Refined Two-Sided Learning Rate Tuning for Robust Evaluation in Federated Learning

Refined Two-Sided Learning Rate Tuning for Robust Evaluation in Federated Learning

Erich Malan, *Graduate Student Member, IEEE*, Valentino Peluso, *Member, IEEE*, Andrea Calimera, *Member, IEEE*, Enrico Macii, *Fellow, IEEE*

***Abstract*—This paper investigates the impact of client and server learning rates on training deep neural networks in Federated Learning (FL). While previous research has primarily focused on optimizing the initial values of these learning rates, we demonstrate that this approach alone is insufficient for maximizing model performance and training efficiency. To address this weakness, we propose a revised two-sided learning rate optimization strategy that integrates learning rate decay schedules as tunable variables and adjusts the learning rate configurations based on the target training budget, allowing for more effective optimization.**

We conduct an extensive experimental evaluation to quantify the improvements offered by our approach. The results reveal that (i) integrating decay schedules into the tuning process leads to significant performance enhancements, and (ii) the optimal configuration of client-server decay schedules is strongly influenced by the training round budget.

Based on these findings, we claim that performance evaluations of new FL algorithms should extend beyond the fine-tuning of the initial learning rate values, as done in the state-of-the-art approach, and include the optimization of decay schedules according to the available training budget.

***Impact Statement*—Federated Learning (FL) enables collaborative model training across distributed devices. However, challenges such as low accuracy and slow convergence persist due to intermittent client participation and data heterogeneity. While many federated optimization algorithms have been proposed, their performance is highly sensitive to the configuration of server and client learning rates. Current tuning practices primarily focus on the initial learning rate values, often underestimating the full potential those algorithms can bring. Through extensive experimental analysis, we show that integrating two-sided learning rate schedules and adjusting learning rate configurations based on the available training budget substantially improves performance. These enhancements enable the discovery of superior configurations, ensuring more reliable evaluations of FL algorithms and optimizations.**

***Index Terms*—Deep learning, federated learning, learning rate optimization, budgeted training.**

I. Introduction

FEDERATED Learning (FL) is a decentralized training strategy that enables multiple clients to collaboratively train a global model under the coordination of a central server. Each client trains a local copy of the global model using its private data and then sends the updates to the server, which aggregates the collected data to assemble an updated global version. By avoiding data sharing among the involved parties, FL streamlines machine learning in compliance with data protection policies [1], thereby encouraging its adoption across different application domains and parties [2]–[4].

Despite considerable progress, an effective FL implementation still faces significant challenges. The first challenge relates to training quality, which is constrained by several structural and environmental factors. FL typically involves a large pool of clients, each possessing statistically heterogeneous datasets. Moreover, server networking resources are often limited, resulting in the selection of only a fraction of available clients for participation in each training round, which exacerbates data heterogeneity, negatively impacting the quality of the global model. The second key challenge concerns the training budget, which encompasses resource usage on both the client and server sides, as well as over the entire network. This budget is commonly defined by the number of training rounds [5], [6]. As the number of rounds required to achieve the desired performance increases, demands on computing power, memory resources, and volume of exchanged data also grow, leading to high fixed and recurring costs.

Considerable research efforts have been directed toward facing these challenges [7]–[10]. However, identifying the optimal FL strategy for a given context and assessing the performance of different implementations remains a significant concern. This difficulty arises from the dependence of FL on many variables, including data heterogeneity across the clients, client participation at each round, and the number of local training iterations [6], [11], [12]. Depending on the operating conditions, the relative performance of different FL algorithms may vary, with no single solution consistently outperforming others across all scenarios [11]. Thus, a key concern is the lack of a consistent evaluation procedure for fair and reliable comparisons. Recent studies attempted to address this gap by investigating the effects of various data distributions and different training variables to cover the widest possible array of representative FL scenarios [6], [11], [12], highlighting that inconsistent experimental settings can lead to misleading conclusions.

Within this context, one critical yet often under-explored factor is the impact of learning rate settings. In state-of-the-art FL implementations—often referred to as FL with two-sided learning rates [8], [10], [13]—learning rate tuning involves the concurrent optimization of two learning rates: the local learning rate used in client-side training and the global learning

Erich Malan, Valentino Peluso, and Andrea Calimera, are with the Department of Control and Computer Engineering, Politecnico di Torino, 10129 Turin, Italy (e-mail: erich.malan@polito.it; valentino.peluso@polito.it; andrea.calimera@polito.it).

Enrico Macii is with the Interuniversity Department of Regional and Urban Studies and Planning, Politecnico di Torino, 10129 Turin, Italy (e-mail: enrico.macii@polito.it).

rate used in server-side updates. Two-sided learning rates can accelerate convergence [13] and enhance the final model performance [5], making this approach a standard practice in recent FL algorithms [14]–[17]. However, variations in learning rate settings can significantly influence experimental outcomes, potentially leading to an underestimation of the achievable performance.

The standard evaluation procedure aims to address this issue through a grid-search exploration over predefined sets of global and local learning rate values to identify the best configuration for each FL algorithm in a given setting [5], [8], [14]–[17]. The model performance achieved with the best configuration serves as a reference for comparison with other algorithms, under the assumption that proper learning rate initialization is sufficient for optimal performance. In this work, we critically review this common assumption and demonstrate that the standard two-sided learning rate tuning procedure is unreliable for accurate performance assessment. Two key factors are currently overlooked in FL research: the impact of learning rate scheduling on training efficiency and final performance, and the need for budget-aware learning rate tuning. Neglecting these aspects can lead to a suboptimal evaluation of a given FL algorithm, as it inhibits the discovery of better-performing operating points. Based on these considerations, we propose an enhanced grid-search-based tuning strategy that introduces two main innovations: (i) extending the configuration space by adding the global and local learning rate schedule profiles as optimization variables, (ii) re-parametrizing the schedule functions to account for training budget constraints.

To quantify the effects of our revised two-sided learning rate tuning, we conducted an extensive experimental campaign on two standard image classification tasks trained on the CIFAR-10 and CIFAR-100 datasets and a causal language modeling task on the Penn Treebank (PTB) corpus. We deployed a dedicated deep neural network architecture for each of these three tasks. The investigation encompassed simulations of three state-of-the-art FL algorithms with varying training budgets. Along with the assessment, we considered multiple training variables, such as data heterogeneity, the number of participating clients, and the synchronization frequency. The analysis of the collected results revealed four main findings: (i) the proposed configuration space (involving the joint search for learning rate initial values and schedules) enables the discovery of better solutions that were hidden with a standard configuration space (where only learning rate initial values are considered); (ii) the achieved performance gains vary depending on the FL algorithm, changing the ranking of the algorithms under evaluation obtained with a standard configuration space; (iii) the best learning rate settings (including global/local initializations and schedules) are a function of the training budget; (iv) the best learning rate decay schedules follow budget-aware decay profiles.

It is worth noting that our contribution focuses on redefining the learning rate configuration space to enable a more robust evaluation of FL algorithms, rather than designing an efficient search algorithm. Nonetheless, the proposed configuration space can be seamlessly integrated into existing FL hyperparameter optimization frameworks [18]–[21], which offer advanced search algorithms to accelerate tuning and improve overall efficiency.

The rest of the paper is organized as follows. Section II reviews previous studies on learning rate tuning in neural network training in both centralized and federated contexts. Section III summarizes the formulation of FL with two-sided learning rates and describes our proposed approach for learning rate tuning. Section IV reports the experimental setup. Section V presents the collected results. Finally, Section VI concludes the paper with a summary of the main findings.

II. Background & Related Works

The learning rate is an important hyperparameter in training neural networks, as it governs the magnitude of the steps taken to minimize the loss function during the optimization process. In standard Stochastic Gradient Descent (SGD) optimization, the learning rate acts as a scalar factor that multiplies the gradient of the loss function. Adaptive optimizers, such as ADAM [22], dynamically adjust the effective step size by rescaling the learning rate based on historical gradient information, which results in a varying step size for each parameter at each iteration. These adaptive optimizers have been employed on both the server side [8] and the client side [23]. Despite their adaptive nature, the initial value of the learning rate remains pivotal in bounding the magnitude of the step size taken during each optimization step.

In Federated Learning (FL), the complexity of learning rate tuning exceeds that of centralized training, as it involves optimizing both the global learning rate on the server and the local learning rate on the client side. Additionally, FL introduces variables such as the number of participating clients and synchronization frequency, which can significantly affect the optimal settings of the two learning rates [7].

Another critical concern is the evolution of the learning rate throughout the training process, specifically the learning rate schedule. The preliminary study in [24] proposed to decay the learning rate to the client side following an inverse time schedule. Empirical results demonstrated the benefits brought by this schedule, although the prevalent choice today is exponential decay [25], [26]. Other works showed that decaying the local learning rate is also advantageous when using adaptive optimization on the server side [8], opening to new optimizations and more advanced scheduling schemes as those presented in [27], where the authors studied the effect of updating the client learning rate at each local training iteration and not just at the beginning of each round. Other works investigated the effect of decaying the learning rate on the server side. For instance, FEDEXP [26] adjusts the global learning rate based on the latest client updates, while FEDHYPER [28] makes use of historical information derived from the client updates. Rather than employing predefined profile functions for learning rate scheduling, these methods define gradient-based schedules that adjust the learning rate according to the progress of model updates. These approaches enhance robustness against variations in the initial learning rate and simplify the tuning process. Nevertheless, an incorrect

TABLE I
NOTATIONS.

Notation	Description
R	Total number of communication rounds
r	Current communication round, $r \in [0, R)$
$\mathcal{S}$	Set of clients
$\mathcal{S}^{(r)}$	Subset of clients sampled for training at round r, $S^{(r)} \subset \mathcal{S}$
$\mathbf{w}^{(r)}$	Global model weights at round r
K	Number of local training steps
$\mathcal{D}_s$	Local dataset of the client s
$\mathcal{D}_s^{(r,k)}$	Batch of data sampled at round r and local step k, $k \in [0, K)$ and $\mathcal{D}_s^{(r,k)} \subset \mathcal{D}_s$
$\mathbf{w}_s^{(r,k)}$	Local model weights of client s at round r and local step k
$\mathbf{g}_s^{(r,k)}$	Gradient of the local model of client s at round r and local step k
$\eta_{\text{l}}^{(r)}$	Local learning rate at round r
$\eta_{\text{g}}^{(r)}$	Global learning rate at round r
T	Hyperparameter to control the final value of the learning rate schedule.

Algorithm 1: Workflow of FedAvg with Two-Sided Learning Rates.

1 Randomly Initialize $\mathbf{w}^{(0)}$
2 **for** r = *0,...,*$R-1$ **do**
3 Sample a subset $\mathcal{S}^{(r)}$ of clients
  /* Clients Optimization */
4 **for** $s \in \mathcal{S}^{(r)}$ **do in parallel**
5 $\mathbf{w}_s^{(r,0)} = \mathbf{w}^{(r)}$
6 **for** k = *0,...,*$K-1$ **do**
7 $\mathbf{g}_s^{(r,k)} = \text{GET GRADIENT}(\mathbf{w}_s^{(r,k)}, \mathcal{D}_s^{(r,k)})$
8 $\mathbf{w}_s^{(r,k+1)} = \mathbf{w}_s^{(r,k)} - \eta_{\text{l}}^{(r)} \mathbf{g}_s^{(r,k)}$
9 $\Delta_s^{(r)} = \mathbf{w}_s^{(r,K)} - \mathbf{w}_s^{(r,0)}$
  /* Server Optimization */
10 $\Delta^{(r)} = \frac{1}{\sum_{s \in \mathcal{S}^{(r)}} |\mathcal{D}_s^{(r)}|} \sum_{s \in \mathcal{S}^{(r)}} |\mathcal{D}_s^{(r)}| \Delta_s^{(r)}$
11 $\mathbf{w}^{(r+1)} = \mathbf{w}^{(r)} + \eta_{\text{g}}^{(r)} \Delta^{(r)}$

initialization of the learning rate can still lead to substantial performance degradation.

An important consideration is the relationship between learning rate scheduling and resource budget. Prior studies on centralized training indicated that the optimal learning rate schedule should be a function of the total number of available training iterations, suggesting the decay from a predefined initial value to near zero [29], [30]. However, the exploration of budget-aware learning rate schedules in federated contexts, especially in conjunction with other joint optimizations, has been largely underexplored. Our study aims to address this gap by extending the concept of budget-aware schedules from centralized training to FL, operating the number of training rounds as a proxy for the available training budget, and exploring joint learning rate optimization on both the client and server sides.

III. TWO-SIDED LEARNING RATE TUNING

In this section, we first review the standard workflow for FL with two-sided learning rates (Section III-A). We then introduce the main variables in our proposed configuration space for learning rate tuning: the initial values of the learning rates (Section III-B) and the decay scheduling functions (Section III-C).

A. FL Workflow

The target of this work is a synchronous FL scheme with partial client participation. The main workflow is outlined in Algorithm 1 using the notations reported in Table I. It is based on the concept of two-sided learning rates [8], [10], [13], with η_{l} as the local learning rate for client training and η_{g} as the global learning rate for server optimization.

The distributed architecture consists of a central server coordinating a set of clients $\mathcal{S}$, where each client s holds a local dataset D_s with private access. At first, the server initializes the global model $\mathbf{w}^{(0)}$ with random weights (line 1). Then, the flow iterates for R communication rounds (lines 2–11). Within each round r, the server samples a subset of clients $\mathcal{S}^{(r)}$ (line 3), and each selected client s downloads the latest updated weights from the server (line 5). The participating clients train the model for a predefined number of iterations K (lines 6–8), obtaining an updated local version of the model $\mathbf{w}_s^{(r,K)}$. Changes of the local model $\Delta_s^{(r)}$ are delivered to the server (line 9), which finally implements the aggregation (line 10) to obtain a new version of the global model $\mathbf{w}^{(r+1)}$ for the next training round (line 11).

According to Algorithm 1, the SGD method is deployed for both the local (line 8) and global (line 11) optimizations, but the presented workflow was conceived to support any other optimizer. Specifically, the optimization step in line 11 can be generalized as follows:

$$\mathbf{w}^{(r+1)} = \text{OPTIMIZE}(\mathbf{w}^{(r)}, \eta_{\text{g}}^{(r)}, \Delta^{(r)}) \tag{1}$$

where OPTIMIZE is any deep learning optimizer for computing the weights of the global model as a function of the aggregated local updates $\Delta^{(r)}$, referred to as "pseudo-gradients". As will be discussed later in the experimental section, we deployed three popular optimization methods in our study: FEDAVG [31], FEDAVGM [7], and FEDADAM [8].

Regardless of the optimization methods adopted on the client and server sides, the final performance and the convergence speed are influenced by the learning rate settings. The

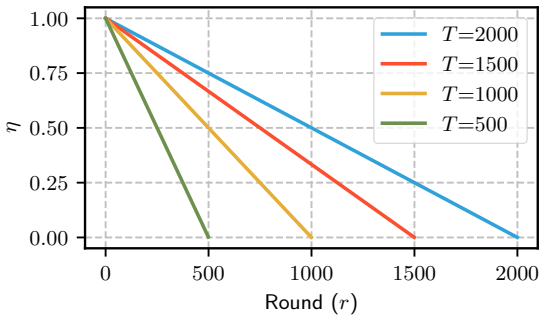


Fig. 1. Learning rate evolution with a *Linear* schedule at varying values of T.

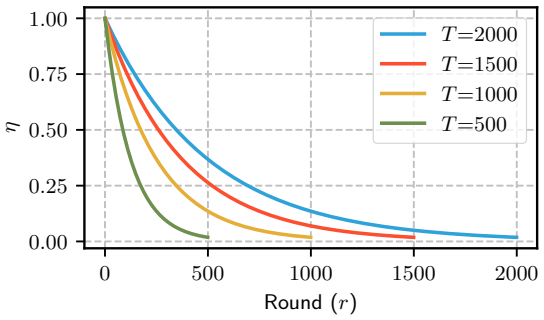


Fig. 2. Learning rate evolution with an *Exponential* schedule at varying values of T.

main step for effective tuning is to define the configuration space, which represents the set of configuration variables to be optimized along with their possible values. Our proposed configuration space for two-sided learning rate optimization includes four variables: the initial values of the global and local learning rates ($\eta_{g,0}$ and $\eta_{l,0}$) and their respective schedules ($\eta_g^{(r)}$ and $\eta_l^{(r)}$). A detailed description of these variables is provided in the following subsections.

B. Learning Rates Initial Value

The initial values of the two learning rates $\eta_{l,0}$ and $\eta_{g,0}$ have a substantial impact on both the final accuracy of the trained models and the convergence speed of the learning process. The range of values that ensures accurate and efficient training depends on the chosen OPTIMIZE function. For instance, adaptive methods like FEDADAM generally require smaller initial values for the global learning rate compared to SGD-like methods [8]. Moreover, some global optimization algorithms exhibit a stronger interdependence between $\eta_{l,0}$ and $\eta_{g,0}$ than others [5]. When the learning rates are initialized incorrectly, the training process may fail to converge, leading to reduced predictive performance. In the worst-case scenario, an incorrect initialization could nullify the training efforts, generating models with predictive capabilities that are no better than random guessing.

In line with common practice, we performed a joint exploration of different combinations of $\eta_{l,0}$ and $\eta_{g,0}$, covering a wide range of initial values (more details in Section IV). This approach aims to identify near-optimal configurations through simultaneous tuning.

C. Learning Rates Schedule

Our proposal aims to extend the configuration space of the two learning rates by integrating the selection of the global and local learning rate schedules. According to Algorithm 1, $\eta_l^{(r)}$ and $\eta_g^{(r)}$ are functions of the current round r, meaning they are both updated using round-based schedules. Despite the existence of alternative approaches that operate gradient-based schedules, such as FEDEXP and FEDHYPER, we focused on round-based schedules due to their adaptability to varying training budgets. However, a comparative analysis between round-based and gradient-based schedules is provided in the experimental section for the sake of completeness.

Among the existing scheduling options, we considered three common choices: *Constant*, *Linear*, and *Exponential* schedules. According to the *Constant* schedule, the learning rate remains equal to its initial value along the entire training process, formally expressed as:

$$\eta(r) = \eta_0, \forall r \in [0, R) \tag{2}$$

In contrast, both the *Linear* and *Exponential* schedules involve a progressive reduction of the learning rate as training proceeds. The main difference between the two lies in their decay speeds. The *Linear* function decays the learning rate at a constant rate, while the *Exponential* function features a slow decay in the early iterations, accelerating toward the end of the training flow.

The *Linear* schedule is formalized as:

$$\eta(r) = \eta_0 \cdot \left(1 - \frac{r}{T}\right) \tag{3}$$

and the *Exponential* schedule as:

$$\eta(r) = \eta_0 e^{-\frac{\gamma r}{T}} \tag{4}$$

where η_0 is the initial value, and γ is a hyperparameter that modulates the decay speed. The parameter T is a positive integer that can be tuned to change the final value of the learning rate. When $T > R$, the learning rate at the last round will be greater than zero; setting $T = R$ implements what we refer to as a budget-aware schedule, where the learning rate decays to zero by the end of the training budget. This latter choice has been proven beneficial in centralized training under budget constraints [29], [30]. Understanding whether it can be extended to FL, how to apply and implement it correctly, and determining which schedules prove most effective are the goals of our study.

To better understand the effect of a budget-aware schedule, the two plots in Fig. 1 and Fig. 2 show the evolution of $\eta(r)$ for the *Linear* and *Exponential* schedules respectively. These plots reference an initial value $\eta_0 = 1$, with $\gamma = 4$ for *Exponential*, and each color represents a different value for T. In round-based learning rate scheduling, the same T is applied, regardless of the budget R. In other words, the learning rate always consistently belongs to a specific curve. For instance, assuming $T = 2000$ (blue curve), with a budget of $R = 500$, the final learning rate value is 0.75 using the *Linear* function or 0.375 for the *Exponential* function. In

budget-aware scheduling, the learning rate follows a distinct curve depending on the actual budget R, always reaching zero at the last round. For instance, when $R = 500$, the learning rate follows the green curve; when $R = 2000$, the learning rate follows the blue curve.

In our experiments, we therefore explored an extended configuration space that includes different schedule pairs of the three profile functions (*Constant*, *Linear*, and *Exponential*) for $\eta_{\mathrm{g}}^{(r)}$ and $\eta_{\mathrm{l}}^{(r)}$, in addition to searching for $\eta_{\mathrm{g},0}$ and $\eta_{\mathrm{l},0}$. Moreover, we explored different round budgets and compared the quality of results obtained by setting a fixed T for all the schedules, regardless of the budget, against budget-aware schedules, i.e., setting $T = R$. More details about the setup can be found in Section IV.

IV. Experimental Setup

Our empirical study was conducted through simulation experiments implemented with PyTorch, version 2.1. The following subsections introduce the implementation details and setups common to all the experiments in our experimental campaign.

A. Benchmarks

The benchmarks used in our experiments consist of two image classification tasks on CIFAR-10 and CIFAR-100 and a causal language modeling task on Penn Treebank (PTB). The following sections describe the dataset structure and model architectures for each task.

a) Image Classification on CIFAR-10: The CIFAR-10 dataset contains 32×32 RGB images from 10 different classes, counting 50,000 samples for training and 10,000 for testing [32]. For this task, we employed a five-layer Convolutional Neural Network (CNN5) architecture from [31]. CNN5 consists of two convolutional layers with 64 filters of size 5×5, followed by two fully-connected layers with 384 and 192 neurons, and a final classification layer. Local training on each client involved standard data augmentation techniques, including random cropping, random horizontal flipping, and normalization, with a batch size of 50. To partition the training data across 100 clients, we followed the approach from [33]. This data allocation procedure simulates realistic scenarios with heterogeneous and imbalanced splits, using a Dirichlet distribution with a concentration parameter of 0.3 to determine the label ratios for each client.

b) Image Classification on CIFAR-100: CIFAR-100 is similar to CIFAR-10 in terms of size and structure but includes 100 classes, with each class containing 600 samples (500 for training and 100 for testing) [32]. We employed the ResNet-20 [34] architecture, replacing batch normalization with group normalization, which is a common practice in federated learning contexts [35]. The data augmentation techniques, the batch size, and the data partitioning strategy across the 100 clients match those applied to CIFAR-10.

c) Word-level Causal Language Modeling on PTB: Word-level causal language modeling focuses on training a model to process text sequences and predict the subsequent word. The PTB dataset [36] includes articles from the Wall Street Journal, featuring 929,000 tokens for training and 82,000 for testing, based on the split used in [37]. The vocabulary encloses 10,000 words along with a special token for indicating the end of a sentence. Both the model and the vocabulary embeddings were trained from scratch, starting with random initialization. The model processes sequences in batches of 10, where each sequence contains 35 tokens per local training iteration. For this task, we employed a two-layer Transformer architecture from [38] with a token embedding size of 200, two attention heads, and a feed-forward network with 200 hidden units. The activation function used is ReLU, and the dropout rate is set at 0.2. The training data was distributed among 100 clients, with each client assigned 250 sequences according to the procedure outlined in [39]. This configuration results in heterogeneous class distributions among clients, as each client has access to a limited subset of words.

B. Local Training

Unless otherwise specified, the same setup for local training was deployed across all benchmarks. At each round, 10 distinct clients out of 100 were randomly selected for training, resulting in a participation ratio of 10%. Each selected client runs 20 training iterations, using cross-entropy as the loss function and Stochastic Gradient Descent (SGD) with a weight decay of 1×10^{-3} as the local optimizer.

C. Evaluation Metrics

We employed the top-1 categorical accuracy as the evaluation metric for the image classification tasks (higher is better) and perplexity for the causal language modeling task (lower is better). In both cases, we probed the evolution of the global model on the test set, calculating the average over the last ten rounds. This averaging is a common practice to reduce the inherent oscillations of the training process and provide a more reliable comparison [8].

V. Results

The experimental analysis is organized into four subsections, each of them aimed to address specific research questions.

- Sec. V-A: Is the initialization of the learning rates sufficient for a reliable assessment of FL algorithms, or does including learning rate schedules in the tuning process provide more accurate assessments?
- Sec. V-B: How do learning rates affect training quality under different training budgets? Can budget-aware tuning of the learning rate settings enhance final model performance?
- Sec. V-C: How do key FL training variables, such as the number of participating clients, the number of local training steps, and data distribution across clients, impact the optimal initial values and schedules of the learning rates? What are the effects of optimal learning rate tuning on performance and efficiency across different FL settings?

TABLE II
CONFIGURATION SPACE OF LEARNING RATE INITIAL VALUES (BASE-10 LOGARITHM) AND SCHEDULES (CON: CONSTANT, EXP: EXPONENTIAL, LIN: LINEAR).

Setting	Grid
$\eta_{\mathrm{g},0}$	{0.5, 0.4, 0.3, 0.2, 0, -0.5, -1, -1.5, -2, -2.5}
$\eta_{\mathrm{l},0}$	{0, -0.5, -1, -1.5, -2, -2.5, -3}
$\eta_{\mathrm{g}}^{(r)}$-$\eta_{\mathrm{l}}^{(r)}$	{Con-Con, Con-Exp, Exp-Con, Exp-Exp, Con-Lin, Lin-Con, Lin-Lin}

TABLE III
COMPARISON BETWEEN THE BEST-PERFORMING CONFIGURATIONS IN A STANDARD CS AND OUR CS FOR EACH BENCHMARK AND GLOBAL OPTIMIZATION METHOD. WE REPORT THE BEST PAIR FOR THE BASE-10 LOGARITHM OF THE INITIAL GLOBAL AND LOCAL LEARNING RATES ($\eta_{\mathrm{g},0}$ AND $\eta_{\mathrm{l},0}$, RESPECTIVELY), THE BEST PAIR FOR GLOBAL AND LOCAL SCHEDULES (CON: CONSTANT, EXP: EXPONENTIAL, LIN: LINEAR), AND THE EVALUATION METRIC ON THE TEST SET (AND, FOR OUR CS, ACCURACY GAINS COMPARED TO STANDARD CS). THE METRIC IS THE TOP-1 ACCURACY IN % (HIGHER IS BETTER) FOR CIFAR-10/100 AND PERPLEXITY (LOWER IS BETTER) FOR PTB.

		Standard CS			Our CS		
Benchmark	**Optimizer**	$\boldsymbol{\eta_{\mathrm{g},0}/\eta_{\mathrm{l},0}}$	**Schedules**	**Metric**	$\boldsymbol{\eta_{\mathrm{g},0}/\eta_{\mathrm{l},0}}$	**Schedules**	**Metric (Gains)**
CIFAR-10 CNN5	FEDAVG	0.3/-0.5	Con-Exp	79.11	0.3/-0.5	Con-Lin	81.81 (+2.70)
	FEDAVGM	0.2/-1.5	Con-Exp	80.93	0.2/-1.5	Con-Lin	83.48 (+2.55)
	FEDADAM	-2/-1.5	Con-Exp	77.88	-1.5/-2	Lin-Con	81.81 (+3.93)
CIFAR-100 ResNet-20	FEDAVG	0.2/-0.5	Con-Exp	43.78	0.4/-0.5	Con-Lin	50.90 (+7.12)
	FEDAVGM	0/-1.5	Con-Exp	51.05	0.2/-1	Con-Lin	56.96 (+5.91)
	FEDADAM	-1.5/-0.5	Con-Exp	52.05	-1.5/-0.5	Con-Lin	55.40 (+3.35)
PTB Transformer	FEDAVG	0.2/-0.5	Con-Exp	117.34	0.2/-0.5	Con-Exp	117.34 (-0.00)
	FEDAVGM	0.0/-1	Con-Exp	124.59	0/-1.5	Con-Lin	115.18 (-9.41)
	FEDADAM	-2/-1	Con-Exp	120.56	-2/-1.5	Con-Lin	115.50 (-5.06)

- Sec. V-D: How does two-sided learning rate tuning with gradient-based schedules compare to round-based schedules in terms of effectiveness and performance?

A. Impact of Schedule Tuning on FL Performance Evaluation

To quantify the impact of two-sided learning rate optimization on the evaluation of FL performance, we conducted a comparative analysis among three standard federated optimization methods: FEDAVG [31], FEDAVGM [7], and FEDADAM [8]. Each method implements the FL framework with two-sided learning rates as outlined in Section III-A. For FEDAVGM, we set the momentum parameter to 0.9; for FEDADAM, we set the first-momentum parameter to 0.9, the second-momentum to 0.99, and the adaptivity to $1\mathrm{e}{-3}$.

Following the standard evaluation protocol, we employed a grid-search approach to identify learning rate settings that maximize the performance of each method. Our primary research question is to determine whether searching for initial learning rate values is sufficient for a reliable assessment and how results may differ if learning rate schedules are included in the tuning process. To address this question, we focused our analysis on a long training session with $R = 2000$ iterations as the round budget, and we compared the results of the three benchmark tasks (introduced in Section IV-A). For each task and scenario, we repeated the grid search over two different configuration spaces:

1) *Standard CS*: This configuration space involves exploring only the initial values of the learning rates, $\eta_{\mathrm{g},0}$ and $\eta_{\mathrm{l},0}$, while assuming fixed decay schedules on both the global and local sides, as done by previous works like [5], [8], [14]–[17]. The sweep range for $\eta_{\mathrm{g},0}$ and $\eta_{\mathrm{l},0}$ is based on the values reported in the first two rows of Table II. For the schedules, we implemented a *Constant* schedule for $\eta_{\mathrm{g}}^{(r)}$ and an *Exponential* schedule for $\eta_{\mathrm{l}}^{(r)}$. This scheduling configuration is commonly found in the literature. For example, in [25], [26], a scaling factor of 0.998 was applied to the local rate after each round, equivalent to employing the exponential schedule function (defined in Section 4) with $T = 2000$ and $\gamma = 4$.
2) *Our CS*: This is our proposed configuration space which uses the same sweep range for $\eta_{\mathrm{g},0}$ and $\eta_{\mathrm{l},0}$ as *Standard CS*, but expands the search including the schedules ($\eta_{\mathrm{g}}^{(r)}$ and $\eta_{\mathrm{l}}^{(r)}$), using the pairs reported in Table II (the schedule names are abbreviated with their first three letters). The resulting search grid consists of 490 distinct configuration points formed by combining 10 initial values for $\eta_{\mathrm{g},0}$, 7 values for $\eta_{\mathrm{l},0}$, and 7 pairs of $\eta_{\mathrm{g}}^{(r)}$-$\eta_{\mathrm{l}}^{(r)}$ for the global and local schedules. For the *Linear* schedule, we consistently set $T = 2000$, and for *Exponential*, we used $T = 2000$ and $\gamma = 4$.

Table III summarizes the results obtained with the two configuration spaces for each benchmark and each federated optimizer, reporting the best-performing configurations identified through the grid search over each configuration space. This includes the best pair of initial values $\eta_{\mathrm{g},0}/\eta_{\mathrm{l},0}$ and the optimal global-local schedule pair, along with the resulting evaluation metric. In *Standard CS*, the schedule pair is always *Con-Exp*, as the schedules are fixed and excluded from the search process.

The collected results reveal the significant role of two-sided schedule functions and their joint optimization. Several key

observations and conclusions can be drawn.

First, the *Con-Exp* schedule pair used in *Standard CS* proves to be suboptimal, as it is dominated by other configurations identified using *Our CS*. On CIFAR-10, *Our CS* resulted in accuracy gains ranging from +2.55% (with FEDAVGM) to +3.93% (with FEDADAM). Interestingly, with FEDAVG, the best-performing configurations in both *Standard CS* and *Our CS* share the same learning rate initialization values. However, simply switching the schedules from Con-Exp to Con-Lin improved accuracy by +2.70%, demonstrating the benefits of two-sided schedule optimization. On CIFAR-100, even higher gains were observed, ranging from +3.35% (with FEDADAM) to +7.12% (with FEDAVG), suggesting that two-sided schedule optimization has a greater impact when tasks become more complex. Similar trends were noted for the PTB benchmark, where *Our CS* reduced model perplexity by 9.41 in the best case (with FEDAVGM).

Second, the learning rate initialization values and the scheduling functions are tightly coupled and interdependent. In many analyzed cases, the values of $\eta_{\mathrm{g},0}$ and $\eta_{\mathrm{l},0}$ in *Our CS* differ from those in *Standard CS*, indicating that optimal results can be achieved through concurrent tuning of both the initial values and the schedule functions. For instance, with FEDAVGM on CIFAR-100, the best-performing initialization pair shifts from (0/-1.5) in *Standard CS* to (0.2/-1) in *Our CS*.

Third, two-sided schedule optimization affects how the deployed optimization algorithms rank relative to one another. For CIFAR-10 in *Standard CS*, FEDAVG outperforms FEDADAM (79.11% vs. 77.88%), but they achieve the same accuracy with *Our CS* (81.81%). For CIFAR-100, FEDADAM achieves the best performance in *Standard CS* (52.05%), while FEDAVGM produces better results in *Our CS* (56.96%). For PTB, FEDAVG achieves the best result in *Standard CS* (117.34), whereas FEDAVGM emerges as the best choice in *Our CS* (115.18). These findings suggest that (i) when the configuration space does not include multiple global and local schedules, the results of hyperparameter searches may be suboptimal, and, as a consequence, (ii) not considering two-sided schedule optimization can lead to misleading outcomes when comparing different FL methods, including the optimizer algorithms.

To further support our claims and quantify the improvement margins of two-sided learning rate schedule optimization, Table IV presents the best-performing initialization pairs for each schedule pair defined in Table II. The reported analysis focuses on the FEDAVGM optimizer, but similar trends and conclusions apply to FEDAVG and FEDADAM. For CIFAR-10, the top-performing schedule (Con-Lin) achieves 1.19% higher accuracy than the second-best configuration (Lin-Lin), specifically 83.48% vs. 82.29%, and it shows a 4.42% improvement over the worst-performing configuration (Exp-Exp), with accuracy values of 83.48% vs. 79.06%. For CIFAR-100, the accuracy gains range from 0.95% (Con-Lin vs. Lin-Con) to 7.79% (Con-Lin vs. Exp-Exp). For PTB, the perplexity gains range from 1.72 (Con-Lin vs. Lin-Con) to 16.13 (Con-Lin vs. Exp-Con). This analysis emphasizes that learning rate schedules significantly affect the quality of results, highlighting the importance of two-sided schedule tuning for robust and reliable performance assessment.

TABLE IV
BEST-PERFORMING INITIALIZATION PAIR FOR GLOBAL AND LOCAL LEARNING RATES (BASE-10 LOGARITHM OF $\eta_{\mathrm{g},0}$ AND $\eta_{\mathrm{l},0}$, RESPECTIVELY) FOR EACH SCHEDULE PAIR (CON: CONSTANT, EXP: EXPONENTIAL, LIN: LINEAR) IN OUR CS AND THE EVALUATION METRIC ON THE TEST SET. THE METRIC IS THE TOP-1 ACCURACY IN % (HIGHER IS BETTER) FOR CIFAR-10/100 AND PERPLEXITY (LOWER IS BETTER) FOR PTB.

Benchmark	$\eta_{\mathrm{g},0}/\eta_{\mathrm{l},0}$	Schedules	Metric
CIFAR-10 CNN5	-0.5/-1	Con-Con	79.06
	0.2/-1.5	Con-Exp	80.93
	0.2/-1	Exp-Con	80.65
	0/-1	Exp-Exp	80.38
	0.2/-1.5	Con-Lin	83.48
	0/-1	Lin-Con	82.15
	0/-1	Lin-Lin	82.29
CIFAR-100 ResNet-20	0/-1	Con-Con	52.02
	0/-1.5	Con-Exp	51.05
	0.2/-1	Exp-Con	51.09
	0.2/-1.5	Exp-Exp	49.17
	0.2/-1	Con-Lin	56.96
	0.2/-1	Lin-Con	56.01
	0.2/-1	Lin-Lin	55.24
PTB Transformer	0/-2	Con-Con	120.86
	0/-1	Con-Exp	124.59
	-0.5/-1	Exp-Con	131.31
	-0.5/-0.5	Exp-Exp	119.11
	0/-1.5	Con-Lin	115.18
	0/-1.5	Lin-Con	116.90
	-0.5/-1	Lin-Lin	119.69

B. Impact of Budget-Aware Learning Rate Schedules

In this subsection, we focus on understanding how learning rate settings can affect the quality of training under different training budgets and whether tuning the learning rate settings depending on the target budget can be beneficial for the final performance. To this purpose, we repeated the grid-search exploration at different budget constraints, specifically $R \in \{500, 1000, 1500\}$ and considered three alternative grid-search approaches, each of them referring to a different configuration space setting:

1) *Standard CS* with the *Con-Exp* schedule pair and a fixed value of $T = 2000$ regardless of the value of R (as in the previous subsection);
2) *Our CS with* $T = 2000$, which explores the initial values and schedule pairs in Table II, with $T = 2000$ for all the schedules regardless of the budget;
3) *Our CS with* $T = R$, which explores the initial values and schedule pairs in Table II setting a budget-aware configuration for *Linear* and *Exponential*, i.e., with $T = R$.

For all three settings, we deployed the same global optimizer, i.e., FEDAVGM.

Table V collects the best-performing settings for the three benchmarks. At a glance, the best-performing settings found with *Our CS*, both with $T = 2000$ and $T = R$, differ from that found with *Standard CS*. For CIFAR-10 and CIFAR-100, whatever the configuration space adopted, a growing training budget translates into a higher accuracy. However, *Our CS with*

TABLE V
COMPARISON BETWEEN THE BEST-PERFORMING CONFIGURATIONS IN A STANDARD CS, OUR CS WITH T=2000, AND OUR CS WITH T=R ON CIFAR-10 TRAINED WITH FEDAVGM. WE REPORT THE BEST PAIR FOR THE BASE-10 LOGARITHM OF THE INITIAL GLOBAL AND LOCAL LEARNING RATES ($\eta_{g,0}$ AND $\eta_{l,0}$, RESPECTIVELY), THE BEST PAIR FOR GLOBAL AND LOCAL SCHEDULES (CON: CONSTANT, EXP: EXPONENTIAL, LIN: LINEAR), AND THE EVALUATION METRIC ON THE TEST SET (AND, FOR OUR CS, ACCURACY GAINS COMPARED TO STANDARD CS). THE METRIC IS THE TOP-1 ACCURACY IN % (HIGHER IS BETTER) FOR CIFAR-10/100 AND PERPLEXITY (LOWER IS BETTER) FOR PTB.

Benchmark	Standard CS				Our CS with T=2000			Our CS with T=R		
	R	$\eta_{g,0}/\eta_{l,0}$	Schedules	Metric	$\eta_{g,0}/\eta_{l,0}$	Schedules	Metric (Gains)	$\eta_{g,0}/\eta_{l,0}$	Schedules	Metric (Gains)
CIFAR-10 CNN5	500	-0.5/-1	Con-Exp	75.40	0/-1	Exp-Exp	77.67 (+2.27)	0/-1.5	Con-Lin	78.48 (+3.08)
	1000	-0.5/-0.5	Con-Exp	77.48	0/-1	Exp-Exp	79.46 (+1.98)	0/-1	Con-Lin	81.70 (+4.22)
	1500	0/-1	Con-Exp	79.36	0/-1	Lin-Lin	81.26 (+1.90)	0/-1	Con-Lin	83.03 (+3.67)
CIFAR-100 ResNet-20	500	0/-1	Con-Exp	43.18	0/-1	Lin-Lin	44.35 (+1.17)	0/-1	Con-Lin	45.48 (+2.30)
	1000	0/-1	Con-Exp	48.25	0.2/-1	Lin-Con	50.45 (+2.20)	0/-1	Con-Lin	51.46 (+3.21)
	1500	0/-1.5	Con-Exp	49.68	0.2/-1	Lin-Con	53.79 (+4.11)	0.2/-1	Con-Lin	54.30 (+4.62)
PTB Transformer	500	0/-1	Con-Exp	122.76	0/-1	Con-Exp	122.76 (-0.00)	0/-0.5	Con-Exp	117.42 (-5.34)
	1000	0/-1	Con-Exp	122.63	0/-1.5	Con-Lin	119.94 (-2.69)	-0.5/-0.5	Con-Exp	118.21 (-4.42)
	1500	0/-1	Con-Exp	123.43	0/-1.5	Con-Lin	116.23 (-7.20)	0/-1	Con-Exp	116.24 (-7.19)

$T = R$ achieves the best performance, with gains ranging from 3.08% to 4.22% for CIFAR-10, and from 2.30% to 4.62% in CIFAR-100. Counterintuitively, for PTB trained with *Standard CS*, the best perplexity (122.63) is observed under a lower budget (100 rounds) and worsens slightly as the budget increases (123.43 at 1500 rounds). This indicates potential convergence issues due to suboptimal configurations, which can be fixed using *Our CS*, where perplexity tends to decrease as the budget grows. Compared to *Standard CS*, substantial improvements were obtained using *Our CS with* $T = 2000$, from 2.69 at $R = 1000$ to 7.20 at $R = 1500$, and even more with *Our CS with* $T = R$, from 5.34 to 7.19. It is noteworthy that a close-to-optimum performance (117.42) can already be achieved under the lowest training budget with *Our CS with* $T = R$. These findings indicate that optimal learning rate schedules can accelerate convergence.

The analysis supports our claims, suggesting that (i) suboptimal learning rate schedules, like that used in *Standard CS*, may nullify the advantages of longer training, resulting in a waste of resources, (ii) fine-tuning the learning rate settings depending on the budget, as done with budget-aware scheduling, leads to better performance.

C. Impact of Training Variables on Learning Rate Settings

The optimal initial values and schedules of the learning rates are influenced by other training variables. Therefore, we extend our analysis considering the effects of three key FL variables that are closely linked to model performance and training costs [6]: the number of participating clients ($|\mathcal{S}^{(r)}|$), the number of local training steps per round (K), and the data distribution across clients. The number of participating clients serves as a knob for resource allocation in FL systems. We considered lower values of $|\mathcal{S}^{(r)}|$ (2 and 5) to assess the impact of learning rate tuning in resource-constrained environments and quantify the achievable gains. Moreover, since participating clients are often battery-powered devices with an energy budget that limits the number of local training iterations, we explored different values of K. Specifically, we focused on smaller K values (5 and 10) to emulate more stringent constraints. Finally, the level of statistical heterogeneity in clients' datasets is another critical factor influencing both performance and training costs. We explored lower values of α (0.1 and 0.2) to simulate data partitions with higher heterogeneity. Overall, the selected values for these three variables cover real-world deployments characterized by constrained resources and high data heterogeneity across clients, which are known to adversely affect training quality.

Tables VI, VII, and VIII collect the results for the CNN-5 model trained on CIFAR-10 using FEDAVGM under varying round budgets of 500, 1000, 1500, and 2000. Beyond reaffirming the conclusions presented in Sections V-A and V-B—specifically, the significance of optimizing schedules for maximizing accuracy and the interplay between initial learning rate values and schedules—we also observed other noteworthy insights.

Training with budget-aware schedules achieves the highest accuracy in all cases. We noted substantial accuracy gains, particularly in more challenging conditions, such as +6.72% in training with two active clients for 1000 rounds (second row of Table VI) and +4.25% in training under high statistical heterogeneity with 500 rounds (first row of Table VIII). To further appreciate the value of the improvements brought by budget-aware schedules, one can note the example in Table VI with $|\mathcal{S}^{(r)}|$=5. With this setup, the maximum accuracy with *Standard CS* is 81.86% at 2000 rounds, but *Our CS* with $T = R$ reaches the same accuracy level in just 1000 rounds, which is a substantial saving. Similar trends can also be found in Tables VII and VIII, indicating that tuning the learning rate schedules affects accuracy and training costs.

D. Comparison between Round-based and Gradient-based Schedules

This subsection investigates the performance of two-sided learning rate tuning using fixed schedule settings based on recently proposed gradient-based schedules: FEDEXP [26], and FEDHYPER [28]. These gradient-based schedules dynamically adjust learning rates throughout training based on model updates, offering a more adaptive approach compared to traditional fixed schedules.

TABLE VI

COMPARISON BETWEEN THE BEST-PERFORMING CONFIGURATIONS IN A STANDARD CS, OUR CS WITH T=2000, AND OUR CS WITH T=R ON CIFAR-10 TRAINED WITH FEDAVGM AT A VARYING NUMBER OF PARTICIPATING CLIENTS ($|\mathcal{S}^{(r)}|$). WE REPORT THE BEST PAIR FOR THE BASE-10 LOGARITHM OF THE INITIAL GLOBAL AND LOCAL LEARNING RATES ($\eta_{g,0}$ AND $\eta_{l,0}$, RESPECTIVELY), THE BEST PAIR FOR GLOBAL AND LOCAL SCHEDULES (CON: CONSTANT, EXP: EXPONENTIAL, LIN: LINEAR), AND THE TOP-1 ACCURACY ON THE TEST SET (AND, FOR OUR CS, ACCURACY GAINS COMPARED TO STANDARD CS). RESULTS ON CIFAR-10.

$\lvert\mathcal{S}^{(r)}\rvert$	Standard CS				Our CS with T=2000			Our CS with T=R		
	R	$\eta_{g,0}/\eta_{l,0}$	Schedules	Metric	$\eta_{g,0}/\eta_{l,0}$	Schedules	Metric (Gains)	$\eta_{g,0}/\eta_{l,0}$	Schedules	Metric (Gains)
2	500	-1/-1	Con-Exp	65.97	-0.5/-1	Exp-Exp	69.34 (+3.37)	-0.5/-1	Con-Lin	71.02 (+5.05)
	1000	-1/-0.5	Con-Exp	71.95	-0.5/-1	Exp-Con	73.04 (+1.09)	-0.5/-1	Con-Lin	78.67 (+6.72)
	1500	-0.5/-1	Con-Exp	74.17	-0.5/-1	Exp-Con	75.66 (+1.49)	-0.5/-1	Con-Lin	77.88 (+3.71)
	2000	-0.5/-1	Con-Exp	76.75	-0.5/-1	Con-Lin	80.34 (+3.59)	-0.5/-1	Con-Lin	80.34 (+3.59)
5	500	-0.5/-1	Con-Exp	70.68	0/-1.5	Exp-Con	71.52 (+0.84)	0/-1.5	Con-Lin	75.28 (+4.60)
	1000	-0.5/-1	Con-Exp	77.09	0/-1.5	Exp-Con	79.29 (+2.22)	0/-1.5	Con-Lin	81.86 (+4.77)
	1500	0/-1.5	Con-Exp	79.75	0/-1.5	Lin-Lin	80.51 (+0.76)	0/-1.5	Con-Lin	83.01 (+3.26)
	2000	0/-1.5	Con-Exp	81.86	0/-1	Con-Lin	83.02 (+1.36)	0/-1	Con-Lin	83.02 (+1.36)

TABLE VII

COMPARISON BETWEEN THE BEST-PERFORMING CONFIGURATIONS IN A STANDARD CS, OUR CS WITH T=2000, AND OUR CS WITH T=R ON CIFAR-10 TRAINED WITH FEDAVGM AT A VARYING NUMBER OF LOCAL TRAINING STEPS (K). WE REPORT THE BEST PAIR FOR THE BASE-10 LOGARITHM OF THE INITIAL GLOBAL AND LOCAL LEARNING RATES ($\eta_{g,0}$ AND $\eta_{l,0}$, RESPECTIVELY), THE BEST PAIR FOR GLOBAL AND LOCAL SCHEDULES (CON: CONSTANT, EXP: EXPONENTIAL, LIN: LINEAR), AND THE TOP-1 ACCURACY ON THE TEST SET (AND, FOR OUR CS, ACCURACY GAINS COMPARED TO STANDARD CS). RESULTS ON CIFAR-10.

K	Standard CS				Our CS with T=2000			Our CS with T=R		
	R	$\eta_{g,0}/\eta_{l,0}$	Schedules	Metric	$\eta_{g,0}/\eta_{l,0}$	Schedules	Metric (Gains)	$\eta_{g,0}/\eta_{l,0}$	Schedules	Metric (Gains)
5	500	-0.5/-0.5	Con-Exp	71.63	0/-1	Exp-Exp	72.55 (+0.92)	0/-1	Con-Lin	74.35 (+2.72)
	1000	-0.5/-0.5	Con-Exp	76.82	0/-1.5	Lin-Lin	77.36 (+0.54)	-0.5/-0.5	Con-Lin	79.41 (+2.59)
	1500	-0.5/-0.5	Con-Exp	77.54	0/-1	Lin-Lin	78.88 (+1.34)	0/-1	Con-Lin	82.13 (+4.59)
	2000	-0.5/-0.5	Con-Exp	79.74	0/-1.5	Con-Lin	82.18 (+2.44)	0/-1.5	Con-Lin	82.18 (+2.44)
10	500	-0.5/-0.5	Con-Exp	73.65	0/-1	Exp-Con	75.36 (+1.71)	0/-1	Con-Lin	78.22 (+4.57)
	1000	0/-1	Con-Exp	78.84	0/-1	Lin-Lin	79.93 (+1.09)	0/-1	Con-Lin	81.92 (+3.08)
	1500	0/-1	Con-Exp	81.16	0/-1	Lin-Lin	82.03 (+0.87)	0/-1	Con-Lin	83.20 (+2.04)
	2000	0/-1	Con-Exp	81.54	0/-1.5	Con-Lin	83.03 (+1.49)	0/-1.5	Con-Lin	83.03 (+1.49)

TABLE VIII

COMPARISON BETWEEN THE BEST-PERFORMING CONFIGURATIONS IN A STANDARD CS, OUR CS WITH T=2000, AND OUR CS WITH T=R ON CIFAR-10 TRAINED WITH FEDAVGM AT VARYING DATA DISTRIBUTION (LOWER α DENOTES HIGHER STATISTICAL HETEROGENEITY). WE REPORT THE BEST PAIR FOR THE BASE-10 LOGARITHM OF THE INITIAL GLOBAL AND LOCAL LEARNING RATES ($\eta_{g,0}$ AND $\eta_{l,0}$, RESPECTIVELY), THE BEST PAIR FOR GLOBAL AND LOCAL SCHEDULES (CON: CONSTANT, EXP: EXPONENTIAL, LIN: LINEAR), AND THE TOP-1 ACCURACY ON THE TEST SET (AND, FOR OUR CS, ACCURACY GAINS COMPARED TO STANDARD CS). RESULTS ON CIFAR-10.

α	Standard CS				Our CS with T=2000			Our CS with T=R		
	R	$\eta_{g,0}/\eta_{l,0}$	Schedules	Metric	$\eta_{g,0}/\eta_{l,0}$	Schedules	Metric (Gains)	$\eta_{g,0}/\eta_{l,0}$	Schedules	Metric (Gains)
0.1	500	-0.5/-0.5	Con-Exp	68.28	0/-1.5	Exp-Con	69.71 (+1.43)	0/-1.5	Lin-Con	72.53 (+4.25)
	1000	-0.5/-0.5	Con-Exp	73.37	-0.5/-0.5	Con-Exp	73.37 (+0.00)	0/-1.5	Lin-Con	76.46 (+3.09)
	1500	-0.5/-0.5	Con-Exp	75.09	0/-1.5	Lin-Lin	76.60 (+1.61)	0/-1.5	Lin-Con	78.36 (+3.27)
	2000	-0.5/-0.5	Con-Exp	75.68	0/-1.5	Lin-Con	79.62 (+3.94)	0/-1.5	Lin-Con	79.62 (+3.94)
0.2	500	-0.5/-1	Con-Exp	74.48	0/-1.5	Exp-Con	76.42 (+1.94)	0/-1.5	Con-Lin	77.83 (+3.35)
	1000	-0.5/-0.5	Con-Exp	77.33	0/-1.5	Lin-Lin	78.77 (+1.44)	0/-1.5	Con-Lin	80.11 (+2.78)
	1500	-0.5/-0.5	Con-Exp	78.50	0/-0.5	Lin-Lin	80.76 (+2.26)	0/-1.5	Con-Lin	81.23 (+2.73)
	2000	0/-1.5	Con-Exp	80.06	0/-1	Con-Lin	82.35 (+2.29)	0/-1	Con-Lin	82.35 (+2.29)

We provide a quantitative comparison among three grid-search explorations with the following configuration spaces:

1) FEDEXP *CS*, that just focuses on optimizing the initial learning rate values $\eta_{g,0}$ and $\eta_{l,0}$ while employing the FEDEXP schedule. We adopted the FEDEXP-M version, setting $\varepsilon = 1e-3$ and using an *Exponential* local schedule $\eta_l^{(r)}$ with $T = 2000$ and $\gamma = 4$.
2) FEDHYPER *CS*, that searches for $\eta_{g,0}$ and $\eta_{l,0}$ while employing the FEDHYPER scheduler. We adopted the FEDHYPER-G variant, using a bound value of 3 and a *Constant* local schedule $\eta_l^{(r)}$.
3) *Our CS*, that handles both initialization values and round-based schedules as defined in Table II.

In both FEDEXP *CS* and FEDHYPER *CS*, the grids for the initial learning rates are those reported in the first two rows of Table II. The total number of explored initialization pairs is 70, derived from 10 values for $\eta_{g,0}$ and 7 values for $\eta_{l,0}$. For the three configuration spaces, we deployed FEDAVGM as the global optimizer.

Since FEDEXP and FEDHYPER were designed to be robust

TABLE IX
ROBUSTNESS TO LEARNING RATE INITIALIZATION VALUES IN FEDEXP CS, FEDHYPER CS, AND OUR CS (CON-LIN SCHEDULES). COUNT OF SUCCESSFUL INITIALIZATION PAIRS, MINIMUM, MEAN, AND MAXIMUM TOP-1 ACCURACY (IN %) ON THE TEST-SET OVER THE CONFIGURATION SPACE. RESULTS ON CIFAR-10 TRAINED WITH FEDAVGM FOR 2000 ROUNDS.

	FEDEXP CS	**FEDHYPER CS**	**Our CS**
Count	30	50	30
Min. Top-1	77.72	70.84	49.39
Mean Top-1	79.21	76.80	75.81
Max. Top-1	80.50	79.45	83.48

TABLE X
ROBUSTNESS TO LEARNING RATE INITIALIZATION VALUES IN FEDEXP CS, FEDHYPER CS, AND OUR CS (CON-LIN SCHEDULES). COUNT OF SUCCESSFUL INITIALIZATION PAIRS, MINIMUM, MEAN, AND MAXIMUM TOP-1 ACCURACY (IN %) ON THE TEST-SET OVER THE CONFIGURATION SPACE. RESULTS ON CIFAR-100 TRAINED WITH FEDAVGM FOR 2000 ROUNDS.

	FEDEXP CS	**FEDHYPER CS**	**Our CS**
Count	50	60	47
Min. Top-1	10.77	15.31	10.51
Mean Top-1	32.93	41.01	36.61
Max. Top-1	44.29	52.88	56.96

TABLE XI
ROBUSTNESS TO LEARNING RATE INITIALIZATION VALUES IN FEDEXP CS, FEDHYPER CS, AND OUR CS (CON-LIN SCHEDULES). COUNT OF SUCCESSFUL INITIALIZATION PAIRS, MAXIMUM, MEAN, AND MINIMUM PERPLEXITY (PPL) ON THE TEST-SET OVER THE CONFIGURATION SPACE. RESULTS ON PTB TRAINED WITH FEDAVGM FOR 2000 ROUNDS.

	FEDEXP CS	**FEDHYPER CS**	**Our CS**
Count	30	50	45
Max. PPL	211.22	416.15	639.01
Mean PPL	176.37	222.09	261.03
Min. PPL	133.55	132.17	115.18

against variations in learning rate initialization values, we incorporated this aspect into our analysis. Specifically, we counted the number of initialization pairs that resulted in successful model training[1]. Additionally, we computed the minimum, mean, and maximum performance on the test set for these successful configurations. The results are presented in Tables IX–XI. For comparison with round-based scheduling, the column *Our CS* collects similar statistics from trainings configured by grid-searching over the same set of initialization pairs used in FEDEXP/FEDHYPER *CS* and using round-based schedules. For each benchmark, we reported these statistics for the best-performing schedule: *Con-Lin* for CIFAR-10/100 and *Con-Exp* for PTB.

The collected results confirm the effectiveness of gradient-based schedules in mitigating the negative impact of sub-optimal initialization pairs. For instance, in CIFAR-10 (Table IX), FEDHYPER *CS* achieves a higher number of successful runs compared to *Our CS* (50 vs. 30). Furthermore, both FEDEXP *CS* and FEDHYPER *CS* show lower variability across different initialization pairs, as indicated by the smaller gap

[1]A successful training gets an accuracy that exceeds random guessing

TABLE XII
TOP-1 ACCURACY (AND LOSS FROM THE LAST COLUMN) OF THE BEST-PERFORMING CONFIGURATIONS IN A FEDEXP CS, FEDHYPER CS, AND OUR CS WITH T=R (CON-LIN SCHEDULES). RESULTS ON CIFAR-10 TRAINED WITH FEDAVGM.

R	**FEDEXP CS**	**FEDHYPER CS**	**Our CS with T=R**
500	59.81 (-18.76)	75.52 (-2.96)	78.48 (0.00)
1000	73.24 (-8.46)	77.00 (-4.70)	81.70 (0.00)
1500	77.98 (-5.05)	78.05 (-4.98)	83.03 (0.00)

TABLE XIII
TOP-1 ACCURACY (AND LOSS FROM THE LAST COLUMN) OF THE BEST-PERFORMING CONFIGURATIONS IN A FEDEXP CS, FEDHYPER CS, AND OUR CS WITH T=R (CON-LIN SCHEDULES). RESULTS ON CIFAR-100 TRAINED WITH FEDAVGM.

R	**FEDEXP CS**	**FEDHYPER CS**	**Our CS with T=R**
500	21.77 (-23.71)	40.18 (-5.30)	45.48 (0.00)
1000	32.54 (-18.92)	48.42 (-3.04)	51.46 (0.00)
1500	40.25 (-14.05)	51.41 (-2.89)	54.30 (0.00)

TABLE XIV
PERPLEXITY (AND LOSS FROM THE LAST COLUMN) OF THE BEST-PERFORMING CONFIGURATIONS IN A FEDEXP CS, FEDHYPER CS, AND OUR CS WITH T=R (CON-EXP SCHEDULES). RESULTS ON PTB TRAINED WITH FEDAVGM.

R	**FEDEXP CS**	**FEDHYPER CS**	**Our CS with T=R**
500	174.76 (+57.34)	152.38 (+34.96)	117.42 (0.00)
1000	128.20 (+9.99)	120.96 (+2.75)	118.21 (0.00)
1500	130.32 (+14.08)	125.02 (+8.78)	116.24 (0.00)

between minimum and maximum accuracy compared to *Our CS*. However, the highest accuracy level is attained with the round-based schedules in *Our CS*, showing an accuracy gain of 2.98% compared to FEDEXP *CS* and 4.03% over FEDHYPER *CS*. Similar trends can be inferred in the analyses of CIFAR-100 (Table X) and PTB (Table XI). However, gradient-based schedules exhibit higher performance variability across different learning rate initialization settings, suggesting that their robustness may vary with the dataset. For instance, in CIFAR-100, the gap between the most and least accurate configurations in FEDEXP *CS* is substantial, precisely 33.52% (10.77% vs. 44.29%) compared to just 2.78% on CIFAR-10. This analysis indicates that configuration spaces with gradient-based schedules can simplify the tuning process and provide higher robustness to sub-optimal initializations, albeit with some performance trade-offs. On the other hand, configuration spaces with round-based schedules may require more extensive searches but can yield superior performance.

Moreover, gradient-based schedules do not adapt to the available training budget. We evaluated the configuration spaces with gradient-based schedules under varying training budgets to understand the implications of this limitation. Tables XII–XIV show the collected results, reporting the test performance for each round budget and the best-performing configurations in *Our CS* with $T = R$. Gradient-based schedules suffer from significant performance losses across all benchmarks and round budgets. This is evident in FEDEXP *CS*, which experiences up to a 23.71% decrease in accuracy

for $R = 500$ on the CIFAR-100 dataset. FEDHYPER *CS* also shows a substantial drop of 5.20%. This analysis provides further evidence of the efficiency of budget-aware schedules in constrained training scenarios.

VI. CONCLUSIONS

This work aimed to demonstrate the significance of two-sided learning rate schedule optimization in evaluating federated learning (FL) performance. We showed that optimizing the initial learning rate values alone is insufficient for achieving the best results, and considerable optimization improvements exist when learning rate schedules are treated as direct variables, particularly under constrained training resources.

Our extensive experimental campaign emphasized the need for revisited local and global learning rate tuning. An optimal setting tailored to the round budget can significantly enhance training efficiency, an aspect largely overlooked in existing FL research but paramount to improving FL's applicability across various contexts and scenarios. Additionally, our parametric analyses revealed that tuning learning rate schedules can lead to previously unexplored resource savings. Thus, the validation and assessment of any future FL optimization method or algorithm should include the optimization of learning rate schedules, with re-tuning whenever the training budget varies. Failing to do so risks yielding incomplete or misleading conclusions.

Building upon these findings, we envision future work focusing on new search algorithms capable of efficiently navigating the proposed configuration space, further enhancing FL performance. Moreover, exploring other types of learning rate schedules could yield additional benefits in terms of accuracy and convergence speed in FL.

REFERENCES

[1] H. Woisetschläger, A. Erben, B. Marino, S. Wang, N. D. Lane, R. Mayer, and H. Jacobsen, "Federated learning priorities under the european union artificial intelligence act," *arXiv:2402.05968*, 2024.

[2] V. Rey, P. M. S. Sánchez, A. H. Celdrán, and G. Bovet, "Federated learning for malware detection in iot devices," *Comput. Networks*, vol. 204, p. 108693, 2022.

[3] S. Ramaswamy, R. Mathews, K. Rao, and F. Beaufays, "Federated learning for emoji prediction in a mobile keyboard," *arXiv:1906.04329*, 2019.

[4] A. Hard, K. Rao, R. Mathews, F. Beaufays, S. Augenstein, H. Eichner, C. Kiddon, and D. Ramage, "Federated learning for mobile keyboard prediction," *arXiv:1811.03604*, 2018.

[5] J. Wang *et al.*, "A field guide to federated optimization," *arXiv:2107.06917*, 2021.

[6] M. Morafah, W. Wang, and B. Lin, "A practical recipe for federated learning under statistical heterogeneity experimental design," *IEEE Trans. Artif. Intell.*, vol. 5, no. 4, pp. 1708–1717, 2024.

[7] T. H. Hsu, H. Qi, and M. Brown, "Measuring the effects of non-identical data distribution for federated visual classification," *arXiv:1909.06335*, 2019.

[8] S. J. Reddi, Z. Charles, M. Zaheer, Z. Garrett, K. Rush, J. Konečný, S. Kumar, and H. B. McMahan, "Adaptive federated optimization," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2021.

[9] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated optimization in heterogeneous networks," in *Proc. Mach. Learn. Syst. (MLSys)*, 2020, pp. 429–450.

[10] S. P. Karimireddy, S. Kale, M. Mohri, S. J. Reddi, S. U. Stich, and A. T. Suresh, "SCAFFOLD: stochastic controlled averaging for federated learning," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2020, pp. 5132–5143.

[11] Q. Li, Y. Diao, Q. Chen, and B. He, "Federated learning on non-iid data silos: An experimental study," in *Proc. Int. Conf. Data Eng. (ICDE)*, 2022, pp. 965–978.

[12] S. Vahidian, M. Morafah, C. Chen, M. Shah, and B. Lin, "Rethinking data heterogeneity in federated learning: Introducing a new notion and standard benchmarks," *IEEE Trans. Artif. Intell.*, vol. 5, no. 3, pp. 1386–1397, 2024.

[13] H. Yang, M. Fang, and J. Liu, "Achieving linear speedup with partial worker participation in non-iid federated learning," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2021.

[14] Y. Wang, L. Lin, and J. Chen, "Communication-efficient adaptive federated learning," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2022, pp. 22 802–22 838.

[15] N. Dhawan, N. Mitchell, Z. Charles, Z. Garrett, and G. K. Dziugaite, "Leveraging function space aggregation for federated learning at scale," *Trans. Mach. Learn. Res.*, vol. 2024, 2024.

[16] Y. J. Cho, D. Jhunjhunwala, T. Li, V. Smith, and G. Joshi, "Maximizing global model appeal in federated learning," *Trans. Mach. Learn. Res.*, vol. 2024, 2024.

[17] I. Tenison, S. A. Sreeramadas, V. Mugunthan, E. Oyallon, I. Rish, and E. Belilovsky, "Gradient masked averaging for federated learning," *Trans. Mach. Learn. Res.*, vol. 2023, 2023.

[18] K. A. Bonawitz *et al.*, "Towards federated learning at scale: System design," in *Proc. Mach. Learn. Syst. (MLSys)*, 2019, pp. 374–388.

[19] H. Zhang, L. Fu, M. Zhang, P. Hu, X. Cheng, P. Mohapatra, and X. Liu, "Federated learning hyperparameter tuning from a system perspective," *IEEE Internet Things J.*, vol. 10, no. 16, pp. 14 102–14 113, 2023.

[20] J. M. P. Ullauri, X. Zhang, A. Bravalheri, R. Nejabati, and D. Simeonidou, "Federated hyperparameter optimisation with flower and optuna," in *Proc. ACM Symp. Appl. Comput. (SAC)*, 2023, pp. 1209–1216.

[21] Z. Wang, W. Kuang, C. Zhang, B. Ding, and Y. Li, "Fedhpo-bench: A benchmark suite for federated hyperparameter optimization," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2023, pp. 35 908–35 948.

[22] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2015.

[23] Y. Sun, L. Shen, H. Sun, L. Ding, and D. Tao, "Efficient federated learning via local adaptive amended optimizer with linear speedup," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 12, pp. 14 453–14 464, 2023.

[24] X. Li, K. Huang, W. Yang, S. Wang, and Z. Zhang, "On the convergence of fedavg on non-iid data," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2020.

[25] D. A. E. Acar, Y. Zhao, R. M. Navarro, M. Mattina, P. N. Whatmough, and V. Saligrama, "Federated learning based on dynamic regularization," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2021.

[26] D. Jhunjhunwala, S. Wang, and G. Joshi, "Fedexp: Speeding up federated averaging via extrapolation," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2023.

[27] K. Mo, C. Chen, J. Li, H. Xu, and C. J. Xue, "Two-dimensional learning rate decay: Towards accurate federated learning with non-iid data," in *Proc. Int. Jt. Conf. Neural Netw. (IJCNN)*, 2021, pp. 1–7.

[28] Z. Wang, J. Wang, and A. Li, "Fedhyper: A universal and robust learning rate scheduler for federated learning with hypergradient descent," 2024.

[29] M. Li, E. Yumer, and D. Ramanan, "Budgeted training: Rethinking deep neural network training under resource constraints," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2020.

[30] J. Chen, C. R. Wolfe, and T. Kyrillidis, "REX: revisiting budgeted training with an improved schedule," in *Proc. Mach. Learn. Syst. (MLSys)*, 2022, pp. 64–76.

[31] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proc. Int. Conf. Artif. Intell. Stat. (AISTATS)*, 2017, pp. 1273–1282.

[32] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," 2009.

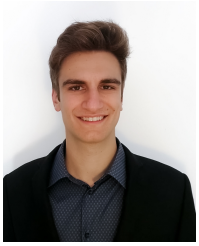
[33] Z. Wang, W. Zhang, X. Wu, and X. Wang, "Matched averaging federated learning gesture recognition with wifi signals," in *Proc. 7th Int. Conf. Big Data Comput. Commun. (BigCom)*, 2021, pp. 38–43.

[34] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2016, pp. 770–778.

[35] K. Hsieh, A. Phanishayee, O. Mutlu, and P. B. Gibbons, "The non-iid data quagmire of decentralized machine learning," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2020, pp. 4387–4398.

[36] M. P. Marcus, B. Santorini, and M. A. Marcinkiewicz, "Building a large annotated corpus of english: The penn treebank," *Comput. Linguistics*, vol. 19, no. 2, pp. 313–330, 1993.

[37] W. Zaremba, I. Sutskever, and O. Vinyals, "Recurrent neural network regularization," *arXiv:1409.2329*, 2014.
[38] Y. Zhou, Q. Ye, and J. Lv, "Communication-efficient federated learning with compensated overlap-fedavg," *IEEE Trans. Parallel Distributed Syst.*, vol. 33, no. 1, pp. 192–205, 2022.
[39] S. Yuan, B. Cao, Y. Sun, Z. Wan, and M. Peng, "Secure and efficient federated learning through layering and sharding blockchain," *IEEE Trans. Netw. Sci. Eng.*, vol. 11, no. 3, pp. 3120–3134, 2024.



Erich Malan (Graduate Student Member, IEEE) received the M.Sc. degree in Data Science and Engineering from the Politecnico di Torino in 2021. He is currently pursuing the Ph.D. degree with the Department of Control and Computer Engineering, Politecnico di Torino. His main research interest focuses on distributed artificial intelligence.

Valentino Peluso (Member, IEEE) received the M.Sc. degree in electronic engineering and the Ph.D. degree in computer engineering from the Politecnico di Torino. He is currently an Assistant Professor with the Department of Control and Computer Engineering, Politecnico di Torino. His main research interest focuses on design automation of digital circuits, low-power embedded systems, energy-efficient edge artificial intelligence, distributed and federated learning in IoT networks.

Andrea Calimera (Member, IEEE) received the M.Sc. degree in Electronic Engineering and the Ph.D. in Computer Engineering from the Politecnico di Torino, where he is currently a Full Professor of Computer Engineering. His research interests include design automation of digital circuits and embedded systems and applications, focusing on synthesis and optimization techniques for low-power consumption, energy efficiency, and reliability.

Enrico Macii (Fellow, IEEE) received the Laurea degree in electrical engineering from the Politecnico di Torino, Turin, Italy, the Laurea degree in computer science from the Università di Torino, Turin, and the Ph.D. degree in computer engineering from the Politecnico di Torino received respectively in 1990, 1991, and 1995. He is currently a Full Professor of Computer Engineering with the Politecnico di Torino. His research interests include the design of electronic digital circuits and systems, with a particular emphasis on low-power consumption aspects.