FEDBN: EVALUATING ROBUSTNESS UNDER MIXED-DOMAIN AND NOISY CLIENTS

Arpnik Singh
Concordia University
Student ID: 40305871
arpnik.singh@live.concordia.ca

Mayank Parmar
Concordia University
Student ID: 40269385
m_p28762@live.concordia.ca

ABSTRACT

Federated Batch Normalization (FedBN) addresses feature-shift non-IID data in federated learning by maintaining Batch Normalization statistics locally at each client. While prior work shows that FedBN outperforms standard aggregation methods such as FedAvg and FedProx under clean, domain-shifted settings, its behavior under more realistic forms of client heterogeneity remains poorly understood. In this work, we reproduce the original FedBN results on the Office-Caltech benchmark and systematically extend them to challenging scenarios involving asymmetric client-side noise and mixed-domain clients that violate the one-domain-per-client assumption. We evaluate FedBN against FedAvg and FedProx under Gaussian input noise, label noise, and unseen-domain generalization on Office-Home, and further conduct normalization ablations comparing BatchNorm, GroupNorm, and LayerNorm. Our results show that FedBN remains robust under noisy single-domain clients but loses its advantage when multiple domains are mixed within clients, revealing a critical dependency on domain-aligned normalization statistics. Moreover, we demonstrate that FedBN's gains are intrinsically tied to Batch Normalization and do not generalize to alternative normalization schemes. These findings clarify the practical limits of FedBN and highlight the need for domain-aware or noise-robust normalization strategies in real-world federated learning. The full implementation and experimental code are publicly available at https://github.com/Arpnik/FedBN.

1 Introduction

Federated Learning (FL) enables multiple clients to collaboratively train machine learning models without sharing raw data, thereby preserving data privacy and complying with regulatory constraints. In a typical FL setting, a central server coordinates training across a set of clients, each holding its own local dataset. The objective is to learn a global model by minimizing the weighted empirical risk across clients:

$$\min_{w} \sum_{k=1}^{K} \frac{n_k}{n} \mathcal{L}_k(w) \tag{1}$$

where K denotes the number of clients, n_k is the number of samples on client k, $n = \sum_k n_k$, and $\mathcal{L}_k(w)$ represents the local loss function. While this formulation assumes a unified objective, in practice the client data distributions are often non-independent and identically distributed (non-IID), leading to optimization instability and degraded generalization.

The Federated Averaging (FedAvg McMahan et al. [2017]) algorithm addresses this problem by performing multiple local stochastic gradient descent (SGD) steps on each client and aggregating the resulting model parameters via weighted averaging at the server. Despite its simplicity and communication efficiency, FedAvg is sensitive to data heterogeneity, as local updates may drift in different directions when client distributions vary significantly. This issue becomes particularly pronounced when clients differ in label distributions, data quantity, or feature characteristics.

To improve robustness under heterogeneous settings, FedProx extends FedAvg by introducing a proximal regularization term that constrains local updates to remain close to the current global model. Specifically, each client minimizes the objective:

$$\mathcal{L}_k(w) + \frac{\mu}{2} \left\| w - w^{(t)} \right\|^2 \tag{2}$$

where $w^{(t)}$ denotes the global model at communication round t, and μ controls the strength of the proximal constraint. This formulation reduces divergence among client updates and improves convergence stability. However, FedProx primarily addresses optimization heterogeneity and does not explicitly account for feature distribution shifts across clients.

Recent studies have emphasized that non-IID data in federated learning arises not only from label skew but also from feature shift, particularly in multi-domain and cross-environment scenarios. In such cases, the input distributions observed by clients differ substantially, even when labels are shared. To address this challenge, Federated Batch Normalization (FedBN) proposes a structural modification to model aggregation by keeping batch normalization (BN) parameters local to each client while aggregating only the remaining model parameters. Given BN parameters (μ_k, σ_k) that capture client-specific feature statistics, FedBN avoids averaging these quantities across clients, thereby preserving domain-specific normalization while sharing feature extractors and classifiers.

Empirical results have shown that FedBN Li et al. [2021] significantly outperforms FedAvg and FedProx under feature-shifted but clean client distributions, particularly on multi-domain benchmarks such as Office-Caltech. However, existing evaluations rely on assumptions that limit real-world applicability. In particular, prior work typically assumes that each client corresponds to a single domain and that local datasets are free from noise. In realistic federated environments, clients may observe data from multiple domains due to changing contexts, and data quality may vary due to sensor noise, annotation errors, or imperfect data collection processes.

In this work, we reproduce the original FedBN experiments and extend them by evaluating robustness under mixed-domain and noisy client settings. We construct heterogeneous clients containing data from multiple domains and introduce asymmetric noise patterns, including Gaussian input noise and label noise at varying levels. We evaluate FedAvg, FedProx Li et al. [2020], and FedBN across clean domains, corrupted test sets, and unseen domains from the Office-Home dataset. Furthermore, we perform ablation studies on client composition, noise intensity, and normalization strategies, comparing Batch Normalization and Group Normalization to analyze sensitivity and stability.

Through this systematic empirical study, we aim to provide a critical assessment of FedBN beyond its original assumptions, offering insights into the conditions under which its advantages persist or degrade. Our findings contribute to a deeper understanding of robustness in federated learning and inform the design of more resilient algorithms under realistic data heterogeneity.

2 Literature Review

2.1 Federated Learning under Data Heterogeneity

Federated Learning (FL) enables multiple clients to collaboratively train a shared model while keeping their local data decentralized and private. Most federated optimization methods are formulated as a weighted empirical risk minimization objective McMahan et al. [2017]:

$$\min_w \sum_{k=1}^{K} \frac{n_k}{n} \mathcal{L}_k(w), \tag{3}$$

where K denotes the number of clients, n_k is the number of samples on client k, $n = \sum_{k=1}^{K} n_k$, and $\mathcal{L}_k(w)$ represents the local loss function. This formulation implicitly assumes that client data distributions are similar. However, in real-world federated systems, client datasets are often non-independent and identically distributed (non-IID), which can lead to optimization instability and degraded generalization.

The Federated Averaging (FedAvg) algorithm addresses this objective by allowing each client to perform multiple local stochastic gradient descent (SGD) updates and aggregating the resulting models via weighted averaging at the server McMahan et al. [2017]:

$$w^{(t+1)} = \sum_{k=1}^{K} \frac{n_k}{n} w_k^{(t+1)}. \tag{4}$$

Despite its simplicity and communication efficiency, FedAvg is sensitive to data heterogeneity, as local updates may drift toward different optima when client distributions differ substantially in terms of label proportions, data quantity, or feature characteristics.

2.2 Optimization Heterogeneity and FedProx

To mitigate the divergence caused by heterogeneous client updates, FedProx extends FedAvg by introducing a proximal regularization term that constrains local optimization Li et al. [2020]. Specifically, each client minimizes the following objective:

$$\mathcal{L}_k(w) + \frac{\mu}{2} \left\| w - w^{(t)} \right\|^2, \tag{5}$$

where $w^{(t)}$ denotes the global model at communication round t, and μ controls the strength of the proximal constraint. This formulation reduces divergence among client updates and improves convergence stability under heterogeneous data and system conditions. However, FedProx primarily addresses optimization heterogeneity and does not explicitly account for feature distribution shifts across clients.

2.3 Feature Shift and Multi-Domain Federated Learning

Beyond label skew and data imbalance, feature distribution shift has been identified as a major source of non-IID data in federated learning, particularly in multi-domain settings. Feature shift commonly arises when clients observe data collected under different environments, sensors, or acquisition conditions. This form of heterogeneity can be formalized as:

$$P_k(X) \neq P_{k'}(X), \quad \text{while} \quad P_k(Y \mid X) \approx P_{k'}(Y \mid X), \tag{6}$$

indicating that input distributions differ across clients even when label semantics are shared. In such scenarios, standard federated optimization methods often fail to learn representations that generalize across domains.

Federated Batch Normalization (FedBN) was proposed to explicitly address feature-shift non-IID data by modifying the aggregation strategy for batch normalization layers Li et al. [2021]. Instead of averaging batch normalization parameters across clients, FedBN keeps these parameters local while aggregating the remaining network weights. This separation of parameters can be expressed as:

$$w = \{w_{\text{shared}},\ \mu_k, \sigma_k\}, \tag{7}$$

where w_{shared} denotes globally shared parameters and (μ_k, σ_k) represent client-specific batch normalization statistics. By preserving domain-specific normalization behavior, FedBN has been shown to outperform FedAvg and FedProx on multi-domain benchmarks such as Office-Caltech under clean client distributions Li et al. [2021].

2.4 Mixed-Domain Clients and Realistic Federated Settings

While prior evaluations of FedBN assume that each client corresponds to a single domain, real-world federated systems often involve clients collecting data from multiple domains due to evolving environments or usage patterns. In such cases, the input distribution on client k can be modeled as a mixture of domain distributions:

$$P_k(X) = \sum_{d=1}^{D} \pi_{k,d}\, P_d(X), \tag{8}$$

where $\pi_{k,d}$ denotes the proportion of data from domain d on client k. This mixed-domain setting violates the assumptions underlying earlier FedBN analyses and raises important questions regarding the robustness of local normalization strategies when domain boundaries are no longer clearly defined.

2.5 Robust Federated Learning with Noisy Clients

Another important challenge in federated learning is robustness to noisy or unreliable client data. Noise may arise from label corruption, sensor errors, or imperfect data collection processes and may vary across clients. Label noise on client k can be modeled as:

$$\tilde{Y} = Y \oplus \epsilon_k, \tag{9}$$

where ϵ_k represents a client-specific noise process. Such noise can significantly degrade global model performance, particularly when noisy and clean clients participate simultaneously in the training process.

2.6 Positioning of This Work

In contrast to prior studies, this work jointly examines feature-shift heterogeneity, mixed-domain client distributions, and asymmetric noise within a unified experimental framework. By reproducing the original FedBN results and extending them to more realistic federated scenarios, we provide a critical evaluation of FedBN's robustness beyond its original assumptions. Our study complements existing work on optimization heterogeneity, normalization-based federated learning, and robustness to noisy clients, offering new insights into the practical applicability and limitations of FedBN in real-world federated environments.

3 Methodology

3.1 Objective

The goal of our experiments is twofold. First, we reproduce the core results of FedBN on standard multi-domain federated benchmarks to validate the changes in the implementation. Second, we critically evaluate the robustness of FedBN under more challenging and realistic settings involving mixed-domain clients and asymmetric noise, which were not explored in the original work. We compare FedBN against FedAvg and FedProx across a range of heterogeneity and noise conditions, and analyze whether FedBN's reported advantages persist or degrade under these extensions.

3.2 Datasets

3.2.1 Office-Caltech

We use the Office-Caltech benchmark, consisting of four visual domains: Amazon, Caltech, DSLR, and Webcam, sharing 10 object categories. Following prior work, each domain is treated as a distinct data source. Images are resized to (256×256) and standard data augmentations (random horizontal flip and rotation) are applied during training.

3.2.2 Office-Home (Unseen Evaluation)

To evaluate cross-dataset generalization, we additionally test trained models on the Office-Home dataset, which contains four unseen domains: Art, Clipart, Product, and Real World. Office-Home is never used for training or validation and serves purely as an out-of-distribution generalization benchmark.

3.3 Federated Learning Setup

We consider a standard synchronous federated learning setup with (K) clients. Each client maintains a local model and performs local SGD updates before global aggregation. We fix the number of local epochs to one across all experiments in order to isolate the effects of data heterogeneity, noise, and normalization from client drift induced by extended local training. This choice ensures fair and stable comparisons between aggregation strategies and does not favor any particular method.

3.3.1 3 Aggregation Strategies

- **FedAvg**: Standard parameter averaging
- **FedProx**: FedAvg with proximal regularization ($\mu = 10^{-3}$)
- **FedBN**: Aggregates all parameters except BatchNorm statistics, which remain client-specific

Table 1: Experimental Setup and Training Configuration

Component	Specification
Model Architecture	AlexNet (as in the original FedBN paper)
Normalization	Batch Normalization (default)
Local Epochs	1 per communication round
Optimizer	Stochastic Gradient Descent (SGD)
Batch Size	32
Communication Rounds	300
Evaluation	Classification accuracy on each domain
	Average Accuracy across domains
	Convergence behavior over communication rounds
	Confusion matrices for detailed error analysis

3.4 Evaluation Strategy

We evaluate all federated learning methods under three complementary settings designed to assess in-domain performance, cross-domain generalization, and robustness to data corruption.

3.4.1 In-Domain Evaluation on Office-Caltech

After federated training, we evaluate each model on the clean test split of every Office-Caltech domain (Amazon, Caltech, DSLR, and Webcam) independently. For FedAvg and FedProx, the aggregated server model is evaluated on all domains. For FedBN, each client model is evaluated on its corresponding domain, reflecting the method's use of client-specific normalization statistics. We report classification accuracy and loss for each domain, as well as their average across domains.

3.4.2 Cross-Domain Generalization on Office-Home

To assess generalization to unseen domains, we further evaluate the trained models on the Office-Home dataset, which is never observed during training. Office-Home contains four visually distinct domains (Art, Clipart, Product, and Real World) and serves as a clean out-of-distribution benchmark. For FedAvg and FedProx, the server model is evaluated on all Office-Home domains. For FedBN, each client model is evaluated separately, and we additionally report the best-performing client per domain. This protocol allows us to analyze how client-specific normalization impacts cross-domain transfer.

3.4.3 Metrics and Reporting

Across all evaluation settings, we use the standard cross-entropy loss for multi-class classification and report classification accuracy as the primary performance metric. Both dataset-specific metrics (reported separately for each domain) and their averages across domains are logged throughout training and evaluation using Weights & Biases (W&B) for experiment tracking and comparison. Confusion matrices are computed and logged only at the final evaluation stage to enable fine-grained error analysis without affecting training dynamics or incurring additional computational overhead.

3.4.4 Reproducibility

All experiments fix random seeds for data sampling and noise injection to ensure reproducibility. Further implementation details, including dataset preparation, noise injection, aggregation rules, and evaluation routines, are provided in the released codebase.

4 Experiments

4.1 Baseline Reproduction

4.1.1 Objective

Verify that our implementation matches the behavior reported in the original FedBN paper and establishe a clean reference point, confirming that FedBN outperforms FedAvg and FedProx under standard domain-shift conditions.

Table 2: Mixed-Domain Client Configurations

Component	Dataset Distribution	Noise Specification
Mixed Domains	Client 0: Amazon + DSLR Client 1: Caltech + Webcam	No Noise
Mixed Domains	Client 0: Amazon + Caltech Client 1: DSLR + Webcam	No Noise
Mixed Domains (Asymmetric Noise)	Client 0: Amazon + Caltech Client 1: DSLR + Webcam	Gaussian noise (ratio = 0.3, $\sigma = 0.5$) Gaussian noise (ratio = 0.2, $\sigma = 0.5$) Label noise (30%)

4.1.2 Setup

- 4 clients (each corresponding to a single domain: Amazon, Caltech, DSLR, Webcam)
- No noise injected

4.2 Single-Domain Clients with Asymmetric Noise

4.2.1 Objective

Evaluate the robustness of FedAvg, FedProx, and FedBN under heterogeneous and asymmetric client-side noise, while preserving the standard assumption of one domain per client. This experiment probes whether FedBN's advantage from client-specific normalization persists when clients experience different types and levels of data corruption.

- **Input noise**: Additive Gaussian noise ($\sigma = 0.5$) applied to a fraction of samples
- **Label noise**: Random label flipping with a fixed probability

4.2.2 Setup

- 4 clients, each corresponding to a single domain: Amazon, Caltech, DSLR, and Webcam.
- Asymmetric noise is injected at the client level using a hardcoded configuration (see Table 3)
- Noise is applied only to training and validation data, all clean test sets remain uncorrupted unless explicitly stated.
- All methods use identical model architectures, optimization settings, and communication schedules to ensure fair comparison.

4.3 Mixed-Domain Clients

4.3.1 Objective

Evaluate FedBN under higher heterogeneity by allowing clients to contain multiple domains, violating the assumption of one domain per client (see Table 2).

In standard federated settings, data heterogeneity arises primarily across clients, with each client representing a distinct domain. In contrast, mixed-domain clients introduce heterogeneity within individual clients, causing feature distributions from multiple domains to be jointly normalized and thereby blurring domain-specific statistics.

4.4 Normalization Layer Ablation

4.4.1 Objective

Determine whether FedBN's performance gains are intrinsically tied to Batch Normalization or extend to alternative normalization schemes. By replacing Batch Normalization with Layer Normalization and Group Normalization, this experiment isolates the role of normalization statistics in federated aggregation under heterogeneous and noisy client data.

We adopt a single-domain-per-client configuration (see Table 3) for the normalization ablation to avoid conflating normalization effects with violations of FedBN's domain assumptions. While mixed-domain experiments explicitly

Table 3: Asymmetric Noise Distribution Across Clients

Component	Dataset	Noise
Client 0	Amazon	Gaussian noise applied to 30% of samples
Client 1	Caltech	Label noise applied to 30% of samples
Client 2	DSLR	Gaussian noise applied to 30% of samples Label noise applied to 40% of samples
Client 3	Webcam	No noise applied

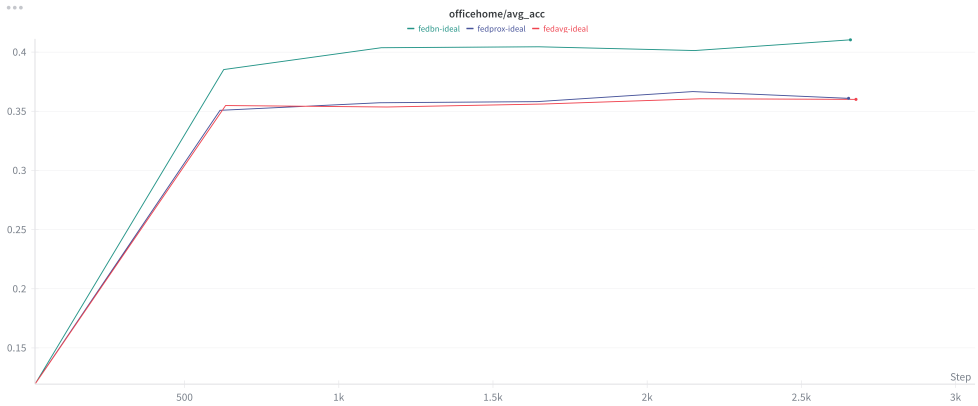


Figure 1: Average Accuracy on unseen Office-Home dataset over communication rounds for FedAvg, FedProx, and FedBN

examine scenarios where FedBN may fail due to blurred domain boundaries within clients, this ablation focuses on whether FedBN's advantage is inherently linked to Batch Normalization itself. By holding the domain structure constant and varying only the normalization layer, we can directly attribute performance differences to the normalization mechanism rather than to changes in data heterogeneity.

4.4.2 Setup

We evaluate three normalization schemes:

- **Batch Normalization (BN)**: BatchNorm maintains running mean and variance statistics with momentum of 0.1 and epsilon of 10^{-5} (PyTorch defaults). It is treated as client-specific under FedBN, while LayerNorm and GroupNorm compute normalization statistics independently of the batch dimension.
- **Layer Normalization (LN)**: In our implementation, LayerNorm is realized via GroupNorm with a single group (`num_groups=1`) for convolutional layers and standard LayerNorm (epsilon of 10^{-5}) for fully connected layers. This approach provides channel-wise normalization compatible with 4D convolutional tensors.
- **Group Normalization (GN)**: Uses 32 channel groups (`num_groups=32`) with epsilon of 10^{-5}, dividing feature channels into fixed groups for normalization.

For each normalization variant, we train models using FedAvg, FedProx, and FedBN under identical training, noise, and optimization settings. This results in a controlled comparison that allows us to assess whether FedBN's performance gains are specific to Batch Normalization or extend to normalization methods that do not rely on batch-level statistics.

5 Result

5.1 Baseline Reproduction

FedBN consistently outperforms FedAvg and FedProx across all domains (both seen and unseen), confirming that our implementation reproduces the results reported in the original FedBN paper. (see Table 4 and Figure 1)

Table 4: Single-Domain Clients with Asymmetric Noise. Test accuracy is reported in the first row, with cross-entropy loss reported in the row below. For Office-Home (unseen domains), FedBN results correspond to the best-performing client per domain and for FedAvg and FedProx results correspond to server model

	Office-Caltech (Seen Domains)				Office-Home (Unseen Domains)				
Method	**Amazon**	**Caltech**	**DSLR**	**Webcam**	**Art**	**Clipart**	**Product**	**Real World**	**Avg.**
FedBN	0.641 1.45	0.467 2.23	0.813 0.32	0.848 0.32	0.349 3.29	0.291 4.56	0.525 1.94	0.473 2.21	0.410 3.00
FedAvg	0.552 1.75	0.467 2.27	0.656 1.38	0.848 0.39	0.300 3.15	0.226 5.10	0.455 2.38	0.470 1.96	0.363 3.15
FedProx	0.547 1.79	0.471 2.28	0.688 1.28	0.864 0.38	0.309 3.16	0.223 5.00	0.457 2.39	0.478 1.94	0.367 3.12

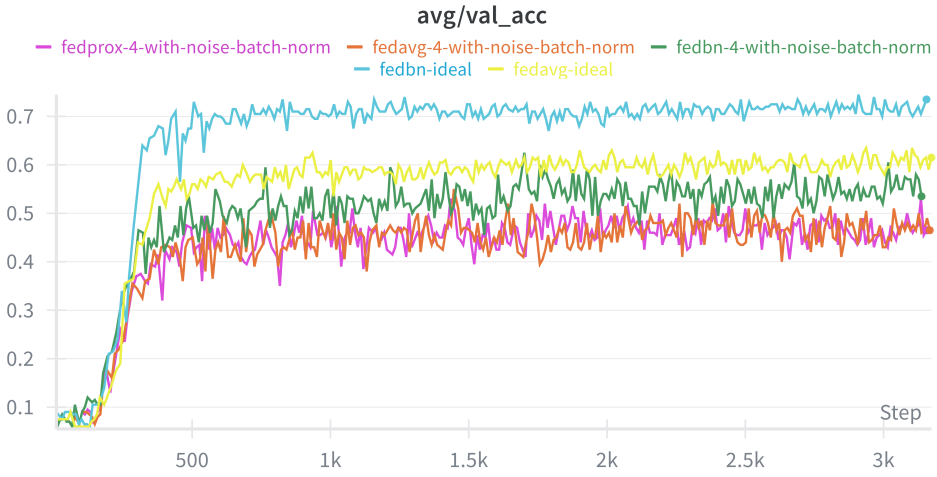


Figure 2: Average validation accuracy on Office-Caltech (seen domains) over communication rounds under asymmetric client-side noise.

5.2 Single-Domain Clients with Asymmetric Noise

As observed in Table 5, under asymmetric client-side noise, FedBN achieves the highest accuracy on three of four seen Office-Caltech domains, confirming that client-specific normalization remains effective when each client corresponds to a single domain. In particular, FedBN significantly outperforms FedAvg and FedProx on Amazon, Caltech, and DSLR, demonstrating robustness to feature-level corruption (Figure 2).

On unseen Office-Home domains, performance differences are more nuanced. While FedAvg attains the highest average accuracy, FedBN achieves the best performance on the Product domain and remains competitive on Real World. This suggests that although FedBN stabilizes feature distributions during training, it does not explicitly mitigate label noise, which can dominate generalization performance on unseen domains when averaged across clients.

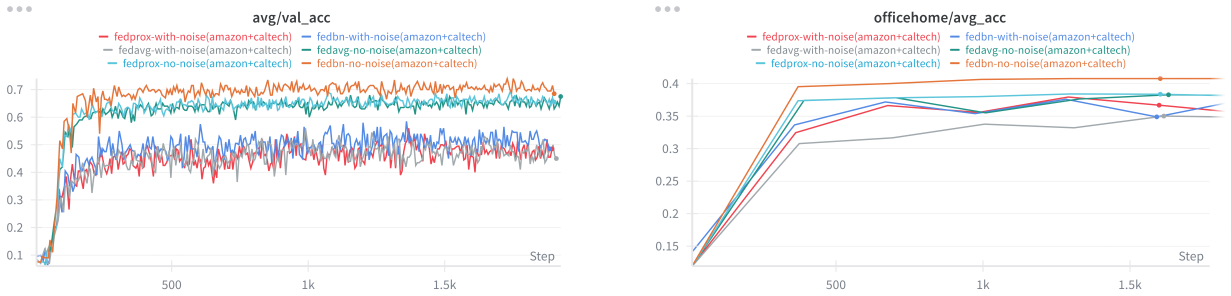
FedBN isolates normalization statistics per client, which stabilizes feature distributions under noisy training data. However, under asymmetric noise, particularly label corruption, the client model with cleaner supervision may generalize better to unseen domains when evaluated via a global server model, as in FedAvg.

5.3 Mixed-Domain Clients

This experiment (Table 6) demonstrates that FedBN's effectiveness is strongly tied to the one-domain-per-client assumption. When clients contain samples from multiple domains, Batch Normalization statistics become entangled across heterogeneous feature distributions, weakening FedBN's key advantage.We observe that:

Table 5: Asymmetric-noise results. Accuracy (top row) and cross-entropy loss (bottom row). For Office-Home, FedBN reports best client performance; FedAvg and FedProx use the server model.

	Office-Caltech (Seen Domains)				Office-Home (Unseen Domains)				
Method	**Amazon**	**Caltech**	**DSLR**	**Webcam**	**Art**	**Clipart**	**Product**	**Real World**	**Avg.**
FedBN	0.5990 1.2902	0.4178 1.9333	0.8125 0.8627	0.8475 0.4630	0.2446 2.2710	0.2526 2.9103	0.5008 1.5890	0.3870 1.8571	0.3357 2.0819
FedProx	0.5573 1.4593	0.4044 1.9399	0.5938 1.3473	0.8644 0.6732	0.2599 2.2845	0.2095 2.9208	0.4110 1.8119	0.3704 1.9363	0.3127 2.2384
FedAvg	0.4896 1.4936	0.4000 2.0344	0.6250 1.3654	0.8814 0.6908	0.3180 2.2136	0.2675 2.5322	0.4307 1.6865	0.4435 1.7491	0.3649 2.0454



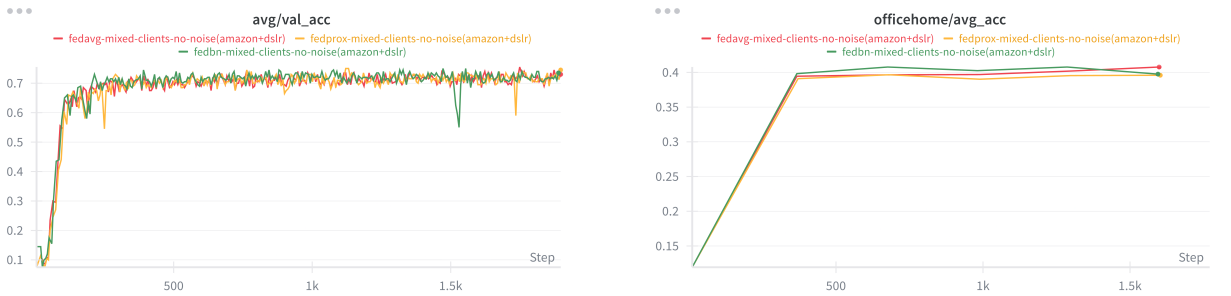
(a) Average validation accuracy on seen domains

(b) Average test accuracy on unseen domains

Figure 3: Mixed-domain client setting with Amazon+Caltech and DSLR+Webcam combined within individual clients.

- Under mixed-domain clients without noise, FedBN no longer consistently outperforms FedAvg or FedProx on either seen or unseen domains, and performance differences across methods become marginal.
- With additional asymmetric noise, all methods degrade, but FedBN loses its relative robustness, indicating that client-specific BatchNorm cannot disentangle domain shift from intra-client heterogeneity.
- Convergence curves (Figure 3 and Figure 4) show increased instability and variance for FedBN in mixed-domain settings, supporting the hypothesis that blurred normalization statistics undermine its optimization behavior.

Overall, these results confirm that FedBN is not inherently robust to within-client heterogeneity.While effective under clean, single-domain clients, its benefits diminish when domain boundaries are violated, suggesting that alternative normalization strategies or domain-aware client partitioning are required for realistic federated deployments with mixed data sources.



(a) Average validation accuracy on seen domains

(b) Average test accuracy on unseen domains

Figure 4: Mixed-domain client setting with Amazon+DSLR and Caltech+Webcam combined within individual clients.

Table 6: Performance comparison under asymmetric noise with mixed-domain clients. Test accuracy is reported in the first row and cross-entropy loss in the second row. Results highlight the effect of violating the one-domain-per-client assumption on Office-Caltech (seen) and Office-Home (unseen) datasets.

	Office-Caltech (Seen Domains)				Office-Home (Unseen Domains)				
Method	**Amazon+Caltech**	**Amazon+DSLR**	**DSLR+Webcam**	**Caltech+Webcam**	**Art**	**Clipart**	**Product**	**Real World**	**Avg.**
FedBN (mixed, no-noise, Amazon+Caltech)	0.5731	–	0.8791	–	0.3333	0.2972	0.5327	0.4801	0.4108
	1.8315	–	0.3482	–	3.4153	4.6476	2.0224	2.1959	3.0703
FedBN (mixed, no-noise, Amazon+DSLR)	–	0.6875	–	0.5599	0.2875	0.3001	0.5236	0.5050	0.4040
	–	1.2340	–	1.7072	3.2329	4.4808	1.9975	1.9295	2.9102
FedBN (mixed, with-noise, Amazon+Caltech)	0.4844	–	0.8242	–	0.2936	0.3031	0.4703	0.4369	0.3760
	1.7330	–	0.8386	–	2.5866	2.9665	1.8065	1.9448	2.3261
FedProx (mixed, no-noise, Amazon+DSLR)	–	0.7009	–	0.5810	0.2844	0.2927	0.5053	0.4967	0.3948
	–	1.2236	–	1.7085	3.3634	4.5940	2.0803	2.0027	3.0101
FedProx (mixed, no-noise, Amazon+Caltech)	0.5420	–	0.8462	–	0.2599	0.2630	0.5114	0.4884	0.3807
	1.8625	–	0.6454	–	3.4474	4.9660	2.0677	2.0269	3.1270
FedProx (mixed, with-noise, Amazon+Caltech)	0.4988	–	0.7473	–	0.2844	0.2645	0.4536	0.4136	0.3540
	1.6738	–	1.0202	–	2.3764	2.6224	1.7740	1.8675	2.1601
FedAvg (mixed, no-noise, Amazon+Caltech)	0.5372	–	0.8462	–	0.2752	0.2630	0.4992	0.4884	0.3815
	1.9080	–	0.6136	–	3.4101	4.9644	2.1521	2.0264	3.1382
FedAvg (mixed, no-noise, Amazon+DSLR)	–	0.6875	–	0.5845	0.2844	0.3076	0.5175	0.5050	0.4036
	–	1.2331	–	1.7212	3.3204	4.6592	2.0628	1.9928	3.0088
FedAvg (mixed, with-noise, Amazon+Caltech)	0.4892	–	0.7143	–	0.2844	0.2719	0.4292	0.4070	0.3481
	1.6983	–	0.9642	–	2.3407	2.4967	1.7346	1.8073	2.0948

5.4 Normalization Layer Ablation

Table 7: Normalization Layer Ablation under Asymmetric Noise: Comparison of BatchNorm, GroupNorm, and LayerNorm across FedAvg, FedProx, and FedBN.

	Office-Caltech (Seen Domains)				Office-Home (Unseen Domains)				
Method	**Amazon**	**Caltech**	**DSLR**	**Webcam**	**Art**	**Clipart**	**Product**	**Real World**	**Avg.**
FedProx (BatchNorm)	0.5573	0.4044	0.5938	0.8644	0.2599	0.2095	0.4110	0.3704	0.3127
	1.4593	1.9399	1.3473	0.6732	2.2845	2.9208	1.8119	1.9363	2.2384
FedAvg (BatchNorm)	0.4896	0.4000	0.6250	0.8814	0.3180	0.2675	0.4307	0.4435	0.3649
	1.4936	2.0344	1.3654	0.6908	2.2136	2.5322	1.6865	1.7491	2.0454
FedBN (BatchNorm)	0.5990	0.4178	0.8125	0.8475	0.2446	0.2526	0.5008	0.3870	0.3357
	1.2902	1.9333	0.8627	0.4630	2.2710	2.9103	1.5890	1.8571	2.0819
FedAvg (GroupNorm)	0.6198	0.3867	0.7812	0.7966	0.2171	0.2719	0.4277	0.3522	0.3172
	1.2528	1.8541	0.9261	0.6875	2.4430	2.2150	1.8597	2.0236	2.1353
FedProx (GroupNorm)	0.5938	0.3956	0.6562	0.7797	0.2232	0.3120	0.4597	0.3472	0.3355
	1.2157	1.8155	0.9293	0.6900	2.3575	2.2942	1.8178	2.1257	2.1488
FedBN (GroupNorm)	0.5677	0.4400	0.8125	0.7288	0.2538	0.3551	0.4871	0.4219	0.3787
	1.3502	1.6996	0.8133	0.6949	2.4160	2.1043	1.6496	1.7994	2.0113
FedAvg (LayerNorm)	0.5417	0.3333	0.6250	0.6780	0.2385	0.3016	0.4125	0.3256	0.3196
	1.4399	1.9082	1.1434	0.9820	2.3804	2.1366	1.9297	2.1434	2.1475
FedProx (LayerNorm)	0.5677	0.3600	0.5938	0.6271	0.2416	0.2927	0.4110	0.3439	0.3223
	1.4268	1.8561	1.1676	1.0790	2.4391	2.2797	1.9289	2.1457	2.1984
FedBN (LayerNorm)	0.5417	0.3600	0.6562	0.6610	0.2080	0.2823	0.3714	0.3339	0.2970
	1.4983	1.9119	1.1387	1.0946	2.4658	2.2052	2.0484	2.1307	2.2106

Table 7 reports the impact of replacing Batch Normalization with Group and Layer Normalization across FedAvg, FedProx, and FedBN under asymmetric noise. Several consistent trends emerge.

First, FedBN's advantage is strongly tied to Batch Normalization. When BatchNorm is used, FedBN achieves the best or near-best performance on most seen domains and remains competitive on unseen Office-Home domains. This confirms that FedBN benefits specifically from maintaining client-specific batch statistics, which helps mitigate domain shift and noise-induced distribution mismatch. When BatchNorm is replaced by LayerNorm or GroupNorm—both of which do not rely on batch-level statistics—FedBN's relative advantage largely disappears.

(a) Average validation accuracy on seen domains

(b) Average test accuracy on unseen domains

Figure 5: Normalization ablations results

Second, GroupNorm improves stability for FedAvg and FedProx compared to BatchNorm in several cases, particularly on seen domains. This is expected, as GroupNorm avoids noisy batch statistics and is more robust under small batch sizes and heterogeneous data. However, despite these gains, GroupNorm-based variants do not consistently outperform FedBN with BatchNorm, especially in cross-domain generalization, indicating that robustness alone does not replace the benefit of client-specific normalization.

Third, LayerNorm consistently underperforms BatchNorm and GroupNorm across all methods. This degradation is most pronounced on unseen domains, suggesting that per-sample normalization removes useful feature-scale information needed for transfer across domains. This effect is amplified in federated settings where feature distributions differ significantly across clients.

Overall (Figure 5), these results demonstrate that FedBN's performance gains are not generic to all normalization schemes, but are intrinsically linked to Batch Normalization and its ability to capture domain-specific feature statistics at the client level. Normalization methods that eliminate batch-dependent statistics reduce inter-client variance but also eliminate the mechanism through which FedBN mitigates domain shift, thereby narrowing the performance gap between aggregation strategies.

6 Limitations and Future Work

- Experiments are limited to Office-Caltech and Office-Home using a single backbone (AlexNet); evaluating larger datasets and modern architectures (e.g., ResNet, ViT) would strengthen generality.
- We fix local training to 1 epoch per round to isolate heterogeneity effects; varying local update lengths may expose additional interactions between client drift and normalization.
- Noise is injected using predefined asymmetric configurations; future work should consider adaptive, adversarial, or temporally evolving noise to better reflect real-world federated settings.

7 Conclusion

In this work, we systematically evaluated the robustness and limitations of FedBN under increasingly realistic federated learning conditions, including asymmetric client-side noise, mixed-domain clients, and normalization layer ablations. Starting from a faithful reproduction of the original FedBN results, we confirmed its effectiveness under standard domain-shift settings with one domain per client. We then showed that FedBN remains robust to heterogeneous and asymmetric noise when the one-domain-per-client assumption holds, particularly on seen domains, by stabilizing feature distributions through client-specific Batch Normalization. However, when this assumption is violated by mixing multiple domains within individual clients, FedBN's advantage consistently degrades, revealing its sensitivity to intra-client heterogeneity and entangled normalization statistics. Finally, through controlled normalization ablations, we demonstrated that FedBN's gains are intrinsically tied to Batch Normalization; replacing it with GroupNorm or LayerNorm largely eliminates its benefits. Together, these findings clarify both the strengths and failure modes of FedBN, highlighting that its effectiveness critically depends on domain-aligned client partitioning and batch-dependent normalization, and motivating future work on domain-aware client grouping and alternative normalization strategies for realistic federated deployments.

References

Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 1273–1282, 2017.

Xiaoxiao Li, Meirui Jiang, Xiaofei Zhang, Michael Kamp, and Qi Dou. Learning on non-iid features via local batch normalization. In *International Conference on Learning Representations (ICLR)*, 2021. arXiv:2102.07623.

Tian Li, Anit Kumar Sahu, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. In *Proceedings of Machine Learning and Systems (MLSys)*, 2020.