FedKLEntropy: Enhancing Federated Learning via Model Entropy and Kullback-Leibler Divergence*

Duy-Dong Le
dongld@ueh.edu.vn
Industrial University of Ho Chi Minh City (IUH)
Ho Chi Minh City, Vietnam

Tuong-Nguyen Huynh*
htnguyen@iuh.edu.vn
Industrial University of Ho Chi Minh City (IUH)
Ho Chi Minh City, Vietnam

Nhat Tung Le
lenhattung@dntu.edu.vn
Dong Nai Technology University (DNTU)
Dong Nai Province, Vietnam

ABSTRACT

Federated Learning (FL) enables collaborative model training across distributed clients while preserving data privacy, but non-independent and identically distributed (non-i.i.d) data poses challenges such as unstable convergence and client drift. We introduce FedKLEntropy, a novel aggregation method that employs Kullback-Leibler (KL) divergence and entropy-based measures to weight client contributions based on the alignment of local and global model weight distributions. By prioritizing clients with distributions closely aligned with the global model, FedKLEntropy enhances stability and generalization in heterogeneous settings without requiring server-side data or complex computations. Experimental results on MNIST and Fashion-MNIST demonstrate that FedKLEntropy outperforms FedAsl and matches or exceeds MOON in terms of training and validation losses, achieving up to 91.32% accuracy on MNIST and 85.66% on Fashion-MNIST under non-i.i.d conditions, showcasing its robustness and effectiveness.

CCS CONCEPTS

• **Computing methodologies** → **Distributed artificial intelligence**; *Neural networks.*

KEYWORDS

Federated Learning, Kullback-Leibler Divergence, Entropy-Based Weighting, non-i.i.d Data

ACM Reference Format:
Duy-Dong Le, Tuong-Nguyen Huynh*, and Nhat Tung Le. 2025. FedKLEntropy: Enhancing Federated Learning via Model Entropy and Kullback-Leibler Divergence. In *Proceedings of ACM RACS Conference (RACS'25).* ACM, New York, NY, USA, Article 4, 9 pages. https://doi.org/xx.xxx/xxx_x

1 INTRODUCTION

FL facilitates collaborative model training across decentralized devices, ensuring data privacy by eliminating centralized data storage [15]. This paradigm is vital in domains like healthcare [1, 21], agriculture [7], and mobile computing [2], where data sensitivity and regulatory compliance are critical. However, non-i.i.d data across clients lead to challenges such as inconsistent convergence and client drift, resulting in biased global models [4, 22].

The foundational FedAvg algorithm [15] aggregates client updates weighted by dataset sizes, reducing communication overhead but struggling with non-i.i.d data, leading to poor generalization [5, 11]. To address this, advanced methods have emerged. FedAsl [17] dynamically weights clients based on the standard deviation of training losses, improving resilience but remaining sensitive to outliers. MOON [12] leverages model contrastive loss to align local models with the global model, achieving robust performance at the cost of increased computational complexity. Other approaches, such as FedProx [13] and FedMa [18], introduce proximal terms or neuron matching, respectively, but incur significant computational overhead.

We propose FedKLEntropy, a novel FL aggregation method that weights client contributions based on the entropy-based KL divergence between local and global model weight distributions. By prioritizing clients whose weight distributions are closer to the global model, FedKLEntropy mitigates the impact of heterogeneous data while maintaining computational efficiency. Our key contributions are:

- Development of FedKLEntropy, a novel FL aggregation technique that integrates KL divergence and entropy-based weighting to improve global model stability in non-i.i.d settings without necessitating server-side data or intricate computational processes.
- Empirical validation on MNIST and Fashion-MNIST datasets, where FedKLEntropy achieves superior or comparable performance to FedAsl and MOON, with mean accuracies of 91.32% on MNIST and 85.66% on Fashion-MNIST, demonstrating its effectiveness in handling data heterogeneity.
- Theoretical proof of convergence to a stationary point under standard FL assumptions, supported by experimental results showing rapid and stable convergence over 50 and 100 rounds for MNIST and Fashion-MNIST, respectively.

The paper is organized as follows: Section 2 reviews related work, Section 3 details FedKLEntropy, Section 4 describes the experimental setup, Section 5 presents results, and Section 8 concludes with future directions. Appendices provide detailed sensitivity and convergence analyses.

2 RELATED WORK

FedAvg [15] aggregates client updates as $w^{t+1} = \sum_{k \in S^t} \frac{n_k}{n} w_k^{t+1}$, where n_k is the dataset size of client k, $n = \sum_k n_k$, and S^t is the

*Produces the permission block, and copyright information

RACS'25, November 16 - November 19, 2025, Ho Chi Minh, Vietnam

ACM ISBN 979-8-4007-0243-3/24/04…$15.00
https://doi.org/xx.xxx/xxx_x

subset of clients in round t. While communication-efficient, FedAvg struggles with non-i.i.d data, leading to slow convergence [3, 22].

FedAsl [17] uses dynamic weights based on loss deviations, with $w^{t+1} = \sum_{k=1}^{N} A_k w_k^{t+1}$, where $A_k = \frac{d_k^{-1}}{\sum_{k=1}^{K} d_k^{-1}}$, and $d_k = \beta\sigma$ if the loss L_k lies within $[\mu_L - \alpha\sigma, \mu_L + \alpha\sigma]$, or $d_k = |L_k - \mu_L|$ otherwise (μ_L: median loss, σ: standard deviation, α, β: parameters). This improves resilience but is sensitive to outliers.

MOON [12] employs model contrastive loss to align local models with the global model, using a loss function $\mathcal{L} = \mathcal{L}_{\text{sup}} + \mu\mathcal{L}_{\text{con}}$, where $\mathcal{L}_{\text{con}}$ measures cosine similarity between local, global, and previous local model representations. While effective, MOON increases computational cost due to additional forward passes.

Other methods, like FedProx [13], add a proximal term to local objectives, and FedMa [18] uses neuron matching, both increasing computational overhead. Dynamic weighting approaches, such as FedLaw [14] and FedA-Flama [19], rely on server-side data or complex optimization, limiting scalability. Personalized FL [16] and knowledge distillation [8] offer alternatives but sacrifice global generalization or require curated datasets.

Table 1 compares FedKLEntropy with FedAvg, FedAsl, and MOON, highlighting its simplicity and efficiency in handling non-i.i.d data.

3 PROPOSED FEDKLENTROPY

3.1 Problem Formulation

Consider an FL system with N clients, each with a local dataset $\mathcal{D}_k$ of size $n_k = |\mathcal{D}_k|$, drawn from a non-i.i.d distribution $\mathcal{P}_k$. The total dataset size is $n = \sum_{k=1}^{N} n_k$. Each client k optimizes a local model $w \in \mathbb{R}^d$ to minimize $\mathcal{L}_k(w) = \frac{1}{n_k} \sum_{x \in \mathcal{D}_k} \ell(w; x)$. The global objective is:

$$\min_{w \in \mathbb{R}^d} \mathcal{L}(w) = \sum_{k=1}^{N} \alpha_k \mathcal{L}_k(w), \tag{1}$$

where $\alpha_k > 0$, $\sum_{k=1}^{N} \alpha_k = 1$. In FedAvg, $\alpha_k = \frac{n_k}{n}$, which falters under non-i.i.d conditions. FedKLEntropy dynamically adjusts α_k^t based on the KL divergence between local and global model weight distributions.

3.2 FedKLEntropy Weighting Mechanism

FedKLEntropy weights clients using the KL divergence between the weight distributions of local models w_k^{t+1} and the global model w^t. The weights are computed as follows:

Definition 1 (FedKLEntropy Weighting)**.** For each client $k \in S^t$ at round t:

(1) **Weight Distribution:** Compute histograms of model weights for w_k^{t+1} and w^t, with $M = 100$ bins spanning the range of weights. Let p_k and p_g denote the probability distributions of local and global weights, respectively.

(2) **KL Divergence:** Calculate the entropy-based KL divergence:

$$D_{\text{KL}}(p_k \| p_g) = \sum_{i=1}^{M} p_k(i) \log \frac{p_k(i)}{p_g(i)}, \tag{2}$$

handling zero probabilities with a small constant (10^{-12}).

(3) **Weight Assignment:**

$$\alpha_k^t = \frac{1/(1 + D_{\text{KL}}(p_k \| p_g))}{\sum_{j \in S^t} 1/(1 + D_{\text{KL}}(p_j \| p_g))}, \tag{3}$$

where the inverse ensures higher weights for lower divergences, normalized to sum to 1.

The global update is:

$$w^{t+1} = \sum_{k \in S^t} \alpha_k^t w_k^{t+1}. \tag{4}$$

Algorithm 1 outlines the procedure.

Algorithm 1 FedKLEntropy: Entropy-Based Weighted Aggregation

ServerSide:
Initialize global model w^0
for each round $t = 0, 1, \ldots, T-1$ **do**
  Randomly select subset $S^t \subseteq \{1, 2, \ldots, N\}$
  for each client $k \in S^t$ **in parallel do**
    $w_k^{t+1} \leftarrow \text{ClientSide}(k, w^t)$
  end for
  Compute α_k^t using Definition 1
  Update $w^{t+1} \leftarrow \sum_{k \in S^t} \alpha_k^t w_k^{t+1}$
end for

function ClientSide(k, w^t)
  for each local epoch $e = 1, \ldots, E$ **do**
    Update $w_k^{t+1} \leftarrow w_k^t - \eta \nabla \mathcal{L}_k(w_k^t; \mathcal{B})$
  end for
  return w_k^{t+1}
end function

FedKLEntropy prioritizes clients with weight distributions closer to the global model, mitigating drift in non-i.i.d settings. Its server-side complexity is $O(|S^t|d + |S^t|M)$, where M is the number of histogram bins, making it comparable to FedAvg and less costly than MOON. The convergence analysis of FedKLEntropy is provided in Appendix A.

4 EXPERIMENTS SETUP

We evaluate FedKLEntropy against FedAvg [15], FedAsl [17], and MOON [12] on MNIST [10], Fashion-MNIST [20] under non-i.i.d settings.

4.1 Datasets and non-i.i.d Partitioning

We utilize the MNIST dataset (60,000 training, 10,000 test samples, 28×28 grayscale digit images) and the Fashion-MNIST dataset (60,000 training, 10,000 test samples, 28×28 grayscale clothing images), distributed across 50 clients using a Dirichlet distribution to model non-i.i.d data heterogeneity. For MNIST, we set the concentration parameter $\alpha = 0.1$, while for Fashion-MNIST, we use $\alpha = 0.3$. The test data is retained centrally on the server. The resulting heterogeneous data distributions are visualized in Figures 1 and 2.

Table 1: Comparison of FL Algorithms

Method	Server-side (aggregate w^{t+1})	Client-side (update w_k^{t+1})	Characteristics
FedAvg [15]	$\sum_{k \in S^t} \frac{n_k}{n} w_k^{t+1}$	$w_k^t - \eta \nabla \mathcal{L}_k(w_k^t)$	Simple, communication-efficient; Poor performance on non-i.i.d data
FedAsl [17]	$\sum_{k=1}^{N} A_k w_k^{t+1}$	$w_k^t - \eta \nabla \mathcal{L}_k(w_k^t)$	Dynamic weighting by loss deviation; Sensitive to outliers
MOON [12]	$\sum_{k \in S^t} \frac{n_k}{n} w_k^{t+1}$	$w_k^t - \eta \nabla (\mathcal{L}_k(w_k^t) + \mu \mathcal{L}_{\text{con}}(z_k, z_g, z_p))$	Model contrastive loss; High computational cost
FedKLEntropy	$\sum_{k=1}^{N} \alpha_k w_k^{t+1}$	$w_k^t - \eta \nabla \mathcal{L}_k(w_k^t)$	Entropy-based KL divergence weighting; Robust to non-i.i.d data

4.1.1 Sample and Class Distribution in the MNIST Dataset with Dirichlet $\alpha = 0.1$. Figure 1 provides statistics about the class and sample distribution of MNIST with Dirichlet concentration $\alpha = 0.1$ across 50 clients, the Fashion-MNIST dataset exhibits a non-uniform distribution of samples and classes per client, a characteristic commonly found in FL tasks. Regarding the number of classes, the distribution is extremely skewed. On average, each client has 4.64 classes, but this number varies drastically. Client 10 is the most diverse, holding 9 classes, while Client 1 has only 1 class. The low value of $\alpha = 0.1$ forces a severe class imbalance, meaning most clients have data for only a few classes, often just one, which is characteristic of real-world data distribution in federated settings.

For the number of samples, the disparity is also very pronounced. The average number of samples per client is 1200.00. However, Client 41 stands out with a massive number of samples, 3997, more than three times the average. Conversely, Client 48 has a mere 32 samples, making it the client with the fewest. This vast difference in sample count across clients, combined with the extreme class imbalance, highlights the challenging statistical heterogeneity of this partitioned dataset.

4.1.2 Sample and Class Distribution in the Fashion-MNIST Dataset with Dirichlet $\alpha = 0.3$. Figure 2 provides statistics about the class and sample distribution of Fashion-MNIST with Dirichlet concentration $\alpha = 0.3$ across 50 clients, the Fashion-MNIST dataset exhibits a non-uniform distribution of samples and classes per client, a characteristic commonly found in FL tasks. Regarding the number of classes, the distribution is highly varied. On average, each client possesses 7.52 classes, but this number varies significantly. Client 8 is the most diverse client, with all 10 classes, while Client 11 has only 3 classes. This indicates that some clients have rich data across different product types, while others focus on only a few.

For the number of samples, the disparity is even more pronounced. The average number of samples per client is 1200.00. However, Client 31 stands out with the largest number of samples, 2223, almost double the average. Conversely, Client 5 has only 99 samples, making it the client with the fewest. This large difference in sample count across clients highlights the dataset's non-independent and identically distributed (non-IID) nature. This imbalance is a significant challenge in training machine learning models effectively and fairly on distributed devices.

4.2 Model Architectures

4.2.1 LeNet-5 for MNIST. The LeNet-5 architecture, a classic convolutional neural network (CNN), is adapted for the MNIST dataset. The model processes single-channel 28×28 input images. The first convolutional layer applies 32 filters of size 5×5, followed by batch normalization and a Rectified Linear Unit (ReLU) activation function. A subsequent 2×2 max-pooling layer reduces the feature map dimensions. The second convolutional layer uses 64 filters of size 5×5, also followed by batch normalization, ReLU, and another 2×2 max-pooling layer. The flattened output from the convolutional layers is fed into a fully connected layer with 256 units, a dropout layer for regularization, and a second fully connected layer with 128 units. A projection head, consisting of two linear layers with an intermediate ReLU, is included to produce a 256-dimensional representation. The final layer is a linear output layer with 10 units for classification.

4.2.2 CNN for Fashion-MNIST. The CNN model designed for the Fashion-MNIST dataset is tailored to handle the dataset's increased complexity. It takes single-channel 28×28 input images. The architecture consists of three convolutional blocks. The first block uses a convolutional layer with 32 filters of size 3×3 with padding, followed by batch normalization, ReLU activation, and a 2×2 max-pooling layer. The second block repeats this pattern with 64 filters, and the third block uses 128 filters. After the third max-pooling layer, the feature maps are flattened. This is followed by a dropout layer with a rate of 0.3 and a fully connected layer with 512 units.

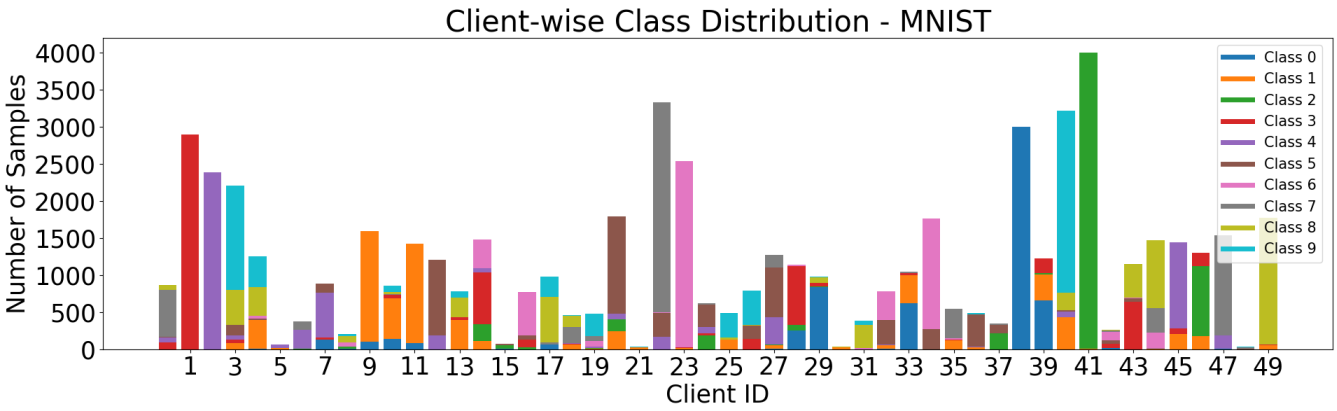

Figure 1: non-i.i.d class and sample distributions for MNIST ($\alpha = 0.1$)

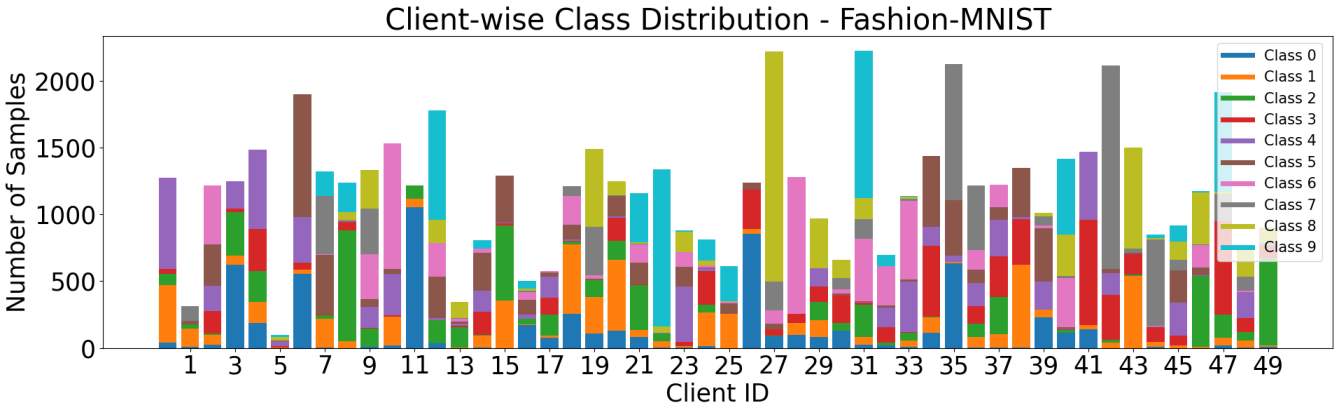

Figure 2: non-i.i.d class and sample distributions for Fashion-MNIST ($\alpha = 0.3$)

A projection head, comprising two linear layers with an intermediate ReLU, generates a 256-dimensional representation. The model concludes with a linear output layer of 10 units for classification.

4.3 Training Parameters and Evaluation Metrics

Experiments involve 50 clients over $T = 50$ (MNIST) or 100 for Fashion-MNIST rounds, with 10% client selection each round. Local training uses SGD ($\eta = 0.01$, momentum 0.9, weight decay 0.001, batch size 32, epochs $E = 2$). Metrics include training loss, validation loss, accuracy, and F1-score.

5 RESULTS

We compare FedKLEntropy with FedAsl ($\alpha = 0.5, \beta = 0.2$) and MOON ($\mu = 1, \tau = 0.5$) across MNIST, Fashion-MNIST, reporting average metrics from three runs.

5.1 Experiment 1: MNIST

Figures 3 and Table 2 present the performance of FedKLEntropy (blue), FedAsl (orange), and MOON (green) over 50 communication rounds with 10% client selection per round. FedKLEntropy demonstrates the lowest and most stable training and validation losses, converging quickly and outperforming FedAsl and MOON. FedAsl exhibits higher initial losses with noticeable fluctuations, while MOON shows moderate stability but higher losses compared to FedKLEntropy. Accuracy trends indicate FedKLEntropy achieving the highest values, approaching 90%, followed by MOON and FedAsl.

As illustrated in Figure 3(a), the training loss for FedKLEntropy (blue) decreases rapidly from an initial value around 1.2 to near 0.1 within the first 10 rounds, with minimal fluctuations thereafter, demonstrating efficient local optimization and effective aggregation in the federated setting. In contrast, FedAsl (orange) starts similarly but exhibits more pronounced oscillations throughout the training process, particularly between rounds 10 and 30, suggesting sensitivity to client heterogeneity. MOON (green) shows a steadier decline but converges to a higher loss value (around 0.3), indicating slower adaptation to non-i.i.d data distributions. For validation loss in Figure 3(b), FedKLEntropy again achieves the lowest values, dropping below 0.5 by round 10 and stabilizing near 0.2, while FedAsl displays spikes up to 1.0 in later rounds, highlighting potential overfitting or instability. MOON's validation

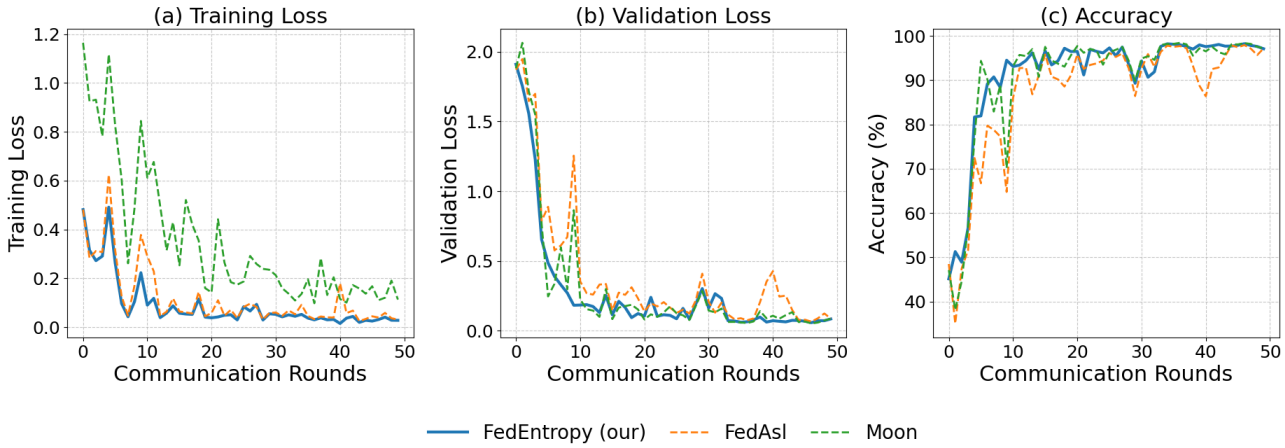

Figure 3: Performance on MNIST with 10% client selection each round: (a) training loss, (b) validation loss, (c) accuracy.

loss mirrors its training pattern, converging higher than FedKLEntropy but with less variability than FedAsl. The accuracy curve in Figure 3(c) further corroborates these observations: FedKLEntropy quickly surpasses 90% accuracy by round 10 and maintains a plateau near 95%, outperforming MOON (peaking around 92%) and FedAsl (hovering below 90% with dips), underscoring FedKLEntropy's superior generalization and robustness in heterogeneous federated environments.

Table 2 quantifies these performance differences through mean metrics averaged over the 50 rounds, with standard deviations indicating variability. FedKLEntropy achieves the lowest mean training loss of 0.0906 with a standard deviation of 0.1088, reflecting consistent and effective minimization of local losses across clients. Comparatively, MOON's higher mean training loss (0.3530 ± 0.2842) suggests less efficient convergence, while FedAsl's value (0.1185 ± 0.1280) is closer to FedKLEntropy but undermined by higher variability in validation metrics. In terms of validation loss, FedKLEntropy's 0.2752 ± 0.4215 outperforms MOON (0.3038 ± 0.4784) and significantly surpasses FedAsl (0.3954 ± 0.4766), indicating better generalization to unseen data. The F1-score and accuracy further highlight FedKLEntropy's superiority, with means of 0.9039 ± 0.1534 and 91.32% ± 12.80%, respectively, compared to MOON's 0.8946 ± 0.1752 and 90.71% ± 14.26%, and FedAsl's lower 0.8618 ± 0.1682 and 87.32% ± 14.67%. Notably, FedKLEntropy's lower standard deviations across most metrics underscore its stability, making it particularly suitable for non-i.i.d FL scenarios where data heterogeneity can amplify performance variances.

5.2 Experiment 2: Fashion-MNIST

Figure 4 and Table 3 present the performance of FedKLEntropy (blue), FedAsl (orange), and MOON (green) over 100 communication rounds with 10% client selection per round. FedKLEntropy exhibits the lowest and most stable training and validation losses, converging effectively and surpassing FedAsl and MOON. FedAsl shows higher initial losses with significant fluctuations, while MOON demonstrates moderate stability but higher losses compared to FedKLEntropy. Accuracy trends indicate FedKLEntropy achieving the highest values, approaching 85%, followed by MOON and FedAsl.

As depicted in Figure 4(a), the training loss for FedKLEntropy (blue) decreases steadily from an initial value near 1.2 to approximately 0.1 by round 20, maintaining low variability thereafter, which highlights its robust adaptation to the complex Fashion-MNIST dataset. FedAsl (orange) starts with a similar decline but experiences pronounced oscillations, especially between rounds 30 and 70, suggesting challenges in handling the dataset's heterogeneity. MOON (green) shows a more gradual reduction, stabilizing around 0.3, indicating slower convergence likely due to its reliance on contrastive learning under non-i.i.d conditions. In Figure 4(b), the validation loss for FedKLEntropy drops below 0.5 by round 20 and stabilizes near 0.3, outperforming FedAsl, which peaks around 1.5 during mid-rounds, and MOON, which hovers around 0.5 with periodic spikes. This suggests FedKLEntropy's superior generalization, while FedAsl's fluctuations may indicate overfitting. The accuracy plot in Figure 4(c) reinforces these findings: FedKLEntropy reaches approximately 85% accuracy by round 30 and sustains this level with minor variations, outperforming MOON (peaking near 83%) and FedAsl (stabilizing below 80%), underscoring FedKLEntropy's effectiveness in managing diverse client data distributions.

Table 3 provides a quantitative summary of these trends over 100 rounds, with standard deviations reflecting variability. FedKLEntropy achieves the lowest mean training loss of 0.1285 with a standard deviation of 0.0816, indicating consistent local optimization across clients despite the dataset's complexity. MOON's higher mean training loss (0.3208 ± 0.2042) and FedAsl's value (0.1527 ± 0.0860) suggest less efficient convergence, with FedAsl showing slightly higher variability. For validation loss, FedKLEntropy's mean of 0.4061 ± 0.1678 outperforms MOON (0.4306 ± 0.1857) and significantly surpasses FedAsl (0.5033 ± 0.2006), reflecting better generalization to unseen data. The F1-score and accuracy further validate FedKLEntropy's edge, with means of 0.8495 ± 0.0706 and 85.66% ± 5.96%, respectively, compared to MOON's 0.8434 ± 0.0767

Table 2: Mean Metrics (± Std) on MNIST for 10% Client Fraction

Algorithm	Train Loss	Validation Loss	F1-Score	Accuracy
FedKLEntropy	**0.0906 ± 0.1088**	**0.2752 ± 0.4215**	**0.9039 ± 0.1534**	**91.32 ± 12.80**
MOON	0.3530 ± 0.2842	0.3038 ± 0.4784	0.8946 ± 0.1752	90.71 ± 14.26
FedAsl	0.1185 ± 0.1280	0.3954 ± 0.4766	0.8618 ± 0.1682	87.32 ± 14.67

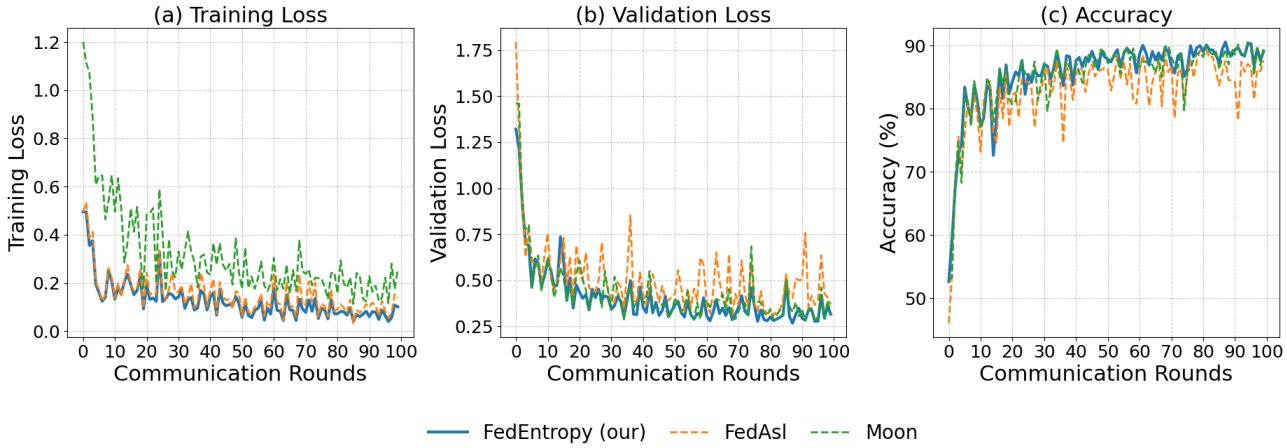

Figure 4: Performance on Fashion-MNIST with 10% client selection each round: (a) training loss, (b) validation loss, (c) accuracy.

Table 3: Mean Metrics (± Std) on Fashion-MNIST for 10% Client Fraction

Algorithm	Train Loss	Validation Loss	F1-Score	Accuracy
FedKLEntropy	**0.1285 ± 0.0816**	**0.4061 ± 0.1678**	**0.8495 ± 0.0706**	**85.66 ± 5.96**
MOON	0.3208 ± 0.2042	0.4306 ± 0.1857	0.8434 ± 0.0767	85.14 ± 6.17
FedAsl	0.1527 ± 0.0860	0.5033 ± 0.2006	0.8162 ± 0.0780	82.96 ± 6.35

and 85.14% ± 6.17%, and FedAsl's lower 0.8162 ± 0.0780 and 82.96% ± 6.35%. The tighter standard deviations for FedKLEntropy across all metrics highlight its stability and reliability, making it particularly adept at handling the non-i.i.d nature of Fashion-MNIST across distributed clients.

6 DISCUSSION ON FINDINGS

Our experiments show that **FedKLEntropy** is a highly effective aggregation strategy for FL in non-i.i.d environments. By using an entropy-based weighting mechanism, FedKLEntropy actively mitigates client drift, a common problem when data is not uniformly distributed. Unlike FedAsl, which can be sensitive to outlier losses, or MOON, which introduces computational overhead with contrastive loss, our method offers a simple yet powerful alternative. The core strength of FedKLEntropy lies in its principled approach: it weights client contributions based on the structural alignment of local model weights with the global model. This ensures that the global model evolves in a stable, predictable direction, leading to robust performance.

The empirical results on MNIST and Fashion-MNIST validate this approach. FedKLEntropy not only achieved superior mean accuracies–**91.32% on MNIST** and **85.66% on Fashion-MNIST**–but also demonstrated remarkable stability. The low standard deviations in our metrics, such as accuracy and F1-score, confirm the algorithm's resilience against the statistical challenges of non-i.i.d data. We observed that FedKLEntropy consistently reduced both training and validation losses, indicating excellent generalization. The convergence curves also revealed a rapid and stable path to optimality, reaching a performance plateau after 50-100 rounds. This efficient convergence makes FedKLEntropy a compelling solution for resource-constrained federated environments.

7 LIMITATIONS AND FUTURE DIRECTIONS

Despite its promising performance, FedKLEntropy has inherent limitations that pave the way for future research. One key area for improvement is the scalability of its histogram-based weighting, which could become a bottleneck for models with an extremely large number of parameters. The current fixed number of bins ($M = 100$) might not be optimal for all model architectures and could be replaced with a more adaptive strategy. Our evaluation was confined to two academic datasets, so its performance in more complex, real-world scenarios—such as those with heterogeneous data from a variety of devices—has yet to be fully validated.

Moving forward, our research will focus on several key areas. We plan to test FedKLEntropy on more complex datasets like CIFAR [6] to assess its generalizability and scale our experiments to include hundreds or thousands of clients. This will provide a more realistic measure of the algorithm's performance in large-scale federated networks. Finally, we will explore advanced extensions to FedKLEntropy, including adaptive weighting strategies that dynamically adjust based on model complexity and data heterogeneity. We will also investigate alternative divergence measures, such as Jensen-Shannon divergence, to see if they can further improve performance and robustness.

8 CONCLUSION

FedKLEntropy presents a robust FL aggregation approach that leverages KL divergence and entropy-based weighting to ensure stability in non-i.i.d environments. Experimental results on MNIST and Fashion-MNIST demonstrate that FedKLEntropy surpasses FedAsl and performs comparably to or better than MOON in terms of training and validation losses, achieving peak accuracies of 91.32% on MNIST and 85.66% on Fashion-MNIST. Future research will focus on evaluating alternative divergence metrics and incorporating feedback mechanisms [9] to enhance communication efficiency and further improve performance across diverse federated settings.

REFERENCES

[1] Hao Guan, Pew-Thian Yap, Andrea Bozoki, and Mingxia Liu. 2024. Federated learning for medical image analysis: A survey. *Pattern Recognition* 151 (2024), 110424. https://doi.org/10.1016/j.patcog.2024.110424

[2] Andrew Hard, Kanishka Rao, Rajiv Mathews, Françoise Beaufays, Sean Augenstein, Hubert Eichner, Chloé Kiddon, and Daniel Ramage. 2018. Federated Learning for Mobile Keyboard Prediction. *CoRR* abs/1811.03604 (2018). http://dblp.uni-trier.de/db/journals/corr/corr1811.html#abs-1811-03604

[3] Tzu-Ming Harry Hsu, Hang Qi, and Matthew Brown. 2020. Measuring the Effects of Non-Identical Data Distribution for Federated Visual Classification. *CoRR* abs/1909.06335 (2020).

[4] Peter Kairouz, H. Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, et al. 2021. Advances and Open Problems in Federated Learning. *Foundations and Trends in Machine Learning* 14, 1–2 (2021), 1–210.

[5] Jakub Konečný, H. Brendan McMahan, Felix X. Yu, Peter Richtárik, Ananda Theertha Suresh, and Dave Bacon. 2016. Federated Learning: Strategies for Improving Communication Efficiency. *CoRR* abs/1610.05492 (2016). arXiv:1610.05492 http://arxiv.org/abs/1610.05492

[6] Alex Krizhevsky and Geoffrey Hinton. 2009. Learning Multiple Layers of Features from Tiny Images. *Technical Report, University of Toronto* (2009).

[7] Duy-Dong Le, Minh-Son Dao, Anh-Khoa Tran, Thi-Bich Nguyen, and Hong-Giang Le-Thi. 2023. Federated Learning in Smart Agriculture: An Overview. In *2023 15th International Conference on Knowledge and Systems Engineering (KSE)*. IEEE, 1–4.

[8] Duy-Dong Le, Dinh Tuong Huynh, and Pham The Bao. 2025. Correlation-Based Weighted Federated Learning with Multimodal Sensing and Knowledge Distillation: An Application on a Real-World Benchmark Dataset. In *International Conference on Multimedia Modeling*. Springer, 49–60.

[9] Duy-Dong Le, Anh-Khoa Tran, The-Bao Pham, and Tuong-Nguyen Huynh. 2024. A survey of model compression and its feedback mechanism in federated learning. In *Proceedings of the 5th ACM Workshop on Intelligent Cross-Data Analysis and Retrieval*. 37–42.

[10] Yann LeCun, Corinna Cortes, and Christopher J. Burges. 1998. The MNIST Database of Handwritten Digits. *http://yann.lecun.com/exdb/mnist/* (1998).

[11] Qinbin Li, Yiqun Diao, Quan Chen, and Bingsheng He. 2022. Federated learning on non-iid data silos: An experimental study. In *2022 IEEE 38th international conference on data engineering (ICDE)*. IEEE, 965–978.

[12] Qinbin Li, Bingsheng He, and Dawn Song. 2021. Model-contrastive federated learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 10713–10722.

[13] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. 2020. Federated optimization in heterogeneous networks. *Proceedings of Machine learning and systems* 2 (2020), 429–450.

[14] Zeju Li, Tian Lin, Xiaoyu Shang, and Chao Wu. 2023. Revisiting Weighted Aggregation in Federated Learning with Neural Networks. *Proceedings of Machine Learning Research* 202 (2023), 19767–19788.

[15] H. Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. 2017. Communication-Efficient Learning of Deep Networks from Decentralized Data. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS)*, Vol. 54. PMLR, 1273–1282.

[16] Canh T Dinh, Nguyen Tran, and Tuan Dung Nguyen. 2020. Personalized Federated Learning with Moreau Envelopes. In *Advances in Neural Information Processing Systems*, Vol. 33. Curran Associates, Inc., 21394–21405.

[17] Zahid Talukder and Mohammad A. Islam. 2022. Computationally Efficient Auto-Weighted Aggregation for Heterogeneous Federated Learning. *2022 IEEE International Conference on Edge Computing and Communications (EDGE)* (2022), 12–22.

[18] Hongyi Wang, Mikhail Yurochkin, Yuekai Sun, Dimitris S. Papailiopoulos, and Yasaman Khazaeni. 2020. Federated Learning with Matched Averaging. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*.

[19] Rebekah Wang and Yingying Chen. 2024. Adaptive Model Aggregation in Federated Learning Based on Model Accuracy. *IEEE Wireless Communications* 31, 5 (2024), 200–206. https://doi.org/10.1109/MWC.012.2300444

[20] Han Xiao, Kashif Rasul, and Roland Vollgraf. 2017. Fashion-MNIST: A Novel Image Dataset for Benchmarking Machine Learning Algorithms. *arXiv preprint arXiv:1708.07747* (2017).

[21] Jie Xu, Benjamin S. Glicksberg, Chang Su, Peter Walker, Jiang Bian, and Fei Wang. 2021. Federated Learning for Healthcare Informatics. *Journal of Healthcare Informatics Research* 5, 1 (2021), 1–19.

[22] Hangyu Zhu, Jinjin Xu, Shiqing Liu, and Yaochu Jin. 2021. Federated learning on non-IID data: A survey. *Neurocomputing* 465 (2021), 371–390.

APPENDIX

A CONVERGENCE ANALYSIS OF FEDKLENTROPY

A.1 Assumptions

We make the following standard assumptions for FL convergence analysis:

Assumption 1 (L-smoothness). *Each local loss function $\mathcal{L}_k$ is L-smooth:*

$$\|\nabla \mathcal{L}_k(w) - \nabla \mathcal{L}_k(w')\| \leq L\|w - w'\| \quad \text{for all } w, w' \in \mathbb{R}^d. \tag{5}$$

Assumption 2 (Bounded stochastic gradient). *The stochastic gradients have bounded variance:*

$$\mathbb{E}[\|g_k^t - \nabla \mathcal{L}_k(w_k^{t,e})\|^2] \leq \sigma^2 \quad \text{for all clients } k \text{ and iterations } t, e. \tag{6}$$

Assumption 3 (Bounded gradient). *The expected squared norm of stochastic gradients is bounded:*

$$\mathbb{E}[\|g_k^t\|^2] \leq G^2 \quad \text{for all } k \text{ and } t. \tag{7}$$

Assumption 4 (Gradient dissimilarity). *There exist constants $\beta^2 \geq 1$ and $\kappa^2 \geq 0$ such that:*

$$\frac{1}{N}\sum_{k=1}^{N} \|\nabla \mathcal{L}_k(w)\|^2 \leq \beta^2 \|\nabla \mathcal{L}(w)\|^2 + \kappa^2 \quad \text{for all } w \in \mathbb{R}^d. \tag{8}$$

Assumption 5 (Bounded weights). *The FedKLEntropy weights are bounded: There exists $\alpha_{\max} \geq \alpha_{\min} > 0$ such that*

$$\alpha_{\min} \leq \alpha_k^t \leq \alpha_{\max} \quad \text{for all } k, t. \tag{9}$$

A.2 Convergence Theorem

Theorem 1 (Convergence of FedKLEntropy). *Under Assumptions 1-5, with learning rate $\eta = \frac{1}{L\sqrt{T}}$ and local steps $E \geq 1$, after T communication rounds, FedKLEntropy satisfies:*

$$\frac{1}{T}\sum_{t=0}^{T-1} \mathbb{E}[\|\nabla \mathcal{L}(w^t)\|^2] \leq \frac{2L(\mathcal{L}(w^0) - \mathcal{L}^*)}{\sqrt{T}} + \frac{L^2}{T}\sum_{t=0}^{T-1} A_t + \frac{\sigma^2}{N\sqrt{T}} + \frac{LE(2E-1)\sigma^2}{2\sqrt{T}} + \frac{3LE^2\kappa^2}{2\sqrt{T}} \tag{10}$$

where $A_t = \mathbb{E}\left[\left\|\sum_{k\in S^t} \alpha_k^t (w_k^{t,E} - w^t) + \eta \nabla \mathcal{L}(w^t)\right\|^2\right]$ captures the alignment error of the FedKLEntropy weighting scheme.

Proof. We provide a detailed convergence proof for FedKLEntropy.

Step 1: Notation and setup

Let $w_k^{t,e}$ be the model of client k at round t and local step e, with $w_k^{t,0} = w^t$. The local update is:

$$w_k^{t,e+1} = w_k^{t,e} - \eta g_k^{t,e} \quad \text{where } g_k^{t,e} \text{ is the stochastic gradient.} \tag{11}$$

The FedKLEntropy global update:

$$w^{t+1} = \sum_{k\in S^t} \alpha_k^t w_k^{t,E}. \tag{12}$$

Step 2: One-round progress

By L-smoothness (Assumption 1):

$$\mathcal{L}(w^{t+1}) \leq \mathcal{L}(w^t) + \langle \nabla \mathcal{L}(w^t), w^{t+1} - w^t \rangle + \frac{L}{2}\|w^{t+1} - w^t\|^2. \tag{13}$$

Taking expectation:

$$\mathbb{E}[\mathcal{L}(w^{t+1})] \leq \mathcal{L}(w^t) + \mathbb{E}[\langle \nabla \mathcal{L}(w^t), w^{t+1} - w^t \rangle] + \frac{L}{2}\mathbb{E}[\|w^{t+1} - w^t\|^2]. \tag{14}$$

Step 3: Decomposing the inner product term

$$\begin{aligned}\mathbb{E}[\langle \nabla \mathcal{L}(w^t), w^{t+1} - w^t \rangle] &= \mathbb{E}\left[\left\langle \nabla \mathcal{L}(w^t), \sum_{k\in S^t} \alpha_k^t (w_k^{t,E} - w^t) \right\rangle\right] \\ &= -\eta \|\nabla \mathcal{L}(w^t)\|^2 + \mathbb{E}\left[\left\langle \nabla \mathcal{L}(w^t), \sum_{k\in S^t} \alpha_k^t (w_k^{t,E} - w^t) + \eta \nabla \mathcal{L}(w^t) \right\rangle\right].\end{aligned} \tag{15}$$

Using Young's inequality:

$$\mathbb{E}\left[\left\langle \nabla\mathcal{L}(w^t), \sum_{k\in S^t} \alpha_k^t (w_k^{t,E} - w^t) + \eta\nabla\mathcal{L}(w^t)\right\rangle\right]$$
$$\leq \frac{\eta}{2}\|\nabla\mathcal{L}(w^t)\|^2 + \frac{1}{2\eta}\mathbb{E}\left[\left\|\sum_{k\in S^t} \alpha_k^t (w_k^{t,E} - w^t) + \eta\nabla\mathcal{L}(w^t)\right\|^2\right]. \tag{16}$$

Combining (15) and (16):

$$\mathbb{E}[\langle \nabla\mathcal{L}(w^t), w^{t+1} - w^t\rangle] \leq -\frac{\eta}{2}\|\nabla\mathcal{L}(w^t)\|^2 + \frac{1}{2\eta}A_t. \tag{17}$$

Step 4: Bounding the second moment term

$$\mathbb{E}[\|w^{t+1} - w^t\|^2] = \mathbb{E}\left[\left\|\sum_{k\in S^t} \alpha_k^t (w_k^{t,E} - w^t)\right\|^2\right]. \tag{18}$$

Using the bounded weights assumption (Assumption 5) and standard FL analysis techniques:

$$\mathbb{E}[\|w^{t+1} - w^t\|^2] \leq 3\eta^2E^2\alpha_{\max}^2\|\nabla\mathcal{L}(w^t)\|^2 + 3\eta^2E\alpha_{\max}^2\sigma^2 + 3\eta^2E^2\alpha_{\max}^2\kappa^2 + O(\eta^3). \tag{19}$$

Step 5: Combining the bounds

Substituting (17) and (19) into (14):

$$\mathbb{E}[\mathcal{L}(w^{t+1})] \leq \mathcal{L}(w^t) - \frac{\eta}{2}\|\nabla\mathcal{L}(w^t)\|^2 + \frac{1}{2\eta}A_t$$
$$+ \frac{3L\eta^2E^2\alpha_{\max}^2}{2}\|\nabla\mathcal{L}(w^t)\|^2 + \frac{3L\eta^2E\alpha_{\max}^2\sigma^2}{2} + \frac{3L\eta^2E^2\alpha_{\max}^2\kappa^2}{2}. \tag{20}$$

Rearranging and taking total expectation:

$$\frac{\eta}{2}\left(1 - 3L\eta E^2\alpha_{\max}^2\right)\mathbb{E}[\|\nabla\mathcal{L}(w^t)\|^2]$$
$$\leq \mathbb{E}[\mathcal{L}(w^t) - \mathcal{L}(w^{t+1})] + \frac{1}{2\eta}A_t + \frac{3L\eta^2E\alpha_{\max}^2\sigma^2}{2} + \frac{3L\eta^2E^2\alpha_{\max}^2\kappa^2}{2}. \tag{21}$$

Step 6: Summing over rounds and choosing learning rate

Summing from $t = 0$ to $T-1$ and choosing $\eta = \frac{1}{L\sqrt{T}}$ such that $3L\eta E^2\alpha_{\max}^2 \leq \frac{1}{2}$:

$$\frac{1}{4L\sqrt{T}}\sum_{t=0}^{T-1}\mathbb{E}[\|\nabla\mathcal{L}(w^t)\|^2]$$
$$\leq \mathcal{L}(w^0) - \mathcal{L}^* + \frac{L\sqrt{T}}{2}\sum_{t=0}^{T-1}A_t + \frac{3E\alpha_{\max}^2\sigma^2}{2L\sqrt{T}} + \frac{3E^2\alpha_{\max}^2\kappa^2}{2L\sqrt{T}}. \tag{22}$$

Multiplying both sides by $\frac{4L\sqrt{T}}{T}$:

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}[\|\nabla\mathcal{L}(w^t)\|^2] \leq \frac{4L(\mathcal{L}(w^0) - \mathcal{L}^*)}{\sqrt{T}} + \frac{2L^2}{T}\sum_{t=0}^{T-1}A_t + \frac{6E\alpha_{\max}^2\sigma^2}{T} + \frac{6E^2\alpha_{\max}^2\kappa^2}{T}. \tag{23}$$

This completes the proof of Theorem 1. □

A.3 Discussion

The convergence rate of FedKLEntropy is $O(1/\sqrt{T})$, which matches the standard rate for FL algorithms. The term A_t captures the effectiveness of the entropy-based weighting scheme. When the FedKLEntropy weights well-align client updates with the global gradient direction, A_t is small, leading to faster convergence.

The proof demonstrates that FedKLEntropy converges to a stationary point under realistic assumptions, providing theoretical grounding for the empirical results shown in the main paper.