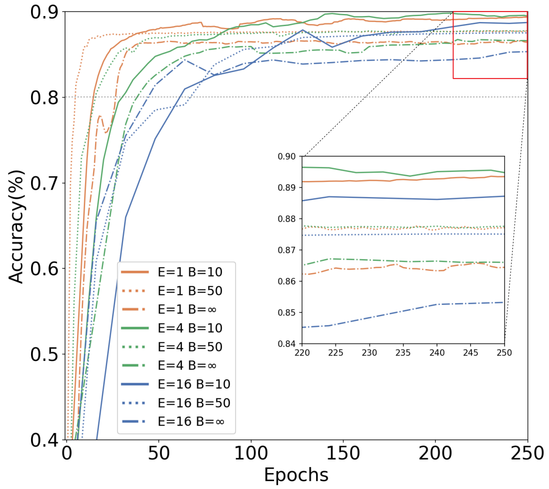


Figure 8: Test set accuracy curve (average of 5 datasets) vs. communication round of using different local updating epochs E and batch size B for FedBN.

E.5 DETAILED STATISTICS OF VARYING LOCAL DATASET SIZE EXPERIMENT

Considering putting all results in one figure (15 lines) might affect readability of the figure, we excluded the statistics of FedAvg in our Fig. 4 (b), the ablation study of our method on the effect of local dataset size. Here, we list the full results in Table 14. It is not too surprising that at Singleset can be the best when the a local client gets a lot of data.

Setting		100%	60%	40%	20%	10%	5%	1%
MNIST	SingleSet	98.09	97.89	97.45	96.24	94.35	90.86	75.28
	FedAvg	98.82	**98.65**	98.19	97.18	95.99	94.01	79.49
	FedBN	**98.93**	98.57	**98.35**	**97.76**	**96.73**	**95.24**	**84.19**
SVHN	SingleSet	86.76	84.62	82.75	76.42	68.54	53.85	12.06
	FedAvg	82.68	80.00	78.33	72.79	64.06	48.47	21.95
	FedBN	**87.31**	**85.05**	**83.55**	**78.43**	**72.10**	**58.88**	**28.30**
USPS	SingleSet	98.87	98.49	97.85	96.94	95.11	93.01	80.11
	FedAvg	98.39	98.17	97.85	97.04	95.38	94.30	81.02
	FedBN	**99.30**	**98.98**	**98.55**	**98.33**	**97.47**	**96.24**	**85.00**
Synth	SingleSet	94.33	92.82	91.02	87.50	80.47	70.61	14.10
	FedAvg	94.07	92.27	90.97	86.95	82.44	72.51	37.22
	FedBN	**94.91**	**93.16**	**91.58**	**88.47**	**83.29**	**74.81**	**40.39**
MNISTM	SingleSet	**92.41**	**91.63**	**89.41**	**84.34**	77.59	66.02	17.46
	FedAvg	90.04	88.74	86.20	83.11	76.11	67.74	40.69
	FedBN	91.66	89.75	88.00	83.84	**79.20**	**70.93**	**44.82**

Table 14: Model performance over varying dataset sizes on local clients

E.6 TRAINING ON UNEQUAL DATASET SIZE

In our benchmark experiment (Section 5.1), we truncate the sample size of the five datasets to their smallest number. This data preprocessing intends to strictly control non-related factors (e.g., imbalanced sample numbers across clients), so that the experimental findings can more clearly reflect the effect of local BN. In this regard, truncating datasets is a reasonable way to make each client have an equal number of data points and local update steps. It is also possible to keep the data sets in their original size (which is unequal), by allowing clients with less data to repeat sampling. In this way, all clients use the same batch size and same local iterations of each epoch. We add results of such a setting with 10% and full original datasize in Table 15 and Table 16 respectively. It is observed that FedBN still consistently outperforms other methods.

Method	SVHN 7943	USPS 743	SynthDigits 39116	MNIST-M 5600	MNIST 5600
FedAvg	87.00	98.01	97.55	88.69	98.75
FedProx	86.75	97.90	97.53	88.86	98.86
FedBN	**89.34**	**98.28**	**97.83**	**90.34**	**98.89**

Table 15: Testing accuracy of each clients when clients' training samples are unequal using 10% of original data. The number of training samples for each client are denoted under their names.

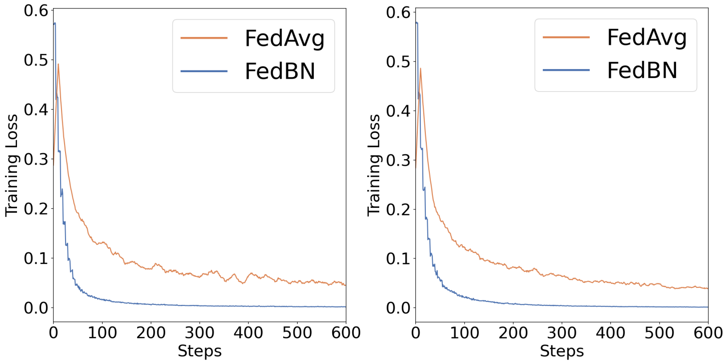
Method	SVHN 79430	USPS 7430	SynthDigits 391160	MNIST-M 56000	MNIST 56000
FedAvg	99.59	92.27	98.71	99.30	95.27
FedProx	99.50	92.12	98.66	99.27	95.44
FedBN	**99.62**	**94.34**	**98.92**	**99.54**	**96.72**

Table 16: Testing accuracy of each clients when clients' training samples are unequal using full size data. The number of training samples for each client are denoted under their names.

F SYNTHETIC DATA EXPERIMENT

Settings We generate data from two-pair of multi-Gaussian distributions. For one pair, samples $(x, 0)$ and $(x, 1)$ are sampled from $\mathcal{N}(-1,\ \Sigma_1)$ and $\mathcal{N}(1,\ \Sigma_1)$ respectively, with coveriance $\Sigma_1 \in \mathbb{R}^{10\times 10}$. For another pair, samples $(\widetilde{x}, 0)$ and $(\widetilde{x}, 1)$ are sampled from $\mathcal{N}(-1,\ \Sigma_2)$ and $\mathcal{N}(1,\ \Sigma_2)$ respectively, with coveriance $\Sigma_2 \in \mathbb{R}^{10\times 10}$. Specifically, we design convariance matrix Σ_1 as an identity diagonal matrix and Σ_2 is different from Σ_1 by having non-zero values on off-diagonal entries. We train a two-layer neural network with 100 hidden neurons for 600 steps using cross-entropy loss and SGD optimizer with 1×10^{-5} learning rate. Denote W_k and b_k are the in-connection weigths and bias term of neuron k. We initialize the model parameters with $W_k \sim \mathcal{N}(0, \alpha^2\mathbf{I}),\ b_k \sim \mathcal{N}(0, \alpha^2)$, where $\alpha = 10$.

Results. The aim of synthetic experiments is to study the behavior of using FedBN with a controlled setup. We achieve 100% accuracy on binary classification for FedAvg and FedBN. Fig. 9 shows comparison of training loss curve over steps using FedAvg and FedBN, presenting that FedBN obtains significantly faster convergence than FedAvg.



(a) client 1 (b) client 2

Figure 9: Training loss on synthetic data. Data in client 1 is generated from Diagonal Gaussian, client 2 is generated from combination of Diagonal Gaussian and Full Gaussian.

G TRANSFER LEARNING AND TESTING ON UNKNOWN DOMAIN CLIENT

In this section, we discuss out-of-domain generalization of FedBN and prove the solutions for the following two scenarios: 1) transferring FedBN to a new unknown domain clients during training; 2) testing a unknown domain client.

If a new center from another domain joins training, we can transfer the non-BN layer parameters of the global model to this new center. This new center will compute its own mean and variance statistics, and learn the corresponding local BN parameters.

	Morpho-global	Morpho-local
FedBN	**92.45**	**94.61**
FedProx	92.35	94.31
FedAvg	91.28	93.55

Table 17: Generalizing the global model to unseen-domain clients.

Testing the global model on a new client with unknown statistics outside federation requires allowing access to local BN parameters at testing time (though BN layers are not aggregated at the global server during training). In this way, the new client can use the averaged trainable BN parameters learned at existing FL clients, and compute the (mean, variance) on its own data. Such a solution is also in line with what was done in recent literature, e.g., SiloBN (Andreux et al., 2020). We conduct the experiment with this solution for FedBN and compared its performance with FedAvg and FedProx. Specifically, we use the digits classification task and treat the two unseen datasets – Morpho-global and Morpho-local from Morpho-MNIST (Castro et al., 2019) as the two new clients. The new clients contain substantially perturbed digits. Specifically, Morpho-global containing thinning and thickening versions of MNIST digits, while Morpho-local changes MNIST by swelling and fractures. The results are listed in Table 17. It is observed that the obtained results from three methods are generally comparable in such a challenging setting, with FedBN presenting slightly higher performance on overall average accuracy.