Adaptive Hybrid Model Pruning in Federated Learning through Loss Exploration

Christian Internò
CITEC, Bielefeld University, Germany
christian.interno@uni-bielefeld.de

Elena Raponi
LIACS, Leiden University, Netherlands
e.raponi@liacs.leidenuniv.nl

Niki van Stein
LIACS, Leiden University, Netherlands
n.van.stein@liacs.leidenuniv.nl

Thomas Bäck
LIACS, Leiden University, Netherlands
T.H.W.Baeck@liacs.leidenuniv.nl

Markus Olhofer
Honda Research Institute EU, Germany
markus.olhofer@honda-ri.de

Yaochu Jin
Westlake University, China
jinyaochu@westlake.edu.cn

Barbara Hammer
CITEC, Bielefeld University, Germany
bhammer@techfak.uni-bielefeld.de

Abstract

The rapid proliferation of smart devices coupled with the advent of 6G networks has profoundly reshaped the domain of collaborative machine learning. Alongside growing privacy-security concerns in sensitive fields, these developments have positioned federated learning (FL) as a pivotal technology for decentralized model training. Despite its vast potential, specially in the age of complex foundation models, FL encounters challenges such as elevated communication costs, computational constraints, and the complexities of non-IID data distributions. We introduce `AutoFLIP`, an innovative approach that utilizes a federated loss exploration phase to drive adaptive hybrid pruning, operating in a structured and unstructured way. This innovative mechanism automatically identifies and prunes model substructure by distilling knowledge on model gradients behavior across different non-IID client losses topology, thereby optimizing computational efficiency and enhancing model performance on resource-constrained scenarios. Extensive experiments on various datasets and FL tasks reveal that `AutoFLIP` not only efficiently accelerates global convergence, but also achieves superior accuracy and robustness compared to traditional methods. On average, `AutoFLIP` reduces computational overhead by 48.8% and communication costs by 35.5%, while improving global accuracy. By significantly reducing these overheads, `AutoFLIP` offer the way for efficient FL deployment in real-world applications for a scalable and broad applicability.

1 Introduction

The proliferation of smart devices at the network edge, coupled with advancements in 6G networks, has created a decentralized setting [31, 47]. Multiple participants store their data locally, which offers an opportunity for collaborative model training, enhancing robustness and generalization. Distributing the computational load across these devices results in faster training times and lower energy

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

consumption compared to centralized approaches [56, 3]. However, collaborative Machine Learning (ML) faces significant challenges [26]. Efficient communication and coordination among participants are crucial, as each device holds only a subset of the data. This requires designing algorithms that minimize data exchange while ensuring high-quality model convergence. Device heterogeneity, including differences in computational power, storage, and bandwidth, further complicates distributed training. Algorithms must adapt to such environments to scale up distributed learning. Privacy and security concerns, along with regulations like the European GDPR [1], the EU AI Act [8] and the U.S. Secure AI [54] Act add another layer of complexity [20]. With sensitive data distributed across various devices, ensuring the privacy of individual data points becomes essential [13]. For example, medical data stored in hospitals and personal devices is valuable for training diagnostic models but is also subject to strict privacy and security regulations [46].

In this context, Federated Learning (FL) [65] emerges as an effective strategy for training always more complex DL models while preserving the privacy of the data. FL facilitates collaborative model training across multiple devices without exposing local data. A central server, i.e., a global model, coordinates this process by aggregating updates from locally trained models, which ensures a secure learning environment. Current FL research focuses on enhancing privacy and adapting ML workflows for specific uses, often with predetermined ML model configurations. Tasks related to computer vision may involve well-known neural network (NN) architectures like VGG-16 [49] (138 million parameters) or ResNet-50 [18] (25.6 million parameters). However, these complex NN risk overfitting, especially with small training data sizes.

In the era of foundation models [3] becoming the norm in machine learning development, FL systems typically expect clients to have high-speed processors and sufficient computational power for local calculations and parameter updates. Yet, many edge devices, such as smartphones, wearable, and sensors, have limited computing and memory capacities, posing a challenge to DL model training systems [20]. Additionally, communicating DL models with millions of parameters presents significant obstacles for FL transmission [48, 2]. Therefore, using FL effectively with edge devices that have limited computational capabilities, while maintaining efficient communication, remains an active research question. FL's effectiveness is further hindered by the prevalence of non-IID data in real-world scenarios [65, 27]. non-IID data refers to the unique statistical properties of each client's dataset, reflecting their varied environments. This creates conflicting training goals for local and global models, leading to convergence towards different local optima. As a result, client model updates become biased, impeding global convergence [65, 27]. These challenges underscore the need for personalized and innovative approaches in FL, particularly in optimizing and compressing models to improve inference time, communication cost, energy efficiency, and complexity, all while maintaining satisfactory accuracy.

Our contribution. We introduce a novel automated federated learning approach via adaptive hybrid pruning (`AutoFLIP`), which uses a novel loss exploration mechanism to automatically prune and compress DL models. In our assumed single-server architecture, each client operates on the same initial deep NN structure that automatically prunes itself at each round, based on the extraction of shared knowledge from the federated loss exploration for an informed model compression. Specifically, by analyzing the variability of gradients during a preliminary local loss exploration phase, which provides insights into gradient behaviors on the loss landscapes across clients, and subsequent information aggregation, the DL models involved in a FL round are hybridly pruned automatically. This strategy allows for dynamically reducing the complexity of the models in FL environments, thereby optimizing performance with limited computational resources at the client level. With our experiments over various datasets, tasks, and realistic non-IID scenarios, we provide strong evidence of the effectiveness and efficiency of `AutoFLIP`.

Reproducibility. Our code for reproducing the experiments is available on Anonymous GitHub.[1]

2 Background and Related Work

Pruning in Deep Learning. Following the assumption that a DL model can contain a sub-network that represents the performance of the entire model after being trained, model pruning is a good strategy to reduce computational requirements of resource-constrained devices [41, 29, 24]. Most pruning approaches balance accuracy and sparsity during the inference stage by calculating the

[1] `https://github.com/ChristianInterno/AutoFLIP`

importance scores of parameters in a well-trained NN and removing those with lower scores. These scores can be derived from weight magnitudes [24, 16], first-order Taylor expansion of the loss function [42, 39], second-order Taylor expansion [29, 17, 40], and other variants [37, 50].

Another recent research direction in NN pruning focuses on improving training efficiency, divided into two categories: pruning at initialization and dynamic sparse training. Pruning at initialization involves pruning the original full-size model before training based on connection sensitivity [30], Hessian-gradient product [55], and synaptic flow [51]. However, since this method does not involve training data, the pruned model may be biased and not specialized for the task. Dynamic sparse training iteratively adjusts the pruned model structure during training while maintaining the desired sparsity [7, 9]. This approach requires memory-intensive operations due to the large search space, making it impractical for resource-constrained devices.

Initial attempts to use pruning for deploying deep neural networks on resource-limited devices have utilized pre-trained CNNs in a centralized setting [61, 33]. However, this approach can lead to reduced data privacy, higher costs, poor adaptation to local conditions, suboptimal performance on diverse data, and latency in real-time applications.

Hybrid Pruning in Deep Learning. Hybrid pruning techniques combine structured and unstructured pruning strategies to optimize both performance and efficiency in deep neural networks [36, 44, 12, 14]. Structured pruning [19] removes entire units like neurons, filters, or layers, leading to more hardware-efficient designs that are easier to implement on resource-constrained devices. On the other hand, unstructured pruning [32]focuses on removing individual weights, which can achieve higher sparsity levels and further compress the model, though it may require more complex hardware support [40].

Pruning in Federated Learning. The widely accepted FL standard is known as `FedAvg` [38]. It distributes a global model to clients for local training and aggregates it by averaging their parameters. Empirical studies have shown the robustness of this approach, even when handling non-convex optimization problems [5]. As a result, it is commonly used as a standard for evaluating newly developed FL protocols. In this study, we will compare the performance of the proposed `AutoFLIP` method to FedAvg, with different State-of-the-Art (SotA) FL pruning approachs, as tested in [58]. In fact, since data remains locally stored and cannot be shared, traditional centralized pruning approaches that rely on access to training data are not feasible in FL.

In the context of FL, there has been work focused on dynamic active pruning to increase communication efficiency during training. Liu et al. [35], Zhou et al. [64] introduced a method where pruning decisions are made dynamically based on the model's real-time performance evaluation, which significantly reduces the data exchanged during training but adds computational complexity to client devices. Jiang et al. [25] introduced `PruneFL`, a FL method that incorporates adaptive and distributed parameter pruning. Their approach utilizes an unstructured method that does not take advantage of the collective insights of participating clients to develop a cooperative structured pruning strategy. This is in contrast to the objectives of `AutoFLIP`, which seeks to harness client-specific knowledge to facilitate a structured approach to pruning. Lin et al. [34] introduced a novel approach for adaptive per-layer sparsity, however without incorporating any parameter aggregation scheme to reduce the error caused by pruning. This challenge was addressed by Tingting et al. [52] by moving the pruning process to the global model that works on a computationally more powerful server. The pruned model is distributed to each client, where it undergoes training. Subsequently, each client sends back to the server only the updated parameters, restoring the full structure of the model at the server. Although this study includes various parameter selection criteria from the literature, its pruning method does not incorporate the information gathered during model training. This contrasts with our strategy, `AutoFLIP`, which leverages such information to enhance the pruning process. Yu et al. [62] proposed Resource-aware Federated Foundation Models, focusing on integrating large transformer-based models into FL, with the limitation of not exploring other architectures. Our method, `AutoFLIP`, diverges by introducing a pruning strategy that avoids the need for continuous evaluation of parameter significance and is universally applicable across various FL aggregation algorithms and model architectures.

3 Preliminaries

Notation: We consider a total number of C clients. At each FL round, K clients are chosen and trained on different batches of size B for E epochs. The total number of rounds is R, which represents our termination criterion. For the exploration phase,C_{exp} is the number of clients selected, which, in this study, we take as the totality C of available clients. The exploration lasts for E_{exp} epochs.

3.1 Federated Learning

In the conventional FL setting, each client i ($1 \leq i \leq K$) possesses its own data distribution $p_i(x, y)$, where $x \in \mathbb{R}^d$ represents the d-dimensional input vector and $y \in \{1, \ldots, M\}$ is the corresponding label from M classes. Each client has a dataset D_i with N_i data points: $D_i = \{(x_i^{(1)}, y_i^{(1)}), \ldots, (x_i^{(N_i)}, y_i^{(N_i)})\}$. It is assumed that in a non-IID scenario the data distribution $p_i(x, y)$ varies across clients. These data distributions $p_i(x, y)$ are sampled from a family $\mathcal{E}$ of distributions. The objective is for the clients to collaboratively train a global model with parameters W_{global}, which will perform predictions on new data. The global loss function for a data point (x, y) is denoted by $\mathcal{L}(W_{\text{global}}, x, y)$, where the global objective function to be minimized is defined as: $\mathcal{L}(W_{\text{global}}) := \frac{1}{C} \sum_{i=1}^{C} \mathbb{E}_{(x_i, y_i) \sim p_i}[\mathcal{L}(W_{\text{global}}, x_i, y_i)]$, with $\mathbb{E}_{(x_i, y_i) \sim p_i}$ representing the expected loss over the data distribution p_i for each client i with parameters W.

The optimization process involves several key steps:
1. Client Selection: A subset of K clients is selected from the total C clients. **2. Local Training**: Each selected client i performs local training for E epochs using its local dataset D_i. The local training aims to minimize the local objective function $\mathcal{L}(W_i)$ using stochastic gradient descent (SGD): let W_i^r be the local model parameters of client i at round r, the update rule is given by: $W_i^{r+1} = W_i^r - \eta \nabla \mathcal{L}(W_i^r)$, where η is the learning rate. **3. Parameter Aggregation:** After local training, each client sends its updated parameters W_i^{r+1} to the central server. The server aggregates these parameters to form the new global model W_{global}^{r+1} using a weighted average: $W_{global}^{r+1} = \frac{1}{K} \sum_{i=1}^{K} W_i^{r+1}$. This iterative process is repeated for R rounds the termination criterion is met.

3.2 Problem Definition and Objective

i) Mitigates noise and biases in the client trajectories: By selectively pruning model parameters based on their contribution to loss topology variability, we aim to the trajectories of the different clients to help converging the global model more effectively despite the heterogeneous data distributions across clients. **ii) Enhance Computation and Communication Efficiency:** The hybrid pruning serves as a mechanism to decrease the number of model parameters that need to be communicated between clients and the central server. By eliminating less critical parameters and substructure, `AutoFLIP` reduces communication overhead and accelerates the computational process. This not only expedites the overall FL workflow but also makes it more feasible to deploy in resource-constrained scenarios.

4 Methodology

`AutoFLIP` is an automatized FL approach that utilizes informed pruning through a federated client loss exploration process. Inspired by the idea of utilizing agents with similar tasks as *scouts* which explore the conformation of different loss function landscapes from Internò et al. [22], Nikolić et al. [43], `AutoFLIP` introduces a preliminary step to the FL process, which we term *federated loss exploration* phase. Here, a C_{exp} portion of clients (or the totality C), which inherit their model structure from the global model, explore for a number of E_{exp} exploration epochs their loss landscape using its local dataset D_i. Based on this, for each client c_{exp_i}, we compute a local guidance matrix G_{local_i}, which records how important a certain parameter W_i (weight or bias) is in terms of loss variability 1. Afterward, we aggregate the information collected locally in a global pruning guidance matrix PG_{global} on the server 2, which will generate an informed pruning mask to guide the pruning of the client models 3. The pruning workflow of `AutoFLIP` is illustrated in Figure 1. Please note that the initial federated loss exploration, computation of parameter deviations, and definition of local guidance matrices occur only once at the beginning of the FL optimization process as a preliminary

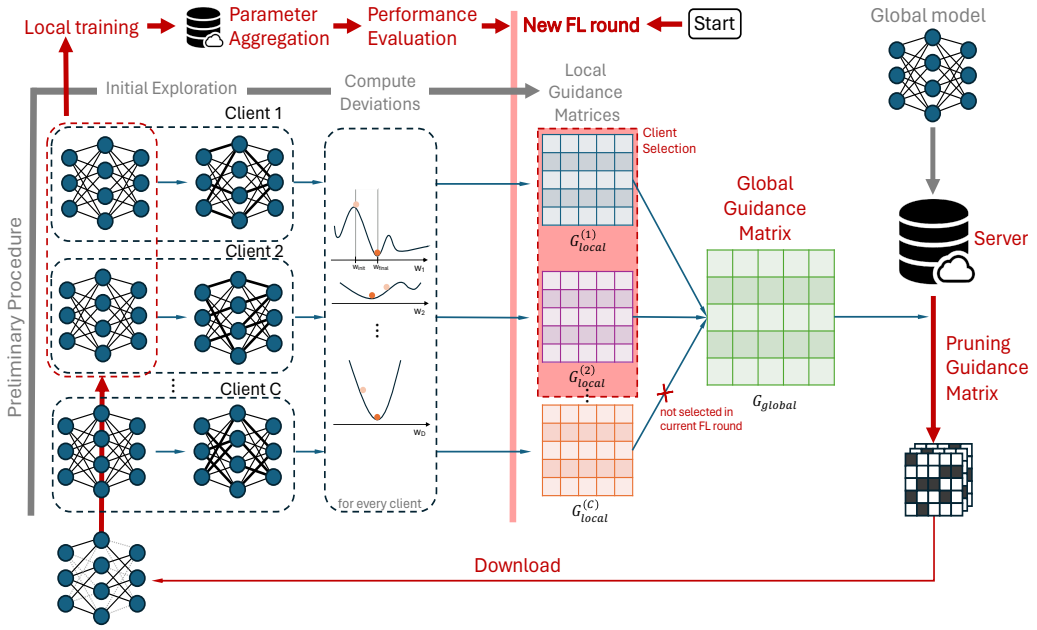

Figure 1: Illustration of the AutoFLIP pruning procedure. The local guidance matrices are computed a priori through the federated exploration phase. The global guidance matrix is computed by the server by aggregating the elements of the local guidance matrices corresponding to the clients participating in each FL round. The pruning mask is downloaded by the participant clients. All steps preliminary to the FL procedure are denoted in gray, while the steps intrinsic to the FL procedure with pruning are denoted in red.

procedure. In contrast, the global guidance matrix and subsequent pruning strategy are automatically redefined in each FL round, considering the clients participating in that round.

To summarize, the iterative procedure consists of **(1)** pruning local models using the updated pruning guidance matrix, **(2)** training the pruned local models, **(3)** aggregating the model parameters and **(4)** evaluating performance and updating the pruning guidance matrix in each FL round.

4.1 Federated Loss Exploration

In AutoFLIP, the model initialization phase is augmented by a crucial federated loss exploration phase, allowing clients to explore their loss function landscapes. We envision each client as an explorer that delves into different regions of their loss landscape. Through this exploration, they can identify crucial dimensions and those that can be disregarded based on their experience by quantifying gradient variability during the exploration. In other words, how much the loss is steep in that direction. Subsequently, they transmit this knowledge to the server, which updates a pruning guidance mask PG_{global}. This knowledge contained in the mask is then distilled among participating clients in each FL round to guide the evolution of client model structures within an informed pruning session.

To construct the mask PG_{global}, we begin with an initial exploration phase conducted on C_{exp} clients. In this study, we consider $C_{\text{exp}} = C$. In our study, we let explore the clients for $E_{\text{exp}} = 150$ epochs. Appendix A provides ablation studies for C_{exp} and E_{exp}. For each model parameter we evaluate its evolution in the search space during the loss exploration. This evaluation is conducted by calculating the deviation $D_{i,m}$ for the mth parameter of a client model i as the squared difference between the initial ($W_{i,m}^{\text{Initial}}$) and final ($W_{i,m}^{\text{Final}}$) parameter values after E_{exp} epochs of exploration:

$$D_{i,m} = (W_{i,m}^{\text{Initial}} - W_{i,m}^{\text{Final}})^2. \tag{1}$$

Using stochastic gradient descent for exploration, the deviation $D_{i,m}$ in Eq. (1) serves as a measure of gradient variability on the loss landscape for parameter m during the preliminary exploration

phase before the actual FL procedure. The greater the variation in the parameter space, the faster the improvements in loss: the update rule for a parameter in stochastic gradient descent is $W_{i,m}^{(e_{\text{exp}}+1)} = W_{i,m}^{(e_{\text{exp}})} - \eta \nabla \mathcal{L}_i \left(W_{i,m}^{(e_{\text{exp}})}; D_i \right)$), where $W_{i,m}^{(e_{\text{exp}})}$ and $W_{i,m}^{(e_{\text{exp}}+1)}$ are the values of the parameter m at the exploration epochs e_{exp} and $e_{\text{exp}} + 1$, η is the learning rate, and $\nabla \mathcal{L}_i \left(W_{i,m}^{(e_{\text{exp}})} \right)$ is the gradient of the loss function of client i with respect to the parameter m at epoch e_{exp} using its local dataset D_i. Given the gradient update rule, the deviation in $W_{i,m}$ from the initial to the final exploration epoch E_{exp} can be approximated to $W_{i,m}^{\text{Final}} - W_{i,m}^{\text{Initial}} \approx -\eta \sum_{t=1}^{E_{\text{exp}}} \nabla \mathcal{L}_i \left(W_{i,m}^{(t)}; D_i \right)$. To ensure non-negativity and highlight larger deviations more severely, we take the square of this value. This squared deviation measure $D_{i,m}$ approximates the square of the sum of gradients affecting the parameter evolution, indicating the significance of parameter updates on loss variability during the exploration phase. By squaring the sum of the gradients, we ensure that the deviation measure is always non-negative and that larger deviations are highlighted more severely than smaller ones.

The C_{exp} clients compile these deviations into a local matrix G_{local}, whose entries are the deviations for the model parameters. At each FL round, where only K clients are involved, the server aggregates the G_{local} matrices associated to those client to formulate G_{global} through a normalization process:

$$G_{\text{global}} = \frac{1}{K} \sum_{k=1}^{K} \frac{G_{\text{local}_k} - \min(G_{\text{local}})}{\max(G_{\text{local}}) - \min(G_{\text{local}})} \tag{2}$$

Here, the minimum and maximum values are taken over all G_{local_k} matrices for $k = 1, \ldots, K$. Each element of G_{global} thus represents the mean normalized deviation for each parameter, scaled between 0 and 1. This process ensures that no single client's G_{local} disproportionately influences G_{global} due the possible presence of outliers in terms of deviations $D_{i,m}$. A value closer to 0 indicates minimal deviation, suggesting gradient stability during the exploration, hence scarce relevance of the parameter itself. Conversely, values near 1 highlight significant parameter deviations, pointing to more dynamic and potentially insightful areas of the loss landscape. Then, a binarization process is applied to G_{global} where elements below T_{p} are set to 0 and those above are set to 1:

$$PG_{\text{global},m} = \begin{cases} 0 & \text{if } G_{\text{global},m} < T_{\text{p}} \\ 1 & \text{otherwise} \end{cases} \tag{3}$$

The threshold T_{p} directly determines the compression ratio of the model by setting the proportion of parameters to be pruned. T_{p} can be seen has a hard constraint on model size, which is sometimes needed due to devices' resource constraints. Given their smaller influence, parameters corresponding to 0 are marked for pruning, whereas those marked with 1 are retained, indicating important search directions within the model parameter space. During each FL round, the K participating clients update PG_{global} by incorporating their G_{local} deviation values derived from the initial loss exploration phase.

To select an appropriate T_{p}, consider the desired compression ratio for the model. This ratio reflects the extent to which the model needs to be compressed while maintaining acceptable performance. By carefully selecting T_{p} based on the desired compression ratio and empirical validation, we can achieve a well-balanced model that is both efficient and accurate, tailored to the specific needs of the FL task. In appendix B we conduct ablation studies on T_{p}.

The Proposed `AutoFLIP` Framework. Here our aim is to argument how the parameter pruning mechanism based on loss exploration enters a general FL edge training framework. Algorithm 1 provides an overview of the entire framework of the proposed `AutoFLIP` algorithm for FL. It is composed by the following steps.

Server initialization (Line 1). The server is initialized with a global model that it is sent to all the clients. At this stage, the total number of clients undergoing exploration, the number of exploration epochs, and the pruning threshold are also decided.

Exploration phase (Lines 2–3).

The preliminary exploration phase aimed at understanding the relevance of each parameter (weight or bias) in view of loss improvement starts. For each client participating (in this study we select all the available clients), a local guidance matrix storing parameter deviations is computed.

Mask update (Lines 5–7). A FL round starts. The server selects K clients that participate in the

Algorithm 1 `AutoFLIP` Algorithm

1: **Server Initialization:** Initial matrix $W_{\text{global}}^{(0)}$, number of clients for exploration C_{exp}, exploration epochs E_{exp}, pruning threshold T_{p}, FL rounds R, training epochs E, number of selected clients per round K
2: Server selects C_{exp} clients for exploration
3: $G_{\text{local}_i} = (W_i^{\text{Initial}} - W_i^{\text{Final}})^2, \forall i \in [1, C_{\text{exp}}]$
4: **for** round $r = 1$ to R **do**
5: Server selects K clients
6: Compute $G_{\text{global}}^{(r)}$ using Eq. (2)
7: Compute mask $PG_{\text{global}}^{(r)}$ using Eq. (3)
8: **for** client $k = 1$ to K **do**
9: $W_{k,\text{pruned}}^{(r)} = W_k^{(r)} \odot PG_{\text{global}}^{(r)}$
10: **for** each local epoch $e = 0$ to $E - 1$ **do**
11: $W_{k,\text{pruned}}^{(e+1)} = W_{k,\text{pruned}}^{(e)} - \eta \nabla L_k \left(W_{k,\text{pruned}}^{(e)} \right)$
12: **end for**
13: **end for**
14: $W_{\text{global}}^{(r)} = \frac{1}{K} \sum_{k=1}^{K} W_{k,\text{pruned}}^{(E)}$ ▷ This can be replaced with other FL aggregation algorithms
15: **end for**

round. Only the local guidance matrices of those clients are considered to compute a global guidance matrix, which is then used to generate a binary mask for pruning. The mask contains ones only for the parameters with normalized deviations higher than a prescribed threshold T_p.

Pruning (Lines 8–9). During each round, clients use the pruning mask to compress their models. This happens through element-wise multiplication between their weight matrix and PG_{global} at that FL round. Parameters aligned with a 0 in PG_{global} are pruned; those corresponding to a 1 are kept.

FL round with reduced client models (Lines 10–14). The standard algorithm FedAvg [38] is usedin this framework but AutoFLIP can be applied to other SotA FL aggregation algorithm. The pruned clients are trained. The server receives the local model updates and, upon aggregation, proceeds to update the global model with the FL aggregation strategy. Once updated, the global model is either ready for the next communication round or deemed ready for deployment if the convergence criteria are satisfied.

4.2 Robustness and Efficiency of `AutoFLIP`

Referring to [11, 10, 57, 60], we base our convergence guarantees on a federated stochastic aggregation scheme. The authors' assumptions on Lipschitz smoothness, convexity of local loss functions, unbiased gradient estimators, finite client answering times, and specific client aggregation weights form the theoretical backbone of `AutoFLIP`. These conditions ensure that the learning process remains stable and converges efficiently even in the presence of non-IID data distributions. With `AutoFLIP`, at each round, each client experiences the same pruning strategy with the pruning mask PG_{global}, resulting in a substantial decrease in the variance ($\sigma^2_{\Delta W}$) previously defined in Section 3.2 of weight updates for the global model. This uniform pruning strategy minimizes discrepancies in weight adjustments across clients by focusing updates on critical weights identified during the federated loss exploration phase. The reduction in variance helps to alleviate the bias caused by the non-IID setting, as shown in the work of [65], thus promoting better global convergence.

Furthermore, [59, 45, 53, 23] provide a theoretical foundation for which pruned NNs can effectively learn signals. They demonstrate that pruning preserves the signal's magnitude in features and reduces noise, leading to improved generalization. These studies highlight that pruning, when done correctly, does not degrade the model's capacity to learn but rather focuses the learning on more relevant features. By focusing on parameters with significant contributions to the loss topology, `AutoFLIP` ensures that the essential features are retained. As illustrated in Figure 2 for different NN, the parameters in G_{global} with minimal variability during the federated loss exploration phase are pruned, while those exhibiting high deviations are retained. Note that higher frequencies are recorded for smaller deviation values, indicating that many parameters, according to our pruning strategy, are non important. The high density of those parameters lead to the weights of entire channels or layers

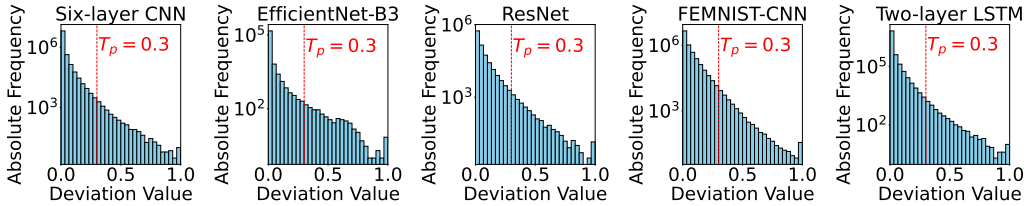

Figure 2: Distribution of parameter deviations in G_{global} after exploration. Absolute frequency in log-scale is shown for each normalized deviation. Higher frequencies are recorded for smaller deviation values, indicating that many parameters are irrelevant for loss improvement.

being set to zero. In such cases, we delete the entire substructure of the NN, resulting in a structured network reduction. This leads to an improvement in computing efficiency as detteiled in appendix E.

`AutoFLIP` enhances communication efficiency in FL by reducing model sizes transmitted between clients and the server, thus lowering bandwidth requirements for FL rounds. Its selective updating mechanism ensures only essential parameters, those significantly affecting global model convergence, are communicated. The appendices F, G and H provide a detailed analysis of how `AutoFLIP` accelerates inference and improves training efficiency, demonstrating its significant role in lowering computational costs and boosting FL's overall applicability.

5 Experiments

Inspired by [15], we benchmark `AutoFLIP` across established datasets to evaluate its robustness in various non-IID environments. We explore three distinct partitioning approaches for creating strongly non-IID conditions: a **Pathological non-IID scenario**, which involves clients using data from two distinct classes, employing MNIST with a six-layer CNN (7,628,484 parameters) and CIFAR10 with EfficientNet-B3 (10,838,784 parameters), a **Dirichlet-based non-IID scenario**, which utilizes the Dirichlet distribution to distribute data among clients, with varying class counts per client, using CIFAR100 with ResNet (23,755,900 parameters), and a **LEAF non-IID scenario**, which adopts the LEAF benchmark [4] with FEMNIST and Shakespeare datasets. For FEMNIST, a CNN architecture with 13,180,734 parameters is used. For Shakespeare, we consider a two-layer LSTM model with 5,040,000 parameters. Further details on these scenarios are provided in Appendix C.

5.1 Experimental Setup and Results

We evaluate `AutoFLIP` against both `FedAvg` without any model compression and with SotA algorithm e.g PruneFL [25], and FL-pruning with various parameter selection criteria: Random, L1, L2, Similarity, and BN mask, as described in [58]. The experimental setup involves $C = 20$ (for LEAF non-IID scenario we employ $C = 730$ for Shakespeare and $C = 660$ for FEMNIST) clients, a batch size $B = 350$, a local update epochs $E = 5$, and a learning rate $\eta = 0.0003$ over 200 total rounds R with $K = 5$ (for LEAF non-IID scenario $K = 20$) clients selected per round. We incorporate a server momentum of 0.9 and use an SGD optimizer with weight decay. The exploration phase consists of up to $E_{\text{exp}} = 150$ epochs, and the pruning threshold is set to $T_{\text{p}} = 0.3$. Data is divided into 80% for training and 20% for testing, with global model performance assessed by the average prediction accuracy on the test sets. To ensure statistical validity, each experiment is repeated 10 times. We measure the compression rate to evaluate model size reduction and its impact. Experiments were conducted with an Intel Xeon X5680, 128 GB of DDR4 RAM, and an NVIDIA TITAN X GPU.

Pathological non-IID) Here, `AutoFLIP` achieves an average client compression rate of x1.74. At each round, we remove on average 3244298 parameters of the six-layer CNN for each participant client. For the EfficientNet-B3, we obtain an average compression rate of x2.1 with 5677458 deleted parameters. For a fair comparison with the baselines, we ensure that the number of parameters pruned matches the compression ratio of AutoFLIP, quantified as 42% for the six-layer CNN and 52.38% for EfficientNet-B3.

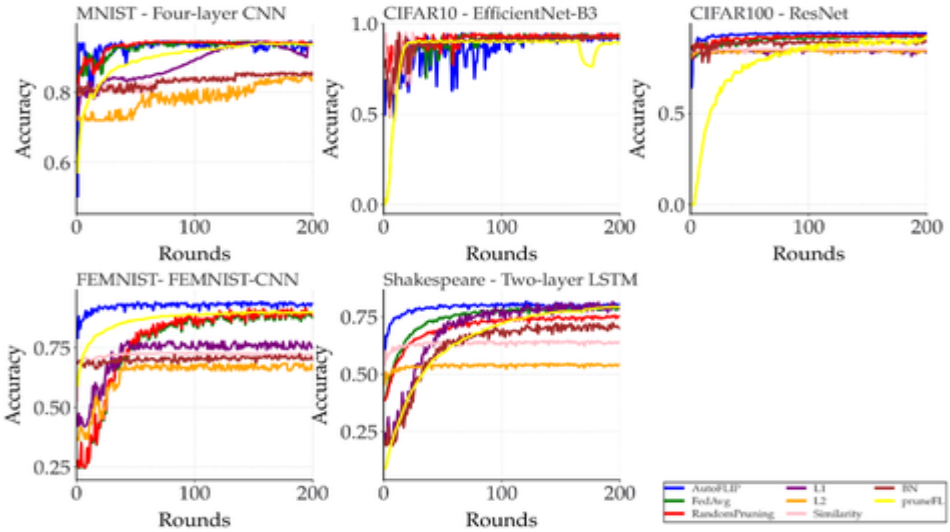

Figure 3: Average accuracy convergence profiles for the global model within the FL framework.

The first two subplots in Figure 3 show the evolution of global model accuracy during the FL rounds for the four-layer CNN with the MNIST dataset and for EfficientNet-B3 with the CIFAR10 dataset. Refer to Appendix D for the evolution of the loss metric. In the case of the MNIST, the early rounds of FL show that `AutoFLIP` achieves slightly higher accuracy compared to both `FedAvg` and the other FL pruning strategies, among which RandomPruning emerges as the top performer. This indicates a faster convergence rate for our proposed method. However, the performance of the baselines soon becomes comparable, with no clear superiority as the FL procedure progresses. We attribute this to the simplicity of the prediction tasks on the MNIST dataset compared to the excessive complexity of the four-layer CNN, which already possesses extremely good prediction capabilities that cannot be further enhanced by pruning. For the CIFAR10 dataset, we do not observe any advantage in using AutoFLIP over the other baselines. Surprisingly, all methods exhibit severe fluctuations in the accuracy convergence profiles up to FL round 100, after which they stabilize and become comparable.

Dirichlet-based non-IID) For ResNet, AutoFLIP achieves an average compression rate for the clients of x1.58, with 8,720,520 parameters pruned on average out of 23,755,900 total parameters. Hence, we adjust the percentage of parameters to be pruned to 36.71% for the different baselines.

The third subplot in Figure 3 illustrates the evolution of the global model accuracy during the FL rounds for ResNet on CIFAR100. Here, `AutoFLIP` exhibits a performance enhancement throughout the considered training rounds. At round 200, it achieves an accuracy of 0.987, compared to 0.918 for FedAvg and 0.925 for PruneFL. This enhancement signifies the robustness of `AutoFLIP`, showcasing its ability to maintain elevated performance levels when integrated with larger-complex neural networks and larger datasets.

LEAF non-IID) In this scenario, `AutoFLIP` achieves an average compression rate of x1.8 for 5858104 client parameters pruned out of 13180734. Hence, we adjust the number of parameters to be pruned for the different baselines to 44%. As observed in the last two subplots of Figure 3 for the FEMNIST and Shakespeare datasets, `AutoFLIP` consistently outperforms the other pruning strategies by a significant margin.

What stands out is the initial acceleration in convergence speed observed for `AutoFLIP`, firmly establishing it as a superior choice over`FedAvg` and the other FL baselines. Furthermore, this superiority persists throughout the entire FL training procedure. The final average accuracy values are 0.985 for `AutoFLIP`, 0.905 for FedAvg, and 0.935 for RandomPruning on the FEMNIST dataset. For the Shakespeare dataset, the values are 0.815, 0.783, and 0.738, respectively. Here, even L1 proves to be competitive, reaching a final accuracy equal to 0.802. However, it demonstrates inferior initial convergence.

6 Conclusion and Limitations

We introduced AutoFLIP, an innovative federated learning (FL) approach that employs informed pruning to optimize deep learning (DL) models on clients with limited computational resources. Through extensive experiments in various non-IID scenarios, AutoFLIP has demonstrated its ability to achieve better accuracy and significantly reduce computational and communication overheads. It enhances global convergence in federated settings and shows remarkable adaptability and scalability across diverse DL model architectures and multi-class datasets, particularly as the complexity of tasks increases.

Limitations. AutoFLIP shows promise but has limitations. It is primarily tested in the popular efficient single-server setting, not accounting for multi-server or hierarchical environments with diverse client capabilities and model structure. Our tests also assume standard conditions without data label noise.

Future Research Directions. AutoFLIP underscores its potential for future research avenues, such as leveraging loss exploration for guiding complex Neural Architecture Search (NAS) tasks. Enhancements will focus on refining AutoFLIP's dynamic and adaptive pruning to better client personalization. We aim to perform comparison analysis with other strategies from other domain such us like NAS or Client Dropout. Further, the impact on data privacy and defense against adversarial clients during the federated loss exploration phase has to assessed. Research will also explore the extension of AutoFLIP to more complex DL architectures and its integration into real-world applications across various domains such as healthcare and mobile computing.

Broader Impact. AutoFLIP enhances sustainability and efficiency in FL, reducing the energy footprint of training deep learning models. Its utility in sensitive sectors like healthcare and finance emphasizes its societal importance. However, deploying AutoFLIP requires careful consideration of ethical issues, including data privacy and biases. Proactive management and regulation are crucial to ensure its positive societal impact and responsible integration into critical fields.

References

[1] Regulation (eu) 2016/679 of the european parliament and of the council of 27 april 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing directive 95/46/ec (general data protection regulation) (text with eea relevance), May 2016.

[2] M. Asad, S. Shaukat, D. Hu, Z. Wang, E. Javanmardi, J. Nakazato, and M. Tsukada. Limitations and future aspects of communication costs in federated learning: A survey. *Sensors*, 23(17), 2023. ISSN 1424-8220. doi: 10.3390/s23177358.

[3] R. Bommasani and D. A. et al. On the opportunities and risks of foundation models, 2022. URL https://arxiv.org/abs/2108.07258.

[4] S. Caldas, S. M. K. Duddu, P. Wu, T. Li, J. Konečný, H. B. McMahan, V. Smith, and A. Talwalkar. Leaf: A benchmark for federated settings, 2019.

[5] R. Das, A. Acharya, A. Hashemi, S. Sanghavi, I. S. Dhillon, and U. Topcu. Faster non-convex federated learning via global and local momentum. In *Conference on Uncertainity in Artificial Intelligence (UAI) (UAI)*, 2022.

[6] L. Deng. The mnist database of handwritten digit images for machine learning research [best of the web]. *IEEE Signal Processing Magazine*, 2012.

[7] T. Dettmers and L. Zettlemoyer. Sparse networks from scratch: Faster training without losing performance. *CoRR*, abs/1907.04840, 2019. URL http://arxiv.org/abs/1907.04840.

[8] European Union. The eu ai act: All you need to know in 2024. https://bigid.com/resources/blog/eu-ai-act-2024/, 2024. Accessed: 2024-09-25.

[9] U. Evci, T. Gale, J. Menick, P. S. Castro, and E. Elsen. Rigging the lottery: Making all tickets winners. In H. D. III and A. Singh, editors, *Proceedings of the 37th International Conference*

on Machine Learning, volume 119 of *Proceedings of Machine Learning Research*, pages 2943–2952. PMLR, 13–18 Jul 2020. URL https://proceedings.mlr.press/v119/evci20a.html.

[10] A. Fallah, A. Mokhtari, and A. Ozdaglar. Personalized federated learning with theoretical guarantees: A model-agnostic meta-learning approach. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 3557–3568. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/24389bfe4fe2eba8bf9aa9203a44cdad-Paper.pdf.

[11] Y. Fraboni, R. Vidal, L. Kameni, and M. Lorenzi. A general theory for federated optimization with asynchronous and heterogeneous clients updates. *J. Mach. Learn. Res.*, 24:110:1–110:43, 2022. URL https://api.semanticscholar.org/CorpusID:249889335.

[12] X. Geng et al. Complex hybrid weighted pruning method for accelerating convolutional neural networks. *Scientific Reports*, 14(1):5570, 2024. doi: 10.1038/s41598-024-55942-5.

[13] M. M. Grynbaum and R. Mac. The times sues openai and microsoft over a.i. use of copyrighted work. https://www.nytimes.com/2023/12/27/business/media/new-york-times-open-ai-microsoft-lawsuit.html, 2023. Accessed: 2023-12-27.

[14] C. Guo and P. Li. Hybrid pruning method based on convolutional neural network sensitivity and statistical threshold. *Journal of Physics: Conference Series*, 2171(1):012055, jan 2022. doi: 10.1088/1742-6596/2171/1/012055. URL https://dx.doi.org/10.1088/1742-6596/2171/1/012055.

[15] S.-J. Hahn, M. Jeong, and J. Lee. Connecting low-loss subspace for personalized federated learning. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. ACM, aug 2022.

[16] S. Han, H. Mao, and W. J. Dally. Deep Compression: Compressing Deep Neural Networks with Pruning, Trained Quantization and Huffman Coding. 10 2015.

[17] B. Hassibi and D. Stork. Second order derivatives for network pruning: Optimal brain surgeon. In S. Hanson, J. Cowan, and C. Giles, editors, *Advances in Neural Information Processing Systems*, volume 5. Morgan-Kaufmann, 1992. URL https://proceedings.neurips.cc/paper_files/paper/1992/file/303ed4c69846ab36c2904d3ba8573050-Paper.pdf.

[18] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition, 2015.

[19] Y. He, X. Zhang, and J. Sun. Channel pruning for accelerating very deep neural networks. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 1389–1397, 2017. doi: 10.1109/ICCV.2017.417.

[20] K. Hoffpauir, J. Simmons, N. Schmidt, R. Pittala, I. Briggs, S. Makani, and Y. Jararweh. A survey on edge intelligence and lightweight machine learning support for future applications and services. *J. Data and Information Quality*, 15(2), jun 2023. ISSN 1936-1955. doi: 10.1145/3581759.

[21] T.-M. H. Hsu, H. Qi, and M. Brown. Measuring the effects of non-identical data distribution for federated visual classification, 2019.

[22] C. Internò, M. Olhofer, Y. Jin, and B. Hammer. Federated loss exploration for improved convergence on non-iid data. In *2024 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8, 2024. doi: 10.1109/IJCNN60899.2024.10651455.

[23] B. Isik, T. Weissman, and A. No. An information-theoretic justification for model pruning. In G. Camps-Valls, F. J. R. Ruiz, and I. Valera, editors, *Proceedings of The 25th International Conference on Artificial Intelligence and Statistics*, volume 151 of *Proceedings of Machine Learning Research*, pages 3821–3846. PMLR, 28–30 Mar 2022. URL https://proceedings.mlr.press/v151/isik22a.html.

[24] S. A. Janowsky. Pruning versus clipping in neural networks. *Phys. Rev. A*, 39:6600–6603, Jun 1989. doi: 10.1103/PhysRevA.39.6600. URL https://link.aps.org/doi/10.1103/PhysRevA.39.6600.

[25] Y. Jiang, S. Wang, V. Valls, B. J. Ko, W.-H. Lee, K. K. Leung, and L. Tassiulas. Model Pruning Enables Efficient Federated Learning on Edge Devices. *IEEE Transactions on Neural Networks and Learning Systems*, 34(12):10374–10386, Dec. 2023. ISSN 2162-237X, 2162-2388. doi: 10.1109/TNNLS.2022.3166101.

[26] P. Kairouz, H. B. McMahan, B. Avent, A. Bellet, M. Bennis, A. N. Bhagoji, K. Bonawitz, Z. Charles, G. Cormode, R. Cummings, R. G. L. D'Oliveira, H. Eichner, S. E. Rouayheb, D. Evans, J. Gardner, Z. Garrett, A. Gascón, B. Ghazi, P. B. Gibbons, M. Gruteser, Z. Harchaoui, C. He, L. He, Z. Huo, B. Hutchinson, J. Hsu, M. Jaggi, T. Javidi, G. Joshi, M. Khodak, J. Konečný, A. Korolova, F. Koushanfar, S. Koyejo, T. Lepoint, Y. Liu, P. Mittal, M. Mohri, R. Nock, A. Özgür, R. Pagh, M. Raykova, H. Qi, D. Ramage, R. Raskar, D. Song, W. Song, S. U. Stich, Z. Sun, A. T. Suresh, F. Tramèr, P. Vepakomma, J. Wang, L. Xiong, Z. Xu, Q. Yang, F. X. Yu, H. Yu, and S. Zhao. Advances and open problems in federated learning, 2021. URL https://arxiv.org/abs/1912.04977.

[27] S. P. Karimireddy, S. Kale, M. Mohri, S. Reddi, S. Stich, and A. T. Suresh. SCAFFOLD: Stochastic controlled averaging for federated learning. Proceedings of Machine Learning Research, 2020.

[28] A. Krizhevsky. Learning multiple layers of features from tiny images. *University of Toronto*, 2012.

[29] Y. LeCun, J. Denker, and S. Solla. Optimal brain damage. In D. Touretzky, editor, *Advances in Neural Information Processing Systems*, volume 2. Morgan-Kaufmann, 1989. URL https://proceedings.neurips.cc/paper_files/paper/1989/file/6c9882bbac1c7093bd25041881277658-Paper.pdf.

[30] N. Lee, T. Ajanthan, and P. Torr. SNIP: SINGLE-SHOT NETWORK PRUNING BASED ON CONNECTION SENSITIVITY. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=B1VZqjAcYX.

[31] K. B. Letaief, W. Chen, Y. Shi, J. Zhang, and Y.-J. A. Zhang. The roadmap to 6g: Ai empowered wireless networks. *IEEE Communications Magazine*, 57(8):84–90, 2019. doi: 10.1109/MCOM.2019.1900271.

[32] T. Liang, J. Glossner, L. Wang, S. Shi, and X. Zhang. Pruning and quantization for deep neural network acceleration: A survey, 2021. URL https://arxiv.org/abs/2101.09671.

[33] M. Lin, R. Ji, Y. Wang, Y. Zhang, B. Zhang, Y. Tian, and L. Shao. HRank: Filter Pruning using High-Rank Feature Map, Mar. 2020. arXiv:2002.10179 [cs].

[34] R. Lin, Y. Xiao, T.-J. Yang, D. Zhao, L. Xiong, G. Motta, and F. Beaufays. Federated Pruning: Improving Neural Network Efficiency with Federated Learning, Sept. 2022. arXiv:2209.06359 [cs].

[35] S. Liu, G. Yu, R. Yin, J. Yuan, L. Shen, and C. Liu. Joint Model Pruning and Device Selection for Communication-Efficient Federated Edge Learning. *IEEE Transactions on Communications*, 70(1):231–244, Jan. 2022. ISSN 1558-0857. doi: 10.1109/TCOMM.2021.3124961. Conference Name: IEEE Transactions on Communications.

[36] Z. Liu, J. Li, Z. Shen, G. Huang, S. Yan, and C. Zhang. Learning efficient convolutional networks through network slimming. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 2736–2744, 2017. doi: 10.1109/ICCV.2017.177.

[37] C. Louizos, M. Welling, and D. P. Kingma. Learning sparse neural networks through l_0 regularization. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018. URL https://openreview.net/forum?id=H1Y8hhg0b.

[38] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas. Communication-efficient learning of deep networks from decentralized data, 2023.

[39] P. Molchanov, S. Tyree, T. Karras, T. Aila, and J. Kautz. Pruning convolutional neural networks for resource efficient inference. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. URL https://openreview.net/forum?id=SJGCiw5gl.

[40] P. Molchanov, A. Mallya, S. Tyree, I. Frosio, and J. Kautz. Importance estimation for neural network pruning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11264–11272, 2019. doi: 10.1109/CVPR.2019.00932.

[41] M. C. Mozer and P. Smolensky. Skeletonization: A technique for trimming the fat from a network via relevance assessment. In D. Touretzky, editor, *Advances in Neural Information Processing Systems*, volume 1. Morgan-Kaufmann, 1988. URL https://proceedings.neurips.cc/paper_files/paper/1988/file/07e1cd7dca89a1678042477183b7ac3f-Paper.pdf.

[42] M. C. Mozer and P. Smolensky. Skeletonization: A technique for trimming the fat from a network via relevance assessment. In D. Touretzky, editor, *Advances in Neural Information Processing Systems*, volume 1. Morgan-Kaufmann, 1988. URL https://proceedings.neurips.cc/paper_files/paper/1988/file/07e1cd7dca89a1678042477183b7ac3f-Paper.pdf.

[43] D. Nikolić, D. Andrić, and V. Nikolić. Guided Transfer Learning, Mar. 2023. arXiv:2303.16154 [cs].

[44] A. Onan, S. Korukoğlu, and H. Bulut. A hybrid ensemble pruning approach based on consensus clustering and multi-objective evolutionary algorithm for sentiment classification. *Information Processing Management*, 53(4):814–833, 2017. ISSN 0306-4573. doi: https://doi.org/10.1016/j.ipm.2017.02.008. URL https://www.sciencedirect.com/science/article/pii/S0306457316301480.

[45] W. T. Redman, M. FONOBEROVA, R. Mohr, Y. Kevrekidis, and I. Mezic. An operator theoretic view on pruning deep neural networks. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=pWBNOgdeURp.

[46] N. Rieke, J. Hancox, W. Li, F. Milletarì, H. R. Roth, S. Albarqouni, S. Bakas, M. N. Galtier, B. A. Landman, K. Maier-Hein, S. Ourselin, M. Sheller, R. M. Summers, A. Trask, D. Xu, M. Baust, and M. J. Cardoso. The future of digital health with federated learning. *npj Digital Medicine*, 3(1):119, Sep 2020. ISSN 2398-6352. doi: 10.1038/s41746-020-00323-1. URL https://doi.org/10.1038/s41746-020-00323-1.

[47] W. Saad, M. Bennis, and M. Chen. A vision of 6g wireless systems: Applications, trends, technologies, and open research problems. *IEEE Network*, 34(3):134–142, 2020. doi: 10.1109/MNET.001.1900287.

[48] N. Shlezinger, S. Rini, and Y. C. Eldar. The communication-aware clustered federated learning problem. In *2020 IEEE International Symposium on Information Theory (ISIT)*, 2020.

[49] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition, 2015.

[50] S. P. Singh and D. Alistarh. Woodfisher: Efficient second-order approximation for neural network compression. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 18098–18109. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/d1ff1ec86b62cd5f3903ff19c3a326b2-Paper.pdf.

[51] H. Tanaka, D. Kunin, D. L. K. Yamins, and S. Ganguli. Pruning neural networks without any data by iteratively conserving synaptic flow. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA, 2020. Curran Associates Inc. ISBN 9781713829546.

[52] W. Tingting, C. Song, and P. Zeng. Efficient federated learning on resource-constrained edge devices based on model pruning. *Complex & Intelligent Systems*, 9, 06 2023. doi: 10.1007/s40747-023-01120-5.

[53] M. Tukan, L. Mualem, and A. Maalouf. Pruning neural networks via coresets and convex geometry: Towards no assumptions. In A. H. Oh, A. Agarwal, D. Belgrave, and K. Cho, editors, *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=btpIaJiRx6z.

[54] U.S. Congress. Senators introduce bill on secure ai act of 2024 to congress. https://www.dataguidance.com/news/usa-senators-introduce-bill-secure-ai-act-2024-congress, 2024. Accessed: 2024-09-25.

[55] C. Wang, G. Zhang, and R. Grosse. Picking winning tickets before training by preserving gradient flow. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=SkgsACVKPH.

[56] S. Wang, T. Tuor, T. Salonidis, K. K. Leung, C. Makaya, T. He, and K. Chan. Adaptive federated learning in resource constrained edge computing systems. *IEEE Journal on Selected Areas in Communications*, 37(6):1205–1221, 2019. doi: 10.1109/JSAC.2019.2904348.

[57] Y. Wang, X. Zhang, M. Li, T. Lan, H. Chen, H. Xiong, X. Cheng, and D. Yu. Theoretical convergence guaranteed resource-adaptive federated learning with mixed heterogeneity. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '23, page 2444–2455, New York, NY, USA, 2023. Association for Computing Machinery. ISBN 9798400701030. doi: 10.1145/3580305.3599521. URL https://doi.org/10.1145/3580305.3599521.

[58] T. Wu, C. Song, and P. Zeng. Efficient federated learning on resource-constrained edge devices based on model pruning. *Complex & Intelligent Systems*, 9(6):6999–7013, 2023. ISSN 2198-6053. doi: 10.1007/s40747-023-01120-5.

[59] H. Yang, Y. Liang, X. Guo, L. Wu, and Z. Wang. Theoretical characterization of how neural network pruning affects its generalization, 2023. URL https://openreview.net/forum?id=dn6_PK73hAY.

[60] L. Yin, S. Lin, Z. Sun, R. Li, Y. He, and Z. Hao. A game-theoretic approach for federated learning: A trade-off among privacy, accuracy and energy. *Digital Communications and Networks*, 10(2):389–403, 2024. ISSN 2352-8648. doi: https://doi.org/10.1016/j.dcan.2022.12.024. URL https://www.sciencedirect.com/science/article/pii/S2352864823000056.

[61] Z. You, K. Yan, J. Ye, M. Ma, and P. Wang. Gate Decorator: Global Filter Pruning Method for Accelerating Deep Convolutional Neural Networks, Sept. 2019. arXiv:1909.08174 [cs, eess].

[62] S. Yu, J. P. Muñoz, and A. Jannesari. Bridging the gap between foundation models and heterogeneous federated learning, 2023.

[63] S. Yu, P. Nguyen, A. Anwar, and A. Jannesari. Heterogeneous federated learning using dynamic model pruning and adaptive gradient. In *2023 IEEE/ACM 23rd International Symposium on Cluster, Cloud and Internet Computing (CCGrid)*, pages 322–330, Los Alamitos, CA, USA, may 2023. IEEE Computer Society. doi: 10.1109/CCGrid57682.2023.00038. URL https://doi.ieeecomputersociety.org/10.1109/CCGrid57682.2023.00038.

[64] G. Zhou, K. Xu, Q. Li, Y. Liu, and Y. Zhao. AdaptCL: Efficient Collaborative Learning with Dynamic and Adaptive Pruning, June 2021. arXiv:2106.14126 [cs].

[65] H. Zhu, J. Xu, S. Liu, and Y. Jin. Federated learning on non-iid data: A survey. *Neurocomputing (Amsterdam)*, 465:371 – 390, 2021. ISSN 0925-2312. doi: 10.1016/j.neucom.2021.07.098.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction precisely summarize the main contributions and scope of the paper, detailing the experimental approach, the significance of the results, and the potential impact. This alignment ensures that readers are immediately informed of the paper's aims and achievements.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We have dedicated a subsection within the "6 Discussion and Limitations" section of our paper specifically to address the limitations of our study.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [NA]

Justification: Our paper focuses on empirical evaluations and practical implementations of `AutoFLIP` in federated learning environments, and does not delve into theoretical proofs or formulations. Thus, this question is not applicable as our contributions are primarily experimental and do not involve new theoretical results.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: Our paper provides a detailed description of the experimental settings, including the complete framework (Section 4 and 4.1), hyperparameters, and pubblic datasets used (Section 5, 5.1 and Appendix B). We ensure full reproducibility by providing access to our code through supplementary materials and an anonymous GitHub repository (https://anonymous.4open.science/r/AutoFLIP-D283).

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We ensure full reproducibility by providing access to our code through supplementary materials and an anonymous GitHub repository (https://anonymous.4open.science/r/AutoFLIP-D283).

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: This information is documented in Section 4 and Appendix A for experimental framework and hyperparameter ablation studies. Section 5 and its subsections, along with Appendix B, elaborate on the dataset configurations and the specifics of the public datasets utilized.

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: The paper includes error bars for each experiment, indicating the variability and reliability of the results. As detailed in Section 5.1, each experiment was conducted multiple times, and error bars represent the standard deviation across these runs, providing statistical consistency.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: Section 5.1 of our paper provides detailed information about the computing resources used for the experiments, including the type of CPUs and GPUs, the amount of RAM available, and the specific machine configurations.

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

Answer: [Yes]

Justification: We have ensured that all experimental procedures, data handling, and methodologies are conducted responsibly, with a focus on maintaining privacy and fairness. We have considered potential impacts and ensured that our research does not facilitate misuse or harm. Additionally, all data used in our experiments are from publicly available datasets or are generated through simulations that comply with ethical standards. The methods proposed are designed to enhance federated learning settings, aligning with ethical guidelines regarding the use of artificial intelligence in sensitive and resource-constrained environments.

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: In Section 6, titled "Discussion and Limitations," we specifically added the broader impacts of our work in a dedicated subsection.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: Our research does not involve the release of models or datasets that pose a high risk for misuse, such as pretrained language models or image generators.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: Our research utilizes publicly available datasets, and we have duly cited the original sources in the manuscript.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [Yes]

Justification: The paper introduces a new federated learning framework `AutoFLIP`, which we have documented extensively in the supplementary material provided alongside the publication. We provide a publicly accessible anonymous GitHub repository that contains the code, a README file detailing the setup and execution instructions, and a license file specifying the usage rights.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA]

Justification: The paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.

- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

A Ablation Study on C_{exp} and E_{exp}

We perform an ablation study to assess the sensitivitiy of our method to the parameters C_{exp} and E_{exp}. The number of explorer clients C_{exp} influences the comprehensiveness of G_{global} in capturing the intricacies of the clients loss landscapes. The depth of the exploration phase, quantified by the number of exploration epochs E_{exp} influences loss function surface understanding and G_{global}'s knowledge depth.

In particular, we check how the average accuracy and loss for the global model predictions vary for $C_{\text{exp}} \in \{0.25, 0.5, 0.75, 2.0\}$ and for $E_{\text{exp}} \in \{150, 300, 500, 750, 1000\}$. We do this for datasets FEMINST the LEAF non-IID scenario 4. It is possible to observe in Figure 4, the x-axis represents the E_{exp} parameter. Each distinct plot corresponds to a different C_{exp}, with the highest accuracy achieved distinctly highlighted. In particular, our findings reveal that even a conservative value of C_{exp} can boost the accuracy. The influence of C_{exp} is substantial, with higher counts of explorer clients resulting in improved initial accuracy, indicative of a more robust G_{global} at the outset of the learning process. A discernible trend suggests that increasing the E_{exp} value generally leads to an improvement in accuracy. However, at higher values, the increase is reduced, indicating a saturation point beyond which additional exploration epochs no longer improve accuracy.

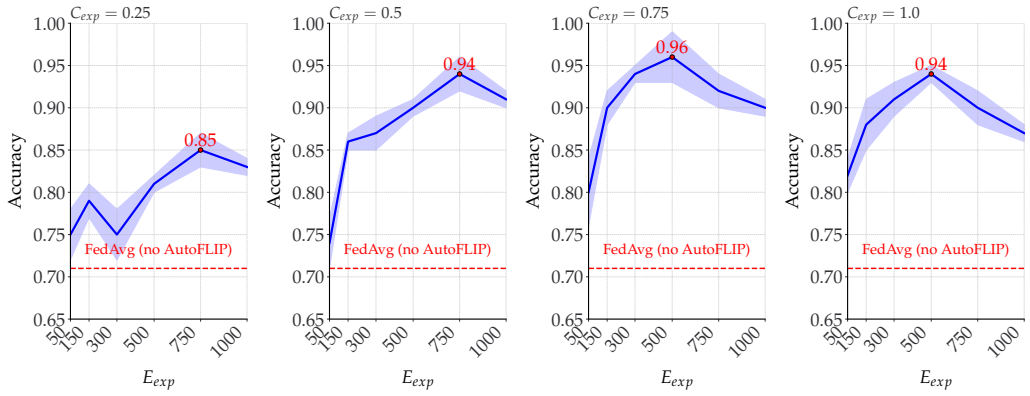

Figure 4: Ablation on the total number of clients for exploration C_{exp} and the total number of exploration epochs E_{exp} for FEMNIST/pathological non-IID data, based on average accuracy.

B Ablation Study on T_p

We perform an ablation study to assess the sensitivitiy of our method to the pruning threshold parameter T_p. In particular, we check how the average accuracy and loss for the global model predictions vary for $T_p \in \{0.1, 0.2, 0.3, 0.4, 0.5\}$. We do this on two datasets: MINST in Figure 5 and CIFAR10 in Figure 6 from the Pathological non-IID scenario. In both cases, $T_p = 0.3$ seems the most convenient choice.

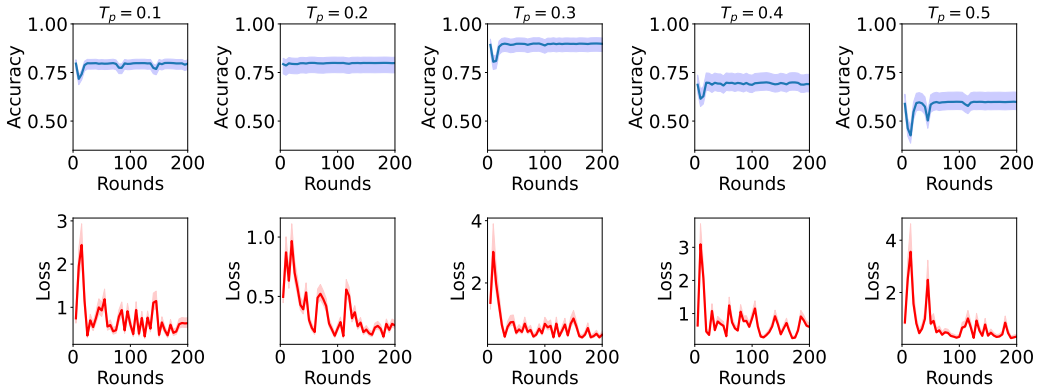

Figure 5: Ablation on T_p for MNIST/non-IID based on average accuracy (top) and loss (bottom).

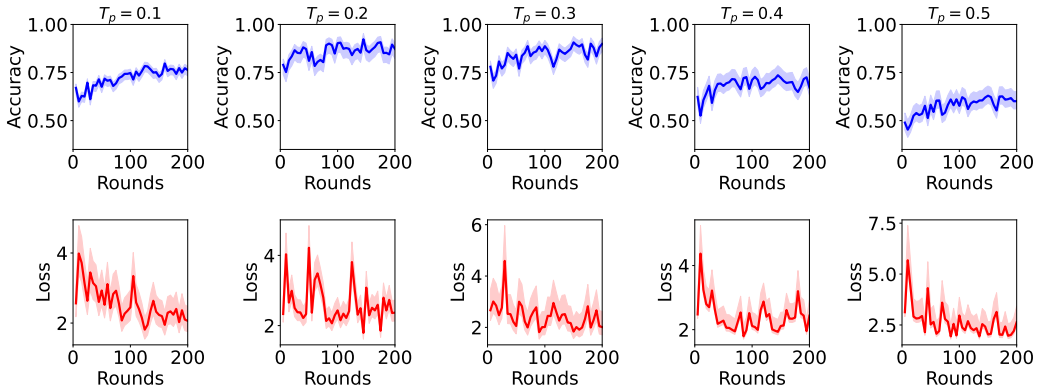

Figure 6: Ablation on T_p for CIFAR10/non-IID based on average accuracy (top) and loss (bottom).

C Partitioning approaches

Pathological non-IID

This experimental configuration is delineated by each client possessing data exclusively from two distinct classes within a broader multi-class dataset. Figure 7 illustrates this "pathological" data partitioning scenario within the CIFAR10 dataset across 20 clients. For our experiments, we select the MNIST dataset [6] with a six-layer CNN (7628484 parameters) and the CIFAR10 dataset [28] with EfficientNet-B3 architecture (10838784 parameters), following the guidelines in [38] and [52].

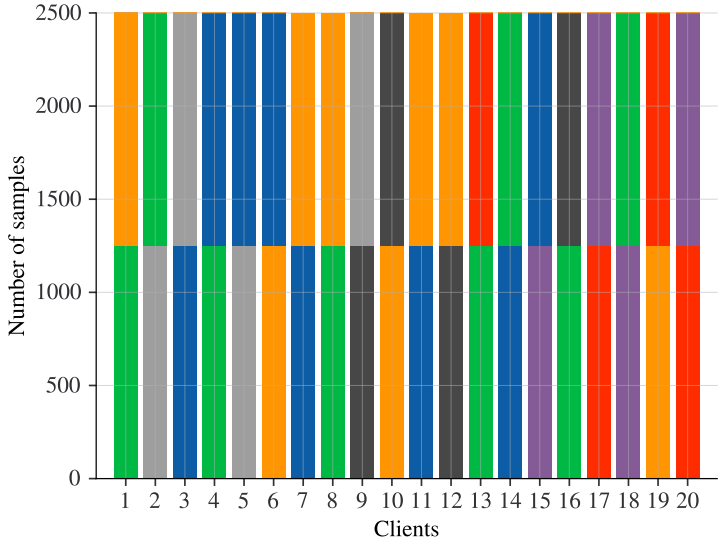

Figure 7: Illustration of pathological data partitioning on CIFAR10 for 20 clients, where each color represents a different class.

Dirichlet-based non-IID

This advanced experimental setup, as introduced by [21], utilizes the Dirichlet distribution, modulated by a concentration parameter α. Let $\boldsymbol{p} = (p_1, p_2, ..., p_N)$ be the class distribution for a given client, where N is the number of classes. The Dirichlet distribution is defined as $\boldsymbol{p} \sim \text{Dir}(\alpha \cdot \mathbf{1}_N)$, where "Dir" denotes the Dirichlet distribution, α is the concentration parameter, and $\mathbf{1}_N$ is a N-dimensional vector of ones. In this context, a low value of α, or $\alpha \to 0$, leads to distributions where most of the probability mass is concentrated on a single class, thereby indicating that each client's data is restricted to a single class. Conversely, as $\alpha \to \infty$, $\boldsymbol{p}$ approaches a uniform distribution, ensuring that the samples are evenly split across all clients. Figure 8 illustrates this "Dirichlet-based non-IID" data partitioning scenario within the CIFAR100 dataset across 20 clients, with individual colors denoting separate classes.

To address the complexities of larger datasets, we have extended our evaluation to include CIFAR100 [28] with a $\alpha = 100$, employing ResNet (23755900 parameters) [18] in alignment with the methodology proposed in [15].

LEAF non-IID

Utilizing the popular LEAF benchmark for FL [4], we selected the FEMNIST and Shakespeare datasets to simulate closer real-world FL scenarios, with each dataset designed for specific tasks. The FEMNIST dataset is defined for a multi-class classification challenge involving 62 distinct classes. Conversely, the Shakespeare dataset is tailored for a next-character prediction task, requiring models to predict the subsequent character from a sequence of 80 characters, thereby testing the model capabilities in sequential data processing and language modeling. The incorporation of the next-character prediction task allows for a comprehensive assessment of `AutoFLIP` adaptability

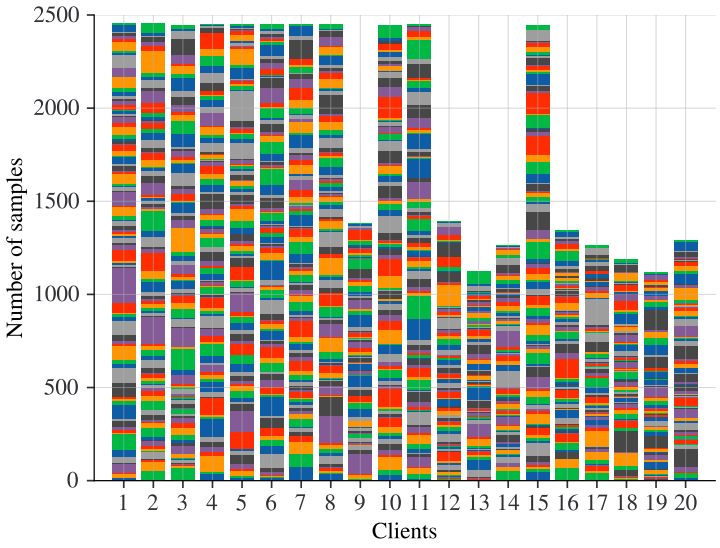

Figure 8: Illustration of Dirichlet-based non-IID data partitioning on CIFAR100 for 20 clients, where each color represents a different class.

and performance across diverse task types and deep neural network architectures, such as Long Short-Term Memory (LSTM) networks.

In our experimental setup, we employed the FEMNIST-CNN architecture, as delineated in [4], for the FEMNIST dataset. For the Shakespeare dataset, we utilized a two-layer (LSTM) (5040000 parameters) model, in accordance with the specifications provided in [38].

D Loss plots

We present in Figure 9 the loss convergence profiles for the global model participating in the FL procedure.for the global model. Here, we compare `AutoFLIP` to the different federated pruning strategies evaluated on both image recognition and text prediction tasks using five distinct datasets: MNIST, CIFAR10, CIFAR100, FEMNIST, and Shakespeare. Due to the varying complexities of each task, we use different model structures for different datasets.

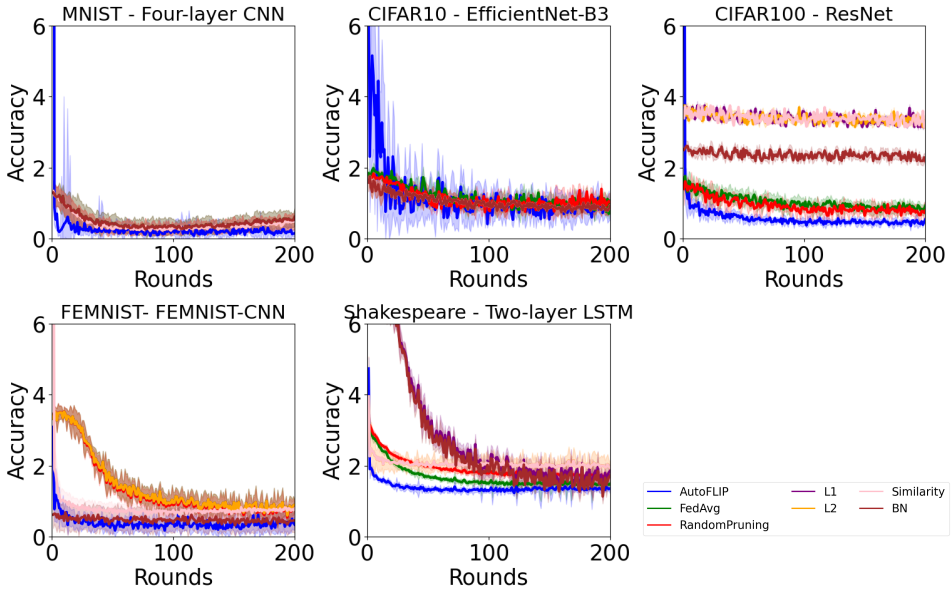

Figure 9: Average loss convergence profiles for the global model within the FL framework.

E Structured Pruning Experiments

We present an evaluation of AutoFLIP using exclusively structured pruning methods. We compare these results with both the hybrid approach of AutoFLIP and the SotA algorithm PruneFL [25]. We conducted experiments on FEMNIST and CIFAR-100, utilizing the same non-IID settings, model architectures, and parameters as described in Sec. 5.

With structured pruning, we focus solely on removing entire filters, channels, or neurons based on the knowledge derived from the federated loss exploration phase. Unlike the hybrid approach, individual weights within structures were not pruned unless the entire substructure was considered unimportant.

Tables 1 and 2 summarize the performance metrics of AutoFLIP with structured pruning compared to AutoFLIP with hybrid pruning, PruneFL, and the standard FedAvg without model compression.

Table 1: Performance Comparison on FEMNIST with Structured Pruning

Metric	AutoFLIP **(Structured)**	AutoFLIP **(Hybrid)**	PruneFL	FedAvg **(No Compression)**
Compression Rate (%)	33.89	–	–	–
Training Time per Client (s)	15.66	–	17.39	21.39
Inference Time per Client (ms)	7.5	–	11.61	14.61
Computation Cost (GFLOPs)	10.08	–	11.28	19.36
Final Test Accuracy (%)	93.9	98.5	89.3	90.5

Table 2: Performance Comparison on CIFAR-100 with Structured Pruning

Metric	AutoFLIP **(Structured)**	AutoFLIP **(Hybrid)**	PruneFL	FedAvg **(No Compression)**
Compression Rate (%)	49.22	–	–	–
Training Time per Client (s)	186.00	–	192.01	257.18
Inference Time per Client (ms)	8.1	–	9.8	15.97
Computation Cost (GFLOPs)	14.39	–	14.98	17.78
Final Test Accuracy (%)	94.9	98.7	90.6	91.8

The structured pruning approach in AutoFLIP resulted in notable reductions in training time per client. For FEMNIST, the training time decreased by approximately 26.8% compared to FedAvg without compression, and by 10.0% compared to PruneFL. For CIFAR-100, the training time decreased by approximately 27.7% compared to FedAvg, and by 3.1% compared to PruneFL.

Inference times per client were significantly reduced. For FEMNIST, the inference time decreased by 48.7% compared to FedAvg, and by 35.4% compared to PruneFL. For CIFAR-100, the inference time decreased by 49.3% compared to FedAvg, and by 17.3% compared to PruneFL.

The computation cost, measured in giga floating-point operations (GFLOPs), was substantially reduced. For FEMNIST, the computation cost decreased by 47.9% compared to FedAvg, and by 10.6% compared to PruneFL. For CIFAR-100, it decreased by 19.0% compared to FedAvg, and by 4.0% compared to PruneFL.

Regarding memory usage and the number of processed parameters, on FEMNIST, the structured pruning version processed approximately 15.3 billion parameters during the entire federated learning procedure, compared to 28.3 billion for FedAvg. On CIFAR-100, the structured pruning version processed approximately 3.9 billion parameters, compared to 9.4 billion for FedAvg. These reductions contribute to decreased memory bandwidth requirements and potential energy savings, which are significant factors in large-scale federated learning deployments.

The accuracy for FEMNIST was 93.9%, which is lower than that of hybrid AutoFLIP (98.5%), but higher than those of PruneFL (89.3%) and FedAvg (90.5%). For CIFAR-100, the final test accuracy was 94.9%, lower than that of hybrid AutoFLIP (98.7%), but higher than those of PruneFL (90.6%) and FedAvg (91.8%). The hybrid AutoFLIP approach achieves superior accuracy by also eliminating individual weights that may contribute to overfitting or biased learning. Thiss demonstrate that the hybrid pruning approach of AutoFLIP offers the best overall performance by combining the benefits of both structured and unstructured pruning.

F Inference acceleration

In this section, we discuss the inference acceleration of `AutoFLIP`. When performing inference on the client's side with the pruned sub-model, we accelerate the inference time and reduce the computational consumption. Figure 10 shows the inference acceleration comparison after applying `AutoFLIP`. Notably, the FLOPs (floating point operations per second) in all the evaluated models are reduced. Table 3 shows that the Six-layer CNN deployed for the pathological non-IID experiment with MNIST, experienced a substantial decrease in computational load, equal to a 41.62% reduction in FLOPs. EfficientNet-B3, used for CIFAR10 in the pathological non-IID experiment, saw further improvements, reaching a FLOPs reduction of 46.44%. The deeper ResNet model, designed for CIFAR100 in the Dirichlet-based non-IID experiment, achieved a significant reduction in FLOPs, over 50%, highlighting the potential of `AutoFLIP` to streamline deep networks for more efficient inference. The FEMNIST-CNN and LSTM models, employed for the LEAF non-IID experiment, showcased a FLOPs reduction equal to 56.49% and 44.44%, respectively.

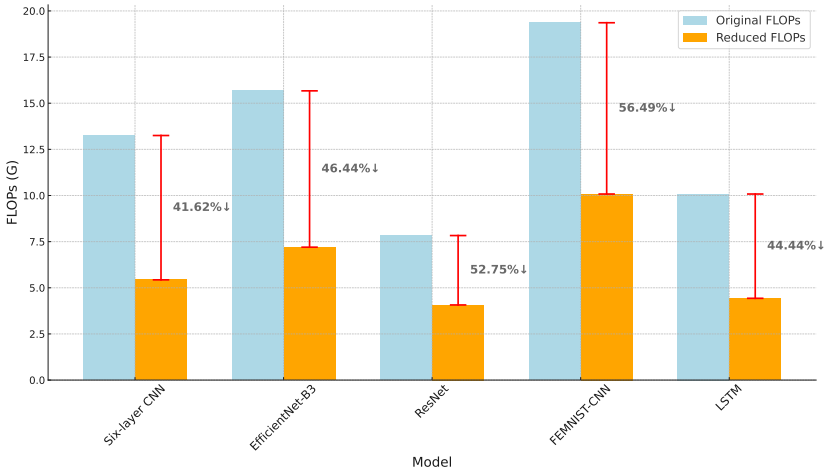

Figure 10: Original FLOPs and reduced FLOPs

Table 3: FLOPs comparison

Model	Compression Rate	Original FLOPs	Reduced FLOPs	FLOPs % Reduced
Six-layer CNN	1.74	13.25 G	5.43 G	41.62% ↓
EfficientNet-B3	2.1	15.67 G	7.20 G	46.44% ↓
ResNet	1.58	7.83 G	4.07 G	52.75% ↓
FEMNIST-CNN	1.8	19.36 G	10.08 G	56.49% ↓
LSTM	1.8	10.08 G	4.43 G	44.44% ↓

G Training efficiency

To ascertain AutoFLIP's impact on enhancing training efficiency within FL frameworks, we delve into an examination of the associated communication costs. For a practical perspective, the deployed models are trained to achieve a 90% accuracy threshold. As presented in [63], the cost function employed for this evaluation is defined as:

$$\text{Cost} = \#\text{ Parameters} \times \#\text{ Rounds to Reach Target Accuracy} \times \#\text{ Clients} \times \text{Sample Rate}.$$

In Table 4, we observe the effectiveness of AutoFLIP in reducing communication costs across various non-IID scenarios with different models and datasets. Notably, the Six-layer CNN model, used in the MNIST dataset for the Pathological non-IID experiment, demonstrated a significant reduction in communication costs by 41.61%, which underscores AutoFLIP's effectiveness in simpler architectures. This efficiency extends to more complex architectures, like EfficientNet-B3 and ResNet, employed for the CIFAR10 and CIFAR100 datasets respectively for the Dirichlet-based non-IID experiment, which also saw notable cost reductions of 30.93% and 29.88%. Similarly, the FEMNIST-CNN and LSTM models, used in the LEAF non-IID experiment, exhibited reductions in communication costs by 19.54% and 19.29%, respectively. These results highlight AutoFLIP's broad applicability and substantial impact on training efficiency across a range of model complexities and dataset types.

Table 4: Comparison of the total communication costs

Model	**Rounds AutoFLIP**	**Rounds NoAutoFLIP**	**Cost AutoFLIP**	**Cost NoAutoFLIP**	**% Cost Reduced**
Six-layer CNN	3	58	189.45 GB	324.43 GB	41.61% $\downarrow$
EfficientNet-B3	27	39	290.26 GB	420.27 GB	30.93% $\downarrow$
ResNet	7	49	712.70 GB	1016.40 GB	29.88% $\downarrow$
FEMNIST-CNN	280	348	369.06 GB	458.69 GB	19.54% $\downarrow$
LSTM	243	301	122.47 GB	151.74 GB	19.29% $\downarrow$

H Computation Cost

To evaluate AutoFLIP's role in reducing computational effort, we investigate the number of parameters processed for a single client. Distinguishing between computational efforts on the global model and the clients is essential, with a particular focus on the client side. For AutoFLIP, each client handles a substantial number of parameters over an additional 150 exploration epochs (E_{exp}). From a practical standpoint, we compare AutoFLIP and FedAvg with RandomPruning with the same compression rate. The models are trained to meet a 90% of global accuracy. We define the computation cost function as:

$$\text{Computation cost for single client} = \text{Total Parameters Processed} \times \# \text{ Epochs} \times \text{Sample Rate}$$

In Table 5, the pathological non-IID experiment with MNIST using the Six-layer CNN model shows a significant reduction in computational cost by 62.51%. This efficiency extends to more complex architectures like EfficientNet-B3 and ResNet, used for the CIFAR10 and CIFAR100 datasets respectively, with cost reductions of 46.41% and 58.22%. Similarly, the FEMNIST-CNN and LSTM models, employed in the LEAF non-IID experiment, demonstrated reductions in computational costs by 45.99% and 29.60% respectively. These results underline AutoFLIP's broad applicability and substantial impact on reducing computational efforts across diverse model architectures and dataset types.

Table 5: Comparison of the total computation costs

Model	**Processed parameters AutoFLIP**	**Processed parameters NoAutoFLIP**	**% Cost Reduced**
Six-layer CNN	535,309,170	1,428,653,880	62.51% ↓
EfficientNet-B3	2,005,175,040	3,740,891,680	46.41% ↓
ResNet	3,919,723,500	9,378,097,500	58.22% ↓
FEMNIST-CNN	15,303,653,440	28,338,578,100	45.99% ↓
LSTM	5,871,600,000	8,335,200,000	29.60% ↓