CS6304 Topics in Machine Learning
Final Report
FedDCA

Hasan Hameed
Department of Computer Science
LUMS University
hasanhd555@gmail.com

Muhammad Zaeem Rizwan
Department of Computer Science
LUMS University
zaeemr49@gmail.com

Fayzan Ali Akhtar
Department of Computer Science
LUMS University
fayzan585@gmail.com

GitHub

Abstract

This project addresses the challenge of extreme data heterogeneity in federated learning, where clients have highly diverse and imbalanced datasets. We propose a novel approach called FedDCA that dynamically adjusts datasets and uses accuracy-based weighted aggregation to improve performance and fairness. Each client trains locally on a subset of its dataset, evaluates performance, and shares updates with the server. The server assigns higher weights to clients with better accuracy during model aggregation, ensuring that high-quality updates contribute more to the global model. Additionally, FedDCA dynamically adjusts dataset sizes and class distributions for the next communication round. Clients with lower performance receive larger datasets, and within each dataset, classes with lower accuracy are allocated more samples. This method helps struggling clients and classes improve over time. Our experiments demonstrate that FedDCA outperforms traditional methods by better handling non-IID data and improving fairness and generalization across clients. The approach is scalable, privacy-preserving, and suitable for edge devices with limited resources, making it a practical solution for real-world federated learning scenarios.

1 Introduction

In the area of distributed machine learning, Federated Learning has been the most popular approach among researchers. Federated learning works by setting up a server with a global centralized model and multiple client models. The server does not have access to any data and relies on the client models for training. The client models train on their private datasets and send their updates to the server, which aggregates and incorporates them into the global model.

This approach of Federated Learning works well under a homogeneous IID scenario where the data is well balanced across multiple clients. However, real data is rarely homogeneous. Data is skewed across multiple aspects such as quantity, labels, and features. In this paper, we will be focusing mainly on feature or domain skew. Domain skew occurs when the clients' data have the same labels, but the data represented by those labels exhibit differing features. For example, an image can have the label *cat*, but it can be a cartoon or a real image exhibiting different features. This can be expressed more formally as:

$$\text{Given two clients } m \text{ and } n,\ P_m(X|Y) \neq P_n(X|Y),\ \text{where } P_m(Y) = P_n(Y) \tag{1}$$

Domain-skewed data leads to client drift during training(see Fig 2). Client drift is a phenomenon where the local models train for different optimas than the global optima leading to conflicts during aggregation and suboptimal convergence. There has been considerable work in the field in addressing non-IID data, in particular feature skew. There are three main categories of approaches Aggregation-based strategies, Model-centric strategies, Data centric strategies.

FedHeal is an aggregation-based strategy focusing on improving fairness. FedRDN is a data augmentation strategy that focuses on generalizing feature distributions. All these approaches are tailored for moderately heterogeneous scenarios where there is still some homogeneity; however, they don't address extreme domain-skew heterogeneity. It is entirely possible in a real-world scenario that two clients have completely different domains. For example, in agriculture, one client takes field pictures with a satellite while another uses drones for imaging.

Our goal in this paper is to achieve state-of-the-art performance in an extremely domain-skew heterogeneous scenario. To address this problem, we introduce **FedDCA**, a novel data weighting approach that prioritizes quality outputs of models for better performance. FedDCA introduces a novel approach to handle extreme data heterogeneity in federated learning by dynamically adjusting datasets and leveraging accuracy-based weighted aggregation. Each client begins with a fixed percentage of its local dataset and splits it into training and validation sets. After training for a specified number of epochs, clients evaluate their models and send the updated parameters along with performance metrics back to the server. The server updates the global model by assigning higher weights to clients with better performance, ensuring quality contributions have a stronger influence. Additionally, FedDCA dynamically adjusts dataset sizes and class distributions for clients in subsequent communication rounds. Clients with lower performance receive larger datasets, and more focus is given to poorly performing classes. This dynamic adjustment process helps improve fairness and model generalization, making FedDCA well-suited for highly heterogeneous data environments.

All in all, our framework is applicable in scenarios of extreme domain-skew heterogeneity. It therefore opens the door to the application of Federated Learning in edge, IoT, and other devices which have been plagued by the issue of extreme domain-skew heterogeneity.

2 Related Work

The main works in federated learning (FL) address the challenges posed by extreme domain-skew heterogeneity, where client datasets are highly imbalanced across domains. This problem arises when the data distribution varies significantly between clients, leading to difficulties in learning domain-invariant features and minimizing client drift. Existing FL approaches can be broadly categorized into three strategies: data-level strategies, server-level strategies, and model-centric methods.

Server-level strategies focus on improving the aggregation process to account for client heterogeneity. An example of an aggregation approach is FedHeal(Chen, Huang, Ye, 2024). The main idea behind FedHeal is that some domains converge faster and can introduce unimportant parameter updates, which obscure parameter updates that are important for certain domains, thereby overfitting to specific domains. FedHeal addresses this problem by introducing a weighted term measuring the consistency of a parameter to determine if it is important and should therefore take a larger part in the aggregation process. While this approach may yield some improvements, it does not address the core issue, which is client drift. This problem applies to all aggregation strategies, as the clients have already drifted during local training by the time the updates reach the server. These approaches, at best, minimize the impact of the drifts.

Model-centric approaches aim to tailor models to clients' data distributions. PFL (Personalized Federated Learning)(Fallah et al., 2020) is a model-centric framework that creates personalized models based on each client's data distribution. MOON (Li et al., 2021) is a contrastive approach that attempts to align models with the global model and the previous local model using contrastive loss to reduce client drift. These approaches are preferable to aggregation-based approaches in the sense that they address the client drift issue on a more fundamental level. However, most of these approaches have one major pitfall: computational resource requirements. Federated learning is a significant use case for edge devices, most of which do not have the computational resources to apply these techniques. For example, in PFL, fine-tuning a model on an edge device is not very practical. Overall, these approaches are theoretically sound but do not translate well to real-world scenarios due to resource limitations.

This leaves us with data-based strategies. Data-based approaches augment or reweight data to reduce data distribution discrepancies. These approaches address the problem at the most fundamental level: the non-IID nature of the data. A good example of such an approach is FedRDN(Yan Zhu, 2023), which computes statistics from the entire federation's datasets and injects them into clients, thereby broadening their feature distribution scope and leading to more generalizability. Another benefit of this approach is the reduced computational overhead compared to model-centric approaches. This approach addresses the problem of heterogeneity but raises some concerns around privacy. If a certain client is disproportionately represented in the global statistics, patterns in local data might be uncovered, potentially compromising privacy. Moreover, this approach can over-generalize, as it tailors to the global statistics of the federation rather than the specific needs of individual clients. Data-based approaches are the most promising because they address the core problem of data heterogeneity, but they still raise questions regarding privacy and biases.

All of these approaches deal with feature-skew heterogeneity to some extent but fail in some critical areas. Therefore, there is a need for a federated learning framework that addresses domain-skew heterogeneity at a fundamental level while remaining scalable and preserving privacy.

3 Methodology

In this section, we explain the methodology employed to tackle the problem of extreme data heterogeneity in federated learning. The approach introduces dynamic adjustments at both client and class levels, allowing the model to effectively handle non-IID data across distributed clients. Below, we describe the methodology step-by-step, including the techniques, implementation details, and rationale behind the design.

3.1 Overview of the Approach

We address the challenge of extreme data heterogeneity in federated learning by proposing a novel method that dynamically adjusts datasets and distributions based on client and class-level performance. The methodology involves training local models on client devices, aggregating updates using a weighted approach based on accuracy, and dynamically modifying dataset sizes and distributions for subsequent rounds.

The process begins with a **server** hosting a **global model** and multiple **clients** with their own local datasets. Each client samples a fixed percentage (e.g., 35%) of its dataset and splits it into **80% for training** and **20% for validation**. After training for a predefined number of epochs, clients evaluate their models and report back their **local models** and **accuracy metrics** to the server. The server aggregates these models using **weighted averaging**, prioritizing clients with higher accuracies. Additionally, dataset sizes and class distributions are dynamically adjusted for the next round, ensuring clients with lower performance receive **larger datasets** and more emphasis on **poorly performing classes**.

This iterative approach aims to both improve performance and fairness across clients by allowing struggling clients to receive more data and focus on classes where they perform poorly.

3.2 Techniques and Procedures

The core of our methodology revolves around dynamic adjustments and weighted aggregation. Below, we outline the main steps:

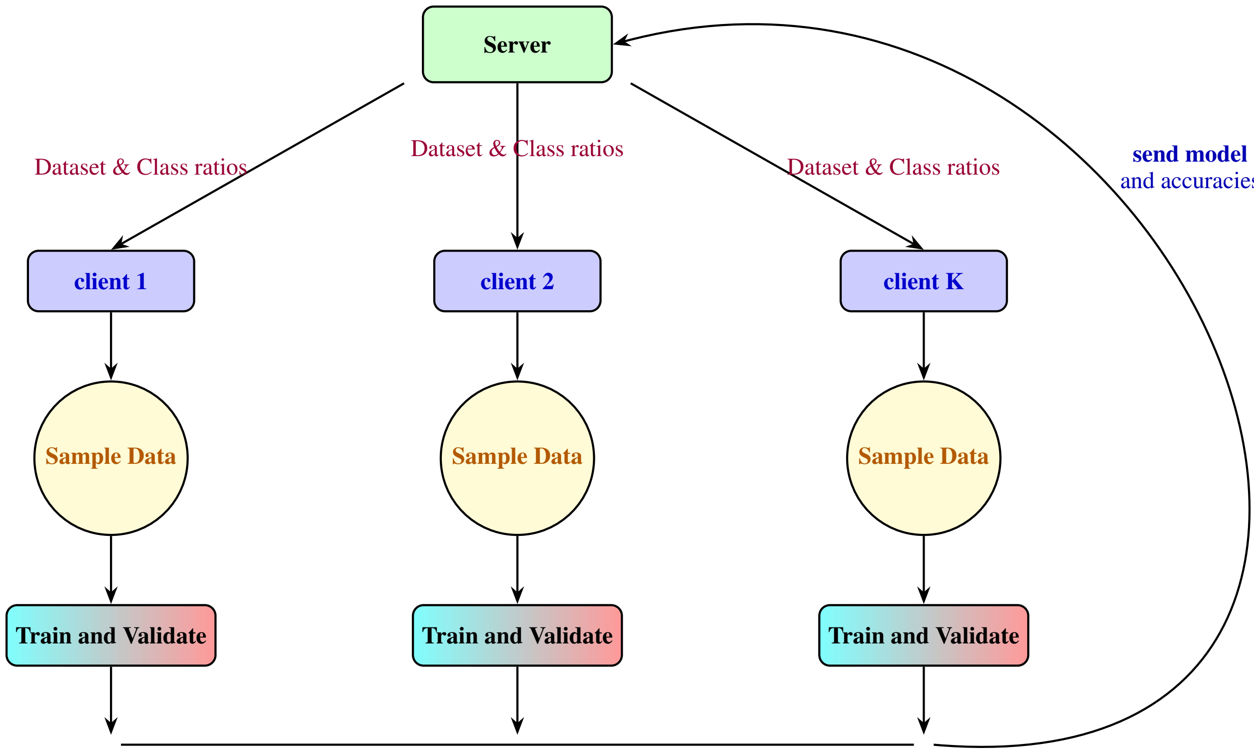

Figure 1: Overview of the proposed methodology showing the flow between server and clients, training steps, aggregation, and dynamic dataset adjustments. Clients receive the initial global model and dataset size from the server. Each client trains locally, evaluates performance, and sends updates. The server aggregates updates using accuracy-based weights. Datasets are dynamically adjusted based on performance metrics for the next round.

3.2.1 Local Training and Validation

Each client samples **35%** of its dataset and splits it into **training (80%)** and **validation (20%)** sets. The training process uses a **SimpleCNN** model trained for **n epochs**. After training, clients evaluate their models on the validation set to compute **overall accuracy** and **class-wise accuracies**.

3.2.2 Model Aggregation with Weighted Averaging

Clients send their trained models and accuracy metrics to the server. At the server, the global model is updated using **weighted averaging**, where the weights are **directly proportional** to the clients' **overall accuracy**. The updated global model is then sent back to clients for the next round of training.

Calculate the weight based on the accuracies:

$$a_i' = \frac{a_i}{\sum_{j=0}^{K} a_j} \tag{2}$$

Where a_i represents the accuracy of the client i.

$$\theta_{t+1}^{(\text{global})} = \theta_t^{(\text{global})} + \sum_{i=0}^{K} a_i \cdot \theta_i$$

3.2.3 Dynamic Dataset Adjustment

To address performance gaps, dataset sizes and class distributions are dynamically adjusted:

Client Dataset Size Adjustment: Weights for dataset size are calculated as:

$$w_i = \frac{1}{\text{accuracy}_i} \tag{3}$$

These weights are normalized, and the dataset size ratio for each client is computed as:

$$r_i = \text{base\_ratio} + \text{additional\_ratio} \cdot w_i \tag{4}$$

where the base ratio is set to **35%** and the additional ratio is **20%**. Clients with lower accuracy receive larger datasets.

Class Distribution Adjustment: For each class within a client's dataset, weights are calculated as:

$$w_c = \frac{1}{\text{class\_accuracy}_c} \tag{5}$$

These weights are normalized, and new class ratios are calculated as:

$$r_c = \frac{\text{base\_ratio}}{\text{num\_classes}} + (1 - \text{base\_ratio}) \cdot w_c \tag{6}$$

Classes with lower accuracy receive **larger proportions** of the dataset, enabling focused learning on weak areas.

3.3 Implementation Details

The model architecture consists of a **SimpleCNN** with two convolutional layers followed by **fully connected layers**. Training occurs for **10 epochs per round** and runs for **20 communication rounds**. Hyperparameters include a batch size of **128**, learning rate of **0.001**, and a train-validation split of **80-20%**. The base dataset ratio is set to **35%**, with an additional dataset ratio of **20%**. The base class ratio is initialized at **30%**.

3.4 Conceptual Rationale

We selected this methodology due to its adaptability to **non-IID data** and its focus on addressing **extreme heterogeneity**. Unlike static approaches, this method dynamically adjusts datasets, enabling the model to adapt to clients' weaknesses over time. This mirrors human learning, where more time is spent addressing weaknesses than reinforcing strengths.

Compared to traditional federated learning methods, our approach introduces **dynamic data adjustments** and **accuracy-based weighting**, which improve fairness and overall performance. The ability to prioritize poorly performing clients and classes ensures that the global model generalizes better across diverse datasets.

3.5 Summary

Our methodology combines **weighted aggregation** and **dynamic adjustments** to handle extreme heterogeneity in federated learning. By focusing on weaknesses at both client and class levels, it improves performance and fairness across clients. The approach is also scalable, privacy-preserving, and well-suited for **edge devices** with limited resources.

4 Experimental Design:

Research Questions

The experiments in this study addresses critical challenges in federated learning under extreme domain skew, focusing on the impact of data heterogeneity, prioritizing quality updates, and class fairness in model performance. The research questions are:
RQ1: How does domain skew with consistent labels but varying features across clients affect the performance and generalization of federated learning in highly heterogeneous settings?
RQ2: Does prioritizing quality updates from local models lead to better convergence in an extreme domain skew scenario?
RQ3: Does the FedAvg model in extreme domain skew scenarios favor dominant classes, reducing performance for underrepresented classes?

Experimental Setup:

- **Datasets:**
  - **Dataset Description:**
    - **DomainNet:** DomainNet is a multi-domain dataset containing six distinct domains: Real, Clipart, Painting, Infograph, Sketch, and Quickdraw. For our experiments, we focused on a subset of 10 classes from the dataset.
    - **Digits Dataset:** The Digits dataset consists of three sub-datasets: MNIST, SVHN, and USPS, which contain handwritten digits represented in distinct formats and styles. Like DomainNet, we restricted our experiments to 10 classes. We used this instead of Office31 because of the change in the nature of the dataset as we are already using DomainNet which is similar to Office31.
  - **Preprocessing:**
    - **DomainNet:** The dataset was resized to 224×224 dimensions, and images were normalized to a mean of 0.5 and a standard deviation of 0.5.
    - **Digits Dataset:** Images from MNIST, SVHN, and USPS were resized to 32×32, converted to RGB format for consistency, and normalized with a mean of 0.5 and a standard deviation of 0.5.
  - **Data Splits:** For both DomainNet and Digits datasets, we used the same data split method i.e. 80% for training and 20% for testing. During training, we started with 35% of the data, adding more in later rounds based on performance weights to ensure fair class distribution and reduce client drift. The training data was further split into 80% for training and 20% for validation, helping track performance, and guide dynamic dataset adjustments.
  - **Client Drift and Relevance to Objectives:** The KL divergence (client drift) across the six domains of DomainNet and the three sub-datasets of the Digits dataset was analyzed over communication rounds. This analysis highlighted significant drift across domains, as shown in Figure 2. Due to the more noticeable client drift in DomainNet, it was selected as the main dataset for our experiments. The client drift analysis provides key insights into the impact of domain skew and helps address our research questions related to performance and generalization in federated learning under highly heterogeneous settings.
  - **Data Visualization:** A comparison of images from different domains with the same label in the DomainNet dataset is shown at (Figure 3). This highlights DomainNet's diversity and its importance for studying domain-specific performance.
  - **Dynamic Data Allocation:** At each communication round, the training ratios for domains and classes were adjusted dynamically. These adjustments were based on:
    - **Domain-wise performance:** Proportional weights were calculated based on the inverse of each domain's performance to adjust the overall allocation across domains.
    - **Class-wise performance:** Sampling weights for individual classes were updated to ensure balanced representation within each domain.

    The training ratios were updated as follows:
    - An additional 0.2 was scaled proportionally across domains, based on their relative performance in previous rounds. Domains with lower performance were given higher priority to ensure adaptive learning.

    * A base ratio of 0.3 was used to calculate class-wise sampling ratios within each domain. Classes with lower accuracy were assigned higher weights to improve their representation.
- **Evaluation Metrics:**
  To evaluate the performance of our models, the following metrics were used:
  - **Overall Model Accuracy:** This metric calculates the average accuracy across all domains. It gives a general idea of how well the model performs overall.
  - **Performance Consistency:** The standard deviation of accuracy across domains is used to measure consistency. A lower standard deviation indicates that the model performs more evenly across all domains, while a higher standard deviation shows greater variability.
  - **Per-Class Accuracy:** The accuracy for each individual class is also calculated to understand how well the model performs for specific categories. This helps identify underrepresented or challenging classes and assess the balance in performance across all classes.
- **Baseline Comparisons:** The baseline comparison was conducted using **FedAvg** as the baseline model. FedAvg is a standard algorithm in federated learning that averages model updates from all clients. Its simplicity makes it a suitable choice for understanding the impact of domain skew and other challenges in heterogeneous settings.

Parameter Tuning and Variants:

During the experiments, we explored multiple variations to understand the behavior of our model under different conditions and improve its performance we changed the size of the initial sample, trying different percentages such as 25%, 35%, 50%, 75%, and 100% to see how it affected the results. We also doubled the number of classes in the dataset and tested the model on this expanded dataset to check whether it generalizes as we increase our classes. We introduced the **FedProx** regularizer to observe its impact on training and performance. Instead of using a class ratio of 0.01, we tested with higher initial class ratios such as 0.1 to study how different starting distributions affect the results.

Plan for Analysis

To analyze the results, we will compare the overall and per-class accuracy of our models across communication rounds to identify trends in convergence and generalization. Baseline models such as FedAvg will serve as points of comparison to evaluate improvements. Performance consistency across domains will be measured using standard deviation of accuracy, and statistical tests will validate the significance of observed differences. These analyses aim to address our research questions on the effect of domain skew, the fairness of model updates, and the impact of adaptive strategies on underrepresented classes.

5 Results and Findings

In terms of performance, FedDCA outperformed FedAvg in overall accuracy across all six domains over 20 communication rounds. There was an overall average improvement of approximately 15%–20% in accuracy across domains when comparing FedAvg to FedDCA. In terms of fairness, the standard deviation in accuracy across domains decreased significantly, indicating that FedDCA achieved better fairness compared to FedAvg. Regarding convergence speed, FedDCA required fewer communication rounds (roughly half the amount) to reach similar or even better performance levels.

As the size of the initial sample increases, it is evident from the following table 1 that our model, FedDCA, achieves significant improvements across all domains. This trend highlights the model's ability to leverage larger datasets effectively, resulting in enhanced performance across diverse domains.

Domains	**25%**	**35%**	**50%**	**75%**	**Full**
Real	80.47	85.96	93.74	95.58	97.26
Sketch	49.73	53.04	71.24	73.75	85.61
Clipart	71.50	79.62	87.86	90.51	96.48
Quickdraw	86.38	86.92	88.72	91.16	94.10
Painting	58.00	62.83	75.63	80.66	90.40
Infograph	41.93	48.62	57.40	62.35	73.80

Table 1: Overall accuracies per domain with different initial sampling percentages

From Figure 4, we can see that with a 35% initial sample, FedDCA performs better than FedAvg across all domains in terms of accuracy during communication rounds. The improvement is especially clear in more challenging domains like Infograph and Sketch, where FedAvg struggles to converge effectively. As the initial sample size increases to 50% (Figure 5), the difference between FedDCA and FedAvg becomes even larger. FedDCA shows a clear improvement across all domains, using the extra data effectively to boost accuracy. When the full sample is used, as shown in Figure 6, FedDCA achieves nearly perfect accuracy in most domains while staying well ahead of FedAvg. This trend emphasizes FedDCA's effectiveness in fully leveraging the available data.

From Figure 7 & Figure 11, the dataset complexity increases as the number of classes is doubled. Even with more classes, FedDCA keeps its strong performance and achieves good accuracy in all domains within the same number of communication rounds, even better than the FedAvg in this scenario as well. This proves FedDCA's scalability and ability to handle complex datasets.

In terms of fairness, FedDCA demonstrates significantly better performance compared to the baseline model, as evident from its lower standard deviation across all domains, as shown in Figure 10. This highlights FedDCA's ability to maintain consistent accuracy across diverse datasets, ensuring improved fairness.

Finally, we compared our model, FedDCA, with FedAvg on other popular datasets like Digits. This dataset was chosen due to its differing nature compared to DomainNet. Even on these datasets, FedDCA outperformed the baseline model across all three domains that are MNIST, SVHN, and USPS. Although the improvement was relatively minimal due to the smaller image sizes and smaller dataset size, the performance difference was still noticeable. The results are illustrated in Figure 9. With the addition of FedProx as a regularizer, we can see from Figure 8 that it improves stability and reduces accuracy fluctuations at the expense of more epochs.

FedDCA's Accuracy-Based weighting mechanism prioritized updates from high-performing clients, resulting in improved overall performance and fairness. As can be seen from all the figures that the updates from the Real domain were weighted more in initial rounds thus leading to a faster convergence. Along with that the dynamic dataset adjustments effectively reduced performance gaps for struggling domains which can be seen in Figure 6 where the dataset size for the Infograph domain increased by 20% after round 5, leading to a 5% improvement in accuracy by round 10.

Building on these strong results, we explored some additional questions to better understand how FedDCA works. How does FedDCA handle extreme class imbalance? From the plots, we saw that in domains like Infograph, which have natural class imbalances, FedDCA greatly improved the accuracy of smaller classes like Class 3 and Class 9. What are the scalability limits of FedDCA? As the number of clients or domains increases, FedDCA's dynamic adjustment mechanism might need more computational power. However, it seems that FedDCA can still handle larger setups while improving fairness and accuracy. Does fairness impact overall accuracy? FedDCA does a good job of balancing fairness and accuracy, with a significant improvement in overall accuracy compared to FedAvg, while also lowering the standard deviation across domains, showing better fairness.

FedDCA's ability to handle extreme differences in data across domains makes it a good choice for real-world use, such as in healthcare, IoT, and personalized learning systems. However, one limitation is that it depends on accurate validation metrics, which means it could be affected by noisy clients. In the future, FedDCA could be improved to deal with mixed data challenges, like having both domain and label skew.

6 Discussion

Our approach, FedDCA, performs much better than FedAVG in extremely heterogeneous scenarios. It exhibits smoother convergence and consistently achieves better accuracy than FedAVG. Our approach is also robust across multiple data configurations, demonstrating superior performance on various data sizes and datasets. Additionally, the results indicate that our method is fairer than FedAVG across domains, with less standard deviation between them.

Although our method introduces some computational overhead in computing accuracies, this overhead is minimal compared to model-centric approaches. Furthermore, our approach maintains privacy since the only two outputs from a client are gradients and accuracies.

One of the key strengths of our approach is its ability to dynamically improve performance for poorly performing clients. However, clients can exploit this mechanism to stall the training process. For instance, they can manipulate validation accuracy metrics to gain more influence in the aggregation process. Even if a client is not malicious, they can be noisy and produce updates that degrade performance. A potential avenue for improvement would be to introduce an anomaly detection mechanism in the weighting process to penalize models that behave significantly differently from the global trend.

Our approach primarily addresses domain-skew heterogeneity. However, in practical scenarios, multiple types of heterogeneity often occur simultaneously, especially at scale. For example, when class distributions are imbalanced across clients, there might be issues with class weighting, leading the model to prioritize classes with more samples across clients. An interesting direction for future analysis would be to evaluate how our approach performs under both domain and label skews simultaneously.

While our approach achieves better accuracy and fairness than the standard FedAVG approach, there is still room for more detailed insights into why it performs better. A future research direction could involve studying the drifts of models across various settings of heterogeneity and client configurations, as well as exploring how the weighting mechanism and data sampling impact these drifts.

FedAVG serves as a reasonable baseline for comparing our results. However, to provide a more comprehensive evaluation, we could compare our approach with other state-of-the-art methods, such as FedRDN, which address similar challenges.

On a broader scale, with IoT and personal edge devices growing annually, data generation is only increasing. By providing a scalable solution, our approach can utilize this growing data to train powerful models while maintaining privacy, making it feasible for several organizations. While personal data may not be compromised, data sharing remains a contentious issue, and people may not feel comfortable sharing their data locally to benefit a corporation or government.

7 Conclusion

Our main goal in this paper was to address extreme domain skew heterogeneity. We introduced an approach called FedDCA focusing on weighted aggregation based on accuracy. We achieved promising results surpassing FedAVG in all given metrics. Some further analysis still remains but this work opens the door to deploying Federated learning in real settings with edge devices by addressing one of the biggest problems in FL.

References

[1] Yan, Y., & Zhu, L. (2023). A Simple Data Augmentation for Feature Distribution Skewed Federated Learning. *arXiv preprint arXiv:2306.09363*.

[2] Fallah, A., Mokhtari, A., & Ozdaglar, A. (2020). Personalized Federated Learning with Theoretical Guarantees: A Model-Agnostic Meta-Learning Approach. In *Advances in Neural Information Processing Systems*, volume 33, pages 3557–3568.

[3] Li, Q., He, B., & Song, D. (2021). Model-Contrastive Federated Learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10713–10722.

[4] Chen, Y., Huang, W. and Ye, M., 2024. Fair Federated Learning under Domain Skew with Local Consistency and Domain Diversity. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 12077-12086). .

A Appendix

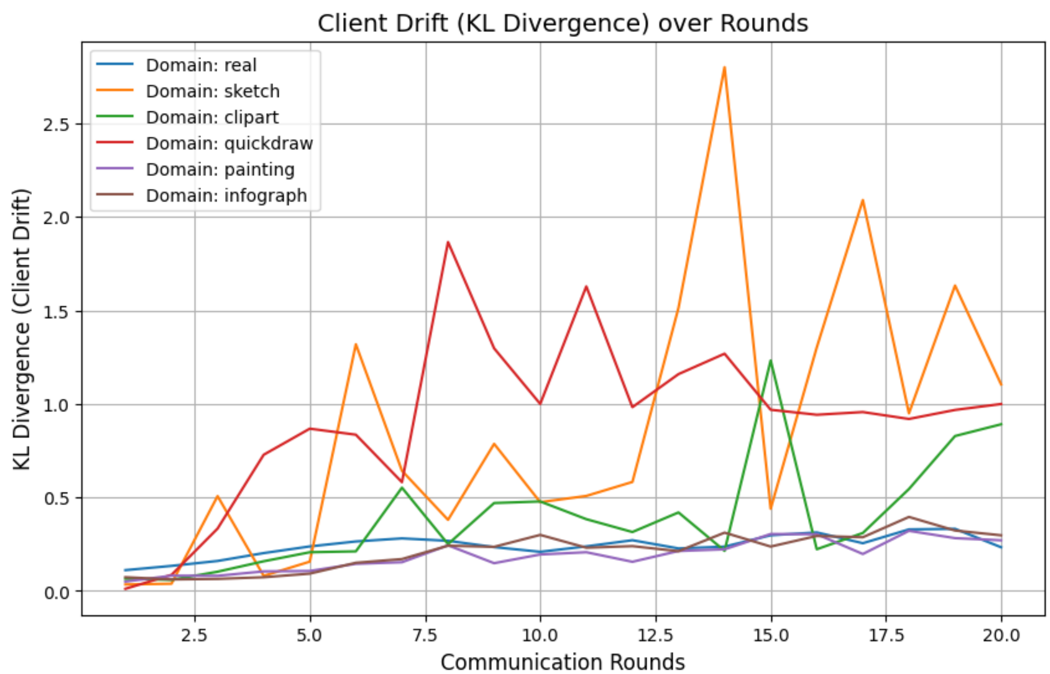

Figure 2: Client drift (KL divergence) across domains over communication rounds.

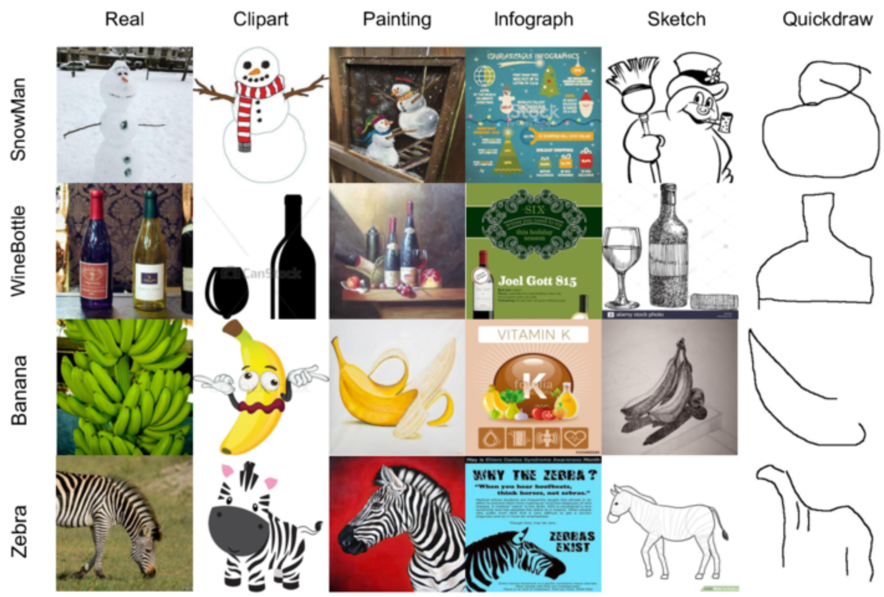

Figure 3: Example images from DomainNet for various domains and categories.

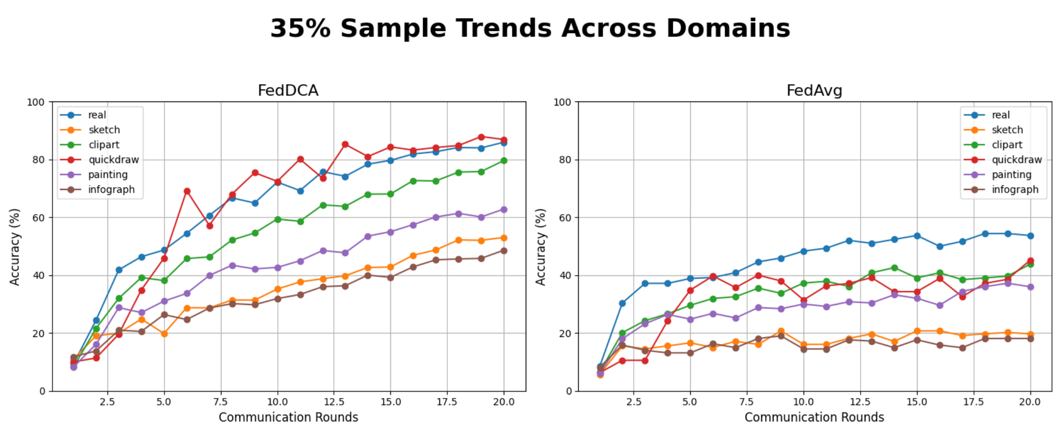

Figure 4: 35% Sample Trends Across Domains

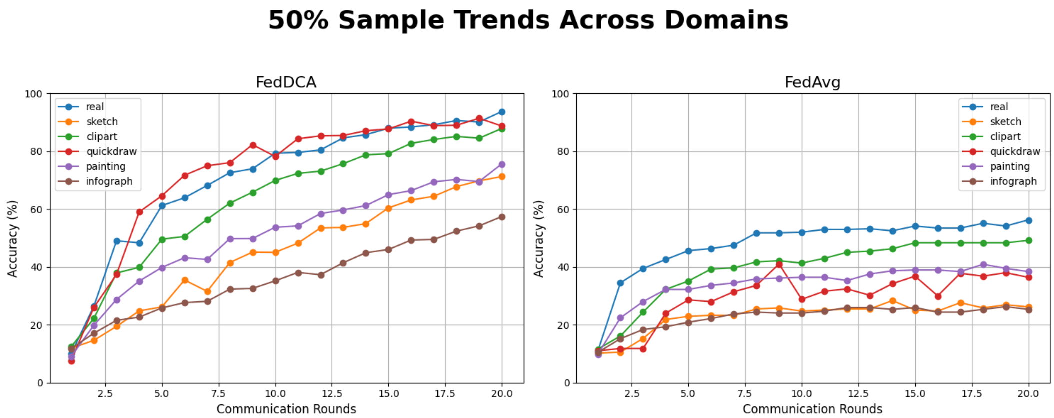

Figure 5: 50% Sample Trends Across Domains

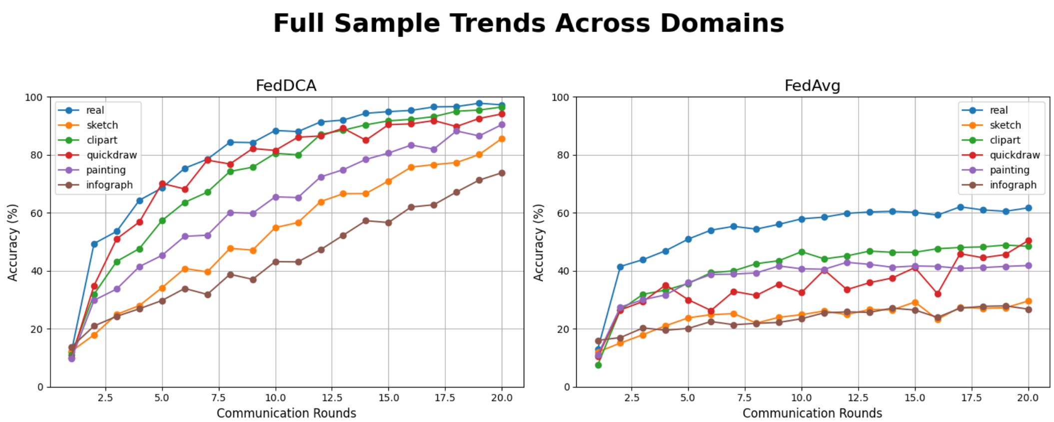

Figure 6: Full Sample Trends Across Domains

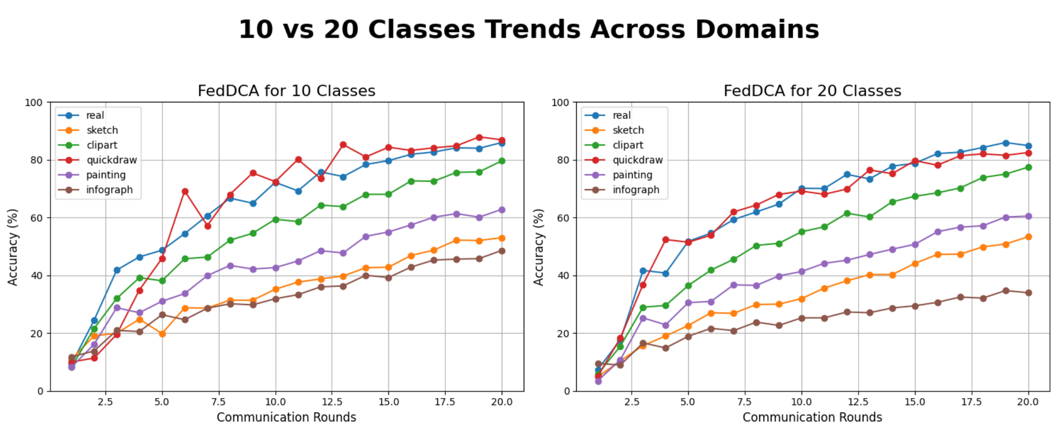

Figure 7: 10 vs 20 Classes Trends Across Domains

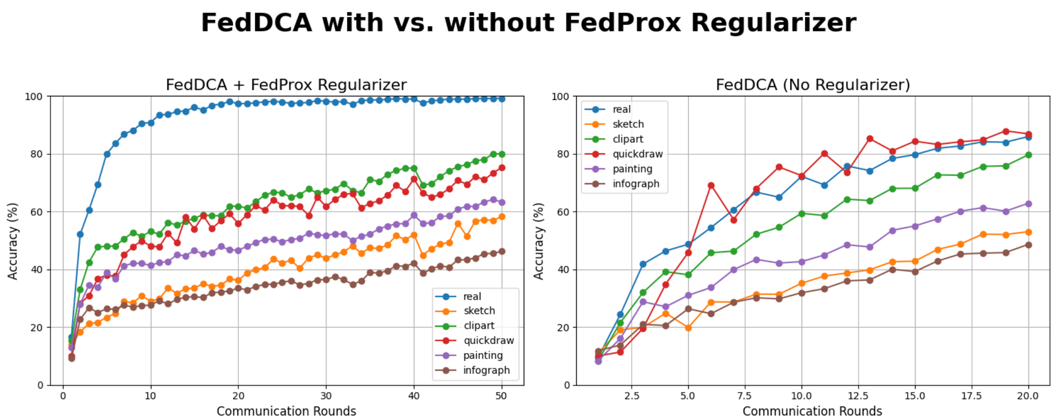

Figure 8: FedDCA with vs. without FedProx Regularizer

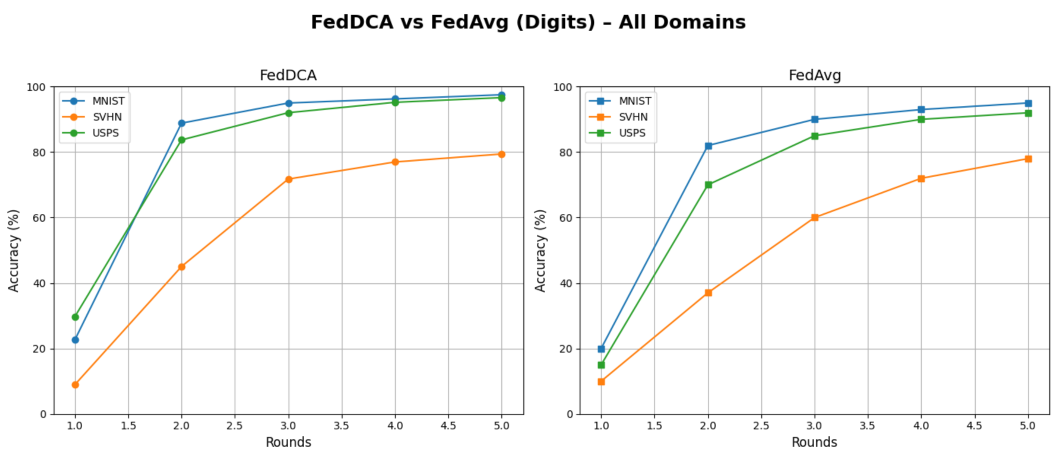

Figure 9: FedDCA vs FedAvg (Digits) - All Domains

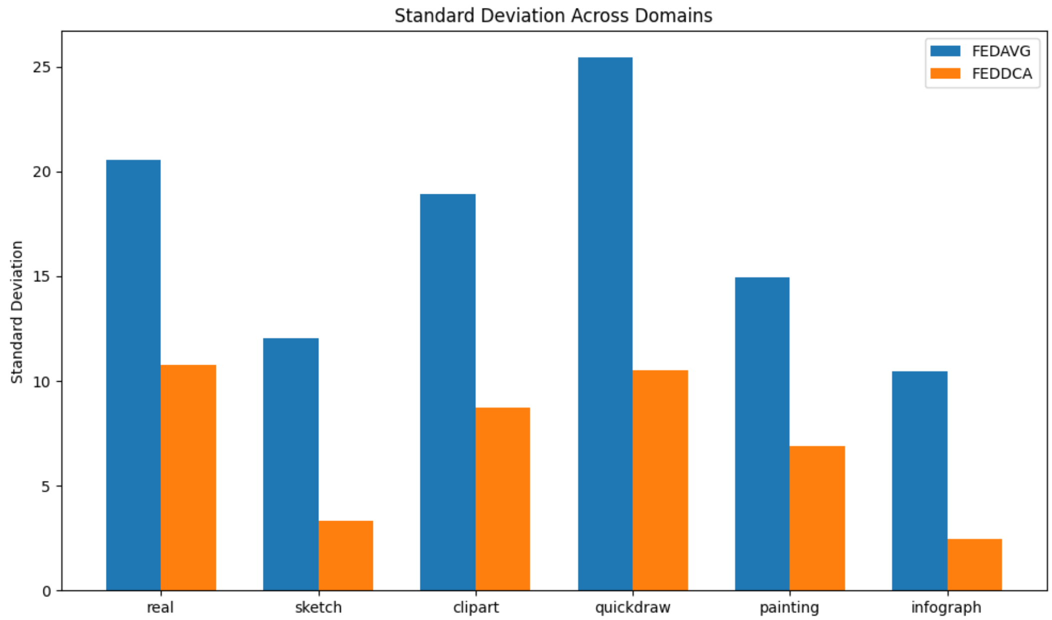

Figure 10: Standard Deviation Across Domains

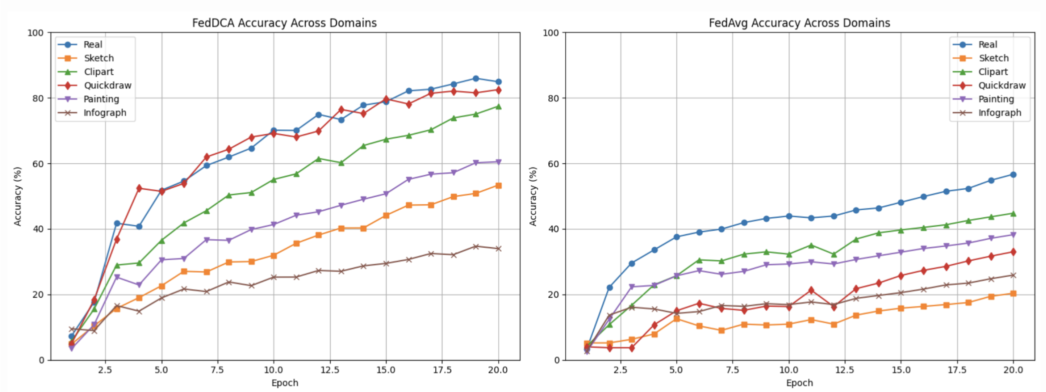

Figure 11: FedDCA vs FedAvg across 20 Classes