FLASH: Federated Learning Across Simultaneous Heterogeneities

Xiangyu Chang[1], Sk Miraj Ahmed[1,2], Srikanth V. Krishnamurthy[1], Basak Guler[1], Ananthram Swami[3], Samet Oymak[4], Amit K. Roy-Chowdhury[1]
[1]University of California, Riverside, [2]Brookhaven National Laboratory
[3]DEVCOM Army Research Laboratory, [4]University of Michigan
Email: [1]{cxian008, krish@cs, basakg, amitrc@ece}@ucr.edu, [2]sahme047@ucr.edu, [3]ananthram.swami.civ@army.mil, [4]oymak@umich.edu

***Abstract*—The key premise of federated learning (FL) is to train ML models across a diverse set of data-owners (clients), without exchanging local data. An overarching challenge to this date is client heterogeneity, which may arise not only from variations in data distribution, but also in data quality, as well as compute/communication latency. An integrated view of these diverse and concurrent sources of heterogeneity is critical; for instance, low-latency clients may have poor data quality, and vice versa. In this work, we propose FLASH (Federated Learning Across Simultaneous Heterogeneities), a lightweight and flexible *client selection* algorithm that outperforms state-of-the-art FL frameworks under extensive sources of heterogeneity, by trading-off the statistical information associated with the client's data quality, data distribution, and latency. FLASH is the first method, to our knowledge, for handling all these heterogeneities in a unified manner. To do so, FLASH models the learning dynamics through contextual multi-armed bandits (CMAB) and dynamically selects the most promising clients. Through extensive experiments, we demonstrate that FLASH achieves substantial and consistent improvements over state-of-the-art baselines—as much as 10% in absolute accuracy—thanks to its unified approach. Importantly, FLASH also outperforms federated aggregation methods that are designed to handle highly heterogeneous settings and even enjoys a performance boost when integrated with them.**

***Index Terms*—Federated Learning, Client heterogeneity, Client selection, Multi-armed Bandits, Noise-robust training**

I. Introduction

Federated Learning (FL) is a distributed learning paradigm where multiple clients collaborate to train a model without exchanging raw data. Training is coordinated by a central server, who selects clients [1]–[7] in each round to update the aggregated global model [1], [8]–[13]. While FL offers advantages in privacy, reduced communication costs, and scalability, it faces unique challenges due to its distributed nature, particularly in handling client heterogeneity, ensuring fairness and robustness, and balancing model accuracy with privacy.

Client heterogeneity is a central challenge in FL, manifesting through non-homogeneous label distribution [8], unreliable label assignment [6], and latency [3]. This heterogeneity degrades model accuracy [1] and increases training resources required. While existing research suggests that informed client selection can address these issues [1]–[7], current methods typically handle only one or two types of heterogeneities [3]. This limitation, combined with the need for client selection algorithms that integrate seamlessly with federated aggregation strategies, motivates our research question:

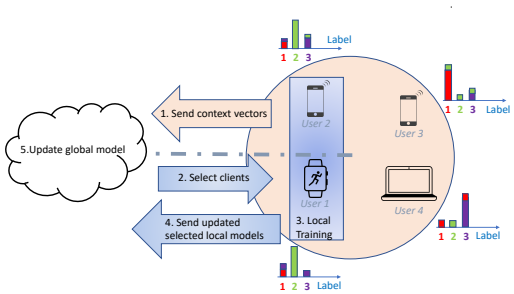


Fig. 1: Problem setup for FLASH: Building upon the standard federated learning setup of a global model learned from updates from local clients, we consider the setting where the labels of the data at the clients are imprecise (mismatched colors for the labels at each client indicate noise in those labels), the distribution of the data classes across the clients is non-uniform (height of the bars for each label class at each client), and the latencies of clients are variable (e.g. diverse devices, varying communication distance, etc.). We term these variations as heterogeneities in the data. FLASH is built upon a contextual multi-armed bandit approach, which selects the optimal set of users to update the global model, where the context vectors of clients capture their various heterogeneities. The main steps (1-4) of FLASH are illustrated in the figure.

***Q:** How can we select clients systematically under diverse and concurrent sources of heterogeneity to facilitate faster training and better accuracy? Can we combine the benefits of our client selection method with existing aggregation methods?*

Main contribution. Our algorithm FLASH addresses this challenge by explicitly modeling client heterogeneity as a context vector that summarizes client characteristics, including distributional heterogeneity, label noise, and straggler latency. FLASH employs Contextual Multi-Armed Bandits (CMAB) to select clients that maximize improvement in the global optimization objective. It offers three key advantages:

• **Simultaneous and diverse heterogeneities:** FLASH uniquely addresses multiple concurrent sources of heterogeneity through contextual variables, facilitating optimal trade-offs between diverse heterogeneities in complex scenarios.
• **Contextual and interpretable framework:** The novel CMAB framework employs contextual features to predict client contributions to global accuracy, with ablation studies confirming its ability to emphasize relevant features as heterogeneity changes.
• **Significant Performance Improvement:** FLASH achieves up to **10% improvement** in accuracy over state-of-the-art baselines and demonstrates superior performance when combined with various federated learning aggregation methods.

II. Methodology: FLASH Algorithm

Our goal is to efficiently select clients to train a global model subject to multiple sources of client heterogeneity. The key idea is that the central server uses each client's contextual information to select those that most improve the global model's accuracy. Since the server lacks a priori knowledge of client latencies, data diversity, or label noise, it must dynamically select the most contributive clients, as shown in Fig. 1. Multi-Armed Bandits (MAB) [14] provide an effective framework for such decision-making problems. The client selection policy must be determined concurrently with training, which makes approaches with higher sample complexity infeasible. Experimentally, we find that more complex approaches (e.g., using neural net-based bandits) can slow down optimization and harm accuracy (see Fig 4). This motivates our choice of a sample-efficient contextual MAB (CMAB) [15] framework that rapidly adapts to federated optimization dynamics by incorporating client heterogeneity statistics and prior rewards in its context vector (see Algo. 1).

A. Contextual MAB with Thompson-Sampling

Let $[m] := \{1, 2, \ldots, m\}$ denote the set of clients and $\boldsymbol{\mathcal{S}}_t$ be the set of all feasible subsets in round t. At round t, a learner observes m d-dimensional context vectors $\{\boldsymbol{x}_t(1), \ldots, \boldsymbol{x}_t(m)\} \subseteq \mathbb{R}^d$ corresponding to the m arms (Algorithm 1, lines 3-4). The learner chooses a super arm $S_t \in \boldsymbol{\mathcal{S}}_t$ containing M_t clients and observes rewards $\boldsymbol{r}_t = \{r_t(i)\}_{i \in S_t}$, receiving total reward $R_t(S_t) = \sum_{i \in S_t} r_t(i)$. For linear bandits, the expected reward follows: $\mathbb{E}\left[r_t(i) \mid \boldsymbol{x}_t(i)\right] = \boldsymbol{\theta}_*^\top \boldsymbol{x}_t(i)$.

Thompson Sampling Procedure (Algorithm 2). Using previous selections' history $\{(S_\tau)_{\tau=1}^{t-1}, \boldsymbol{X}_{t-1}^{\text{all}}, \boldsymbol{r}_{t-1}^{\text{all}}\}$, we estimate $\boldsymbol{\theta}_*$ via ridge regression (Algorithm 2, lines 2-3): $\hat{\boldsymbol{\theta}}_t = \boldsymbol{V}_t^{-1} \boldsymbol{X}_{t-1}^{\text{all}^\top} \boldsymbol{r}_{t-1}^{\text{all}}$ where $\boldsymbol{V}_t = \boldsymbol{X}_{t-1}^{\text{all}^\top} \boldsymbol{X}_{t-1}^{\text{all}} + \lambda \mathbf{I}$. Thompson Sampling models the distribution of $\boldsymbol{\theta}$ at time t as $\mathcal{N}(\hat{\boldsymbol{\theta}}_t, \gamma_t^2 \boldsymbol{V}_t^{-1})$ with $\gamma_t = \lambda^{1/2} + \sqrt{d \ln\left(\frac{1+tm}{\delta}\right)}$, drawing a sample $\hat{\boldsymbol{\theta}}_{\text{new}}$ to score clients by expected rewards $\hat{r}_t(i) = \boldsymbol{x}_t(i)^\top \hat{\boldsymbol{\theta}}_{\text{new}}$ (Algorithm 2, lines 4-6). For computational efficiency, $\boldsymbol{V}_t$ is updated incrementally as $\boldsymbol{V}_{t+1} \leftarrow \boldsymbol{V}_t + \sum_{i \in S_t} \boldsymbol{x}_i \boldsymbol{x}_i^\top$ (Algorithm 1, lines 13-14).

Algorithm 1 FLASH: Heterogeneity-aware Client Selection

1: **Input:** Initial model $\boldsymbol{w}^0$, Number of FL rounds n, Number of clients m, Split local dataset $\{\mathcal{D}_i\}_{i=1}^m$ into local training and validation sets $[\mathcal{T}_i; \mathcal{V}_i]_{i=1}^m$, Number of clients to select $(M_t)_{t=0}^n$, Exploration strength $\gamma_t \geq 0$, Regularization strength λ, Confidence δ
2: **Output:** Final model $\boldsymbol{w}^n$
3: Initialize MAB parameter estimation $\hat{\boldsymbol{\theta}}_0 \leftarrow 0$
4: $S_0 \leftarrow [m]$
5: $\boldsymbol{b}_0 \leftarrow 0$
6: $\boldsymbol{V}_0 \leftarrow \lambda \boldsymbol{I}_d$
7: **for** rounds $t = 0, 1, ..., n-1$ **do**
8: $\quad \gamma_t = \lambda^{1/2} + \sqrt{d \ln\left(\frac{1+tm}{\delta}\right)}$
9: $\quad$ **Server:** Send $\boldsymbol{w}^t$ to all clients $i \in [m]$
10: $\quad$ **for** client $i \in S_t$ **do**
11: $\quad\quad$ Download global model $\boldsymbol{w}^t$
12: $\quad\quad \boldsymbol{w}_i^{t+1} \leftarrow \texttt{LocalTraining}(\boldsymbol{w}^t, \mathcal{T}_i)$
13: $\quad$ **end for**
14: $\quad$ **for** all clients $i \in [m]$ **do**
15: $\quad\quad$ Measure the duration τ
16: $\quad\quad \boldsymbol{x}_t(i) \leftarrow \texttt{GetContext}(\mathcal{V}_i, \mathcal{T}_i, \tau)$
17: $\quad$ **end for**
18: $\quad$ *FedAvg*: $\boldsymbol{w}^{t+1} \leftarrow \sum_{i \in S_t} (N_i/N) \boldsymbol{w}_i^{t+1}$
// $\boldsymbol{X}_t \leftarrow$ Concatenate $(\boldsymbol{x}_t(i))_{i \in S_t}$
19: $\quad \boldsymbol{r}_t \leftarrow \texttt{GlobalModelEvaluation}(\mathcal{V}, \boldsymbol{w}^t)$
// Concatenate $\boldsymbol{X}_t^{\text{all}} \leftarrow [\boldsymbol{X}_t, \boldsymbol{X}_t^{\text{all}}]$
// Concatenate $\boldsymbol{r}_t^{\text{all}} \leftarrow [\boldsymbol{r}_t, \boldsymbol{r}_t^{\text{all}}]$
20: $\quad \texttt{scores}_t, \boldsymbol{V}_{t+1}, \boldsymbol{b}_{t+1} \leftarrow$ TSSCORES$(\{\boldsymbol{x}_i\}_{i \in [m]}, \{r_i\}_{i \in S_t}, \boldsymbol{V}_t, \boldsymbol{b}_t, \gamma_t, S_t)$
21: $\quad S_{t+1} \leftarrow$ top_M_{t+1}_indices$(\texttt{scores}_t)$
22: **end for**
23: **return** Final model $\boldsymbol{w}^n$
24: // Final model is evaluated on a global test dataset $\mathcal{G}$

Properties of FLASH. The selected clients influence the global model, causing both rewards and context vectors to change in subsequent rounds (Algorithm 1, lines 8-12). As FLASH samples a client more frequently, the client's context vector changes, making it more likely that a different client will be selected in future rounds, preventing repeatedly choosing the same clients.

B. Noise Robust Training

To reduce the impact of noisy labels, FLASH employs pseudo-labeling techniques and measures model performance on both actual and pseudo-labels. For a K-class classification problem with dataset $\mathcal{D} = \{\boldsymbol{a}^i, \boldsymbol{y}^i\}_{i=1}^n$, we aim to learn a model $p(\cdot; \boldsymbol{w}^t) : \mathcal{X} \to \mathcal{Y}$. The local dataset $\mathcal{D}_i$ is split into training set $\mathcal{T}_i$ and validation set $\mathcal{V}_i$. We generate soft pseudo-labels using model output $\boldsymbol{z} := p(\boldsymbol{a}; \boldsymbol{w}^t)$ for samples from $\mathcal{T}_i$ (Algorithm 1, line 7).

Algorithm 2 TSSCORES: Thompson Sampling-based client scores

Input: Data $\{\boldsymbol{x}_i\}_{i\in[m]}, \{r_i\}_{i\in S_t}$, current parameter $\boldsymbol{V}, \boldsymbol{b}$, exploration strength $\gamma_t \geq 0$, selected clients S_t
Output: Client selection $\texttt{scores} \in \mathbb{R}^m$
// Equivalent to $\boldsymbol{V} \leftarrow {\boldsymbol{X}^{\text{all}}}^\top \boldsymbol{X}^{\text{all}} + \lambda \boldsymbol{I}_d$
$\boldsymbol{V} \leftarrow \boldsymbol{V} + \sum_{i\in S_t} \boldsymbol{x}_i \boldsymbol{x}_i^\top$
$\boldsymbol{b} \leftarrow \boldsymbol{b} + \sum_{i\in S_t} r_i \boldsymbol{x}_i$
$\hat{\boldsymbol{\theta}} \leftarrow \boldsymbol{V}^{-1}\boldsymbol{b}$
$\hat{\boldsymbol{\theta}}_{new}$ is sampled from $\mathcal{N}(\hat{\boldsymbol{\theta}}, \gamma_t^2 \boldsymbol{V}^{-1})$
for all clients $i \in [m]$ **do**
    $\texttt{scores}(i) \leftarrow \hat{\boldsymbol{\theta}}_{new}^\top \boldsymbol{x}(i)$
end for
return $\texttt{scores}, \boldsymbol{V}, \boldsymbol{b}$

The regular cross-entropy loss $\mathcal{L}_{CE}$ and reverse cross-entropy loss $\mathcal{L}_{RCE}$ are:

$$\mathcal{L}_{CE}(\mathcal{D}) = -\frac{1}{|\mathcal{D}|} \sum_{(\boldsymbol{a},\boldsymbol{y})\in\mathcal{D}} \sum_{k=1}^{K} [\boldsymbol{y}]_k \log[p(\boldsymbol{a}; \boldsymbol{w}^t)]_k, \quad (1)$$

$$\mathcal{L}_{RCE}(\mathcal{D}) = -\frac{1}{|\mathcal{D}|} \sum_{(\boldsymbol{a},\boldsymbol{y})\in\mathcal{D}} \sum_{k=1}^{K} [p(\boldsymbol{a}; \boldsymbol{w}^t)]_k \log[\boldsymbol{y}]_k \quad (2)$$

The noise-robust loss combines these losses:

$$\mathcal{L}_{robust}(\mathcal{D}_i) = \mathcal{L}_{CE}(\mathcal{T}_i) + \alpha \mathcal{L}_{CE}(\mathcal{P}_i) + \beta \mathcal{L}_{RCE}(\mathcal{T}_i) \quad (3)$$

where α controls overfitting and β allows flexible exploration of RCE robustness. For computational stability with one-hot labels, we define $\log 0 = A$ where $A < 0$ is a constant.

C. Reward and Context Vector for FLASH

The reward is defined as the average pseudo-label CE loss change rate: $r_t = \frac{|\mathcal{L}^t_{robust} - \mathcal{L}^{t-1}_{robust}|}{\tau_{t-1}}$ (Algorithm 1, line 11). The context vector for client i in round t is $\boldsymbol{x}_t(i) = [\mathcal{L}^t_{robust}/\mathcal{L}^1_{robust}, \mathcal{L}^t_{CE}(\mathcal{V}_i)/\mathcal{L}^1_{CE}(\mathcal{V}_i), \tau_{t-1}, r_{t-1}]$ (Algorithm 1, line 3), incorporating:

- **Local training loss ratio** $\mathcal{L}^t_{robust}/\mathcal{L}^1_{robust}$: Normalized training loss reflecting relative change
- **Local validation loss ratio** $\mathcal{L}^t_{CE}(\mathcal{V}_i)/\mathcal{L}^1_{CE}(\mathcal{V}_i)$: Measure of overfitting and data heterogeneity
- **Duration** τ_{t-1}: Computation time, simulated using shifted-exponential distribution
- **Previous reward** r_{t-1}: Included for more accurate prediction of reward changes.

III. Theoretical Analysis of Linear Contextual Bandits for Client Selection

A. Modeling Assumptions for Regret Analysis

For the purpose of the regret analysis presented herein, we adopt the following standard modeling assumptions:

1) **Linear Expected Reward**: The expected reward $\mathbb{E}[r_t(i)|\boldsymbol{x}_t(i)]$ for any client $i \in [m]$ (where m is the total number of clients) given its d-dimensional context vector $\boldsymbol{x}_t(i) \in \mathbb{R}^d$ at round $t \in [0, n-1]$ (where n is the total number of FL rounds) is assumed to be linear. This is modeled as $\mathbb{E}[r_t(i)|\boldsymbol{x}_t(i)] = {\theta^*}^\top \boldsymbol{x}_t(i)$, where θ^*_i is an unknown true d-dimensional parameter vector specific to client i. This per-client parameterization allows for a fine-grained analysis within this theoretical framework.
2) **Conditionally R-sub-Gaussian Noise**: The observed reward $r_t(i)$ from client i at round t is $r_t(i) = {\theta^*}^\top \boldsymbol{x}_t(i) + \eta_t(i)$. The noise term $\eta_t(i)$ is assumed to be conditionally R-sub-Gaussian,, for some constant $R > 0$.
3) **Bounded Norms**: The L_2 norms of the context vectors and the true parameter vectors are assumed to be bounded. That is, for all clients i and rounds t, $||\boldsymbol{x}_t(i)||_2 \leq L$ and $||\theta^*_i||_2 \leq S$, for some positive constants L and S.

B. Cumulative Regret Definition

The performance of the client selection algorithm is evaluated using cumulative regret, denoted $\mathcal{R}_n$. Let S^*_t be the optimal set of M_t clients that would be chosen at round t by an oracle with full knowledge of all true parameter vectors θ^*_i. Let S_t be the set of M_t clients selected by the learning algorithm at round t. The cumulative regret over n rounds is:

$$\mathcal{R}_n = \sum_{t=0}^{n-1} \left[\sum_{i\in S^*_t} \mathbb{E}[r_t(i)|\boldsymbol{x}_t(i)] - \sum_{i\in S_t} \mathbb{E}[r_t(i)|\boldsymbol{x}_t(i)] \right]$$

C. Regret Bound for a LinUCB-type Selector

Consider a LinUCB-type client selection algorithm operating under the assumptions outlined above. If the exploration parameter α_t used for computing the Upper Confidence Bound (UCB) for client scores at round $t \in [0, n-1]$ is defined as: $\gamma_t = R\sqrt{d \log\left(\frac{1+tL^2/\lambda}{\delta}\right)} + \lambda^{1/2} S$, where d is the dimension of the context vectors $\boldsymbol{x}_t(i)$, $\lambda > 0$ is the regularization strength (as used in $V_0 = \lambda I_d$ within the FLASH MAB initialization), and $\delta \in (0, 1)$ is a confidence parameter (an input to FLASH, used in Alg. 1 for γ_t).

Then, for a strategy that selects M_t clients per round (where M_t is the number of clients to select, an input to FLASH) over n total communication rounds, the cumulative regret $\mathcal{R}_n$ from Alg. 1 can be proven (similarly as in [15]) to be upper bounded by (with probability $1-\delta$):

$$\mathcal{R}_n \leq \gamma_n \|\boldsymbol{x}_n(i)\|_{\boldsymbol{V}_n^{-1}} \leq M_n \log\left(1 + \frac{nL^2}{5\lambda}\right) \cdot \left(R\sqrt{d \log\left(\frac{1+nL^2/\lambda}{\delta}\right)} + \lambda^{1/2} S\right)$$

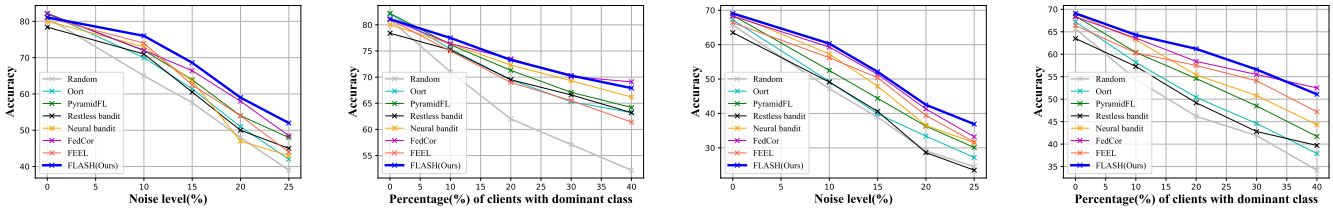
Where $\boldsymbol{V}_n^{-1}$ comes from Alg. 2. This bound indicates that the regret grows sub-linearly with the number of rounds n, demonstrating the learning capability of such an approach.

	FedAvg [1]	FedProx [8]	FedBiO [9]	FedDF [10]	FedNova [11]	SCAFFOLD [12]	RHFL [13]	Average
Random [1]	57.1 ± 2.4	64.3 ± 2.5	65.2 ± 2.7	65.5 ± 3.5	67.8 ± 3.2	69.3 ± 2.6	64.7 ± 2.7	64.8
Oort [2]	65.3 ± 1.6	66.5 ± 1.4	67.2 ± 1.1	70.1 ± 1.2	67.2 ± 1.1	69.0 ± 1.4	65.2 ± 1.4	67.2
PyramidFL [3]	67.1 ± 1.1	68.7 ± 1.0	69.1 ± 1.3	68.0 ± 1.5	69.3 ± 1.3	70.1 ± 1.6	66.5 ± 1.2	68.4
Restless bandit [4]	66.6 ± 2.9	63.2 ± 2.2	66.1 ± 2.5	67.6 ± 2.4	62.4 ± 2.3	63.2 ± 1.6	64.3 ± 2.0	64.7
Neural bandit [5]	69.3 ± 2.6	71.2 ± 1.8	69.3 ± 2.3	70.4 ± 2.4	69.0 ± 2.1	71.8 ± 2.2	65.9 ± 2.5	69.6
FedCor [6]	70.1 ± 1.4	73.2 ± 1.5	73.8 ± 1.2	69.5 ± 1.6	71.8 ± 1.6	72.8 ± 1.4	64.8 ± 1.6	70.8
FEEL [7]	65.5 ± 1.1	69.4 ± 1.5	67.3 ± 1.6	69.7 ± 1.3	70.4 ± 1.4	64.6 ± 1.2	65.3 ± 1.5	67.4
FLASH (Ours)	70.3 ± 1.1	71.6 ± 1.3	72.7 ± 1.2	72.2 ± 1.1	73.5 ± 1.4	73.7 ± 1.3	69.2 ± 1.3	71.8

	FedAvg [1]	FedProx [8]	FedBiO [9]	FedDF [10]	FedNova [11]	SCAFFOLD [12]	RHFL [13]	Average
Random [1]	57.6 ± 2.7	61.1 ± 3.1	61.3 ± 3.4	60.3 ± 2.5	59.6 ± 3.1	57.2 ± 2.2	68.3 ± 2.4	60.8
Oort [2]	61.4 ± 1.8	66.8 ± 1.4	66.4 ± 1.2	67.6 ± 1.3	65.8 ± 1.2	66.3 ± 1.5	69.4 ± 1.4	66.3
PyramidFL [3]	63.9 ± 1.2	66.1 ± 1.1	65.7 ± 1.2	64.5 ± 1.4	65.7 ± 1.4	66.6 ± 1.6	70.0 ± 1.3	66.0
Restless bandit [4]	60.5 ± 3.0	60.7 ± 2.6	64.1 ± 2.4	58.9 ± 2.3	54.3 ± 1.9	60.2 ± 1.6	69.0 ± 2.1	61.1
Neural bandit [5]	63.6 ± 2.4	68.7 ± 2.2	66.8 ± 2.4	63.4 ± 2.1	67.6 ± 2.0	68.2 ± 1.7	71.4 ± 2.5	67.1
FedCor [6]	66.6 ± 1.5	71.4 ± 1.3	66.4 ± 1.2	63.6 ± 1.8	67.9 ± 1.4	70.3 ± 1.7	69.1 ± 1.8	67.9
FEEL [7]	62.5 ± 1.4	64.9 ± 1.7	59.2 ± 1.5	67.0 ± 1.6	65.5 ± 2.1	62.3 ± 2.3	71.5 ± 1.5	64.7
FLASH (Ours)	68.2 ± 1.2	69.0 ± 1.4	71.5 ± 1.4	72.2 ± 1.2	69.8 ± 1.3	71.9 ± 1.4	74.2 ± 1.5	70.9

	FedAvg [1]	FedProx [8]	FedBiO [9]	FedDF [10]	FedNova [11]	SCAFFOLD [12]	RHFL [13]	Avg
Random [1]	47.6 ± 3.9	49.8 ± 2.7	50.7 ± 3.1	49.2 ± 2.6	48.1 ± 3.0	46.1 ± 2.8	57.0 ± 2.4	49.8
Oort [2]	51.2 ± 2.5	54.7 ± 2.3	56.3 ± 2.1	56.9 ± 2.4	55.5 ± 2.2	56.0 ± 2.2	57.6 ± 2.5	55.4
PyramidFL [3]	52.4 ± 2.6	54.4 ± 2.3	53.9 ± 2.2	52.5 ± 2.0	53.9 ± 2.3	54.9 ± 2.1	58.1 ± 2.1	54.3
Restless bandit [4]	46.2 ± 3.5	49.4 ± 2.9	52.6 ± 3.2	47.8 ± 3.1	49.3 ± 3.1	48.8 ± 2.7	55.3 ± 2.9	49.9
Neural bandit [5]	52.6 ± 2.8	53.0 ± 2.6	54.3 ± 2.5	52.0 ± 2.2	56.2 ± 2.4	56.8 ± 2.4	56.9 ± 2.3	54.5
FedCor [6]	51.7 ± 1.9	54.9 ± 2.1	58.0 ± 2.3	52.7 ± 2.4	56.4 ± 2.1	58.8 ± 2.3	57.7 ± 1.9	55.9
FEEL [7]	51.2 ± 2.5	53.4 ± 2.1	57.7 ± 2.2	55.5 ± 2.0	54.0 ± 2.1	50.8 ± 2.0	59.7 ± 2.3	54.6
FLASH (Ours)	56.4 ± 2.4	57.6 ± 2.1	60.0 ± 1.8	60.8 ± 2.2	58.3 ± 2.1	60.4 ± 1.7	62.7 ± 1.6	59.4

TABLE I: Combining selection-based algorithms with aggregation-based algorithms, we compare the accuracy (in %) of FLASH on CIFAR-10 dataset. Upper: 30% non-IIDness, Middle: 15% label noise, Lower: 30% non-IIDness and 15% label noise. Rows are client selection strategies, columns are aggregation strategies. We highlighted the best and second best one in each column. The average shows that our client selection strategy works better than other selection strategies when combined with a variety of aggregation methods.



(a) (b) (c) (d)

Fig. 2: Best global model test accuracy on CIFAR10 (a-b) and FEMNIST (c-d) dataset for different selection algorithms (with FedAvg aggregation) as the noise level (a,c), and non-IIDness of the data distribution (a,d) are varied.

IV. Experimental Evaluation

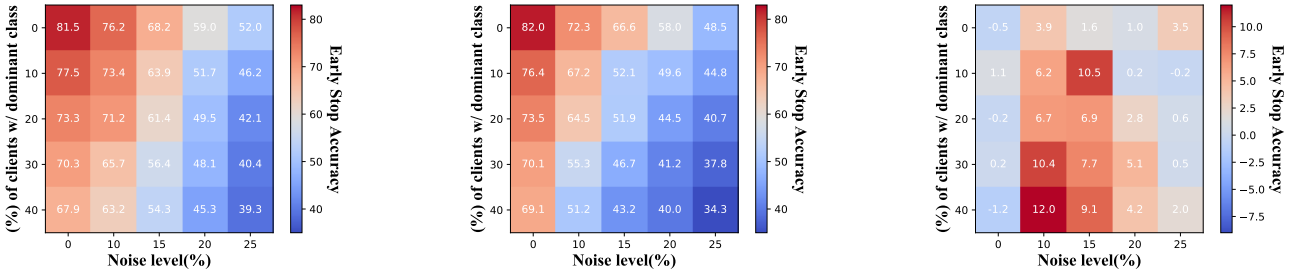
We analyze FLASH's performance under different conditions, focusing on datasets, heterogeneity modeling, and comparative analysis with baselines. In all experiments, we report the *early-stop accuracy* achieved during federated optimization.

A. Datasets, Heterogeneity Models, and Baselines

Datasets. We evaluate FLASH on two widely-used FL datasets: CIFAR-10 [16] with 60,000 32x32 color images in 10 classes, and FEMNIST [17], a hand-written digits dataset with 62 classes built by partitioning Extended MNIST based on writers.

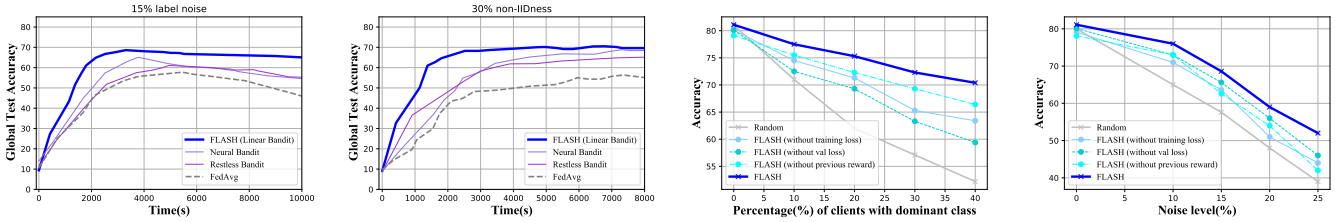
Modeling Heterogeneity. We model three types of heterogeneity: 1) Non-IIDness: For heterogeneous clients, 80% of data comes from a single class while 20% from other classes; homogeneous clients maintain uniform distribution. 2) Label noise: Noise levels follow a Beta distribution $B(\alpha_{Beta}, \beta_{Beta})$ with α_{Beta} in $\{5, 10, 15, 20, 25\}$ and $\beta_{Beta} = 100 - \alpha_{Beta}$, following [18], [19]. 3) Latency: Device execution times follow a shifted exponential distribution [20].

For non-IID modeling, we assign a pronounced data skew (e.g., 80% from one class) to a controlled percentage of clients (e.g., "30% non-IIDness" explicitly means 30% of clients exhibit this 80/20 profile). This method provides direct, intuitive control over the proportion of clients with significant skew [18], which is advantageous for studying the direct impact on client selection algorithm performance as this quantity of heterogeneity varies. While the Dirichlet distribution (dir(α)) is a common alternative, controlling the exact proportion of clients with a specific high degree of skew (e.g., a dominant class holding >80% of data) is less direct and requires precise α tuning. For instance, numerical simulations indicate that achieving approximately 40% dominant-class clients requires $\alpha \approx 0.06 - 0.07$, while $\alpha = 0.01$ yields about 89% such clients, and $\alpha = 0.10$ results in roughly 12%. Our experiments, which



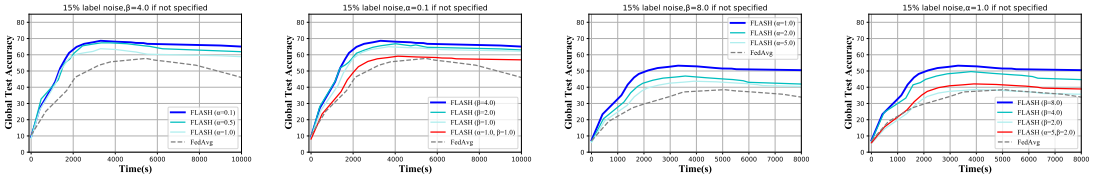
(a) FLASH Accuracy (b) FedCor Accuracy (c) FLASH - FedCor Accuracy Gap

Fig. 3: Heatmaps demostrate the best test accuracy that FLASH and FedCor (state-of-the-art) can achieve under varying levels of combination of two heterogeneities with FedAvg aggregation: (a) FLASH, (b) FedCor, and (c) FLASH-FedCor. The larger the area of the red and orange regions, the better the corresponding algorithm performs on more heterogeneities. The advantage of FLASH over FedCor is more visible when the problem involves both label noise and non-IIDness. The improvement is 3.76% improvement on average over all noise/non-IID levels and can be more than 10%.



(a) (b) (c) (d)

Fig. 4: (a-b) Test accuracy for FLASH (with FedAvg) applying different types of multi-armed bandits under different settings of heterogeneity on CIFAR dataset: (a) under noisy setting (b) under non-IID setting. These figures depict the training time required and the achieved global model accuracy when replacing FLASH's linear bandit with other types of bandit. It is clear that FLASH achieves the same accuracy with far less training time. (c-d) Ablation studies of the context vector elements of FLASH (with FedAvg) on CIFAR: Best global model test accuracy as the (c) non-IIDness, (d) noise level of the data distribution is varied. These figures illustrate the potential performance degradation of the global model when specific context vector elements are removed.



(a) CIFAR10 (b) CIFAR10 (c) FEMNIST (d) FEMNIST

Fig. 5: Test accuracy for FLASH (with FedAvg), applying different combinations of (α, β) in $\mathcal{L}_{robust}$.$A = -4$ is fixed for all the dataset: (a-b) CIFAR10, (c-d) FEMNIST. The choice of (α, β) are: (a) $\alpha = \{0.1, 0.5, 1.0\}, \beta = 4.0$, (b) $\alpha = 0.1, \beta = \{4.0, 2.0, 1.0\}$ and $(\alpha, \beta) = (1.0, 1.0)$,(c) $\alpha = \{0.1, 0.5, 1.0\}, \beta = 8.0$, (d) $\alpha = 1.0, \beta = \{8.0, 4.0, 2.0\}$ and $(\alpha, \beta) = (5, 1.0)$

examine scenarios with up to 40% skewed clients, therefore correspond to challenging non-IID conditions (equivalent to dir(α) with $\alpha < 0.1$), confirming a rigorous evaluation environment.

Baseline methods. We compare FLASH with combinations of recent client selection strategies and aggregation methods. Selection methods focus on client contribution [5], [7], client drift [6], and communication delay [2], [3]. Aggregation methods aim to optimize model aggregation [10]–[12] and reduce client drift [8], [9], [13].

B. Implementation Details

We use $m = 50$ clients for CIFAR-10 and $m = 3550$ for FEMNIST, selecting $M_t = 0.2 \cdot m$ clients per round for maximum $n = 1500$ rounds. Algorithm parameters are set to $\delta = 5 \times 10^{-2}$ and $\lambda = 1$. All clients use ResNet-18 trained locally for 5 epochs with Adam optimizer.

For $\mathcal{L}_{robust}$, we fix $A = -4$ and tune (α, β). A moderately large α is recommended for challenging datasets, while β's impact varies with dataset complexity (see Fig. 5). For latency simulation, we model duration as $\tau_t = \max_{i \in S_t}\{T_i\}$ where $T_i - \alpha_T N_i \sim \text{Exp}(1/\lambda_T N_i)$, with $\alpha_T = 1$ and λ_T varying in $\{1, 10, 100\}$.

C. Results and Analysis

Generalizability: Table I shows FLASH provides better average performance across different aggregation strategies. While some methods like FedCor [6] excel with specific

strategies, FLASH demonstrates consistent performance across various conditions.

Heterogeneity Analysis: Figures 2 and 3 demonstrate FLASH's effectiveness under various heterogeneity levels. While some algorithms may outperform FLASH in specific scenarios, FLASH shows superior performance with multiple concurrent heterogeneities.

Bandit Algorithm Comparison: Fig. 4 (a,b) show that more complex MAB algorithms like Neural Bandit [5] or Restless Bandit [4] don't improve performance but increase computational overhead, indicating the feasibility and optimality of our linear bandit application.

Context Vector Analysis: Our ablation studies (Fig. 4, (c,d)) reveal that local training loss is crucial for noisy datasets, while local validation loss is more important in non-IID settings. Previous round reward contributes to stable improvement across various settings. The "Duration" feature effectively addresses latency heterogeneity.

Different combinations of (α, β, A) in $\mathcal{L}_{robust}$: The RCE term in the loss (Eqn. 5) can be further simplified:

$$\begin{aligned}\mathcal{L}_{RCE}(\boldsymbol{a}, y) &= -\sum_{k=1}^{K}[p(\boldsymbol{a}; \boldsymbol{w}_i^t)]_k \log[y]_k \\ &= -A\sum_{k\neq y}[p(\boldsymbol{a}; \boldsymbol{w}_i^t)]_k = -A(1 - [p(\boldsymbol{a}; \boldsymbol{w}_i^t)]_{k=y}).\end{aligned}$$

Since tuning A is equivalent to scaling β, we fix $A = -4$ and only tune hyperparameters (α, β). For parameter tuning in $\mathcal{L}_{robust}$, both α and β require careful consideration:

Parameter α: Large α leads to overfitting, while small α reduces overfitting in $\mathcal{L}_{CE}$ but slows convergence (similar to using only $\mathcal{L}_{RCE}$). A moderately large α is recommended, especially for challenging datasets like FEMNIST.

Parameter β: Its impact depends on dataset complexity and α selection: For simple datasets (e.g., CIFAR10) with well-chosen α: β has minimal impact on training (shown for $\beta = 1, 2, 4$ in Fig. 5(b)). For challenging datasets (e.g., FEMNIST) or poorly-chosen α: results are highly sensitive to β choice, as demonstrated in Fig. 5(b) and (d).

V. Conclusions

We have addressed an open, but critically important, problem in federated learning, namely how to simultaneously deal with multiple kinds of heterogeneities that arise across local clients. These include latencies, noisy labels at the clients, and varying data distributions across the clients. We proposed FLASH – a flexible client selection algorithm that automatically incorporates rich contextual information associated with the heterogeneity at the clients via contextual multi-armed bandits. On two of the most commonly-used datasets, FLASH shows significant performance improvements over existing client selection methods, especially when multiple heterogeneities are present simultaneously. Moreover, we showed the generalizability of FLASH when combined with a variety of global aggregation methods.

References

[1] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Artificial intelligence and statistics*. PMLR, 2017, pp. 1273–1282.

[2] Fan Lai, Xiangfeng Zhu, Harsha V Madhyastha, and Mosharaf Chowdhury, "Oort: Efficient federated learning via guided participant selection.," in *Operating Systems Design and Implementation*, 2021.

[3] Chenning Li, Xiao Zeng, Mi Zhang, and Zhichao Cao, "Pyramidfl: A fine-grained client selection framework for efficient federated learning," in *Proceedings of the 28th Annual International Conference on Mobile Computing And Networking*, 2022, pp. 158–171.

[4] Michal Yemini, Amir Leshem, and Anelia Somekh-Baruch, "The restless hidden markov bandit with linear rewards and side information," *IEEE Transactions on Signal Processing*, vol. 69, 2021.

[5] Hangrui Cao, Qiying Pan, Yifei Zhu, and Jiangchuan Liu, "Birds of a feather help: Context-aware client selection for federated learning," in *International Workshop on Trustable, Verifiable and Auditable Federated Learning in Conjunction with AAAI (FL-AAAI)*, 2022.

[6] Minxue Tang, Xuefei Ning, Yitu Wang, Jingwei Sun, Yu Wang, Hai Li, and Yiran Chen, "Fedcor: Correlation-based active client selection strategy for heterogeneous federated learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2022, pp. 10102–10111.

[7] Jinke Ren, Yinghui He, Dingzhu Wen, Guanding Yu, Kaibin Huang, and Dongning Guo, "Scheduling for cellular federated edge learning with importance and channel awareness," *IEEE Transactions on Wireless Communications*, vol. 19, no. 11, pp. 7690–7703, 2020.

[8] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith, "Federated optimization in heterogeneous networks," *arXiv preprint arXiv:1812.06127*, 2018.

[9] Junyi Li, Feihu Huang, and Heng Huang, "Communication-efficient federated bilevel optimization with local and global lower level problems," *arXiv preprint arXiv:2302.06701*, 2023.

[10] Tao Lin, Lingjing Kong, Sebastian U. Stich, and Martin Jaggi, "Ensemble distillation for robust model fusion in federated learning," *arXiv preprint arXiv: 2006.07242*, 2020.

[11] Jianyu Wang, Qinghua Liu, Hao Liang, Gauri Joshi, and H Vincent Poor, "Tackling the objective inconsistency problem in heterogeneous federated optimization," *Advances in neural information processing systems*, vol. 33, pp. 7611–7623, 2020.

[12] Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank Reddi, Sebastian Stich, and Ananda Theertha Suresh, "Scaffold: Stochastic controlled averaging for federated learning," in *International Conference on Machine Learning*. PMLR, 2020.

[13] Xiuwen Fang and Mang Ye, "Robust federated learning with noisy and heterogeneous clients," in *Proceedings of the IEEE/CVF CVPR*, June 2022, pp. 10072–10081.

[14] Michael N Katehakis and Herbert Robbins, "Sequential choice from several populations.," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 92, pp. 8584, 1995.

[15] Lijing Qin, Shouyuan Chen, and Xiaoyan Zhu, "Contextual combinatorial bandit and its application on diversified online recommendation," in *Proceedings of the 2014 SIAM International Conference on Data Mining*. SIAM, 2014, pp. 461–469.

[16] Alex Krizhevsky, Geoffrey Hinton, et al., "Learning multiple layers of features from tiny images," *Master's thesis*, 2009.

[17] Sebastian Caldas, Peter Wu, Tian Li, Jakub Konečnỳ, H Brendan McMahan, Virginia Smith, and Ameet Talwalkar, "Leaf: A benchmark for federated settings," *arXiv preprint arXiv:1812.01097*, 2018.

[18] Diego Ortego, Eric Arazo, Paul Albert, Noel E O'Connor, and Kevin McGuinness, "Towards robust learning with different label noise distributions," in *2020 25th International Conference on Pattern Recognition (ICPR)*. IEEE, 2021, pp. 7020–7027.

[19] Paul Albert, Diego Ortego, Eric Arazo, Noel E O'Connor, and Kevin McGuinness, "Addressing out-of-distribution label noise in webly-labelled data," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2022, pp. 392–401.

[20] Wenqi Shi, Sheng Zhou, and Zhisheng Niu, "Device scheduling with fast convergence for wireless federated learning," in *IEEE International Conference on Communications (ICC)*. IEEE, 2020, pp. 1–6.