Federated Learning

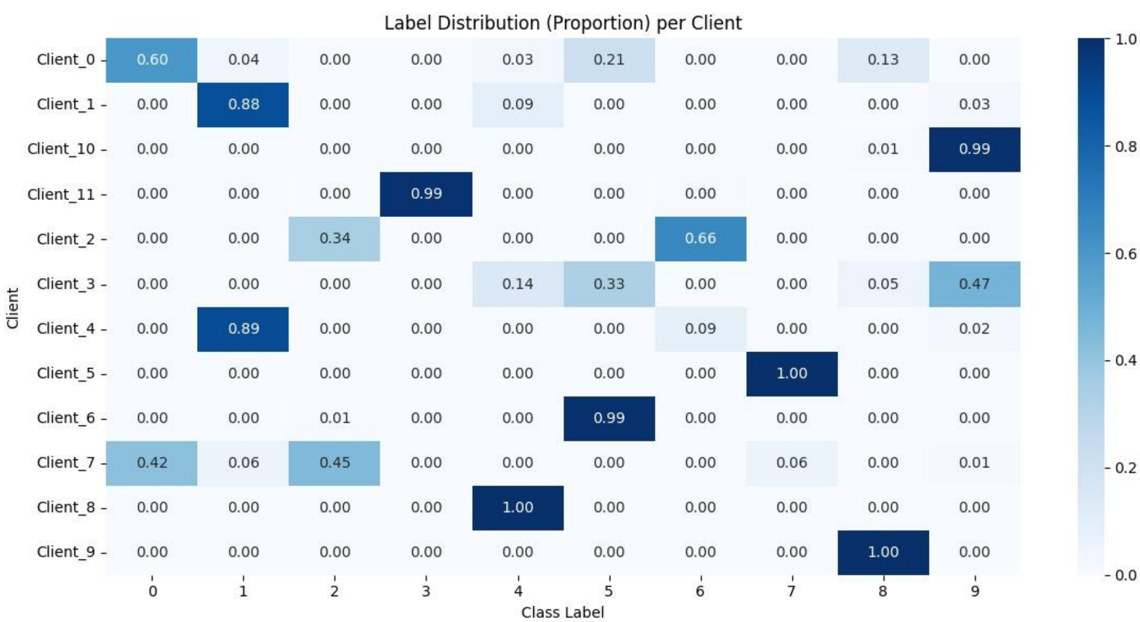
Team Avengers | June 2025

1. Per-Client Data Analysis

In a federated learning (FL) setting, clients are equipped with local datasets that often vary significantly in size and distribution. This heterogeneity poses key challenges to the convergence and generalization ability of global models.

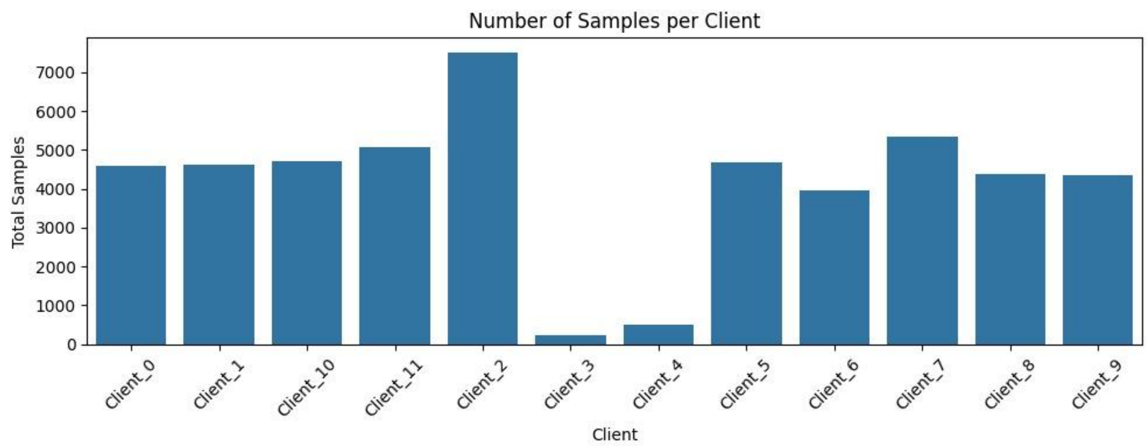
Label Distribution

Each client's dataset was analyzed for class imbalance. The dataset used is CIFAR-10 partitioned using a Dirichlet distribution with α=0.05, which induces high non-IIDness. Bar plots for each client show that:
- Some clients have samples from only 2–3 classes.
- Certain classes are completely missing from some clients' datasets.



Sample Size Per Client

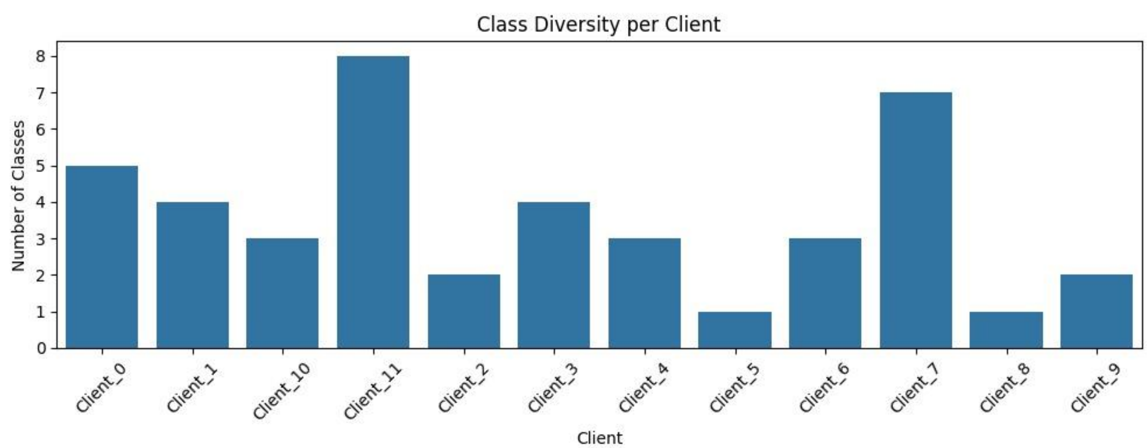
Client sample sizes range significantly, with the smallest client having fewer than 500 samples and the largest having over 2,500. This imbalance can bias the global model toward over-represented clients.



Class Diversity

Using a heatmap of client-class presence:
- Some clients cover as few as 2 classes, while others have moderate diversity.
- Global coverage is preserved, but local diversity is poor.

Conclusion: The data is strongly non-IID and unbalanced, necessitating careful client sampling and aggregation strategies.



2. Client Training Profiling

To understand training behavior under different conditions, we profiled clients locally using batch sizes: 4, 8, 16, 32. The model used was a CNN on CIFAR-10, trained over 100 mini-batches per batch size.

Metrics Recorded

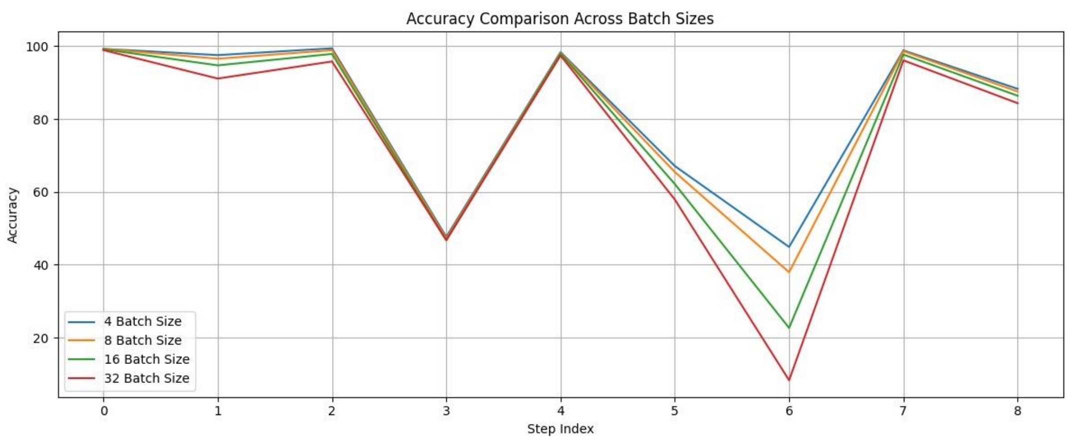
- Minibatch Training Time (seconds)
- Memory Usage (MB), recorded using `top` and PyTorch utilities.

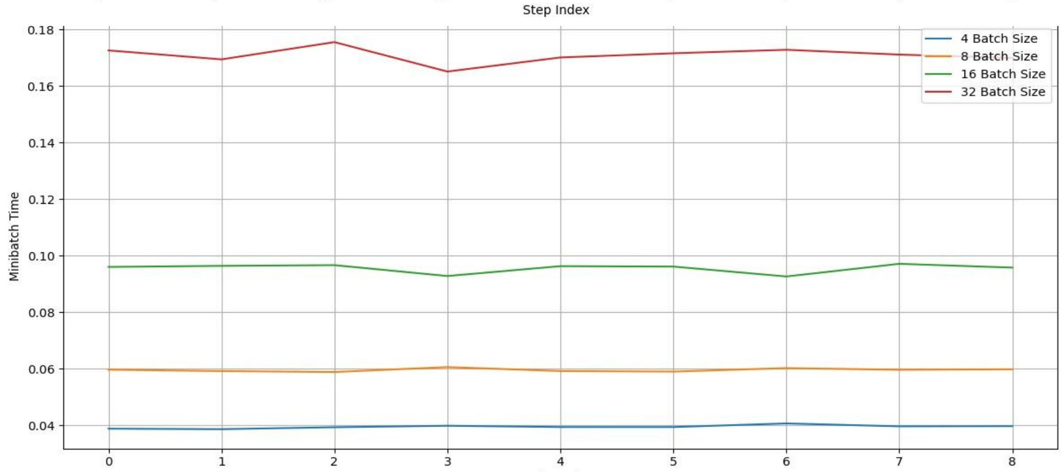
Observations

- Memory Usage: Increased steadily with batch size. Batch size 32 approached GPU memory limits.
- Training Time per Minibatch: Decreased initially but plateaued or even increased slightly for batch size 32 due to I/O and preprocessing overhead.

Batch Size	Accuracy (%)	Minibatch Time (s)	Total Time (s)
4	82.4	0.039	32.3
8	81.1	0.060	24.4
16	78.4	0.096	19.9
32	75.2	0.171	17.7

Conclusion: Batch size 16 offered the best balance of speed and memory usage.





3. Results and Discussion

3.1 Visual Summary

Plots were generated for:
- Class distribution per client
- Heatmaps of class coverage
- Training time and memory usage vs batch size

These plots make the following patterns evident:
- Skewed label distributions
- Sparse class coverage on several clients
- Nonlinear training time behavior with increasing batch size

3.2 Analysis

- Non-IIDness leads to local optima, impacting global convergence.
- Skewed sample sizes make global updates biased toward larger clients.
- Small batches are inefficient; very large ones trigger memory issues and latency.

3.3 Challenges

- Client Data Heterogeneity: Local models overfit their class subset.
- Memory Limitations: Higher batch sizes constrained by GPU VRAM.
- Training Variability: Training time fluctuated due to OS scheduling, I/O delays.

3.4 Solutions & Strategies

- Client Selection: Sampling diverse clients each round.
- Adaptive Aggregation: Weight updates based on effective data size.
- Batch Tuning: Use profiling to dynamically adjust local training hyperparameters.

4. Conclusion and Future Work

This study highlighted the significance of:
- Understanding local data distributions
- Profiling training to identify bottlenecks
- Designing FL strategies robust to client heterogeneity

Key Findings:
- Batch size 16 is optimal for current hardware and dataset size.
- Non-IID distributions significantly affect learning dynamics.

Future Directions:
- Implement class-aware sampling to ensure class balance per round.
- Use meta-learning for personalization.
- Apply gradient noise scaling for client-specific learning rates.