

FedStein: Enhancing Multi-Domain Federated Learning Through James-Stein Estimator

Sunny Gupta¹, Nikita Jangid², Amit Sethi³

Indian Institute of Technology Bombay
sunnygupta@iitb.ac.in¹, nikitajangid@iitb.ac.in², aset@iitb.ac.in³

Abstract

Federated Learning (FL) facilitates data privacy by enabling collaborative in-situ training across decentralized clients. Despite its inherent advantages, FL faces significant challenges of performance and convergence when dealing with data that is not independently and identically distributed (non-i.i.d.). While previous research has primarily addressed the issue of skewed label distribution across clients, this study focuses on the less explored challenge of multi-domain FL, where client data originates from distinct domains with varying feature distributions. We introduce a novel method designed to address these challenges – **FedStein**: Enhancing Multi-Domain Federated Learning Through the James-Stein Estimator. FedStein uniquely shares only the James-Stein (JS) estimates of batch normalization (BN) statistics across clients, while maintaining local BN parameters. The non-BN layer parameters are exchanged via standard FL techniques. Extensive experiments conducted across three datasets and multiple models demonstrate that FedStein surpasses existing methods such as FedAvg and FedBN, with accuracy improvements exceeding 14% in certain domains leading to enhanced domain generalization. The code is available at <https://github.com/sunnyinAI/FedStein>

Introduction

Federated learning (FL) represents a transformative paradigm in machine learning, enabling collaborative modelling across decentralized devices while maintaining local data privacy. Unlike traditional centralized methods, FL conducts model training directly on individual devices, transmitting only model updates instead of raw data. This approach not only preserves data privacy, but also aligns with stringent data governance standards, making FL particularly appealing in a variety of domains, including healthcare (Li et al. 2019; Bernecker et al. 2022), mobile devices (Hard et al. 2018; Paulik et al. 2021), and autonomous vehicles (Zhang, Bosch, and Olsson 2021; Posner et al. 2021; Nguyen et al. 2022). However, FL faces significant challenges when applied to data that are not independently and identically distributed (non-i.i.d.) between different clients (Li et al. 2020). These challenges manifest themselves as performance degradation (Tan et al. 2024; Hsieh

et al. 2020; Zhao et al. 2018), instability during training (Wang, Shi, and Chang 2023; Zhuang et al. 2023, 2020), and biases in the resulting models. Most research addressing non-i.i.d. challenges in FL has focused on skewed label distributions, where each client has a different distribution of labels (Chen et al. 2022a; Wang et al. 2020; Hsieh et al. 2020; Li et al. 2020). While this focus is crucial, it often overlooks a critical aspect of real-world FL applications: multi-domain federated learning. In multi-domain FL, client data come from diverse domains, each characterized by unique feature distributions rather than merely differing in label distributions. For example, autonomous vehicles may collect data under various weather conditions or at different times of the day, leading to domain gaps in the images captured by a single client (Yu et al. 2020; Cordts et al. 2016). Similarly, image data from different institutions in healthcare can show significant variation due to differences in equipment and protocols (Bernecker et al. 2022).

The existing solutions fail to effectively address the challenges posed by multi-domain Federated Learning (FL). One notable attempt, FedBN (Li et al. 2021), seeks to mitigate this issue by retaining Batch Normalization (BN) parameters and statistics (Ioffe and Szegedy 2015) locally on each client. However, FedBN is primarily suited for cross-silo FL scenarios, where clients are typically stable organizations, such as healthcare institutions, capable of maintaining state (Karimireddy et al. 2020) and consistently participating in every training round. This requirement for clients to be stateful means that they must preserve BN-related information across training rounds. By contrast, FedBN is less suitable for cross-device FL scenarios, where clients are stateless and only a subset participates in training.

Moreover, BN assumes that the training data originates from a single distribution, ensuring that the mean and variance computed from each mini-batch are representative of the entire dataset (Ioffe and Szegedy 2015).

Although alternative normalization techniques, such as Layer Norm (Ba, Kiros, and Hinton 2016) and Group Norm (Wu and He 2018), have been proposed, their applicability to multi-domain FL has not been thoroughly explored, and they come with limitations such as increased computational overhead during inference.

This paper proposes an approach to address the challenges inherent in multi-domain Federated Learning (FL). Given

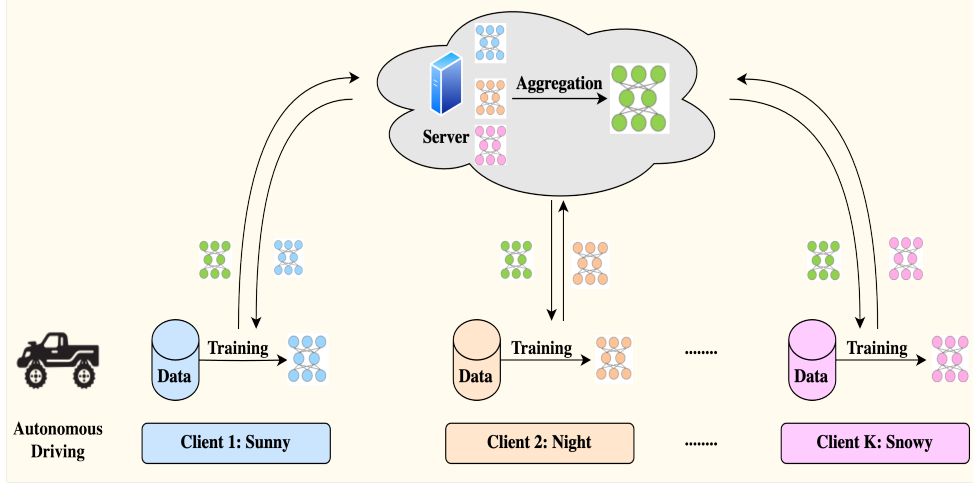


Figure 1: We focus on multi-domain federated learning, where each client possesses data from a specific domain. This framework is highly relevant and useful in real-world applications. For example, autonomous vehicles in different regions gather images under various weather conditions.

the difficulties BN faces in handling multi-domain data and the limitations of alternative normalization techniques, we explore a critical question: Is normalization truly indispensable for learning a robust global model in multi-domain FL? Recent research on normalization techniques using the James-Stein Estimator (Khoshsirar and Kambhamettu 2024) suggests that models can achieve better performance than standard BN by countering differences between the statistics between batches (Ioffe and Szegedy 2015). Building on this insight, we investigate the potential of this methodology within the context of multi-domain FL.

FedStein adheres to the FedAvg (McMahan et al. 2017) protocols for server aggregation and client training. However, unlike existing methods, FedStein selectively aggregates only the essential James-Stein (JS) estimates of Batch Normalization (BN) statistics – namely the mean (μ_{JS}) and variance (σ_{JS}^2) – while eliminating the need to synchronize the BN parameters (γ and β). The parameters of non-BN layers are disseminated conventionally, thereby maintaining the integrity of the model’s performance across diverse domains.

We conducted extensive experiments in three datasets and confirmed that FedStein outperforms state-of-the-art methods in all cases. The global model trained using FedStein achieves an improvement of more than 14% on certain domains compared to the personalized models generated by FedBN (Li et al. 2021).

Our contributions to this study are threefold:

1. We formulate FedStein, a new FL approach to address domain discrepancies among clients in multi-domain Federated Learning (FL).
2. We demonstrate that, unlike traditional methods, FedStein effectively tackles the challenges posed by non-identically distributed data in FL, ensuring a more resilient and accurate model across varied datasets.
3. We show that FedStein is versatile in offering robust sup-

port for cross-silo Federated Learning (FL) environments by introducing JS estimates of BN Statistics. This approach further enhances privacy by not communicating local activation statistics, thereby revealing less sensitive information.

Related Work

Batch Normalization: Batch Normalization (Ioffe and Szegedy 2015) process involves normalizing the inputs of each layer by computing the mean and variance from a mini-batch, followed by scaling and shifting through learnable parameters. Specifically, given a batch of inputs $\{x_1, x_2, \dots, x_m\}$, BN calculates the mean μ_B and variance σ_B^2 as follows:

$$\mu_B = \frac{1}{m} \sum_{i=1}^m x_i \quad (1)$$

$$\sigma_B^2 = \frac{1}{m} \sum_{i=1}^m (x_i - \mu_B)^2 \quad (2)$$

The normalized output is then computed as:

$$\hat{x}_i = \frac{x_i - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}}, \quad (3)$$

To provide the network with the flexibility to adjust the normalization, BN incorporates learnable scaling and shifting parameters, γ and β , applied as follows:

$$y_i = \gamma \hat{x}_i + \beta. \quad (4)$$

where μ and σ^2 , hereafter denoted as BN statistics, are calculated as the running means and variances, respectively, of each channel computed across both spatial and batch dimensions, and γ and β are learned affine renormalization parameters, and where all computations are performed along the channel axis. The term ϵ is a small positive constant added for numerical stability.

Federated Learning with Batch Normalization

In centralized training paradigms, batch normalization (BN) has faced significant challenges when modelling statistics across multiple domains. This limitation has spurred the development of domain-specific BN techniques designed to better accommodate variability in data distributions (Li et al. 2016; Chang et al. 2019). These challenges are further exacerbated in the context of multi-domain Federated Learning (FL), where deep neural networks (DNNs) that rely on BN may struggle to accurately capture the statistical characteristics of diverse domains while attempting to train a unified global model.

A prominent issue addressed for non-i.i.d. Federated learning (FL) is the skewed label distribution, where label distributions vary significantly across clients. This disparity can lead to biased model performance and hinder generalization. To address this challenge, several strategies have been proposed, including specialized operations in BN to tailor the models to the unique data distributions of each client (Li et al. 2020; Zhang et al. 2023; Chen et al. 2022b). For example, SiloBN (Andreux et al. 2020) retains BN statistics locally on each client, ensuring that the normalization process is customized to the specific data distribution of the client. Similarly, FixBN (Zhong et al. 2023) mitigates the problem by training BN statistics during the early stages and subsequently freezing them to maintain consistency.

In contrast, multi-domain FL has received comparatively less attention (Chen et al. 2018; Shen et al. 2022). To address the unique challenges posed by multi-domain FL approaches such as FedBN (Li et al. 2021) and FedNorm (Bernecker et al. 2022) have been developed. These methods retain BN layers locally on clients while aggregating only the remaining model parameters. Similarly, PartialFed (Sun et al. 2021) preserves model initialization strategies on clients, leveraging these strategies to load models in subsequent training rounds.

Alternative Normalization Methods

Despite its widespread effectiveness, BN encounters several limitations in certain scenarios. For example, BN may struggle to accurately capture the statistical properties of training data originating from multiple domains (Li et al. 2016; Chang et al. 2019). To address these limitations, researchers have proposed alternative normalization techniques, such as Group Normalization (GN) (Wu and He 2018) and Layer Normalization (LN) (Ba, Kiros, and Hinton 2016). Although these methods alleviate some of the constraints associated with BN, such as its dependence on batch size, they introduce their own set of challenges. For example, both GN and LN require additional computational overhead during inference, which can limit their practicality, particularly in edge-to-edge deployment scenarios where computational resources are constrained. Recent studies have further highlighted that Batch Normalization (BN) may not perform optimally in Federated Learning (FL) environments, especially under non-i.i.d. data conditions (Hsieh et al. 2020). This sub-optimal performance is primarily because of external covariate shifts (Du et al. 2022) and the mismatch between local and global statistics (Wang et al. 2023). In response to

these challenges, alternative normalization techniques such as Group Normalization (GN) (Hsieh et al. 2020; Casella et al. 2023) and Layer Normalization (LN) (Du et al. 2022; Casella et al. 2023) have been explored. However, these alternatives are not without their drawbacks. GN and LN inherit some of the limitations observed in centralized training, including increased computational complexity and sensitivity to specific hyperparameters.

Normalization with James-Stein Estimator

One notable concern about Equation 1 lies in the estimation of the mean and variance. The conventional approach suggests independently calculating the mean and variance using “usual estimators”. For batch normalization, the estimators are given as follows:

$$\mathbb{E}[x_i] = \frac{1}{n} \sum_{j=1}^n x_{i,j}, \quad (5)$$

$$\text{Var}[x_i] = \frac{1}{n} \sum_{j=1}^n (x_{i,j} - \mathbb{E}[x_i])^2. \quad (6)$$

Given that all the features contribute to a shared loss function, according to Stein’s paradox (Wikipedia 2023), these estimators are inadmissible when $c \geq 3$. Notably, in computer vision networks, it is consistently observed that $c \geq 3$. To address this, a novel method was adopted by (Khosh-sirat and Kambhamettu 2024) to adopt admissible shrinkage estimators, which effectively enhance the estimation of the mean and variance in normalization layers.

Let $X = \{x_1, x_2, \dots, x_c\}$ with unknown means $\theta = \{\theta_1, \theta_2, \dots, \theta_c\}$ and estimates $\hat{\theta} = \{\hat{\theta}_1, \hat{\theta}_2, \dots, \hat{\theta}_c\}$. The basic formula for the James-Stein estimator is:

$$\hat{\theta}_{JS} = \hat{\theta} + s(\mu_{\hat{\theta}} - \hat{\theta}), \quad (7)$$

where $\mu_{\hat{\theta}} - \hat{\theta}$ is the difference between the total mean (average of averages) and each estimated mean, and s is a shrinkage factor. Among the numerous perspectives that motivate the James-Stein estimator, the empirical Bayes perspective (Efron and Morris 1975) is the most insightful. Taking a Gaussian prior on the unknown means leads us to the following formula (James and Stein 1992):

$$\hat{\theta}_{JS} = \left(1 - \frac{(c-2)\sigma^2}{\|\hat{\theta} - v\|_2^2}\right) (\hat{\theta} - v) + v, \quad (8)$$

where $\|\cdot\|_2$ denotes the L_2 norm of the argument, σ^2 is the variance, v is an arbitrarily fixed vector that shows the shrinkage direction, and $c \geq 3$. Setting $v = 0$ results in the following:

$$\hat{\theta}_{JS} = \left(1 - \frac{(c-2)\sigma^2}{\|\hat{\theta}\|_2^2}\right) \hat{\theta}. \quad (9)$$

The above estimator shrinks the estimates towards the origin 0. Using Equation 6 helps to mitigate the ‘mean shift’ problem (Brock et al. 2021a,b). To incorporate this equation into normalization layers, we replace the $\hat{\theta}$ in Equation 6 with the estimated mean and variance derived from the original method. By applying the James-Stein estimator to these estimated statistics, the additional computational overhead is

minimal and can be considered negligible (Khoshsirat and Kambhamettu 2024). Thus, the James-Stein estimator is utilized for both the mean and variance within the normalization layers. In the context of batch normalization, $\mathbb{E}[x]$ and $\text{Var}[x]$ are vectors of length c (where c represents the number of channels in the batch). These vectors can be directly substituted in place of $\hat{\theta}$.

More recently, (Khoshsirat and Kambhamettu 2024) has shown that the mean and variance estimators commonly used in normalization layers are inadmissible. They introduced a novel approach that employs the James-Stein estimator (Stein et al. 1956) to enhance the estimation of mean and variance within these layers. This improved normalization technique consistently yields superior accuracy across a variety of tasks without imposing additional computational burdens. Their method achieves competitive results compared to Batch Normalization (BN) on prominent architectures such as ResNet (He et al. 2016), EfficientNet (Tan and Le 2019), and Swin Transformer v2 (Liu et al. 2022).

Methodology

Batch Normalization offers several key advantages, including the mitigation of internal covariate shift, stabilization of the training process, and acceleration of convergence (Santurkar et al. 2018). BN also reduces the number of iterations required to reach convergence, thereby improving overall performance (Khoshsirat and Kambhamettu 2023). Moreover, it enhances robustness to variations in hyperparameters (Bjorck et al. 2018) and contributes to a smoother optimization landscape (Santurkar et al. 2018). However, the effectiveness of BN is predicated on the assumption that the training data is homogeneous across the dataset, ensuring that the mean (μ) and variance (σ) computed from each mini-batch are representative of the overall data distribution (Ioffe and Szegedy 2015).

James-Stein Estimator

Estimating the mean of a multivariate normal distribution is a fundamental problem in statistics. Typically, the sample mean is employed, which also serves as the maximum-likelihood estimator. However, the James-Stein (JS) estimator, despite being biased, is utilized for estimating the mean of c correlated Gaussian-distributed random vectors with unknown means. The development of the JS estimator is rooted in two pivotal papers, with the initial version introduced by Charles Stein in 1956 (Stein et al. 1956). Stein’s work led to the surprising revelation that the standard mean estimate is admissible when $c \leq 2$ but becomes inadmissible when $c \geq 3$. This breakthrough suggested an improvement by shrinking the sample means towards a central vector of means, a concept commonly referred to as Stein’s paradox or Stein’s example (Efron and Morris 1977).

Applying James-Stein Estimation to Batch Normalization

Implementing James-Stein Normalization (JSNorm) by incorporating the James-Stein estimator into the standard

Algorithm 1: Aggregated Batch Normalization with James-Stein Estimator

Notations:

k : Index for user

l : Index for neural network layer

$w_{0,k}^{(l)}$: Initialized model parameters

E : Local update frequency

T : Total rounds

K : Number of users

Require: Batch of inputs $\{x_1, x_2, \dots, x_m\}$, small constant ϵ , learnable parameters γ, β , dimensionality c

Ensure: Normalized outputs $\{y_1, y_2, \dots, y_m\}$

for each round $t = 1$ to T **do**

for each user k and each layer l **do**

 Update model parameters:

$$w_{t+1,k}^{(l)} \leftarrow \text{SGD}(w_{t,k}^{(l)})$$

end for

if $t \bmod E = 0$ **then**

for each user k and each layer l **do**

if layer l is not BatchNorm **then**

 Aggregate the model parameters:

$$w_{t+1,k}^{(l)} \leftarrow \frac{1}{K} \sum_{k=1}^K w_{t+1,k}^{(l)}$$

else

Do not aggregate BN parameters (γ and β)

 Aggregate the BatchNorm statistics across users:

 Aggregate the batch mean:

$$\mu_{B,\text{agg}} = \frac{1}{K} \sum_{k=1}^K \mu_B^{(k)}$$

 Aggregate the batch variance:

$$\sigma_{B,\text{agg}}^2 = \frac{1}{K} \sum_{k=1}^K \sigma_B^{2(k)}$$

 Adjust the aggregated mean using the James-Stein Estimator:

$$\mu_{JS,\text{agg}} = \left(1 - \frac{(c-2)\sigma_{\mu_{B,\text{agg}}}^2}{\|\mu_{B,\text{agg}}\|^2}\right) \mu_{B,\text{agg}}$$

 Adjust the aggregated variance using the James-Stein Estimator:

$$\sigma_{JS,\text{agg}}^2 = \left(1 - \frac{(c-2)\sigma_{\sigma_{B,\text{agg}}}^2}{\|\sigma_{B,\text{agg}}\|^2}\right) \sigma_{B,\text{agg}}^2$$

for each input x_i in the batch **do**

 Normalize the input using aggregated statistics:

$$\hat{x}_i = \frac{x_i - \mu_{JS,\text{agg}}}{\sqrt{\sigma_{JS,\text{agg}}^2 + \epsilon}}$$

 Apply scaling and shifting:

$$y_i = \gamma \hat{x}_i + \beta$$

end for

end if

end for

end if

end for

return Normalized outputs $\{y_1, y_2, \dots, y_m\}$

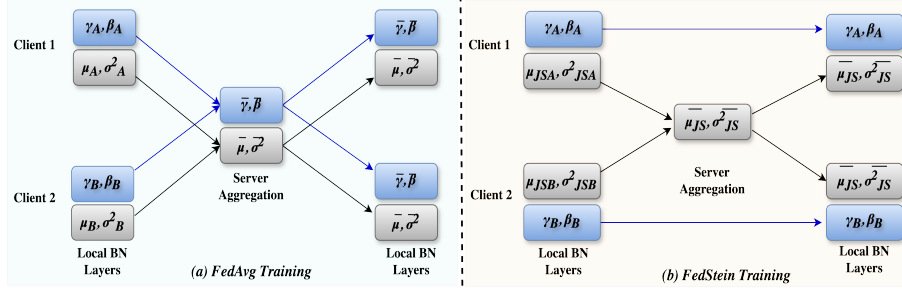


Figure 2: An illustration of two different approaches to multi-centre training of BN layers. This description follows the definitions provided in Eqn 3. Computation flows from left to right. (a) In **FedAvg**, both BN parameters and BN statistics are aggregated into one server, and (b) in **FedStein**, BN parameters are removed, and the JS norms of the BN statistics are aggregated into the server. Non-BN layers are shared in both methods.

Batch Normalization (BN) framework to enhance the robustness of BN in federated learning scenarios, where data distributions can be heterogeneous and high-dimensional, the batch mean μ_B and variance σ_B^2 are adjusted using the James-Stein estimator, resulting in the modified statistics as per Algorithm 1:

JSNorm in Federated Learning

While Batch Normalization (BN) layers are integral components of many modern neural network architectures (He et al. 2016; Hu, Shen, and Sun 2018; Tan and Le 2019), their application in federated learning settings has not been thoroughly investigated and is often overlooked or omitted entirely (McMahan et al. 2017). The naïve implementation of FedAvg, for instance, does not differentiate between the local activation statistics (μ, σ^2) and the trained renormalization parameters (γ, β) , leading to a straightforward aggregation of both at each federated round, as illustrated in Figure 2 (left). This simplistic approach serves as a baseline for using FedAvg in networks that include BN layers. Moreover, BN layers can serve a dual purpose by distinguishing between local and domain-invariant information. Specifically, the BN statistics and the learned BN parameters fulfil different roles (Li et al. 2016): the former encapsulates local domain-specific information, while the latter can be transferred across different domains. In traditional federated learning approaches, all Batch Normalization (BN) parameters are typically treated equally during aggregation. However, this overlooks the distinct roles played by BN statistics (μ, σ^2) and learned parameters (γ, β) . We propose a new method called FedStein, which differentiates between these roles by sharing only the James-Stein (JS) estimates of BN statistics across different federated centres while keeping the learned BN parameters local to each centre. The parameters of non-BN layers are shared using the standard federated learning approach. This method is illustrated in Figure 2 (right). By synchronizing the JS estimates of BN statistics, FedStein enables the federated training of a model that is more robust to the heterogeneity present across different centres, thereby improving the overall model performance in multi-domain federated learning scenarios. By incorporating the James-Stein adjustment, JSNorm provides a more

robust estimation of these statistics by guiding the sample means toward a more centralized mean vector, resulting in improved model performance across clients. This adjustment is particularly advantageous in scenarios involving high-dimensional data or small batch sizes, where standard Batch Normalization (BN) might otherwise face significant challenges. The results show that our improved normalization layers consistently deliver superior accuracy, without incurring any additional computational overhead.

Experimental Evaluation

Results on Cross-silo Federated Learning

Table 1 presents a comprehensive comparison of various methods under cross-silo federated learning (FL) on the Digits-Five, Office-Caltech-10, and DomainNet datasets. The proposed FedStein method consistently outperformed state-of-the-art approaches across most domains and datasets. Notably, FedProx, which incorporates a proximal term into FedAvg, exhibits performance similar to FedAvg. These two methods surpass SingleSet on the Digits-Five dataset but may demonstrate inferior performance compared to SingleSet in certain domains on the more challenging Office-Caltech-10 and DomainNet datasets.

Datasets: We conducted a series of experiments on multi-domain federated learning (FL) using three datasets: Digits-Five (Li et al. 2021), Office-Caltech-10 (Gong et al. 2012), and DomainNet (Peng et al. 2019). The Digits-Five dataset encompasses five distinct sets of 28x28 pixel digit images: MNIST (LeCun et al. 1998), SVHN (Netzer et al. 2011), USPS (Hull 1994), SynthDigits (Ganin and Lempitsky 2015), and MNIST-M (Ganin and Lempitsky 2015), each representing a unique domain. Office-Caltech-10 comprises real-world object images from four domains: three sourced from the Office-31 dataset (WebCam, DSLR, and Amazon) (Saenko et al. 2010), and one from the Caltech-256 dataset (Caltech) (Griffin, Holub, and Perona 2007). DomainNet (Peng et al. 2019), known for its complexity, includes large 244x244 pixel object images spanning six domains: Clipart, Infograph, Painting, Quickdraw, Real, and Sketch.

To simulate realistic scenarios with limited client data, we

Dataset	Domains	SingleSet	FedAvg	FedProx	+GN ^a	+LN ^b	SiloBN	FixBN	FedBN	FedStein (Ours)
Digit-Five	MNIST	94.4	96.2	96.4	96.4	96.4	96.2	96.3	96.3	96.8
	SVHN	67.1	71.6	71.0	76.9	75.2	71.3	71.3	71.1	75.4
	USPS	95.4	96.3	96.1	96.6	96.4	96.0	96.1	96.6	97.1
	SynthDigits	80.3	86.0	85.9	85.6	85.6	86.0	84.8	86.8	86.6
	MNIST-M	77.0	82.5	83.1	83.7	82.2	83.1	83.0	78.6	84.1
	Average	83.1	86.5	86.5	87.8	87.1	86.5	86.5	87.2	88.1
Caltech-10	Amazon	54.5	61.8	59.9	60.8	55.0	60.8	59.2	63.2	64.1
	Caltech	40.2	44.9	44.0	50.8	41.3	44.4	44.0	45.3	47.6
	DSLR	81.3	77.1	76.0	88.5	79.2	76.0	79.2	83.1	83.4
	Webcam	89.3	81.4	80.8	83.6	71.8	81.9	79.6	85.6	93.2
	Average	66.3	66.3	65.2	70.9	61.8	65.8	65.5	69.2	72.1
DomainNet	Clipart	42.7	48.9	51.1	45.4	42.7	51.8	49.2	51.2	58.4
	Infograph	24.0	26.5	24.1	21.1	23.6	25.0	24.5	26.8	27.6
	Painting	34.2	37.7	37.3	35.4	35.3	36.4	38.2	41.5	51.5
	Quickdraw	70.9	44.5	46.1	57.2	46.0	45.9	46.3	64.8	55.4
	Real	51.2	46.8	45.5	50.7	43.9	47.7	46.2	55.2	61.5
	Sketch	33.5	35.7	37.5	36.5	28.9	38.0	37.4	39.6	40.3
	Average	42.9	40.0	40.2	41.1	36.7	40.8	40.3	45.8	49.1

Table 1: Comparison of testing accuracy (%) across different methods on three datasets. The proposed FedStein method consistently outperforms existing approaches in the majority of domains, achieving the highest average testing accuracy across all datasets.

^a+GN means FedAvg+GN, ^b+LN means FedAvg+LN

used a reduced version of the standard digit datasets from (Li et al. 2021), limiting them to 7,438 training samples, divided uniformly among 20 clients to mimic a cross-device FL setup, with a total of 100 clients. For the DomainNet dataset, we focused on 10 classes, each containing 2,000 to 5,000 images, assigning each client images from a single domain to simulate multi-domain FL.

Implementation Details We implemented FedStein using PyTorch (Paszke et al. 2017) and ran experiments on a high-performance cluster with four NVIDIA A6000 GPUs. The models tested included a 6-layer CNN (Li et al. 2021) for Digits-Five, AlexNet (Krizhevsky, Sutskever, and Hinton 2017) and ResNet-18 (He et al. 2016) for Office-Caltech-10, and AlexNet for DomainNet. We used cross-entropy loss for classification error and optimized with SGD. Learning rates were fine-tuned between [0.001, and 0.1] for optimal performance.

Performance Evaluation We evaluated the performance of our proposed FedStein method by benchmarking it against three distinct categories of approaches. The first category includes state-of-the-art methods, which feature advanced techniques for Batch Normalization (BN), such as SiloBN (Andreux et al. 2020), FedBN (Li et al. 2021), and FixBN (Zhong et al. 2023). The second category consists of baseline algorithms, encompassing widely recognized federated learning (FL) algorithms like FedProx (Li et al. 2020), FedAvg (McMahan et al. 2017), and SingleSet, where a model is trained independently on each client using only local data. Lastly, we examined alternative normalization methods, specifically exploring the performance of FedAvg in combination with Group Normalization (FedAvg+GN) and Layer Normalization (FedAvg+LN), where the BN layers were replaced with GN and LN layers, respectively.

SiloBN and FixBN show performance comparable to FedAvg in terms of average accuracy; however, they are not specifically designed for multi-domain FL, and consequently, they achieve only baseline results. In contrast, FedBN, explicitly designed for multi-domain FL, surpasses these methods in overall performance.

Furthermore, our findings indicate that simply replacing BN with GN (FedAvg+GN) can enhance the performance of FedAvg, as GN does not rely on batch-specific domain statistics. FedAvg+GN achieves results comparable to FedBN on the Digits-Five and Office-Caltech-10 datasets. Remarkably, our proposed FedStein method surpasses both FedAvg+GN and FedBN in terms of average accuracy across all datasets. While FedStein trails FedBN by less than 1% in two domains, it outperforms FedBN by more than 10% in specific domains. These results underscore the effectiveness of FedStein in cross-silo FL scenarios.

Results on medical images.

Methods	Center 1	Center 2	Center 3	Center 4	Center 5	Center 6
FedAvg	0.40	0.21	0.37	0.42	0.39	0.43
FedBN	0.31	0.38	0.43	0.39	0.30	0.36
FedStein (Ours)	0.52	0.57	0.65	0.56	0.46	0.62

Table 2: Evaluation on the Fed-ISIC2019 dataset, comprising medical images from six distinct centers, demonstrates that FedStein significantly outperforms both FedAvg and FedBN.

As a further experiment (Table 2) to demonstrate the advantages of our proposed FedStein approach in Feder-

ated Learning (FL) across multi-domains, we extend our analysis to the diagnosis of skin lesions utilizing datasets from the ISIC2019 Challenge (Codella et al. 2018; Combalia et al. 2019) and the HAM10000 dataset (Tschandl, Rosendahl, and Kittler 2018). There are four hospitals represented in these datasets, including one hospital employing three different imaging technologies. Like that described by Flamby (Ogier du Terrail et al. 2022), we have organized the data into six centres: BCN, Vidir-molemax, Vidir-modern, Rosendahl, MSK, and Vienna-dias. Each centre’s data is associated with its domain, ensuring that the images reflect the differences found between many healthcare institutions and imaging methods.

Datasets: The dataset comprises a total of 23,247 images of skin lesions, distributed as follows: BCN contains 9930 training samples and 2483 testing samples; Vidir-molemax includes 3163 training samples and 791 testing samples; Vidir-modern has 2691 training samples and 672 testing samples; Rosendahl provides 1807 training samples and 452 testing samples; MSK comprises 655 training samples and 164 testing samples; and Vienna-dias includes 351 training samples and 88 testing samples.

Implementation Details We conducted experiments using ResNet-18 (He et al. 2016) without pre-training, with local epochs $E = 1$, batch size $B = 64$, and 50 rounds. We used the SGD optimizer with learning rates $\eta = 0.005$ for FedAvg and FedStein, and $\eta = 0.001$ for FedBN, tuning η from $\{0.001, 0.005, 0.01, 0.05\}$. Following the implementation in Flamby, we employed weighted focal loss (Lin et al. 2017) and data augmentations. Table 2 shows that FedStein consistently outperforms FedAvg and FedBN across six healthcare domains, demonstrating its potential for multi-domain healthcare scenarios where data is scarce and fragmented.

Results on Domain Adaptation and Generalization

Methods	Seen Domains			Unseen Domain
	Amazon	Caltech	DSLR	WebCam
FedAvg w/o BN	37.9	34.3	30.9	29.8
FedAvg	59.9	43.2	60.1	53.2
FedBN	64.8	49.0	77.9	60.9
FedStein (Ours)	66.9	53.7	92.3	69.8

Table 3: Comparison of methods on domain generalization using the Office-Caltech-10 dataset.

The experiments for these results were performed on *Office-Caltech10* dataset, with *Amazon (A)*, *Caltech (C)*, and *DSLR (D)* designated as the “seen” domains during the training phase. The *WebCam (W)* domain, conversely, was exclusively reserved for the evaluation phase, functioning as the “unseen” domain in a zero-shot evaluation context. To rigorously assess the model’s performance, we utilized client local models to evaluate the seen domains, while the server global model was applied to test the unseen domain. The results, as documented in Table 3, compellingly illustrate that FedStein not only surpasses other models in performance

across the seen domains but also demonstrates superior generalization capabilities on the unseen domain. These findings underscore the significant advantage of FedStein, particularly in the context of domain generalization.

Conclusion

In this paper, we introduced FedStein, a novel approach for enhancing multi-domain Federated Learning (FL) that relies on the James-Stein (JS) estimation of BN statistics across clients. This method effectively addresses feature shifts in non-i.i.d. data, particularly in multi-domain scenarios where data characteristics can vary significantly across clients. Through extensive experimentation on diverse federated datasets, we demonstrate that FedStein significantly enhances both convergence behaviour and model performance in non-i.i.d. settings. Privacy remains a critical concern in federated learning, and retaining the BN parameters locally at each client in FedStein adds a layer of security, making it more challenging to compromise local data. Our extensive experiments across three multi-domain datasets and models demonstrate that FedStein consistently surpasses state-of-the-art techniques, proving its versatility in both cross-silo and domain generalization scenarios. Future work could explore this method further by evaluating it across a broader range of datasets and model architectures, especially in the context of medical imaging.

References

- Andreux, M.; et al. 2020. Siloed federated learning for multi-center histopathology image analysis. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, 30–39. Springer.
- Ba, J. L.; Kiros, J. R.; and Hinton, G. E. 2016. Layer normalization. In *arXiv preprint arXiv:1607.06450*.
- Bernecker, T.; Peters, A.; Schlett, C. L.; Bamberg, F.; Theis, F.; Rueckert, D.; Weiß, J.; and Albarqouni, S. 2022. Fed-norm: Modality-based normalization in federated learning for multi-modal liver segmentation. *arXiv preprint arXiv:2205.11096*.
- Bjorck, N.; Gomes, C. P.; Selman, B.; and Weinberger, K. Q. 2018. Understanding batch normalization. *Advances in neural information processing systems*, 31.
- Brock, A.; et al. 2021a. High-performance large-scale image recognition without normalization. *International Conference on Machine Learning*.
- Brock, A.; et al. 2021b. Characterizing signal propagation to close the performance gap in unnormalized ResNets. *Advances in Neural Information Processing Systems*.
- Casella, J.; et al. 2023. GN: Group normalization reimagined. *Journal of Machine Learning Research*.
- Chang, W.-G.; You, T.; Seo, S.; Kwak, S.; and Han, B. 2019. Domain-specific batch normalization for unsupervised domain adaptation. In *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition*, 7354–7362.
- Chen, C.; Liu, Y.; Ma, X.; and Lyu, L. 2022a. Calfat: Calibrated federated adversarial training with label skewness.

- Advances in Neural Information Processing Systems*, 35: 3569–3581.
- Chen, Y.; et al. 2022b. Fedcor: Efficient federated learning via proxy model with locality regularization. *Pattern Recognition Letters*.
- Chen, Y.-H.; et al. 2018. Multi-domain image-to-image translation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 1–17.
- Codella, N. C.; Gutman, D.; Celebi, M. E.; Helba, B.; Marchetti, M. A.; Dusza, S. W.; Kalloo, A.; Liopyris, K.; Mishra, N.; Kittler, H.; et al. 2018. Skin lesion analysis toward melanoma detection: A challenge at the 2017 international symposium on biomedical imaging (isbi), hosted by the international skin imaging collaboration (isic). In *2018 IEEE 15th international symposium on biomedical imaging (ISBI 2018)*, 168–172. IEEE.
- Combaila, M.; Codella, N. C.; Rotemberg, V.; Helba, B.; Vilaplana, V.; Reiter, O.; Carrera, C.; Barreiro, A.; Halpern, A. C.; Puig, S.; et al. 2019. Bcn20000: Dermoscopic lesions in the wild. *arXiv preprint arXiv:1908.02288*.
- Cordts, M.; Omran, M.; Ramos, S.; Rehfeld, T.; Enzweiler, M.; Benenson, R.; Franke, U.; Roth, S.; and Schiele, B. 2016. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 3213–3223.
- Du, Y.; et al. 2022. Does covariate shift really matter in deep learning? *Advances in Neural Information Processing Systems*.
- Efron, B.; and Morris, C. 1975. Data analysis using Stein’s estimator and its generalizations. *Journal of the American Statistical Association*, 70(350): 311–319.
- Efron, B.; and Morris, C. 1977. Stein’s paradox in statistics. *Scientific American*, 236(5): 119–127.
- Ganin, Y.; and Lempitsky, V. 2015. Unsupervised domain adaptation by backpropagation. In *ICML*, 1180–1189. PMLR.
- Gong, B.; et al. 2012. Geodesic flow kernel for unsupervised domain adaptation. In *IEEE CVPR*, 2066–2073. IEEE.
- Griffin, G.; Holub, A.; and Perona, P. 2007. Caltech-256 object category dataset. Technical report, California Institute of Technology.
- Hard, A.; Rao, K.; Mathews, R.; Ramaswamy, S.; Beaufays, F.; Augenstein, S.; Eichner, H.; Kiddon, C.; and Ramage, D. 2018. Federated learning for mobile keyboard prediction. *arXiv preprint arXiv:1811.03604*.
- He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.
- Hsieh, K.; Phanishayee, A.; Mutlu, O.; and Gibbons, P. 2020. The non-iid data quagmire of decentralized machine learning. In *International Conference on Machine Learning*, 4387–4398. PMLR.
- Hu, J.; Shen, L.; and Sun, G. 2018. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 7132–7141.
- Hull, J. J. 1994. A database for handwritten text recognition research. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(5): 550–554.
- Ioffe, S.; and Szegedy, C. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, 448–456. pmlr.
- James, W.; and Stein, C. 1992. Estimation with quadratic loss. In *Breakthroughs in statistics: Foundations and basic theory*, 443–460. Springer.
- Karimireddy, S. P.; Kale, S.; Mohri, M.; Reddi, S.; Stich, S.; and Suresh, A. T. 2020. Scaffold: Stochastic controlled averaging for federated learning. In *International conference on machine learning*, 5132–5143. PMLR.
- Khoshsirat, S.; and Kambhamettu, C. 2023. Empowering visually impaired individuals: A novel use of apple live photos and android motion photos. *arXiv preprint arXiv:2309.08022*.
- Khoshsirat, S.; and Kambhamettu, C. 2024. Improving normalization with the james-stein estimator. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2041–2051.
- Krizhevsky, A.; Sutskever, I.; and Hinton, G. E. 2017. ImageNet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6): 84–90.
- LeCun, Y.; et al. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11): 2278–2324.
- Li, T.; Sahu, A. K.; Talwalkar, A.; and Smith, V. 2020. Federated learning: Challenges, methods, and applications. *IEEE Transactions on Big Data*, 7(1): 1–18.
- Li, T.; et al. 2021. Fedbn: Federated learning on non-iid features via local batch normalization. *arXiv preprint arXiv:2102.07623*.
- Li, W.; Milletari, F.; Xu, D.; Rieke, N.; Hancox, J.; Zhu, W.; Baust, M.; Cheng, Y.; Ourselin, S.; Cardoso, M. J.; et al. 2019. Privacy-preserving federated brain tumour segmentation. In *Machine Learning in Medical Imaging: 10th International Workshop, MLMI 2019, Held in Conjunction with MICCAI 2019, Shenzhen, China, October 13, 2019, Proceedings 10*, 133–141. Springer.
- Li, Y.; Wang, N.; Shi, J.; Liu, J.; and Hou, X. 2016. Revisiting batch normalization for practical domain adaptation. *arXiv preprint arXiv:1603.04779*.
- Lin, T.-Y.; Goyal, P.; Girshick, R.; He, K.; and Dollár, P. 2017. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, 2980–2988.
- Liu, Z.; Hu, H.; Lin, Y.; Yao, Z.; Xie, Z.; Wei, Y.; Ning, J.; Cao, Y.; Zhang, Z.; Dong, L.; et al. 2022. Swin transformer v2: Scaling up capacity and resolution. In *Proceedings of the IEEE/CVF conference on CVPR*, 12009–12019.
- McMahan, H. B.; Moore, E.; Ramage, D.; Hampson, S.; and y Arcas, B. A. 2017. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, 1273–1282. PMLR.

- Netzer, Y.; et al. 2011. Reading digits in natural images with unsupervised feature learning. In *Neural Information Processing Systems Workshop*.
- Nguyen, A.; Do, T.; Tran, M.; Nguyen, B. X.; Duong, C.; Phan, T.; Tjiputra, E.; and Tran, Q. D. 2022. Deep federated learning for autonomous driving. In *2022 IEEE Intelligent Vehicles Symposium (IV)*, 1824–1830. IEEE.
- Ogier du Terrail, J.; Ayed, S.-S.; Cyffers, E.; Grimberg, F.; He, C.; Loeb, R.; Mangold, P.; Marchand, T.; Marfoq, O.; Mushtaq, E.; et al. 2022. Flamby: Datasets and benchmarks for cross-silo federated learning in realistic healthcare settings. *Advances in Neural Information Processing Systems*, 35: 5315–5334.
- Paszke, A.; et al. 2017. Automatic differentiation in PyTorch. In *Advances in Neural Information Processing Systems*.
- Paulik, M.; Seigel, M.; Mason, H.; Telaar, D.; Kluivers, J.; van Dalen, R.; Lau, C. W.; Carlson, L.; Granqvist, F.; Van-develde, C.; et al. 2021. Federated evaluation and tuning for on-device personalization: System design & applications. *arXiv preprint arXiv:2102.08503*.
- Peng, X.; et al. 2019. Moment matching for multi-source domain adaptation. In *IEEE International Conference on Computer Vision*, 1406–1415. IEEE.
- Posner, J.; Tseng, L.; Aloqaily, M.; and Jararweh, Y. 2021. Federated learning in vehicular networks: Opportunities and solutions. *IEEE Network*, 35(2): 152–159.
- Saenko, K.; et al. 2010. Adapting visual category models to new domains. In *ECCV*, 213–226. Springer.
- Santurkar, S.; Tsipras, D.; Ilyas, A.; and Madry, A. 2018. How does batch normalization help optimization? *Advances in neural information processing systems*, 31.
- Shen, Z.; et al. 2022. Multi-domain federated learning with communication-efficient aggregation. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 12612–12621. IEEE.
- Stein, C.; et al. 1956. Inadmissibility of the usual estimator for the mean of a multivariate normal distribution. In *Proceedings of the Third Berkeley symposium on mathematical statistics and probability*, volume 1, 197–206.
- Sun, T.; et al. 2021. Partialfed: Personalizing federated learning with client-specific initialization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.
- Tan, M.; and Le, Q. V. 2019. EfficientNet: Rethinking model scaling for convolutional neural networks. In *Proceedings of the 36th International Conference on Machine Learning*, 6105–6114. PMLR.
- Tan, Y.; Chen, C.; Zhuang, W.; Dong, X.; Lyu, L.; and Long, G. 2024. Is heterogeneity notorious? taming heterogeneity to handle test-time shift in federated learning. *Advances in Neural Information Processing Systems*, 36.
- Tschandl, P.; Rosendahl, C.; and Kittler, H. 2018. The HAM10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions. *Scientific data*, 5(1): 1–9.
- Wang, H.; Yurochkin, M.; Sun, Y.; Papailiopoulos, D.; and Khazaeni, Y. 2020. Federated learning with matched averaging. *arXiv preprint arXiv:2002.06440*.
- Wang, J.; et al. 2023. Fedtan: Federated normalization with adaptive tuning and aggregation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.
- Wang, Y.; Shi, Q.; and Chang, T.-H. 2023. Why batch normalization damage federated learning on non-iid data? *IEEE Transactions on Neural Networks and Learning Systems*.
- Wikipedia. 2023. James–Stein Estimator. https://en.wikipedia.org/wiki/JamesStein_estimator. Accessed: 2024-08-15.
- Wu, Y.; and He, K. 2018. Group normalization. In *ECCV*, 3–19.
- Yu, F.; Chen, H.; Wang, X.; Xian, W.; Chen, Y.; Liu, F.; Madhavan, V.; and Darrell, T. 2020. Bdd100k: A diverse driving dataset for heterogeneous multitask learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2636–2645.
- Zhang, H.; Bosch, J.; and Olsson, H. H. 2021. End-to-end federated learning for autonomous driving vehicles. In *2021 International Joint Conference on Neural Networks (IJCNN)*, 1–8. IEEE.
- Zhang, J.; et al. 2023. Client heterogeneity-aware federated learning: Principles and practice. *Journal of Computer Science and Technology*.
- Zhao, Y.; Li, M.; Lai, L.; Suda, N.; Civin, D.; and Chandra, V. 2018. Federated learning with non-iid data. *arXiv preprint arXiv:1806.00582*.
- Zhong, Z.; et al. 2023. Fixbn: Batch normalization requires neither batch nor normalization. *Advances in Neural Information Processing Systems*.
- Zhuang, W.; Gan, X.; Wen, Y.; and Zhang, S. 2023. Optimizing performance of federated person re-identification: Benchmarking and analysis. *ACM Transactions on Multimedia Computing, Communications and Applications*, 19(1s): 1–18.
- Zhuang, W.; Wen, Y.; Zhang, X.; Gan, X.; Yin, D.; Zhou, D.; Zhang, S.; and Yi, S. 2020. Performance optimization of federated person re-identification via benchmark analysis. In *Proceedings of the 28th ACM International Conference on Multimedia*, 955–963.