Team Darts

June 15, 2025

1 Per-Client Data Analysis

The analysis of the non-IID partitioned CIFAR-10 dataset reveals significant heterogeneity in data distribution across clients, characterized by two primary forms of imbalance critical for federated learning scenarios.

1.1 Label Distribution and Class Imbalance

The dataset exhibits extreme non-IID characteristics through deliberate partitioning using Dirichlet distribution with $\alpha = 0.05$. This low alpha value creates highly skewed distributions that mirror real-world federated learning environments where data is naturally heterogeneous across participating devices.

Class Imbalance (Label Skew): The analysis identifies severe class distribution skewness across clients. Several clients operate as single-class entities (e.g., client_5 contains only class 7 data, client_8 exclusively holds class 4), while others demonstrate dominant-class scenarios. For instance, client_0 is heavily dominated by class 0 but maintains minor representations of other classes.

Quantity Imbalance (Size Skew): Substantial variation exists in dataset sizes across clients, ranging from fewer than 1,000 samples to over 7,000 samples per client. This size heterogeneity compounds federated learning challenges by creating scenarios where some clients have significantly more influence on model updates.

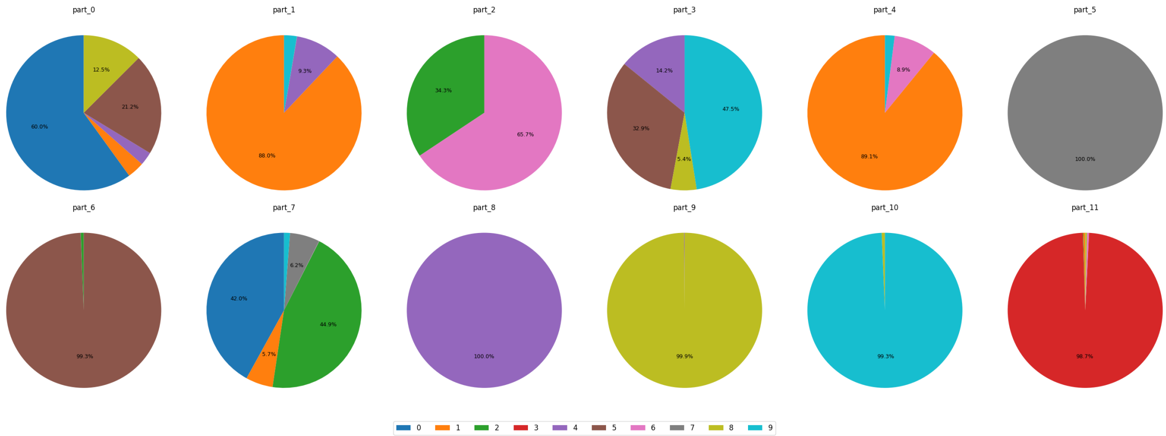
1.2 Client Data Distribution Analysis

The detailed per-client class distribution reveals the extent of data heterogeneity:

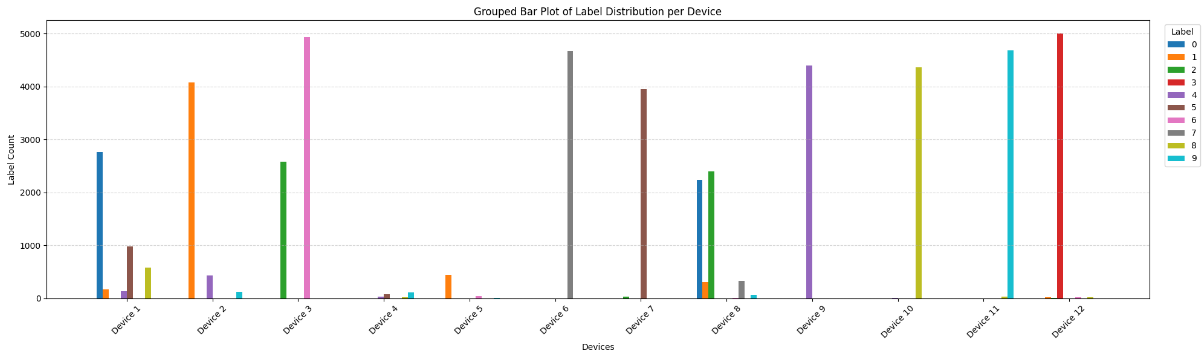
Table 1: Per-Client Data Distribution Summary

Client	Total Samples	Dominant Classes	Distribution Pattern
1	4,607	Class 0 (60%), Class 5 (21%)	Multi-class with strong bias
2	4,632	Class 1 (88%)	Heavily skewed
3	7,513	Class 6 (66%), Class 2 (34%)	Dual-class dominance
4	240	Class 9 (48%), Class 5 (33%)	Smallest client
7	4,668	Class 7 (100%)	Single-class client
9	4,395	Class 8 (100%)	Single-class client

This distribution pattern reflects real-world federated scenarios where different clients may specialize in different data types or have access to limited label diversity. The extreme non-IID nature creates significant challenges for traditional federated averaging approaches.



(a) Class distribution pie charts per client



(b) Per-client class distribution showing label skew

Figure 1: Visualization of non-IID data distribution across clients

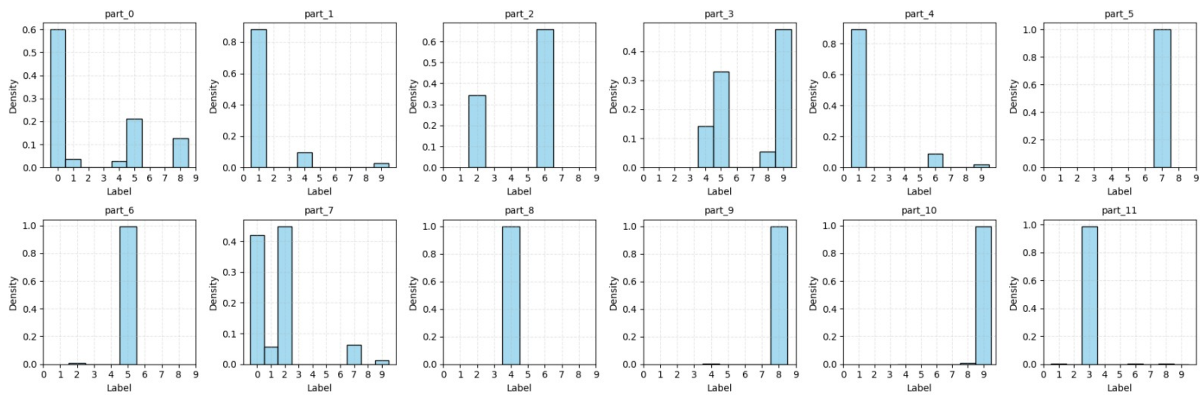


Figure 2: Histogram showing sample count distribution across clients

2 Client Training Performance Profiling

The profiling analysis examines mini-batch training characteristics across different batch sizes to understand computational efficiency and resource utilization patterns in the federated learning context.

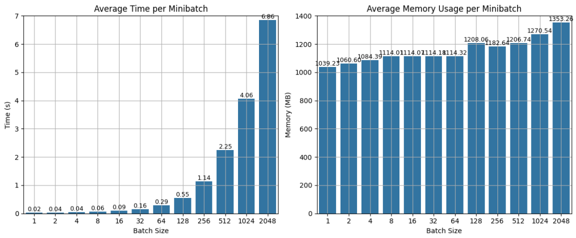
2.1 Variability in Training Metrics

Training Time Variability: Mini-batch training times demonstrate clear correlation with batch size, exhibiting increased duration and higher variability as batch sizes grow. This pattern results from elevated computational loads and system-level factors including I/O operations, memory allocation overhead, and thread scheduling complexities.

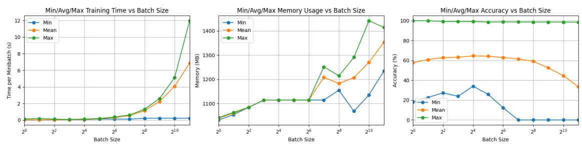
Memory Usage Patterns: Memory consumption follows expected scaling behavior with batch size, showing gradual increases initially followed by steeper growth patterns. The observed variability stems from dynamic memory allocation strategies and background system activities competing for resources.

2.2 Optimal Batch Size Determination

Through comprehensive analysis of the time-memory-accuracy trade-off, **batch size 16** emerges as the optimal choice, balancing computational efficiency, memory optimization, and accuracy preservation while maintaining beneficial stochastic effects for generalization.

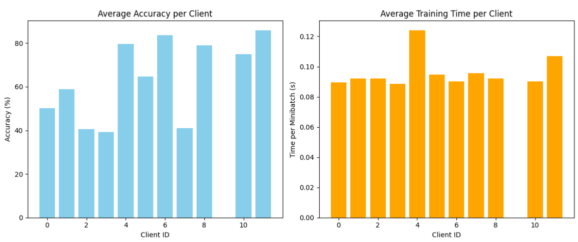


(a) Average training time per mini-batch

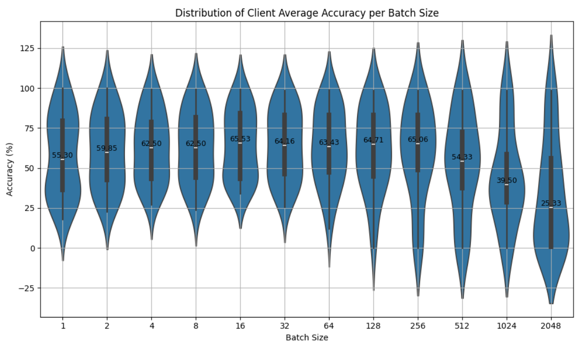


(b) Memory usage statistics across batch sizes

Figure 3: Training performance metrics across different batch sizes



(a) Average training time per client



(b) Client accuracy distribution per batch size

Figure 4: Client-specific training characteristics and accuracy patterns

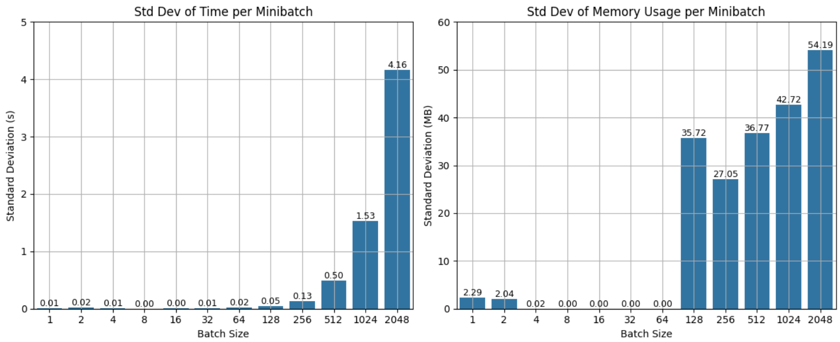


Figure 5: Standard deviation of training time and memory usage across batch sizes

3 Smart Client Selection Strategy

In federated learning, intelligently selecting clients for participation in each training round can significantly impact the performance and convergence of the global model, especially in heterogeneous data settings. To address this, we propose a **Smart Client Selection Strategy** based on the *multi-armed bandit (MAB)* framework, specifically utilizing the **Upper Confidence Bound (UCB)** algorithm with enhancements like ϵ-greedy exploration and initial random rounds.

The strategy treats each client as an arm of a bandit and models the selection process as a reward-driven optimization task. After each training round, the improvement in global validation accuracy is used as a *reward signal*. This reward reflects the client's contribution to the overall model, helping the algorithm learn which clients are most valuable over time.

By combining **UCB scores**—which balance exploration (trying less-selected clients) and exploitation (selecting clients with historically high rewards)—with randomization through ϵ-greedy steps and initial uniform sampling, our method ensures both fair exploration and informed client selection.

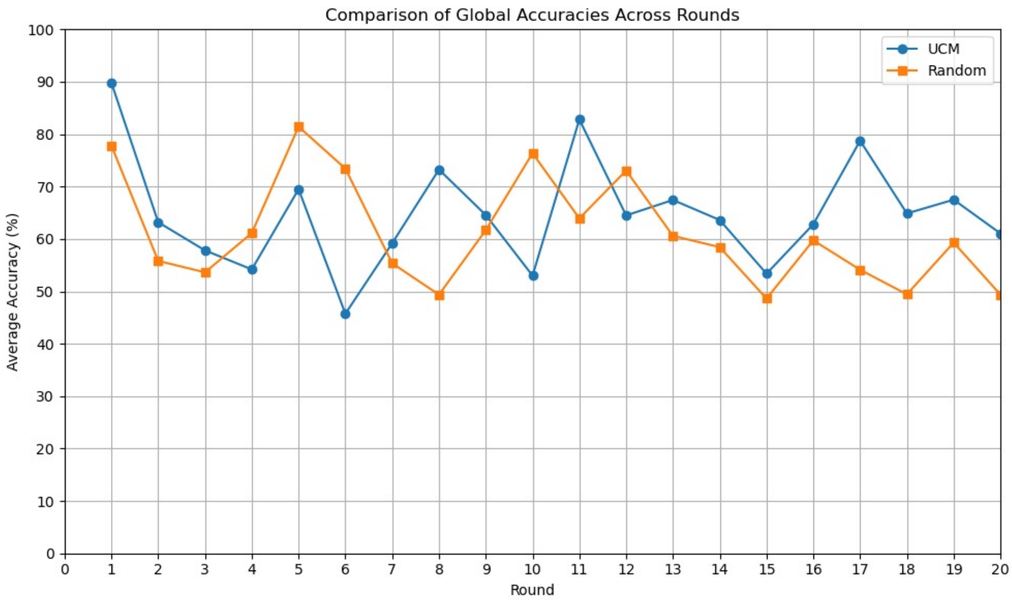


Figure 6: Comparison of global accuracies across 20 rounds using UCM vs. Random client selection

Insight from the Plot:

The plot in Figure 6 visually reinforces the performance trends summarized in Table 2. The UCM (blue line) starts with a strong accuracy, dips midway, and regains performance toward later rounds. In contrast, the Random strategy (orange line) remains less consistent and shows larger fluctuations.

The peaks and troughs in UCM indicate its adaptive learning behavior, adjusting client selection based on the learned reward signals. Despite a temporary drop in performance around Rounds 6–7 and 14–15, UCM demonstrates resilience and recovers with higher accuracy than the random approach in later rounds.

Overall, this figure highlights the strength of intelligent client selection strategies, especially in handling client heterogeneity and ensuring better long-term performance.

Table 2 compares the global accuracy of our proposed UCB-based client selection strategy (UCM) with a naive random selection baseline at various training rounds:

Round	UCM Accuracy (%)	Random Accuracy (%)
1	89.81	77.81
10	53.00	76.37
20	61.05	49.38

Table 2: Comparison of UCM and Random Accuracy across rounds

Observation:
The UCM strategy starts with a strong lead in performance (Round 1), suggesting that the initial clients selected had a highly beneficial impact on the global model. While its accuracy dips at Round 10 due to possible overfitting to specific clients or data skew, it recovers by Round 20 and surpasses the random baseline again. This illustrates the importance of exploration and adaptive learning in client selection, especially in non-IID federated environments. The observed results affirm the potential of MAB-based strategies in dynamically optimizing client participation for more robust global performance.

4 Analysis and Conclusions

4.1 Key Findings

The analysis reveals three critical insights:

1. **Extreme Non-IID Characteristics:** The Dirichlet distribution with $\alpha = 0.05$ creates realistic but challenging federated learning conditions with severe label and quantity imbalances.
2. **Computational Optimization:** Batch size 16 provides optimal balance between training efficiency, memory usage, and model accuracy across the heterogeneous client landscape.
3. **Algorithm Adaptation Necessity:** Standard federated learning algorithms require significant modifications to handle the identified data distribution challenges effectively.

Appendix

.1 Upper Confidence Bound (UCB) Client Selection Algorithm

Notes

- **Reward (r_i):** The reward for client i is the global accuracy gain, defined as the difference in the server's validation accuracy between round t and round $t-1$, normalized to $[0, 1]$.

Algorithm 1 UCB Client Selection with ϵ-Greedy and Initial Random Rounds

Input:
  Client list $\mathcal{C} = \{c_1, c_2, \ldots, c_N\}$ with client IDs $\{0, 1, \ldots, N-1\}$, where N is the total number of clients
  Round number $t \in \{1, 2, \ldots, T\}$
  Number of clients to select per round k
  Exploration parameter $c > 0$
  ϵ-greedy probability $\epsilon \in [0, 1]$
  Number of initial random rounds R_{init}
Initialization:
  For each client $i \in \{0, 1, \ldots, N-1\}$:
    $n_i \leftarrow 0$ ▷ Number of selections for client i
    $r_i \leftarrow []$ ▷ List of rewards for client i
if $t \leq R_{\text{init}}$ or random$(0, 1) < \epsilon$ **then** ▷ Random selection for initial rounds or ϵ-greedy
  Select k clients uniformly at random from $\mathcal{C}$ without replacement
  Update $n_i \leftarrow n_i + 1$ for each selected client i
  return Selected client IDs
end if
UCB Selection:
Compute total selections $s \leftarrow \sum_{i=0}^{N-1} n_i + 10^{-10}$ ▷ Avoid division by zero
for each client $i \in \{0, 1, \ldots, N-1\}$ **do**
  if r_i is empty **then** ▷ Client has no rewards yet
    $\bar{r}_i \leftarrow 0$ ▷ Set average reward to 0
  else
    $\bar{r}_i \leftarrow \frac{\sum_{r \in r_i} r}{|r_i|}$ ▷ Compute average reward
  end if
  $n'_i \leftarrow n_i + 10^{-10}$ ▷ Small epsilon to avoid division by zero
  $\text{UCB}_i \leftarrow \bar{r}_i + c\sqrt{\frac{\ln(s)}{n'_i}}$ ▷ Compute UCB score
end for
Select k clients with the highest UCB_i values
Update $n_i \leftarrow n_i + 1$ for each selected client i
return Selected client IDs

For round $t = 1$, the reward is the absolute validation accuracy.

- **Parameters in Implementation**:
  - $N = 12$ (total clients),
  - $k = 3$ (clients per round),
  - $c = \sqrt{2} \approx 1.414$ (exploration parameter),
  - $\epsilon = 0.1$ (ϵ-greedy probability),
  - $R_{\text{init}} = 5$ (initial random rounds),
  - $T = 100$ (total rounds per batch size).
- **Handling Data Skew**: The global accuracy gain reward accounts for data skew by measuring the impact of each client's contribution on the global model's performance on a shared test dataset.
- **Exploration**: The ϵ-greedy mechanism and initial random rounds ensure sufficient exploration, especially early in the experiment.