When Do Curricula Work in Federated Learning?

Saeed Vahidian[1], Sreevatsank Kadaveru[1], Woonjoon Baek[1], Weijia Wang[1], Vyacheslav Kungurtsev[2],
Chen Chen[3], Mubarak Shah[3], Bill Lin[1]
[3] Center for Research in Computer Vision, University of Central Florida
[2] Czech Technical University
[1] UC San Diego
Saeed@ucsd.edu

Abstract

An oft-cited open problem of federated learning is the existence of data heterogeneity among clients. One pathway to understanding the drastic accuracy drop in federated learning is by scrutinizing the behavior of the clients' deep models on data with different levels of "difficulty", which has been left unaddressed. In this paper, we investigate a different and rarely studied dimension of FL: ordered learning. Specifically, we aim to investigate how ordered learning principles can contribute to alleviating the heterogeneity effects in FL. We present theoretical analysis and conduct extensive empirical studies on the efficacy of orderings spanning three kinds of learning: curriculum, anti-curriculum, and random curriculum. We find that curriculum learning largely alleviates non-IIDness. Interestingly, the more disparate the data distributions across clients the more they benefit from ordered learning. We provide analysis explaining this phenomenon, specifically indicating how curriculum training appears to make the objective landscape progressively less convex, suggesting fast converging iterations at the beginning of the training procedure. We derive quantitative results of convergence for both convex and nonconvex objectives by modeling the curriculum training on federated devices as local SGD with locally biased stochastic gradients. Also, inspired by ordered learning, we propose a novel client selection technique that benefits from the real-world disparity in the clients. Our proposed approach to client selection has a synergic effect when applied together with ordered learning in FL.

1. Introduction

Inspired by the learning principle underlying the cognitive process of humans, curriculum learning (CL) deals with a training paradigm for machine learning models in which the difficulty of the training task is progressively scaled, going from "easy" to "hard". Prior empirical studies have demonstrated that CL is effective in avoiding bad local minima and in improving the generalization results [1, 2]. Also interestingly, another line of work proposes the exact opposite strategy of prioritizing the harder examples first, such as [3, 4, 5]–these techniques are referred to as "anti-curriculum". It is shown that certain tasks can benefit from anti-curriculum techniques. However, in tasks such as object detection [6, 7], and large-scale text models [8] CL is standard practice.

Although the empirical observations on CL appear to be in conflict, this has not impeded the study of CL in machine learning tasks. Certain scenarios [9] have witnessed the potential benefits of CL. The efficacy of CL has been explored in a considerable breadth of applications, including, but not limited to, supervised learning tasks within computer vision [10], healthcare [11], reinforcement learning tasks [12], natural language processing (NLP) [13] as well as other applications such as graph learning [14], and neural architecture search [15].

Curriculum learning has been studied in considerable depth for the standard centralized training settings. However, to the best of our knowledge, our paper is the first attempt at studying the methodologies, applications, and efficacy of CL in a decentralized training setting and in particular for federated learning (FL). In FL, the training time budget and the communication bandwidth are the key limiting constraints, and as demonstrated in [9] CL is particularly effective in settings with a limited training budget. It is an interesting proposition to apply the CL idea to an FL setting, and that is exactly what we explore in our paper (in Section 2).

The idea of CL is agnostic to the underlying algorithms used for federation and hence can be very easily applied to any of the state-of-the-art solutions in FL. Our technique does not require a pre-trained expert model and does not impose any additional synchronization overhead on the system. Also, as the CL is applied to the client, it does not add any additional computational overhead to the server.

Further, we propose a novel framework for efficient client selection in an FL setting that builds upon our idea of CL in FL. We show in Section 4, CL on clients is able to leverage the real-world discrepancy in the clients to its advantage. Furthermore, when combined with the primary idea of CL in FL, it provides compounding benefits.

Contributions: In this paper, we comprehensively assess the efficacy of CL in FL and provide novel insights into the efficacy of CL under the various conditions and methodologies in the FL environment.

We provide a rigorous convergence theory and analysis of FL in non-IID settings, under strongly convex and nonconvex assumptions, by considering local training steps as biased SGD, where CL naturally grows the bias over the iterations, as discussed in Section 5 of the main paper and Section 1 of the **Supplementary Material (SM)**.

We hope to provide comprehensible answers to the following six important questions:

Q1: *Which of the curriculum learning paradigm is effective in FL? And under what conditions?*

Q2: *Can CL alleviate the statistical data heterogeneity in FL?*

Q3: *Does the efficacy of CL in FL depend on the underlying client data distributions?*

Q4: *Whether the effectiveness of CL is correlated with the size of datasets owned by each client?*

Q5: *Are there any benefits of smart client selection? And can CL be applied to the problem of client selection?*

Q6: *Can we apply the ideas of CL to both the client data and client selection?*

We test our ideas on two widely adopted network architectures on popular datasets in the FL community (CIFAR-10, CIFAR-100) under a wide range of choices of curricula and compare them with several global state-of-the-art baselines. We have the following findings:

- CL in FL boosts the classification accuracy under both IID and Non-IID data distributions (Sections 3.1, and 3.2).
- The efficacy of CL is more pronounced when the client data is heterogeneous (Section 3.3).
- CL on client selection has a synergic effect that compounds the benefits of CL in FL (Section 4).
- CL can alleviate data heterogeneity across clients and CL is particularly effective in the initial stages of training as the larger initial steps of the optimization are done on the "easier" examples which are closer together in distribution (Section 5 of main paper, and Section 1 of the SM).
- The efficacy of our technique is observed in both lower and higher data regimes (Section 2 of the SM).

2. Curriculum Components

Federated Learning (FL) techniques provide a mechanism to address the growing privacy and security concerns associated with data collection, all-the-while satiating the need for large datasets for training powerful machine learning models [16, 17]. A major appeal of FL is its ability to train a model over a distributed dataset without needing the data to be collated into a central location for training. In the FL framework, we have a server and multiple clients with a distributed dataset. The process of the federation is an iterative process that involves multiple rounds of back-and-forth communication between the server and the clients that participate in the process [18, 19]. This back-and-forth communication incurs a significant communication overhead, thereby limiting the number of rounds that are pragmatically possible in real-world applications. Curriculum learning is an idea that particularly shines in these scenarios where the training time is limited [9]. Motivated by this idea, we define a curriculum for training in the FL setting. A curriculum consists of three key components:

The scoring function: It is a mapping from an input sample, $x_i \in \mathcal{D} = \{(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)\}$, to a numerical value, $s_i(x_i) \in R^+$. We define a range of scoring functions when we discuss a CL for the FL setting in the subsequent sections. When defining the scoring function of a curriculum in FL, we look for loss-based dynamic measures for the score that update every iteration, unlike the methods proposed in [20] which produce a fixed score for each sample. This is because the instantaneous score of samples changes significantly between iterations, and using a fixed score leads to an inconsistency in the optimization objectives, making training less stable [21]. Also, we avoid techniques like [22] which requires human annotators, as it is not practical in a privacy-preserving framework.

The pacing function: The pacing function $g_\lambda(t)$ determines scaling of the difficulty of the training data introduced to the model at each of the training steps t and it selects the highest scoring samples for training at each step. The pacing function is parameterized by $\lambda = (a, b)$ where a is the fraction of the training budget needed for the pacing function to begin sampling from the full training set, and b represents the fraction of the training set the pacing function exposes to the model at the start of training. In this paper, the full training set size and the total number of training steps (budget) are denoted by N and T, respectively. Further, we consider five pacing function families, including exponential, step, linear, quadratic, and root (sqrt). The expressions we used for the pacing functions are shown in Table 1. we follow [23] in defining the notion of pacing function and use it to schedule how examples are introduced to the training procedure.

The order: Curriculum learning orders sample from the highest score (easy ones) to lowest score, anti-curriculum learning orders from lowest score to highest, and finally, random curriculum randomly samples data in each step regardless of their scores.

2.1. Preliminary

The five function families incorporated, including exponential, step, linear, quadratic, and root, and their expressions can be seen in Table 1 and Fig. 1.

Table 1: The five families of pacing functions we employed in this paper. The parameter a determines the fraction of training time until all data is used. Parameter b sets the initial fraction of the data used.

Pacing Function	Expression
Exponential	$Nb + \frac{N(1-b)}{e^{10}-1}(e^{\frac{10t}{aT}} - 1)$
Step	$Nb + N\lfloor \frac{t}{aT} \rfloor$
Root (Sqrt)	$Nb + \frac{N-b}{\sqrt{aT}}\sqrt{t}$
Linear	$Nb + \frac{N-b}{aT}t$
Quadratic	$Nb + \frac{N-b}{(aT)^2}t^2$

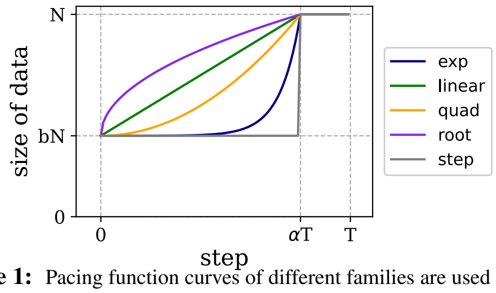


Figure 1: Pacing function curves of different families are used throughout the paper. As shown, the hyperparameter α specifies the fraction of the training step until all data is used in training. The hyperparameter b determines the initial fraction of the data used in training.

3. Experiment

Experimental setting. To ensure that our observations are robust to the choice of architectures, and datasets, we report empirical results for LeNet-5 [24] architecture on CIFAR-10 [25], and ResNet-9 [26] architecture for CIFAR-100 [25] datasets. All models were trained using SGD with momentum. Details of the implementations, architectures, and hyperparameters can be found in Section 7 of the SM.

Baselines and Implementation. To provide a comprehensive study on the efficacy of CL on FL setups, we consider the predominant approaches to FL that train a global model, including FedAvg [16], FedProx [27], SCAFFOLD [28], and FedNova [29]. In all experiments, we assume 100 clients are available, and 10% of them are sampled randomly at each round. Unless stated otherwise, throughout the experiments, the number of communication rounds is 100, each client performs 10 local epochs with a batch size of 10, and the local optimizer is SGD. To better understand the mutual impact of different data partitioning methods in FL and CL, we consider both federated heterogeneous (Non-IID) and homogeneous (IID) settings. In each dataset other than IID data partitioning settings, we consider two different federated heterogeneity settings as in [30, 31]: Non-IID label skew (2), and Non-IID Dir(β).

Algorithm 1: The Curriculum FL Framework

```
Input: $M$ clients indexed by $m$, sampling rate $R \in (0,1]$, number of participating-client $K$, communication rounds $R_C$, server model $f$ with $\theta_g$, pacing function $g_\lambda : [T] \to [N]$, scoring function $s : [N] \to \mathbb{R}$, order $o \in \{$“curriculum”, “anti”, “random”$\}$,

Server executes:
    initialize $f$ with $\theta$
    for each round $t = 0, 1, 2, ...$ do
        $\mathbb{S}_t \leftarrow$ (random set of $K$ clients)
        for each client $m \in \mathbb{S}_t$ in parallel do
            broadcast $\theta_g^t$ to clients
            $\theta_m^{(t)} \leftarrow$ ClientUpdate$(m, \theta_g^{(t)})$
            $\theta_g^{(t+1)} = \sum_{m=1}^{K} \frac{|\mathcal{D}_m|}{\sum_{i=1}^{K} |\mathcal{D}_i|} \theta_m^{(t)}$ ; $\mathcal{D}_m$ is the set of the local data on the client with index $m$.
    return $\theta_g^{t+1}$

ClientUpdate $(m, \theta_g^t)$:
    Obtain the score of each data sample using $\theta_g^t$ and/or $\theta_m^t$ as described in section 3.1
    $(\mathbf{x_1}, \mathbf{x_2}, ..., \mathbf{x_n}) \leftarrow \mathrm{sort}(\{\mathbf{x_2}, ..., \mathbf{x_n}\}, \mathrm{s}, \mathrm{o})$
    for $t = 1, 2, ..., T$ do
        $\theta_m^t \leftarrow \mathrm{train}(\theta_\mathrm{g}^t, \{\mathbf{x_1}, \mathbf{x_2}, ..., \mathbf{x}_{\mathrm{g(t)}}\})$
```

3.1. Effect of scoring function in IID and Non-IID FL

In this section, we investigate five scoring functions. As discussed earlier, in standard centralized training, samples are scored using the loss value of an expert pre-trained model. Given a pre-trained model $f_\theta : \mathcal{X} \to \mathcal{Y}$, the score is defined as $s_i(x_i) = \frac{r_i}{\sum_i r_i}$, where $r_i = \frac{1}{\mathcal{L}(y_i, f_\theta(x_i))}$, with $\mathcal{L}$ being the inference loss. In this setup, a higher score corresponds to an easier sample.

In FL [32], a trusted server broadcasts a single initial global model, θ_g, to a random subset of selected clients in each round. The model is optimized in a decentralized fashion by multiple clients who perform some steps of SGD updates ($\theta_k = \theta_g - \eta\nabla\mathcal{L}_k$). The resulting model is then sent back to the server, ultimately converging to a joint representative model. With that in mind, in our setting, the scores can be obtained by clients either via the global model that they receive from the server, which we name as s_G or by their own updated local model, named s_L or based on the average of the local and global model loss, named as s_{LG}[1].

We further consider another family of scoring that is based on ground truth class prediction. In particular, in each round, clients receive the global model from the server and get the prediction using the received global model and the

[1] Since it produces very similar results to s_G, we skipped it.

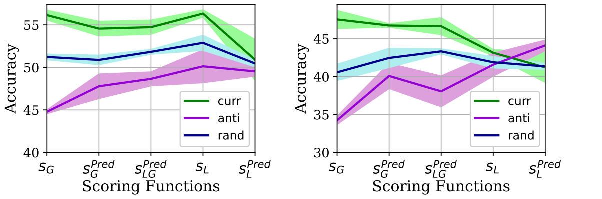


Figure 2: Scoring client samples based on the global model (s_G) provides the most accurate scores for all levels of Non-IIDness. Scoring based on the local model (s_L^{pred}) provides the least accurate scores, especially when data are Non-IID, as it provides the worst accuracy. Evaluating the effect of using different scoring methods on accuracy when the clients employ curriculum, anti-curriculum, or random ordering during their local training on CIFAR-10 with IID data (left) and Non-IID (2) (right). All curricula use the linear pacing function with $a = 0.8$ and $b = 0.2$. We run each experiment three times for 100 communication rounds with 10 local epochs and report the mean and standard deviation (std) for global test accuracy. Note that the results for vanilla FedAvg for the left figure, and the right one are 52.30 ± 0.86, and 41.96 ± 1.77, respectively.

current local model as $\hat{y}_G$ and $\hat{y}_L$, respectively. For those samples whose $\hat{y}_L$ and $\hat{y}_G$ do not match, the client tags them as hard samples and otherwise as easy ones. This scoring method is called s_{LG}^{pred}. Further, ground truth class prediction and scoring can be solely done by the global model or the client's local model, which end up with two other different scoring methods, namely s_G^{pred}, and s_L^{pred} respectively. This procedure is described in Algorithm 1.

Fig. 2 demonstrates what the impact of using these various scoring methods is on the global accuracy when curriculum, anti-curriculum, or random ordering is exploited by the clients in the order in which their CIFAR-10 examples are learned with FedAvg under IID and Non-IID (2) data partitions. The results are obtained by averaging over three randomly initialized trials and calculating the standard deviation (std).

The results reveal that **first,** *the scoring functions are producing broadly consistent results except for s_L^{pred} for both IID and Non-IID and s_L for Non-IID settings. s_G provides the most accurate scores, thereby improving the accuracy by a noticeable margin compared to its peers.* This is quite expected, as the global model is more mature compared to the client model. **Second,** *the curriculum learning improves the accuracy results consistently across different scoring functions.* **Third,** *curriculum learning is more effective when the clients underlying data distributions are Non-IID*. To ensure that the latter does not occur by chance, we will delve into this point in detail in subsection 3.3. Due to the superiority of s_G relative to others, we set the scoring function to be s_G henceforth. We will further elaborate on the precision of s_G compared to an expert model in Section 4.4.

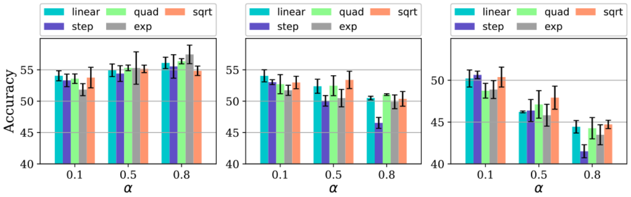


Figure 3: Bigger a values provide better accuracy performance for all pacing function families on IID settings for curriculum learning. But a notable contrast can be seen with random-/anti ordering. The effect of using different pacing function families and their hyperparameter a on accuracy when the clients employ curriculum, anti-curriculum or random ordering during their local training on CIFAR-10 with IID data. We run each experiment three times for 100 communication rounds with 10 local epochs and report the mean and std for global test accuracy. The figures from left to right are for curriculum, random, and anti ones.

3.2. Effect of pacing function and its parameters in IID and Non-IID FL

In order to study the effect of different families of pacing functions along with the hyperparameters $\lambda = (a, b)$, we test the exponential, step, linear, quadratic, and root function families. We further first fix b to 0.2 and let $a \in \{0.1, 0.5, 0.8\}$[2]. The accuracy results are presented in Fig. 3. The complement of this figure for Non-IID is presented in Fig. 11 in the SM. As is evident, for all pacing function families, the trends between the curriculum and the other orderings, i.e., (anti, random)-curriculum are markedly opposite in how they improve/degrade by sweeping a from small values to large ones. The pattern for Non-IID which is presented in the SM is almost similar to that of IID. Values of $a \in [0.5, 0.8]$ produce the best results. As can be seen from Fig. 3, the best accuracy achieved by curriculum learning outperforms the best accuracy obtained by other orderings by a large margin. For example, in the "linear" pacing function, the best accuracy achieved for curriculum learning when $a = 0.8$ is 56.60 ± 0.91 which improved the vanilla results by 4% while that of random when $a = 0.1$ is 52.73 ± 0.81 and improved vanilla by 0.5%. Henceforth, we set $a = 0.8$ and the pacing function to linear. After selecting the pacing function and a the final step is to fix these two and see the impact of b. Now we let all curricula use the linear pacing functions with a = 0.8 and only sweep $b \in \{0.0025, 0.1, 0.2, 0.5, 0.8\}$ and report the results in Fig. 4. Perhaps most striking is that curriculum learning tends to have smaller values of b to improve accuracy, which is in contrast with (random-/anti) orderings. The performance of anti-curriculum shows a significant dependence on the value of b. Further, curriculum learning provides a robust benefit for different values of b and it beats the vanilla FedAvg by $4-7\%$ depending upon the distribution of the data. Henceforth, we fix b to 0.2.

[2] Note that $b \in [0, 1]$. Also, $a = 0$ or $b = 1$ is equivalent to no ordered training, i.e., standard training.

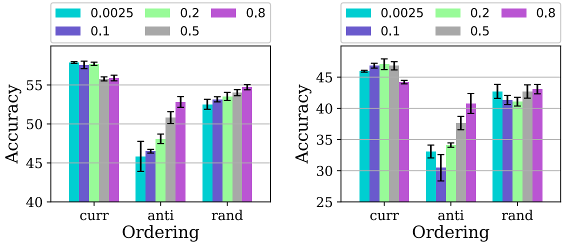


Figure 4: Smaller b values provide better accuracy performance for both IID and Non-IID settings as further corroborate the benefit of employing curriculum learning. Evaluating the effect of hyperparameter b on accuracy when the clients employ curriculum, anti-curriculum, or random ordering during their local training on CIFAR-10 with IID for FedAVg (left), and with Dir(0.05) for FedAvg (right). All curricula use the linear pacing functions with a = 0.8. We run each experiment three times for 100 communication rounds with 10 local epochs and report the mean and std for global test accuracy.

3.3. Effect of level of data heterogeneity

Equipped with the ingredients explained in the preceding section, we are now in a position to investigate the significant benefits of employing curriculum learning in FL when the data distribution environment is more heterogeneous. To ensure a reliable finding, we investigate the impact of heterogeneity in four baselines through extensive experiments on benchmark datasets, i.e., CIFAR-10 and CIFAR-100. In particular, we distribute the data between clients according to Non-IID Dir(β) defined in [33]. In Dir(β), heterogeneity can be controlled by the parameter β of the Dirichlet distribution. Specifically, when $\beta \to \infty$ the clients' data partitions become IID, and when $\beta \to 0$ they become extremely Non-IID.

To understand the relationship between ordered learning and data heterogeneity, we compare the effect of ordered learning on various non-IID scenarios at the clients. In Table 2, we apply our method to the various state-of-the-art (SOTA) baselines (FedAvg [16], FedProx [27], SCAFFOLD [28], FedNova [29]) on CIFAR-10 dataset. The "Vanilla" column in the tables refers to the original implementation of the baseline. We use the Dirichlet distribution with $\beta \in \{0.05, 0.2, 0.9\}$ for the non-IID data distribution, and we use a linear pacing function $(0.8, 0.2)$ and the s_G scoring function. The important highlight of the tables is the differential gain in accuracy of our technique (ordering learning) over the standard "Vanilla" SOTA baselines. To put the absolute accuracy numbers into perspective, an "expert" model with the same network architecture trained in a standard IID centralized non-federated setting on the dataset is about 62%. Using our technique we are able to achieve an accuracy of about 55% in a federated setting. This corresponds to 90% of the expert model's performance. For brevity, we will present results for CIFAR-100 in Section 5 of the SM.

Table 2: Curriculum-learning helps more when training with more severe data heterogeneity across clients. Understanding the benefit of ordered learning with increasing data heterogeneity ($\beta = 0.9 \to 0.05$) when clients are trained on CIFAR-10 with `FedAvg`, `Fedprox`, `FedNova`, and `SCAFFOLD` methods respectively.

Non-IIDness	Curriculum	Anti	Random	Vanilla
Dir($\beta = 0.05$)	46.34 ± 1.55	31.16 ± 3.16	41.91 ± 2.23	39.56 ± 4.91
Dir($\beta = 0.2$)	51.09 ± 0.39	42.34 ± 1.48	46.35 ± 1.44	46.75 ± 0.72
Dir($\beta = 0.9$)	55.36 ± 0.69	46.86 ± 0.35	52.42 ± 0.90	52.19 ± 0.73
Dir($\beta = 0.05$)	47.94 ± 0.96	36.08 ± 1.52	42.62 ± 0.35	41.48 ± 0.29
Dir($\beta = 0.2$)	50.02 ± 0.15	40.92 ± 0.90	46.41 ± 1.12	46.18 ± 0.90
Dir($\beta = 0.9$)	56.48 ± 0.18	48.37 ± 0.91	51.69 ± 0.40	53.07 ± 1.25
Dir($\beta = 0.05$)	43.73 ± 0.09	28.31 ± 1.93	37.81 ± 3.06	31.97 ± 0.90
Dir($\beta = 0.2$)	47.01 ± 1.89	36.55 ± 1.42	44.21 ± 1.00	41.28 ± 0.30
Dir($\beta = 0.9$)	50.74 ± 0.19	41.76 ± 0.90	48.87 ± 0.88	47.230 ± 1.80
Dir($\beta = 0.05$)	45.91 ± 1.17	21.29 ± 1.82	38.27 ± 2.19	41.33 ± 1.30
Dir($\beta = 0.2$)	49.69 ± 1.81	28.69 ± 0.60	45.29 ± 1.93	46.62 ± 0.58
Dir($\beta = 0.9$)	52.05 ± 1.14	30.75 ± 0.79	49.25 ± 0.76	50.24 ± 0.57

The results are surprising: **The benefits of ordered learning increase with data heterogeneity**. The greater the distribution discrepancy between clients, the greater the benefit to curriculum learning.

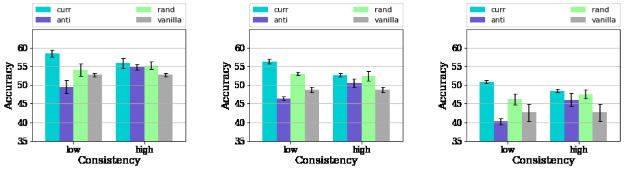
The client heterogeneity can be thought of as a distributional skew [34]. The "easier" data samples have a lower variance and thus the set of "easy" data samples is less skewed from the mean and thereby more IID. In the crucial early phase of the training, the use of the "easy" data samples effectively behaves as training on an IID distribution. This idea is formalized with convergence analysis and quantifying the convergence rates in the Section 5 of the main paper and in Section 1 of the SM.

Therefore, *CL can alleviate the accuracy drop associated with the non-IID client data. The improvement in accuracy is a result of the stable training on near-IID data in the crucial early phases.*

4. Curriculum on Clients

The technique of ordered learning presented in previous sections is designed to exploit the heterogeneity of data at the clients but is not geared to effectively leverage the heterogeneity between the clients that, as we discuss further, naturally emerges in the real world.

In the literature, some recent works have dabbled with the idea of smarter client selection, and many selection criteria have been suggested, such as importance sampling, where the probabilities for clients to be selected are proportional to their importance measured by the norm of update [35], test accuracy [36]. The [37] paper proposes client selection based on local loss where clients with higher loss are preferentially selected to participate in more rounds of federation, which is in stark contrast to [38] in which the clients with a lower loss are preferentially selected. It's clear from the literature that the heterogeneous environment in FL can hamper the overall training and convergence speed [39, 40], but the empirical observations on client selection criteria are either in conflict or their efficacy is minimal. In this section, inspired by curriculum learning, we



(a) IID **(b)** Dir(0.2) **(c)** Dir(0.05)

Figure 5: Consistency in data difficulty at the client hurts the efficacy of curricula. The effect of consistency in the difficulty distribution at the client nullifies the effect of curricula. The values plotted are for FedAvg on CIFAR-10. The standard deviation values of (Low, High) consistency for the IID are $(0.52, 0.01)$, Dir(0.2) are $(0.51, 0.14)$, and Dir(0.05) are $(0.50, 0.13)$. Note that we use Algorithm 2 to construct partitions with varying difficulty, and as detailed in Section 4.3 it is not possible to control the partition difficulty value with arbitrary precision, hence the above minor variations. The Low consistency scenario is generated using $f_{ord} = 0.0$ and the high consistency scenario uses $f_{ord} = 1.0$.

propose a more sophisticated mechanism of client selection that generalizes the above strategies to the FL setting.

4.1. Motivation

In the real world, the distributed dataset used in FL is often generated in-situ by clients, and the data is measured/generated using a particular sensor belonging to the client. For example, consider the case of a distributed image dataset generated by smartphone users using the onboard camera or a distributed medical imaging dataset generated by hospitals with their in-house imaging systems. In such scenarios, as there is a huge variety in the make and quality of the imaging sensors, the data generated by the clients is uniquely biased by the idiosyncrasies of the sensor, such as the noise, distortions, quality, etc. This introduces variability in the quality of the data at clients, in addition to the Non-IID nature of the data. However, it is interesting to note that these effects apply consistently across all the data at the specific client.

From a curriculum point of view, as the data points are scored and ordered by difficulty, which is just a function of the loss value of that data point, these idiosyncratic distortions uniformly affect the loss/difficulty value of all the data at that particular client. Also, it is possible that the difficulty among the data points at the particular client is fairly consistent as the level of noise, quality of the sensor, etc. are the same across the data points. This bias in difficulty varies between clients but is often constant within the same client. Naturally, this introduces a heterogeneity in the clients participating in the federation. In general, this can be thought of as some clients being "easy" and some being "difficult". *We can quantify this notion of consistency in difficulty by the standard deviation in the score of the data points at the client.*

When the standard deviation of the intra-client data is low, i.e., when the difficulty of the data within a client is consistent, we find that the curriculum on FL behaves very differently in these kinds of scenarios. We observe the advantage of curriculum diminishes significantly and has similar efficacy as that of random curricula as shown in Fig 5. The advantage of curriculum can be defined as $A_o = accuracy(o) - accuracy(vanilla)$, where $o \in \{curr, anti, rand\}$.

4.2. Client Curriculum

We propose to extend the ideas of curriculum onto the set of clients, in an attempt to leverage the heterogeneity in the clients. To the best of our knowledge, our paper is the first attempt to introduce the idea of `curriculum on clients` in an FL setting. Many curricula ideas can be neatly extended to apply to the set of clients. In order to define a curriculum over the clients, we need to define a scoring function, a pacing function, and an ordering over the scores. We define the client loss as the mean loss of the local data points at the client (Eq. 1), and the client score can be thought of as inversely proportional to the loss. The ordering is defined over the client scores.

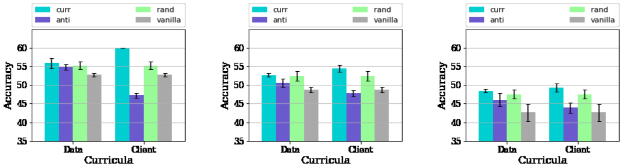
$$\mathcal{L}_k = \frac{1}{\|\mathbb{D}_m\|} \sum_{j}^{\|\mathbb{D}_m\|} l_j \qquad (1)$$

where m is the index for client, $\mathbb{D}_m$ represents the dataset at client m, l_j is the loss of jth data point in $\mathbb{D}_m$.

The pacing function, as in the case of data curricula, is used to pace the number of clients participating in the federation. Starting from a small value, the pacing function gradually increases the number of clients participating in the federation. The action of the pacing function amounts to scaling the participation rate of the clients.

The clients are scored and rank-ordered based on the choice of ordering, then a subset of size $K^{(t)}$ of the K clients is chosen in a rank-ordered fashion. The value $K^{(t)}$ is prescribed by the pacing function. The $K^{(t)}$ clients are randomly batched into mini-batches of clients. These mini-batches of clients subsequently participate in the federation process. Thereby, we have two sets of curricula, one that acts locally on the client data and the other that acts on the set of clients. Henceforth, we will refer to these as the `data curriculum` and the `client curriculum`, respectively. We study the interplay of these curricula in Section 4.5.

Fig. 6 confirms that the benefits of the algorithm severely depend on the data of the client having a diverse set of difficulties. As is evident from Fig. 6, we are able to realize an A_{curr} of $5.67 - 7.19\%$ for the different values of Non-IIDness using our proposed client curriculum algorithm in the scenario with high consistency in the client data where the data curriculum has reduced/no benefits. *This illustrates that the client curriculum is able to effectively overcome the limiting constraint of local variability in data and is able to leverage the heterogeneity in the set of clients to its benefit.*



(a) IID **(b)** Dir(0.2) **(c)** Dir(0.05)

Figure 6: Client curriculum does not suffer from low heterogeneity in the data difficulty and is effective when data curriculum is not. The scenario shown here is the same as the scenario with high local consistency from Fig 5. As we observe the client curriculum is able to overcome the limitations of the data curriculum.

4.3. Difficulty based partitioning

For the experiments in this section, we require client datasets ($\mathbb{D}_m$) of varying difficulty at the desired level of Non-IIDness. In order to construct partitions of varying difficulty, we need to address two key challenges: one, we need a way to accurately assess the difficulty of each of the data points, and two, we need to be able to work with different levels of Non-IIDness. To address the first challenge, we rank the data points in order of difficulty using an a priori trained expert model $\theta_{\mathcal{E}}$ that is trained on the entire baseline dataset and has the same network topology as the global model. As the expert model has the same topology as the model that we intend to train and as it is trained on the entire dataset, it is an accurate judge of the difficulty of the data points. Interestingly, this idea can be extended to be used as a scoring method for curriculum as well. We call this scoring method the expert scoring s_E. We look at this in greater detail in Section 4.4.

To address the second challenge, a possible solution is to first partition the standard dataset into the desired Non-IID partitions using well-known techniques such as the Dirichlet distribution, followed by adding different levels of noise to the samples of the different data partitions. This would partition with varying difficulty; however, doing so would alter the standard baseline dataset, and we would lose the ability to compare the results to known baseline values and between different settings. We would like to be able to compare our performance results with standard baselines, so we require a method that does not alter the data or resort to data augmentation techniques, and we devise a technique that does just that.

Starting with the baseline dataset, we first divide it into the desired Non-IID partitions the same as before, but then instead of adding noise to the dataset, we attempt to reshuffle the data among partitions in such a way that we create "easy" partitions and "hard" partitions. This can be achieved by ordering the data in increasing order of difficulty and distributing the data among the partitions starting from the "easy" data points, all the while honoring the Non-IID class counts of each of the partitions as determined by the Non-IID distribution. The outline is detailed in Algorithm 2. It is noteworthy that, although we are able to generate partitions of varying difficulty, we do not have direct control over the "difficulty" of each of the partitions and hence cannot generate partitions with an arbitrary distribution of difficulty as can be done by adding noise.

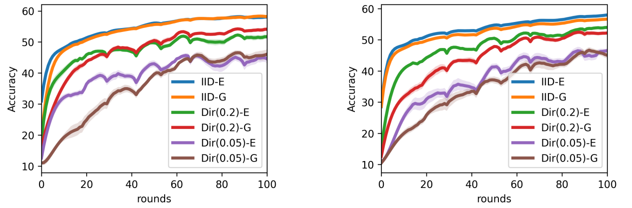
Algorithm 2: Partition Difficulty Distribution

```
Input: partitions $\{\mathbb{P}_0, \mathbb{P}_1, ..., \mathbb{P}_N\}$ of the input dataset $\mathbb{D}$ of $C$ classes indexed by $c$, fraction of each partition to replace $f_{ord} \in (0, 1]$, expert model $\theta_{\mathcal{E}}$
Class prior of partitions and dataset:
    for each partition $i = 0, 1, 2, ..., N$ do
        $\mathbb{P}_i \leftarrow$ count(data points of class $c$ in partition $i$)

    Compute loss ($\mathcal{L}$) for each data point in $\mathbb{D}$ using $\theta_{\mathcal{E}}$
    $\mathbb{D}_c \leftarrow$ argsort($\mathcal{L}_c$)
Reconstitute partition:
    $id_c = cumsum_i(N_{i,c})$

    Distribute $f_{ord}$
    for each partition $i = 0, 1, 2, ..., N$ do
        $\mathbb{P}_i \leftarrow \mathbb{P}_i \cup$ partition($f_{ord} * N_{i,c}$ elements of $\mathbb{D}_c$ beginning at $id_{i,c}$)
    $\mathbb{D}'_c \leftarrow$ remaining elements of $\mathbb{D}_c$

    Distribute remaining $(1 - f_{ord})$
    for each partition $i = 0, 1, 2, ..., N$ do
        $\mathbb{P}_i \leftarrow \mathbb{P}_i \cup$ random($(1 - f_{ord}) * N_{i,c}$ elements of $\mathbb{D}'_c$)
```



(a) Client curriculum **(b)** Data curriculum

Figure 7: Effect of expert scoring s_E and s_G on "curr" curriculum. Plotted here is the evolution of the global model's accuracy over the course of the federation for $\beta \in \{0.05, 0.2\}$ and IID with an ordering of 'curr', using FedAvg. s_G and s_E scoring functions have similar behavior on the Client Curricula (Left) and Data Curricula (Right).

4.4. Expert guided and self-guided curricula

The scoring method s_E, as discussed above, can also be used to guide the learning process in a curriculum learning setting. As the expert model used for scoring shares the same network topology as the global model that we intend to train, and as the expert was trained on the entire dataset, the expert-guided curricula can be thought of as a pedagogical mode of learning.

The global model accuracy at different rounds of the federation is depicted in Fig. 7. We see a clear trend in Fig 7 that s_E outperforms s_G in the initial rounds, but s_G converges to s_E over the rounds. Also, s_G accuracy in the initial rounds very closely approximates the random scoring

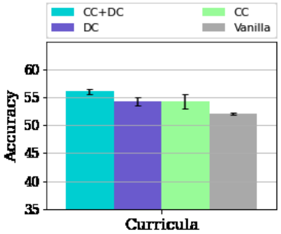


Figure 8: Synergic Effect of Client and Data curricula. CC here refers to Client Curriculum and DC refers to Data Curriculum. The figure shows the synergic effects of the curricula.

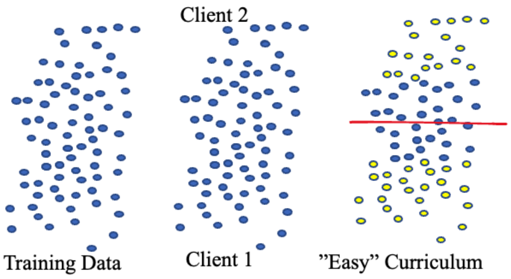


Figure 9: Illustration of skew-based heterogeneous data distribution across clients and curriculum learning mitigation thereof. Note that the easy curricula for the two clients are closer to each other distributionally.

accuracy. The s_G scoring method is a self-guided curriculum, that uses the global model. The global model is just random (noisy) in the initial rounds of the federation, and hence the curricula it produces are also random, thereby closely approximating the performance of the random curriculum. As the global model is refined over time, it becomes better at determining the "true" curricula, eventually converging on the s_E curve. The model trained with s_E benefits from curriculum effects from the first round and thus starts strong.

4.5. Ablation study

In this section, we show the interplay between the `data curriculum` and the `client curriculum` and measure their contributions towards the global model's accuracy. As reported in Fig 8, we observe that the client curriculum and the data curriculum independently outperform the baseline by about $2-3\%$, and we observe a synergic effect of the combination that outperforms both the curricula and the baseline by about 5%.

5. Theoretical Analysis

One can observe, see e.g. [34], that data discrepancy across clients is often manifested as *skew*. As an example, image data is distributed across clients by giving the left division of an image—e.g., the left face of a cat—to one client and the right to another. See Fig. 9 for an illustration.

A simple and transparent way to model this is to use the biased SGD framework. Specifically, there is an underlying objective function of interest $f(x)$, however, each client only has access to a biased stochastic gradient of this function. Uniquely in the case of federated data heterogeneity, the bias adds up to zero across clients (the "right-ness" of one set of images balances the "left-ness" of another). We shall use the notation of [41] although for completeness we acknowledge the predecessor [42]. To the best of our knowledge, we present the first analysis of FedAvg with heterogeneous data using the biased SGD framework, despite how naturally it models the training procedure.

Algorithm 3: Local SGD Model of Algorithm 1

```
Input: $M$ clients indexed by $m$, participating-client number $Q$, communication rounds $T$, local optimization steps $J$, server model $f$ with parameters $\theta_g$
Server executes:
    initialize $f$ with $\theta_g$
    for each round $t = 0, 1, 2, ..., T$ do
        $\mathbb{S}_t \leftarrow$ (random set of $Q$ clients)
        for each client $q \in \mathbb{S}_t$ in parallel do
            broadcast $\theta_g^t$ to clients as $\theta_k^{(t,0)}$
            for $j = 0, 1, 2, ..., J$
                Sample $g_k^{(t,j)} \sim \nabla f(\theta_k^{(t,j)}, \mathcal{D}_k)$
                $\theta_k^{(t,j+1)} \leftarrow \theta_k^{(t,j)} - \alpha^{(t,j)} g_k^{(t,j)}$
            $\theta_g^{(t+1)} = \sum_{k=1}^{K} \frac{|\mathcal{D}_k|}{\sum_{i=1}^{K} |\mathcal{D}_i|} \theta_k^{(t,J)}$
    return $\theta_g^{t+1}$
```

5.1. Formal Convergence Theory Statements

We formalize the notion of distributional skew by making the following assumption on the bias structure associated with each stochastic gradient computation:

Assumption 1 *It holds that $g_k^{(t,j)}$ satisfies,*

$$g_k^{(t,j)} = \nabla f(\theta_k^{(t,j)}) + b_k^{(t,j)}(\theta_k^{(t,j)}) + n_k^{(t,j)}(\theta_k^{(t,j)}, \xi_k^{(t,j)}) \tag{2}$$

where $\|b_k^{(t,j)}(\theta_k^{(t,j)})\|^2 \leq B^{(t,j)}$ for all k, and, for all θ,

$$\sum_{k \in S_t} b_k^{(t,j)}(\theta) = 0 \tag{3}$$

and $\xi_k^{(t,j)}$ is a random variable satisfying,

$$\mathbb{E}_\xi[n_k^{(t,j)}(\theta_k^{(t,j)}, \xi_k^{(t,j)})] = 0 \tag{4}$$

We note that,

$$\begin{aligned} &B^{(0,0)} = B^{(0,1)} = ... = B^{(0,J)} < B^{(1,0)} = B^{(1,1)} = ... \\ &= B^{(1,J)} < ... < B^{(t,0)} = B^{(t,1)} = ... = B^{(t,J)} < B^{(t+1,0)} \\ &= B^{(t+1,1)} = ... = B^{(t+1,J)} < ... < B^{(T,0)} = B^{(T,1)} \\ &= ... = B^{(T,J)} \end{aligned}$$

for **client based** curriculum training, and

$$B^{(0,0)} < B^{(0,1)} < ... = B^{(0,J)} = B^{(1,0)} < B^{(1,1)} < ...$$
$$< B^{(1,J)} = B^{(2,0)} < ...B^{(t,0)} < B^{(t,1)} < ...$$
$$< B^{(t,J)} = B^{(t+1,0)} < B^{(t+1,1)} < ...$$
$$< B^{(t+1,J)}...B^{(T,K-1)} < B^{(T,J)}$$

for **data based** curriculum training.

Now we present two results as depending on the conditions applying to the functions characterizing the optimization. In the first case, we shall consider strongly convex objectives, as characterizing least squares empirical risk minimization of, e.g., linear models. In this scenario, we permit the variance to grow with the parameter size, i.e., we do not assume bounded gradients.

Theorem 1 *Assume that*

- *f is strongly convex with convexity parameter $\mu > 0$*
- *∇f is Lipschitz continuous with Lipschitz constant L*
- *the noise variance satisfies,*

$$\mathbb{E}_\xi \left[\left\| n_k^{(t,j)}(\theta_k^{(t,j)}, \xi_k^{(t,j)}) \right\|^2 \right] \\ \leq M \left\| \nabla f(\theta_k^{(t,j)}) + b_k^{(t,j)}(\theta_k^{(t,j)}) \right\|^2 + \sigma^2$$

- *For all t, j we have ,*

$$\alpha^{(t,j)} \leq \frac{1}{4(3+2M)L}$$

Then it holds that the distance to the solution satisfies, after each averaging step,

$$\mathbb{E} \left\| \hat{\theta}^{(T,0)} - \theta^* \right\|^2 \leq \prod_{t=1}^{T} \prod_{j=0}^{J} (1 - \alpha^{(t,j)}\mu/2) \|\hat{\theta}^{(0,0)} - \theta^*\|^2 \\ + \sum_{t=1}^{T} \sum_{j=0}^{J} \frac{2(\alpha^{(t,j)})^2 [L((3+2M)B^{(t,j)}+3\sigma^3]}{Q} \\ + \sum_{t=1}^{T} \sum_{j=0}^{J} \frac{2\alpha^{(t,j)} L(B^{(t,j)})^2/\mu)}{Q}$$

In studying the form of this result, we note that the overall convergence rate and error resembles the original with an important caveat in regards to the error on account of the bias term. First, the bias term adds an error proportional to the stepsize, thus yielding an asymptotic error bounded from below with the bias. Second, the stepsize can be used to mitigate the error from the bias terms. Indeed, with, e.g., data-based curriculum, if $B^{(t,j)} = O(j^{1/4})$ then $\alpha^{(t,j)} = O(t^{-1} j^{-1/4})$ would mitigate the growing error. It is clear that the standard practice of diminishing stepsizes will result in a lower total error at each iteration for curriculum compared to anti-curriculum. Standard Local SGD guarantees are not preserved regardless, however, with the asymptotic bias depending on the total degree of data heterogeneity, summed in this weighted manner throughout the optimization procedure.

Nonconvex Objectives Now we consider the general case of nonconvex objectives without any additional conditions regarding the growth properties of the objective function to permit generality encompassing the functional properties of neural networks. Using the biased SGD framework and inspired by the structure of the convergence theory of [43], we study the effect of the associated gradient estimate errors.

Theorem 2 *Assume that*

- *$\|\nabla f\|$ is uniformly bounded by G*
- *∇f is Lipschitz continuous with Lipschitz constant L*
- *the noise variance satisfies,*

$$\mathbb{E}_\xi \left[\left\| n_k^{(t,j)}(\theta_k^{(t,j)}, \xi_k^{(t,j)}) \right\|^2 \right] \leq \sigma^2$$

- *f is lower bounded by f_**

Then we obtain the ergodic rate,

$$\sum_{t=0}^{T} \sum_{j=0}^{J} \|\nabla f(\theta^{(t,0)})\|^2 \leq Q(f(\theta^0) - f^*) \\ +2 \sum_{t=0}^{T} \sum_{j=0}^{J} \alpha^{(t,j)} \left(\alpha^{(t,j)} + \sum_{l=j}^{J} \alpha^{(t,l)} \right) LG^2$$

Compared to standard results, we can see that curricula contribute an error that corresponds to the cross terms of the stepsizes, indicating a benefit to annealing the stepsize along local iterations as well as along averaging steps.

The proofs that build up the argument for the main new convergence results we present in the Supplementary Material Section 1, specifically Theorem 1 and 2.

6. Conclusion

In this work, we provided a comprehensive study on the benefit of employing CL in FL under both homogeneous and heterogeneous settings. We further ran extensive experiments on a broad range of curricula and pacing functions over three datasets, CIFAR10 and CIFAR100, and demonstrated that ordered learning can have noticeable benefits in federated training. Surprisingly, we found empirically that CL can be more beneficial when the clients' underlying data distributions are significantly Non-IID. By studying the convergence behavior of FL using a novel biased SGD model based on the observation of data heterogeneity as distributional skew, we were able to theoretically explain this phenomenon. Moreover, we proposed a curriculum on clients for the first time. Our results show that the order in which clients participate in the federation plays an important role in the accuracy performance of the global model. In particular, training the global model in a meaningful order, from the easy clients to the hard ones, using CL can provide performance improvements over the random sampling approach.

References

[1] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. volume 382, pages 41–48. ACM, 2009.

[2] Lu Jiang, Deyu Meng, Qian Zhao, Shiguang Shan, and Alexander G. Hauptmann. Self-paced curriculum learning. In Blai Bonet and Sven Koenig, editors, *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, January 25-30, 2015, Austin, Texas, USA*, pages 2694–2700. AAAI Press, 2015.

[3] Te Pi, Xi Li, Zhongfei Zhang, Deyu Meng, Fei Wu, Jun Xiao, and Yueting Zhuang. Self-paced boost learning for classification. In Subbarao Kambhampati, editor, *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI 2016, New York, NY, USA, 9-15 July 2016*, pages 1932–1938. IJCAI/AAAI Press, 2016.

[4] Xuan Zhang, Gaurav Kumar, Huda Khayrallah, Kenton Murray, Jeremy Gwinnup, Marianna J. Martindale, Paul McNamee, Kevin Duh, and Marine Carpuat. An empirical exploration of curriculum learning for neural machine translation. *CoRR*, abs/1811.00739, 2018.

[5] Xuan Zhang, Pamela Shapiro, Gaurav Kumar, Paul McNamee, Marine Carpuat, and Kevin Duh. Curriculum learning for domain adaptation in neural machine translation. *CoRR*, abs/1905.05816, 2019.

[6] Jihan Yang, Shaoshuai Shi, Zhe Wang, Hongsheng Li, and Xiaojuan Qi. ST3D: self-training for unsupervised domain adaptation on 3d object detection. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, pages 10368–10378, 2021.

[7] Enver Sangineto, Moin Nabi, Dubravko Culibrk, and Nicu Sebe. Self paced deep learning for weakly supervised object detection. *IEEE Trans. Pattern Anal. Mach. Intell.*, 41(3):712–725, 2019.

[8] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21:140:1–140:67, 2020.

[9] Xiaoxia Wu, Ethan Dyer, and Behnam Neyshabur. When do curricula work? In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*, 2021.

[10] Sheng Guo, Weilin Huang, Haozhi Zhang, Chenfan Zhuang, Dengke Dong, Matthew R. Scott, and Dinglong Huang. Curriculumnet: Weakly supervised learning from large-scale web images. In *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part X*, volume 11214, pages 139–154. Springer, 2018.

[11] Amelia Jiménez-Sánchez, Mickael Tardy, Miguel Ángel González Ballester, Diana Mateus, and Gemma Piella. Memory-aware curriculum federated learning for breast cancer classification. *CoRR*, abs/2107.02504, 2021.

[12] Zhipeng Ren, Daoyi Dong, Huaxiong Li, and Chunlin Chen. Self-paced prioritized curriculum learning with coverage penalty in deep reinforcement learning. *IEEE Transactions on Neural Networks and Learning Systems*, 29(6):2216–2226, 2018.

[13] Yi Tay, Shuohang Wang, Anh Tuan Luu, Jie Fu, Minh C. Phan, Xingdi Yuan, Jinfeng Rao, Siu Cheung Hui, and Aston Zhang. Simple and effective curriculum pointer-generator networks for reading comprehension over long narratives. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 4922–4931, 2019.

[14] Chen Gong, Jian Yang, and Dacheng Tao. Multi-modal curriculum learning over graphs. 10(4), 2019.

[15] Yong Guo, Yaofo Chen, Yin Zheng, Peilin Zhao, Jian Chen, Junzhou Huang, and Mingkui Tan. Breaking the curse of space explosion: Towards efficient NAS with curriculum search. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119, pages 3822–3831. PMLR, 2020.

[16] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial Intelligence and Statistics*, pages 1273–1282. PMLR, 2017.

[17] Saeed Vahidian, Mahdi Morafah, and Bill Lin. Personalized federated learning by structured and unstructured pruning under data heterogeneity. In *41st IEEE International Conference on Distributed Computing Systems Workshops, ICDCSW 2021, Washington, DC, USA, July 7-10, 2021*, pages 27–34. IEEE, 2021.

[18] Mahdi Morafah, Saeed Vahidian, Chen Chen, Mubarak Shah, and Bill Lin. Rethinking data heterogeneity in federated learning: Introducing a new notion and standard benchmarks. In *NeurIPS 2022 workshop*, 2022.

[19] Saeed Vahidian, Mahdi Morafah, Chen Chen, Mubarak Shah, and Bill Lin. Rethinking data heterogeneity in federated learning: Introducing a new notion and standard benchmarks. *IEEE Transactions on Artificial Intelligence*, pages 1–13, 2023.

[20] Ziheng Jiang, Chiyuan Zhang, Kunal Talwar, and Michael C. Mozer. Exploring the memorization-generalization continuum in deep learning. *CoRR*, abs/2002.03206, 2020.

[21] Tianyi Zhou, Shengjie Wang, and Jeff A. Bilmes. Curriculum learning by dynamic instance hardness. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

[22] Anastasia Pentina, Viktoriia Sharmanska, and Christoph H. Lampert. Curriculum learning of multiple tasks. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*, pages 5492–5500. IEEE Computer Society, 2015.

[23] Guy Hacohen and Daphna Weinshall. On the power of curriculum learning in training deep networks. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 2535–2544. PMLR, 2019.

[24] Yann LeCun, Bernhard Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne Hubbard, and Lawrence D Jackel. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4):541–551, 1989.

[25] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[26] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[27] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. In *Proceedings of Machine Learning and Systems 2020, MLSys 2020, Austin, March 2-4, 2020*. mlsys.org, 2020.

[28] Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank J. Reddi, Sebastian U. Stich, and Ananda Theertha Suresh. SCAFFOLD: stochastic controlled averaging for federated learning. In *Proceedings of the 37th International Conference on Machine Learning, ICML*, volume 119, pages 5132–5143. PMLR, 2020.

[29] Jianyu Wang, Qinghua Liu, Hao Liang, Gauri Joshi, and H. Vincent Poor. Tackling the objective inconsistency problem in heterogeneous federated optimization. In *Advances in Neural Information Processing Systems*, volume 33, pages 7611–7623. Curran Associates, Inc., 2020.

[30] Saeed Vahidian, Mahdi Morafah, Weijia Wang, Vyacheslav Kungurtsev, Chen Chen, Mubarak Shah, and Bill Lin. Efficient distribution similarity identification in clustered federated learning via principal angles between client data subspaces. In *Thirty-Seventh AAAI Conference on Artificial Intelligence, AAAI 2023, Washington, DC, USA, February 7-14, 2023*, pages 10043–10052. AAAI Press, 2023.

[31] Mahdi Morafah, Saeed Vahidian, Weijia Wang, and Bill Lin. FLIS: clustered federated learning via inference similarity for non-iid data distribution. *CoRR*, abs/2208.09754, 2022.

[32] Jianyi Zhang, Saeed Vahidian, Martin Kuo, Chunyuan Li, Ruiyi Zhang, Guoyin Wang, and Yiran Chen. Towards building the federated GPT: federated instruction tuning. *CoRR*, abs/2305.05644, 2023.

[33] Qinbin Li, Yiqun Diao, Quan Chen, and Bingsheng He. Federated learning on non-iid data silos: An experimental study. *arXiv preprint arXiv:2102.02079*, 2021.

[34] Hangyu Zhu, Jinjin Xu, Shiqing Liu, and Yaochu Jin. Federated learning on non-iid data: A survey. *Neurocomputing*, 465:371–390, 2021.

[35] Wenlin Chen, Samuel Horváth, and Peter Richtárik. Optimal client sampling for federated learning. *CoRR*, abs/2010.13723, 2020.

[36] Ihab Mohammed, Shadha Tabatabai, Ala I. Al-Fuqaha, Faissal El Bouanani, Junaid Qadir, Basheer Qolomany, and Mohsen Guizani. Budgeted online selection of candidate iot clients to participate in federated learning. *IEEE Internet Things J.*, 8(7):5938–5952, 2021.

[37] Yae Jee Cho, Jianyu Wang, and Gauri Joshi. Client selection in federated learning: Convergence analysis and power-of-choice selection strategies. *CoRR*, abs/2010.01243, 2020.

[38] Sai Qian Zhang, Jieyu Lin, and Qi Zhang. A multi-agent reinforcement learning approach for efficient client selection in federated learning. In *Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022, Thirty-Fourth Conference on Innovative Applications of Artificial Intelligence, IAAI 2022, The Twelveth Symposium on Educational Advances in Artificial Intelligence, EAAI 2022 Virtual Event, February 22 - March 1, 2022*, pages 9091–9099. AAAI Press, 2022.

[39] Farzin Haddadpour and Mehrdad Mahdavi. On the convergence of local descent methods in federated learning. *arXiv preprint arXiv:1910.14425*, 2019.

[40] Yuyang Deng, Mohammad Mahdi Kamani, and Mehrdad Mahdavi. Adaptive personalized federated learning. *CoRR*, abs/2003.13461, 2020.

[41] Ahmad Ajalloeian and Sebastian U Stich. On the convergence of sgd with biased gradients. *arXiv preprint arXiv:2008.00051*, 2020.

[42] Belhal Karimi, Blazej Miasojedow, Eric Moulines, and Hoi-To Wai. Non-asymptotic analysis of biased stochastic approximation scheme. In *Conference on Learning Theory*, pages 1944–1974. PMLR, 2019.

[43] Fan Zhou and Guojing Cong. On the convergence properties of a k-step averaging stochastic gradient descent algorithm for nonconvex optimization. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, pages 3219–3227, 2018.

[44] Ahmed Khaled, Konstantin Mishchenko, and Peter Richtárik. Tighter theory for local sgd on identical and heterogeneous data. In *International Conference on Artificial Intelligence and Statistics*, pages 4519–4529. PMLR, 2020.

[45] Sebastian Urban Stich. Local sgd converges fast and communicates little. In *ICLR 2019-International Conference on Learning Representations*, number CONF, 2019.

[46] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. *Proceedings of Machine Learning and Systems*, 2:429–450, 2020.

[47] Mariya Toneva, Alessandro Sordoni, Remi Tachet des Combes, Adam Trischler, Yoshua Bengio, and Geoffrey J. Gordon. An empirical study of example forgetting during deep neural network learning. *CoRR*, abs/1812.05159, 2018.