# FedBalanceTTA – Federated Learning with Balanced Test Time Adaptation

Md Akil Raihan Iftee[1][0009−0001−1459−6365], Rakibul Hasan Rajib[1][0009−0007−3705−9400], Mir Sazzat Hossain[1][0000−0001−6999−6879], A K M Mahbubur Rahman[1][0000−0001−9941−4817], Md. Ashraful Amin[1][0000−0003−2330−9775], Amin Ahsan Ali[1][0000−0002−0129−8705], and Sajib Mistry[2][0000−0001−7513−3789]

[1] Center for Computational & Data Sciences
Independent University, Bangladesh
[2] Curtin University

**Abstract.** Federated learning (FL) enables privacy-preserving model training across decentralized clients with diverse and non-IID data. However, adapting models at test time remains challenging, particularly under class imbalance and domain shifts. Traditional test-time adaptation (TTA) approaches fail in such settings due to biased batch normalization statistics dominated by majority classes. We propose **FedBBN**, a federated TTA framework that incorporates *Balanced Batch Normalization (BBN)* to mitigate prediction bias by treating each class equally during normalization. FedBBN enables client-side unsupervised adaptation and introduces a class-aware aggregation strategy at the server to preserve model robustness across heterogeneous clients. Our proposed method operates without access to labels or raw data at test time, yields personalized models per client, and supports secure aggregation for privacy. Experiments on CIFAR-10-C and CIFAR-100-C benchmarks demonstrate that FedBBN significantly improves robustness and minority-class performance over conventional federated and local adaptation baselines.

**Keywords:** Federated Learning · Test-Time Adaptation · Class Imbalance · Balanced Batch Normalization.

## 1 Introduction

Federated Learning (FL) enables decentralized model training across clients while maintaining data privacy by keeping raw data local. This paradigm is increasingly important in scenarios like mobile computing, healthcare, and IoT, where privacy, personalization, and data distribution shifts are all critical considerations. However, FL systems face several fundamental challenges, particularly at *test time*, where models must generalize to unseen and often unlabeled client data streams with potentially severe *class imbalance* and *domain shifts*.

Standard test-time adaptation (TTA) techniques, which adjust model parameters during inference using unlabeled test data, have shown promise in addressing distribution shifts. A common approach is to update *Batch Normalization*

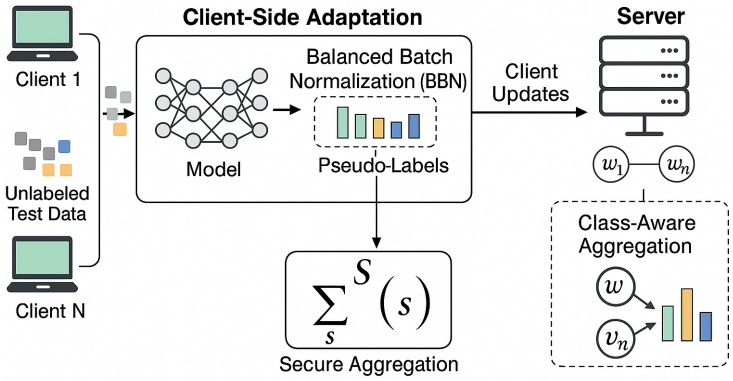


**Fig. 1.** Overview of FedBalanceTTA.

(BN) layers using the statistics of incoming batches. However, this becomes problematic when data is class-imbalanced: the BN statistics become skewed toward majority classes, leading to biased normalization and degraded performance on minority classes. This issue is especially pronounced in FL, where each client may exhibit drastically different data distributions.

To address this, we propose **FedBBN**, a federated TTA framework that integrates *Balanced Batch Normalization* (BBN) at the client level and introduces a novel *class-aware aggregation* strategy at the server. BBN computes normalization statistics on a per-class basis and averages them equally across classes, thereby correcting class imbalance without access to ground-truth labels. Clients adapt their models in an unsupervised manner using pseudo-labels, while the server aggregates updates in a way that accounts for inter-client class skew, ensuring global model robustness and fairness. Our framework operates entirely under standard FL privacy constraints and produces *personalized models* that adapt to each client's local distribution without compromising data security.

In summary, the key contributions of this work are:

- We introduce a federated test-time adaptation framework that incorporates **Balanced Batch Normalization (BBN)** to mitigate prediction bias from class imbalance without requiring labels.
- We develop an **unsupervised client-side adaptation procedure** using BBN and pseudo-label-driven loss functions that personalize models under both domain and label shifts.
- We propose a **class-aware aggregation strategy** that adjusts server-side weighting based on clients' label distribution skew, enhancing robustness and fairness in global model updates.
- We ensure **privacy-preserving personalization** by restricting all sensitive operations (e.g., per-class stats) to the client side and using secure aggregation for federated updates.

## 2 Related Work

### 2.1 Class Imbalance in Federated Learning

Federated Learning (FL) often encounters challenges due to non-IID data distributions and class imbalance across clients. To address these issues, several methods have been proposed: Fed-GraB [5] introduces a self-adjusting gradient balancer that re-weights each client's gradient based on a global long-tail prior, enhancing minority-class accuracy. GBME [7] constructs class-prior proxies from accumulated client gradients, enabling global loss re-balancing without compromising data privacy.

In [9], CReFF identifies biased classifiers as a primary factor for poor performance under heterogeneous long-tailed data. It retrains the classifier on federated features, achieving performance comparable to centrally retrained models. BalanceFL [13] addresses client imbalance through a local update scheme, ensuring each client's model behaves as if trained on a uniform class distribution.

FL-FCR employs calibration weighting and resampling, integrating a calibrated loss during client training and calibration-based resampling at the server, aligning training with true class frequencies [6]. Li et al. [2] propose a probability-corrected loss and shared class prototypes to align model outputs across heterogeneous clients, effectively handling both balanced and imbalanced global data.

### 2.2 Test-Time Adaptation

Test-Time Adaptation (TTA) methods adapt pre-trained models during inference to handle domain shifts or corrupted inputs:

TENT performs entropy minimization on each test batch, updating batch normalization statistics and channel-wise affine parameters online to maximize output confidence [11]. CoTTA maintains a weight-averaged teacher model and applies strong augmentation to stabilize pseudo-labels. It also stochastically resets a fraction of parameters to the source pre-trained values to prevent catastrophic forgetting [12]. RoTTA introduces a Practical TTA setting with temporally changing distributions and correlated sampling. It incorporates robust batch normalization, a category-balanced memory bank, and a time-aware reweighting teacher-student model for adaptation [14]. ROID addresses challenges of universal online TTA by applying diversity weighting, continuously ensembling the source and adapted models, and performing adaptive prior correction on predictions [4].

### 2.3 Bridging TTA with federated or distributed learning

ATP formalizes Test-Time Personalized Federated Learning (TTPFL), where each client locally adapts the global model without labels. It learns adaptive update rates per network module based on cross-client shift information [1]. DynFed uses Adaptive Rate Networks to generate client-specific adaptation rates, refining each client's update rate without sharing raw data and providing convergence guarantees [8].

TTA-FedDG leverages test-time adaptation for federated domain generalization. It mixes features via fast reordering during local training and applies a contrastive teacher-student scheme with selective strong pseudo-labeling at test time [3]. Shao et al. build a federated face anti-spoofing system where clients collaboratively train a general model, and each device minimizes the model's prediction entropy on its new attack data during test time [10].

These methods collectively advance the robustness and adaptability of FL models in the presence of class imbalance and domain shifts.

## 3 Methodology

We propose **FedBBN**, a federated test-time adaptation (TTA) framework that addresses class imbalance and domain shift in decentralized settings. FedBBN operates by enabling each client to adapt its model locally using *Balanced Batch Normalization* (BBN), while a central server aggregates updates in a manner that is robust to inter-client skew.

The core idea is to replace traditional Batch Normalization (BN) layers with a balanced variant that neutralizes the dominance of majority classes in normalization statistics. Given unlabeled test-time data, each client uses pseudo-labels to compute class-wise statistics and perform local adaptation in a fully unsupervised manner.

Let $\mathcal{D}_i = \{x_j\}_{j=1}^{n}$ denote the test data stream for client $i$, and let $f_\theta$ be the model with parameters $\theta$ distributed from the central server. Each client replaces every BN layer in $f_\theta$ with a Balanced BN layer and performs adaptation using the incoming stream.

**Balanced Batch Normalization (BBN):**

At inference time, each client uses the current model to generate pseudo-labels $\hat{y}_j$ for each sample $x_j$. Let $C$ be the number of known classes. For each class $c \in \{1, \ldots, C\}$, the feature vectors belonging to class $c$ are grouped as $\mathcal{X}_c = \{x_j \mid \hat{y}_j = c\}$. The per-class mean and variance are computed as:

$$\mu_c = \frac{1}{|\mathcal{X}_c|} \sum_{x \in \mathcal{X}_c} x, \quad \sigma_c^2 = \frac{1}{|\mathcal{X}_c|} \sum_{x \in \mathcal{X}_c} (x - \mu_c)^2 \tag{1}$$

Rather than computing global statistics based on the entire batch (which would be skewed), BBN computes an unweighted average over all classes:

$$\mu_{BBN} = \frac{1}{C} \sum_{c=1}^{C} \mu_c, \quad \sigma_{BBN}^2 = \frac{1}{C} \sum_{c=1}^{C} \sigma_c^2 \tag{2}$$

The input feature $x$ is then normalized using these balanced statistics as:

$$BN_{balanced}(x) = \gamma \cdot \frac{x - \mu_{BBN}}{\sqrt{\sigma_{BBN}^2 + \epsilon}} + \beta \tag{3}$$

where $\gamma$ and $\beta$ are learnable affine parameters. If a class $c$ has no samples in the batch, the client can either reuse the previous statistics for that class or interpolate using global estimates.

**Local Adaptation:**

Using BBN, each client performs test-time adaptation on its local data stream. The model parameters are updated using unsupervised objectives such as entropy minimization:

$$\mathcal{L}_{ent} = -\frac{1}{n}\sum_{j=1}^{n}\sum_{k=1}^{C} p_\theta(y=k|x_j)\log p_\theta(y=k|x_j) \tag{4}$$

Alternatively, clients can use confident pseudo-labeling:

$$\mathcal{L}_{pl} = -\frac{1}{|\mathcal{H}|}\sum_{j\in\mathcal{H}} \log p_\theta(y=\hat{y}_j|x_j) \tag{5}$$

where $\mathcal{H} = \{j \mid \max_k p_\theta(y=k|x_j) > \tau\}$ is the set of high-confidence samples and $\tau$ is a confidence threshold.

To prevent overfitting to noisy pseudo-labels, a regularization term can be added using an anchor copy of the initial model:

$$\mathcal{L}_{reg} = \|\theta - \theta_0\|_2^2 \tag{6}$$

The total loss is then:

$$\mathcal{L}_{total} = \mathcal{L}_{ent} + \lambda\mathcal{L}_{reg} \tag{7}$$

where $\lambda$ controls the strength of the regularization. Clients update their local models using gradient descent on this objective.

**Server Aggregation:**

Once local adaptation is complete, each client $i$ sends its model update $\Delta\theta_i$ to the server. To address inter-client class imbalance, we use a class-aware weighting scheme. Let $p_{i,c}$ denote the estimated proportion of class $c$ in client $i$'s pseudo-labels. A client weight is computed as:

$$w_i = \frac{1}{\max_c p_{i,c} + \delta} \tag{8}$$

where $\delta$ is a small constant to avoid division by zero. The global model is then updated as:

$$\theta_{global} \leftarrow \frac{\sum_{i=1}^{K} w_i(\theta_0 + \Delta\theta_i)}{\sum_{i=1}^{K} w_i} \tag{9}$$

This downweights clients with highly imbalanced distributions and upweights those with more balanced or minority-class data.

To ensure robustness, the server may optionally apply coordinate-wise median or trimmed mean instead of simple weighted averaging. This protects the global model from adversarial or noisy client updates.

**Privacy Considerations:**

Throughout the process, client privacy is preserved by keeping raw data and per-class statistics local. Only model deltas (e.g., changes to weights and BBN affine parameters) are transmitted. If desired, clients may apply secure aggregation protocols or add differentially private noise to histogram-based summaries (such as class counts) to further obscure sensitive information.

FedBBN thus supports personalization, unsupervised adaptation, and privacy-preserving federated updates, all while explicitly addressing the impact of class imbalance on both client- and server-side learning dynamics.

## 4 Result and Discussion

### 4.1 Dataset

### 4.2 Experimental Setup

### 4.3 Qualitative Result

### 4.4 Quantitative Result

### 4.5 Discussion

## 5 Conclusion

We introduced **FedBBN**, a federated test-time adaptation framework that addresses class imbalance and domain shifts by leveraging Balanced Batch Normalization. FedBBN enables unsupervised, privacy-preserving client-side adaptation and introduces a class-aware server aggregation strategy. Experiments on benchmark datasets show that FedBBN improves robustness and minority-class performance over existing methods. This makes it a practical and scalable solution for real-world federated learning scenarios with non-IID and unlabeled test distributions.

## References

1. Bao, W., Wei, T., Wang, H., He, J.: Adaptive test-time personalization for federated learning. In: Advances in Neural Information Processing Systems (2023), https://arxiv.org/abs/2310.18816
2. Li, F., Others: Federated learning with probability-corrected loss and shared class prototypes. Journal Name (2024)
3. Liang, F., Others: Tta-feddg: Test-time adaptation for federated domain generalization. In: Proceedings of the AAAI Conference on Artificial Intelligence (2024)

**Table 1.** Performance comparison across federated methods and TTA strategies under different class imbalance ratios ($\delta$). Total clients = 10, i.i.d setup

| **Fed Method** | **TTA Method** | **CIFAR-10** | | | **CIFAR-100** | | |
|---|---|---|---|---|---|---|---|
| | | $\delta = 0.001$ | $\delta = 0.01$ | $\delta = 0.1$ | $\delta = 0.001$ | $\delta = 0.01$ | $\delta = 0.1$ |
| FedAvg | Tent | 19.50 | 37.05 | 43.48 | | | |
| | CoTTA | 23.77 | | | | | |
| | ROID | 21.94 | 43.14 | | | | |
| | RoTTA | 64.96 | | | | | |
| | Ours | 65.56 | | | | | |
| FedProx | Tent | | | | | | |
| | CoTTA | | | | | | |
| | ROID | | | | | | |
| | RoTTA | | | | | | |
| | Ours | | | | | | |
| FedAvgM | Tent | | | | | | |
| | CoTTA | | | | | | |
| | ROID | | | | | | |
| | RoTTA | | | | | | |
| | Ours | | | | | | |
| pfedGraph | Tent | | | | | | |
| | CoTTA | | | | | | |
| | ROID | | | | | | |
| | RoTTA | | | | | | |
| | Ours | | | | | | |
| FedAmp | Tent | | | | | | |
| | CoTTA | | | | | | |
| | ROID | | | | | | |
| | RoTTA | | | | | | |
| | Ours | | | | | | |

**Table 2.** Performance comparison across federated methods and TTA strategies under different class imbalance ratios ($\delta$). Total clients = 10, i.i.d setup

| **Fed Method** | **TTA Method** | **CIFAR-10** | | | **CIFAR-100** | | |
|---|---|---|---|---|---|---|---|
| | | $\delta = 0.0$ | $\delta = 1$ | $\delta = 5$ | $\delta = 0.0$ | $\delta = 1$ | $\delta = 5$ |
| FedAvg | Tent | 19.50 | 37.05 | 43.48 | | | |
| | CoTTA | 23.77 | | | | | |
| | ROID | 21.94 | 43.14 | | | | |
| | RoTTA | 64.96 | | | | | |
| | Ours | 65.56 | | | | | |
| FedProx | Tent | | | | | | |
| | CoTTA | | | | | | |
| | ROID | | | | | | |
| | RoTTA | | | | | | |
| | Ours | | | | | | |
| FedAvgM | Tent | | | | | | |
| | CoTTA | | | | | | |
| | ROID | | | | | | |
| | RoTTA | | | | | | |
| | Ours | | | | | | |
| pfedGraph | Tent | | | | | | |
| | CoTTA | | | | | | |
| | ROID | | | | | | |
| | RoTTA | | | | | | |
| | Ours | | | | | | |
| FedAmp | Tent | | | | | | |
| | CoTTA | | | | | | |
| | ROID | | | | | | |
| | RoTTA | | | | | | |
| | Ours | | | | | | |

**Table 3.** Performance comparison under spatial heterogeneity and temporal IID.

| Datasets | Method | *Gaussian* | *Shot* | *Impulse* | *Defocus* | *Glass* | *Motion* | *Zoom* | *Snow* | *Frost* | *Fog* | *Brightness* | *Contrast* | *Elastic* | *Pixelate* | *Jpeg* | *Mean* |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CIFAR10-C | Source | 37.30 | 38.44 | 26.08 | 28.99 | 33.92 | 27.44 | 30.21 | 34.53 | 32.89 | 10.63 | 36.46 | 23.53 | 37.51 | 41.43 | 43.70 | 30.54 |
| | Local | 55.27 | 52.70 | 56.46 | 48.25 | 55.30 | 50.87 | 58.25 | 46.40 | 55.58 | 52.45 | 58.48 | 45.80 | 51.75 | 56.12 | 49.00 | 52.85 |
| | FedAvg | 61.49 | 56.33 | 60.15 | 50.26 | 61.14 | 53.02 | 63.64 | 50.17 | 60.02 | 53.88 | 63.53 | 50.95 | 55.86 | 59.55 | 52.36 | 56.82 |
| | FedAMP | 62.04 | 57.12 | 61.29 | 52.32 | 61.83 | 55.28 | 63.87 | 51.69 | 60.83 | 55.89 | 63.94 | 52.16 | 56.70 | 61.32 | 53.65 | 58.00 |
| | pFedGraph | 61.66 | 56.41 | 61.01 | 51.87 | 61.18 | 54.44 | 63.70 | 51.60 | 60.23 | 55.17 | 63.44 | 52.42 | 56.46 | 60.22 | 53.56 | 57.56 |
| | FedTSA | 62.16 | 57.59 | 61.72 | 52.58 | 61.91 | 55.36 | 63.96 | 50.87 | 61.33 | 56.67 | 64.36 | 51.37 | 57.15 | 61.78 | 53.23 | 58.14 |
| | Ours-v1(FM) | 65.12 | | | | | | | | | | | | | | | |
| CIFAR100-C | Source | 14.05 | 16.64 | 34.76 | 41.60 | 19.35 | 38.15 | 43.32 | 36.48 | 27.63 | 20.96 | 54.91 | 17.24 | 35.02 | 11.45 | 41.73 | 30.22 |
| | Local | 50.68 | 54.84 | 56.29 | 55.56 | 51.77 | 54.06 | 59.04 | 51.24 | 54.80 | 54.43 | 57.43 | 51.30 | 53.15 | 57.56 | 54.87 | 54.47 |
| | FedAvg | 52.14 | 43.19 | 63.30 | 48.91 | 53.90 | 49.31 | 65.61 | 46.71 | 54.31 | 49.38 | 61.93 | 45.47 | 51.57 | 57.00 | 56.34 | 53.27 |
| | FedAMP | 62.04 | 57.12 | 61.29 | 52.32 | 61.83 | 55.28 | 63.87 | 51.69 | 60.83 | 55.89 | 63.94 | 52.16 | 56.70 | 61.32 | 53.65 | 58.00 |
| | pFedGraph | 61.66 | 56.41 | 61.01 | 51.87 | 61.18 | 54.44 | 63.70 | 51.60 | 60.23 | 55.17 | 63.44 | 52.42 | 56.46 | 60.22 | 53.56 | 57.56 |
| | FedTSA | 57.33 | 58.60 | 56.74 | 71.07 | 57.51 | 68.94 | 70.92 | 64.29 | 64.19 | 57.44 | 72.40 | 67.36 | 63.70 | 66.33 | 57.56 | 63.63 |
| | Ours | | | | | | | | | | | | | | | | |

**Table 4.** The experimental scenario and performance comparison of the case study.

| Time | t → | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Client 1-4 | *Gaussian* | *Shot* | *Impulse* | *Defocus* | *Glass* | *Motion* | *Zoom* | *Snow* | *Frost* | *Fog* | *Brightness* | *Contrast* | *Elastic* | *Pixelate* | *Jpeg* | *Mean* |
| Client 5-7 | *Shot* | *Impulse* | *Defocus* | *Glass* | *Motion* | *Zoom* | *Snow* | *Frost* | *Fog* | *Brightness* | *Contrast* | *Elastic* | *Pixelate* | *Jpeg* | *Gaussian* | *Mean* |
| Client 8-10 | *Jpeg* | *Pixelate* | *Elastic* | *Contrast* | *Brightness* | *Fog* | *Frost* | *Snow* | *Zoom* | *Motion* | *Glass* | *Defocus* | *Impulse* | *Shot* | *Gaussian* | *Mean* |
| No-Adapt | 20.85 | 19.55 | 34.65 | 26.70 | 33.40 | 33.45 | 35.45 | 31.80 | 27.15 | 36.30 | 31.75 | 27.70 | 25.80 | 20.40 | 23.75 | 28.58 |
| FedAMP | 48.00 | 54.75 | 61.35 | 53.30 | 60.05 | 61.60 | 62.30 | 59.05 | 58.60 | 58.30 | 52.15 | 54.75 | 54.45 | 51.65 | 54.80 | 56.34 |
| pFedGraph | 47.00 | 51.90 | 61.05 | 51.20 | 58.45 | 62.50 | 63.25 | 59.75 | 55.95 | 58.55 | 50.30 | 54.00 | 52.50 | 50.80 | 53.65 | 55.39 |
| FedTSA | 48.95 | 61.15 | 62.25 | 54.90 | 60.15 | 63.75 | 63.75 | 63.05 | 58.35 | 59.90 | 54.00 | 58.15 | 58.15 | 57.45 | 56.40 | 58.69 |
| Ours | | | | | | | | | | | | | | | | |


4. Marsden, R.A., Döbler, M., Yang, B.: Universal test-time adaptation through weight ensembling, diversity weighting, and prior correction. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV). pp. 2555–2565 (2024)
5. Names, A.: Fed-grab: Self-adjusting gradient balancer for federated learning on long-tailed data. In: Advances in Neural Information Processing Systems (2023)
6. Names, A.: Fl-fcr: Federated learning with frequency calibration and resampling. In: Conference Name (2023)
7. Names, A.: Gbme: Global balanced multi-expert for class-imbalanced federated learning. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (2023)
8. Names, A.: Dynfed: Adaptive rate networks for test-time personalization in federated learning. Transactions on Machine Learning Research (2025)
9. Shang, X., Lu, Y., Huang, G., Wang, H.: Federated learning on heterogeneous and long-tailed data via classifier re-training with federated features. In: Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence. pp. 2218–2224 (2022). https://doi.org/10.24963/ijcai.2022/308
10. Shao, F., Others: Federated face anti-spoofing with test-time adaptation. In: Proceedings of the IEEE International Conference on Automatic Face and Gesture Recognition (2021)
11. Wang, D., Shelhamer, E., Liu, S., Olshausen, B., Darrell, T.: Tent: Fully test-time adaptation by entropy minimization. In: International Conference on Learning Representations (2021), https://arxiv.org/abs/2006.10726
12. Wang, Q., Fink, O., Gool, L.V., Dai, D.: Continual test-time domain adaptation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2022), https://arxiv.org/abs/2203.13591
13. Xian, S., Shen, Y., Jiang, S., Zhao, Z., Yan, Z., Xing, G.: Balancefl: Addressing class imbalance in long-tail federated learning. In: 2022 21st ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN). pp. 271–284 (2022). https://doi.org/10.1109/IPSN54338.2022.00029

14. Yuan, L., Xie, B., Li, S.: Robust test-time adaptation in dynamic scenarios. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2023), https://arxiv.org/abs/2303.13899