On Sampling Strategies for Spectral Model Sharding

Denis Korzhenkov
Qualcomm AI Research*
Amsterdam, The Netherlands
dkorzhen@qti.qualcomm.com

Christos Louizos
Qualcomm AI Research*
Amsterdam, The Netherlands
clouizos@qti.qualcomm.com

Abstract

The problem of heterogeneous clients in federated learning has recently drawn a lot of attention. Spectral model sharding, i.e., partitioning the model parameters into low-rank matrices based on the singular value decomposition, has been one of the proposed solutions for more efficient on-device training in such settings. In this work, we present two sampling strategies for such sharding, obtained as solutions to specific optimization problems. The first produces unbiased estimators of the original weights, while the second aims to minimize the squared approximation error. We discuss how both of these estimators can be incorporated in the federated learning loop and practical considerations that arise during local training. Empirically, we demonstrate that both of these methods can lead to improved performance on various commonly used datasets.

1 Introduction

Due to increasing concerns about user data privacy, a federated approach becomes a viable alternative for common centralized methods of machine learning [25]. Unlike centralized training on powerful servers, the federated setting faces a variety of user devices like personal computers, mobile phones, tablets, etc. which may have different constraints in terms of memory footprint, spare CPU resources, battery levels, and many others [34]. Therefore, in practice, the scenario where each user trains the same model may be impossible.

To tackle this problem, several methods which carefully create smaller sub-models from a larger global model were proposed [6, 17, 1, 44]. The general recipe for each communication round is composed of three steps. Initially, the server receives from each client the information about its capabilities and sends a sub-model of the corresponding size to that client. Afterwards, each of the clients update their respective sub-models for a specific number of steps. Finally, the server collects the fine-tuned models from each client and aggregates them into a single global model.

In order to create sub-models, the recently proposed method called PriSM suggests to conduct singular value decomposition (SVD) of the weight matrices of affine layers and randomly sample the terms from this decomposition [32, 33]. The number of sampled terms is a hyperparameter which depend on the capacity of the client. Despite the fact that PriSM has shown promising results on several datasets, its sampling design remains heuristic.

In this work, we show that by introducing certain first principles, better strategies can be obtained via solutions to specific optimization problems. More specifically, our aim is to obtain the inclusion probabilities for the SVD factors in a way that minimizes the expected, under these probabilities, squared approximation error. In isolation, this requirement leads to the deterministic top-n sampling, as the Eckart–Young–Mirsky theorem implies [8]. However, we propose two additional principles

*Qualcomm AI Research, Qualcomm Technologies Netherlands B.V (Qualcomm AI Research is an initiative of Qualcomm Technologies, Inc.).

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

https://doi.org/10.52202/079017-1968

which lead to two novel strategies, respectively. The first is inspired by the commonly used *inverted* implementation of dropout regularization [37, 24], and searches for the solution in the family of unbiased estimators. The second one takes into account all the (participating in a given round) client specific sub-models, and aims to reconstruct the full weight matrix from all these observations (instead of considering each client separately). Notably, the solutions for both strategies can be presented in closed form. Furthermore, since the sampling of sub-models takes place on the server side, the computation of the optimal probabilities in these settings does not require any additional calculation on potentially weak clients. Finally, we employ such procedures for sampling without replacement which preserve given inclusion probabilities [39].

2 Related Works

Heterogeneity in its different aspects is ubiquitous in real-life setups for federated learning and is therefore a long-standing topic of research [11, 45]. Probably the most well-studied issue is the non-i.i.d. distributed data samples, available on clients' devices [19]. However, the deployment of federated systems also faces the variety of devices which often have different computational constraints [34]. This makes training the same model for each client infeasible and motivates the search for new solutions.

Many popular approaches to tackle this kind of heterogeneity follow one of two directions. Methods based on knowledge distillation in most cases use client models to train a new predictor on the server side [27, 14]. However, these techniques typically require a separate dataset on the server, which may be a serious limitation in practice. In contrast, partial training-based approaches do all the training solely on the client side. To fit the clients' constraints, they sample sub-models which have a size that allows them to be trained on the corresponding device. After local training, these sub-models are aggregated into a global model. Notably, this allows for a final model size that is otherwise infeasible to train in any of the available clients.

Many such methods conduct the sub-sampling in the original model space, e.g., by selecting specific output dimensions on each layer. While HeteroFL [6] simply selected the top-left corner of the weight matrix of the required size, methods such as Federated Dropout [4] and FjORD [17] argued for randomized selection. FedRolex [1] replaced randomness with rolling windows, with the aim of covering all possible sub-models in a shorter period of time.

Recent advances in using factorized low-rank models, both for training from scratch [22, 21, 38, 43] and for fine-tuning [20], found applications in the realm of federated learning [2, 46]. E.g., FLANC used this idea to address heterogeneous computational constraints by representing a weight matrix as a product of two factors, one fixed for all clients and the other having a varying size based on the capacity of each client [30]. Such approaches for model sub-sampling can be generally useful, given the fact that the (stable) rank of the model weights tends, in some cases, to decrease during training [41, 43]. In a similar spirit, FedHM [44] applied truncated SVD to create the sub-models, while PriSM [32, 33] employed a random procedure for choosing the SVD factors. In our work, we propose novel strategies for the selection of sub-models and their training in such scenarios and demonstrate their advantages.

3 Method

3.1 Preliminaries

In the case of cross-device federated learning, low-end clients may not be able to perform the computations with the full weight matrix W in all the layers due to various constraints. A possible remedy for this problem can be to use an approximation for W. In this work we consider a particular case of such approximations, namely sub-sampling the terms from the singular decomposition (SVD) $W = \sum_{i=1}^{N} \lambda_i u_i v_i^T$, where both sets of column vectors $\{u_i \in \mathbb{R}^{c_{out}}\}_i$ and $\{v_i \in \mathbb{R}^{c_{in}}\}_i$ are orthonormal, and the singular values are sorted in a non-increasing order $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_N > 0$. Due to the usage of singular decomposition, we name this type of sub-model sampling as *spectral model sharding*.

More formally, let $0 < r \leq 1$ be the *keep ratio* which depends on the client's capabilities, then the goal is to sample the $n = \lceil Nr \rceil$ terms from the full sum. Following prior works, keep ratio r is shared

https://doi.org/10.52202/079017-1968

across layers [33, 32, 30]. With a sufficiently low value of r, this reduces both communication costs and computational complexity. We introduce a vector of indicators $z = (z_1, \ldots, z_N) \in \{0, 1\}^N$ such that $\sum_{i=1}^{N} z_i = n$, and the corresponding *sub-layer* is created with a weight matrix $\hat{W}$,

$$\hat{W} = \sum_{i=1}^{N} z_i \omega_i \lambda_i u_i v_i^T = \sum_{j=1}^{n} \omega_{\mathcal{I}_j} \lambda_{\mathcal{I}_j} u_{\mathcal{I}_j} v_{\mathcal{I}_j}^T, \tag{1}$$

where $\{\omega_i \in \mathbb{R}\}_i$ are some scalar *auxiliary multipliers*, and $\mathcal{I}$ is the set of selected indices, $\mathcal{I} = \{i : z_i = 1\}$.

If the sampling of indicators z_i is random, we denote the corresponding marginal inclusion probabilities as $\pi_i = \Pr(z_i = 1)$. The joint distribution of indicator variables is interchangeably referred to as $p(z)$ or $p(\mathcal{I})$. Wherever the expectation symbol $\mathbb{E}$ is used in the text, expectation w.r.t. $p(z)$ is assumed. If it is necessary to emphasize that the particular estimator, multiplier, vector, etc. is used by the client c, this is shown with the client index in parentheses, e.g. $\hat{W}^{(c)}$.

Based on the previous models of FedHM [44] and PriSM [32, 33], we consider the following pipeline of sharded model training in the case of federated learning on heterogeneous devices:

1. In the beginning of each communication round, the server performs SVD for the weight matrices of the affine layers. Then, for each client c it randomly samples a subset of indices $\mathcal{I}^{(c)}$ from the decomposition with keep ratio $r^{(c)}$, $\hat{W}^{(c)} = \sum_{j=1}^{n^{(c)}} \omega_{\mathcal{I}_j^{(c)}} u'_{\mathcal{I}_j^{(c)}} v'^T_{\mathcal{I}_j^{(c)}}$. The singular values are absorbed into the vectors, namely $u'_i = \sqrt{\lambda_i} u_i$, $v'_i = \sqrt{\lambda_i} v_i$. Note that in both FedHM and PriSM, the auxiliary multipliers ω_i (which can be used to 'compensate' for the dropped terms) were set to 1. However, since they are actively used in our method, we include them in the description.
2. Each participating client c receives the matrices $U^{(c)} = \left(u'_{\mathcal{I}_1^{(c)}}, \ldots, u'_{\mathcal{I}_n^{(c)}}\right) \in \mathbb{R}^{c_{out} \times n}$ and $V^{(c)} = \left(v'_{\mathcal{I}_1^{(c)}}, \ldots, v'_{\mathcal{I}_n^{(c)}}\right) \in \mathbb{R}^{c_{in} \times n}$ as well as the vector $\omega^{(c)} = \left(\omega_{\mathcal{I}_1^{(c)}}, \ldots, \omega_{\mathcal{I}_n^{(c)}}\right) \in \mathbb{R}^n$ from the server and performs local training with the factorized weight matrix $U^{(c)} \Omega^{(c)} V^{(c)T}$ for some predefined number of epochs. During training $U^{(c)}$ and $V^{(c)}$ are updated while the diagonal matrix $\Omega^{(c)}$ which has values of $\omega^{(c)}$ on the diagonal is kept frozen. When the local training is done, the updated matrices $U^{(c)}$ and $V^{(c)}$ are sent back to the server.
3. The server aggregates each vector from the decomposition separately, i.e. $u'_i \leftarrow \frac{\sum_{c:\, z_i^{(c)}=1} D^{(c)} u_i'^{(c)}}{\sum_{c:\, z_i^{(c)}=1} D^{(c)}}$, where $D^{(c)}$ is the size of the local dataset of the client c and $u_i'^{(c)}$ is the updated value of the i-th vector received from the client c. If the i-th term was not selected by any client in the current round, it remains the same. Vectors $\{v'_i\}_i$ are aggregated in the same way. Afterward, the updated full weight matrix is calculated as $W \leftarrow \sum_{i=1}^{N} u'_i v_i'^T$, and a new communication round begins.

FedHM proposed to use the n terms corresponding to the largest singular values, a deterministic selection instead of random sampling. In contrast, PriSM presented a randomized algorithm, which samples the terms according to the magnitudes of the singular values. It employs the NumPy [12] method `numpy.random.choice` with the option `replace=False`, and the associated probabilities are set proportional to λ_i^k, where k is a hyperparameter depending on the keep ratio r. Sampling in this manner results in a probability law similar to Wallenius' noncentral hypergeometric distribution [42, 5, 10, 9], see Appendix C for details. Such sampling procedures make the analysis of the resulting marginal inclusion probabilities $\{\pi_i\}_i$ complicated [40]. Thus, the statistical properties of the obtained matrix approximation are opaque.

3.2 Sampling Distribution

As opposed to the prior works, we consider the search of an optimal sampling distribution to be the central task. We introduce two assumptions which we use to derive our two different strategies.

https://doi.org/10.52202/079017-1968

3.2.1 Unbiased Estimator

The bias and variance are inherent trade-offs for an estimator. To keep the bias in check, we propose an assumption of the unbiasedness of the sampled estimator $\hat{W}$ of the original weight matrix W, i.e. $\mathbb{E}\hat{W} = W$, where the expectation is taken over the sampling distribution $p(\mathcal{I})$ with the marginal inclusion probabilities $(\pi_1, \dots, \pi_N)$. As shown in Appendix B.1, this requirement necessarily leads to the following values of auxiliary multipliers $\omega_i = \pi_i^{-1}$. This selection of multipliers is known in literature as a Horvitz-Thompson estimator [16] and resembles the common implementation of *inverted dropout* [37, 24] which assumes dividing the non-zeroed activations by the inclusion probability at training time.

With the bias of the estimator taken under control, we aim to reduce the approximation error of the resulting layer. For any input vector $x \in \mathbb{R}^{c_{in}}$, the following inequality holds

$$\left\|Wx - \hat{W}x\right\|_2^2 \leq \left\|W - \hat{W}\right\|_2^2 \cdot \|x\|_2^2 \leq \left\|W - \hat{W}\right\|_F^2 \cdot \|x\|_2^2 , \tag{2}$$

where $\|\cdot\|_F^2$ is a squared Frobenius norm which equals the sum of the squared singular values. Thus, we can control the upper-bound of the expected error by searching for sampling distribution $p(\mathcal{I})$ that minimizes the *Frobenius discrepancy*

$$\min_{p(\mathcal{I})} \quad \mathbb{E}_{p(\mathcal{I})} \left\|W - \hat{W}\right\|_F^2 . \tag{3}$$

Theorem 3.1. *For an unbiased estimator $\hat{W}$ of the type specified in Eq.* (1) *and consisting of n terms, the Frobenius discrepancy can be expressed in terms of the marginal inclusion probabilities*

$$\mathbb{E}\left\|W - \hat{W}\right\|_F^2 = \sum_{i=1}^{N} \lambda_i^2 \left(\pi_i^{-1} - 1\right) , \tag{4}$$

and the optimal set of inclusion probabilities has the following form

$$\pi_i = \begin{cases} 1, & \text{if } i \leq t, \\ \frac{(n-t)\lambda_i}{\sum_{k=t+1}^{N} \lambda_k}, & \text{if } i > t \end{cases} \tag{5}$$

for $i = 1, \dots, N$, where $t \in \{0, \dots, n-1\}$.

Proof. See Appendix B.1. □

As follows from Theorem 3.1, to find the true arguments of the minima of Eq. (4), one needs to sweep over n possible values of t, evaluate the Frobenius discrepancy and select the t with the minimal discrepancy. Note that this procedure takes place on the server side and therefore does not require extra computation on clients' devices.

3.2.2 Collective Estimator

Due to the well-known bias-variance trade-off, unbiased estimators in practice can have too large variance. This motivates us to consider another perspective. Consider the simplifying assumption where C clients participating in the current communication round share the same number of terms n in their respective estimators. We can target 'reconstructing' the full matrix W from the whole set of i.i.d. observations $\left\{\hat{W}^{(c)}\right\}_{1\leq c\leq C}$. The simplest way of obtaining such reconstruction is averaging, i.e. $\bar{W} = \frac{1}{C}\sum_{c=1}^{C} \hat{W}^{(c)}$, and we refer to the matrix $\bar{W}$ as the *collective estimator*. To ensure that our reconstruction is accurate, we aim to find the optimal sampling distribution as well as the set of auxiliary multipliers,

$$\min_{p(\mathcal{I}), \{\omega_i\}_i} \quad \mathbb{E}_{p(\mathcal{I})} \left\|W - \bar{W}\right\|_F^2 . \tag{6}$$

Theorem 3.2. *For a collective estimator $\bar{W}$, the average value of the Frobenius discrepancy can be expressed in terms of just the marginal inclusion probabilities*

$$\mathbb{E}\left\|W - \bar{W}\right\|_F^2 = \sum_{i=1}^{N} \lambda_i^2 \omega_i \pi_i \left(-2 + \frac{\omega_i}{C} + \frac{\omega_i \pi_i (C-1)}{C}\right) + \sum_{i=1}^{N} \lambda_i^2, \tag{7}$$

https://doi.org/10.52202/079017-1968

and the optimal set of inclusion probabilities and auxiliary weights has the following form in the case of $C > 1$

$$\omega_i = \frac{C}{1 + \pi_i\,(C-1)}, \qquad \pi_i = \begin{cases} 1, & \textit{if } i \le t, \\ \frac{\left(n-t+\frac{u}{C-1}\right)\lambda_i}{\sum_{k=t+1}^{t+u} \lambda_k} - \frac{1}{C-1}, & \textit{if } t < i \le t+u, \\ 0, & \textit{if } i > t+u, \end{cases} \tag{8}$$

for $i = 1, \ldots, N$, *where* t *and* u *are integer constants such that* $0 \le t \le n$, *and* $0 \le u \le N - t$. *For* $C = 1$ *the optimal values are* $\pi_i = \omega_i = \mathbb{I}(i \le n)$.

Proof. For $C = 1$ the proof immediately follows from the Eckart–Young–Mirsky theorem [8] which states that truncated SVD provides the best low-rank approximation of the original matrix in terms of Frobenius discrepancy. Since all the estimators defined by Eq. (1) are low-rank approximations, it is impossible to obtain an average error which is strictly lower than the error of truncated SVD. For the case $C > 1$, see Appendix B.2. □

Note that for a group consisting of a single client (i.e. $C = 1$) the optimal solution coincides with top-n sampling used in FedHM method. For larger groups, similarly to the unbiased estimator, sweeping across all possible values of t and u is required on the server side. Nevertheless, this search can be performed in vectorized form.

In practice, to apply Theorem 3.2, the server can cluster the heterogeneous clients into homogeneous groups which share that same keep ratio r, and compute a specific sampling distribution for each group. Such clustering is often employed by methods targeting diverse clients, cf. [30].

3.2.3 Joint Distribution and Sampling

The strategies derived in Secs. 3.2.1 and 3.2.2 provide the values of marginal inclusion probabilities but do not specify the joint distribution, i.e. the co-occurrences of sampled terms. Tillé [39] presented a systematic survey of multiple sampling procedures which preserve the given inclusion probabilities. Notably, each of the methods, while keeping the mean vector $\mathbb{E}\left(z_1, \ldots, z_N\right) = \left(\pi_1, \ldots, \pi_N\right)$ fixed, has different covariance matrices. We follow the authors' recommendation and stick with the Conditional Poisson scheme (CPS) for our experiments. The joint distribution achieved with this method has the maximum entropy among all distributions over samples of a size of exactly n with the same marginals. In theory, such a design should allow the model to better explore the interaction between the singular vectors during training. The reference implementation of this sampling algorithm is provided in the accompanying package[2] for [39].

3.3 Local Training

While the derivations in Sec. 3.2 provide an optimal (in a certain sense) estimator $\hat{W}^{(c)}$ of the full weight matrix W for the client c *before* the local training starts on that client, they do not provide any guidance on how to optimize the received sub-model.

Auxiliary multipliers. As indicated in Sec. 3.1, we follow Khodak et al. [22] and Niu et al. [33] and absorb the square root of the singular value λ_i into both left and right singular vectors, $u_i' = \sqrt{\lambda_i} u_i$, $v_i' = \sqrt{\lambda_i} v_i$. During early stages of our experiments we explored the absorption of the auxiliary multipliers ω_i in the same manner. However, this led to unsatisfactory results. Note that, except for the corner cases $\pi_i \in \{0, 1\}$, the multiplier ω_i is in inverse proportion to λ_i and therefore the magnitude of their product is almost independent of i. Therefore, after the product $\sqrt{\omega_i} \cdot \sqrt{\lambda_i}$ is directly 'baked' into the sub-model weights, it is harder for the clients to distinguish between important and uninformative terms.

Instead, we treat the auxiliary multipliers as scaling factors, frozen for the current round of local training. Hence, the effective weight matrix of the sub-model is trained locally in the factorized form $\hat{W} = U \Omega V^T$ with learnable terms U and V. This could introduce an issue of stale multipliers: during the course of local training the original meaning, along with the guarantees on the desired estimator properties, of the auxiliary multipliers is gradually fading away. However, this was not

[2] `https://cran.r-project.org/package=sampling`

https://doi.org/10.52202/079017-1968

something we observe in practice and any attempts to mitigate this (e.g., with gradual decrease of the scaling factors) did not provide consistent improvement for the model performance.

Effective learning rate. Nevertheless, the chosen way of incorporating the auxiliary multipliers into the model still has a drawback. Namely, the gradient of the scalar loss function L w.r.t. the factors is expressed as $\frac{\partial L}{\partial U} = \frac{\partial L}{\partial \hat{W}} V\Omega, \quad \frac{\partial L}{\partial V} = \left(\frac{\partial L}{\partial \hat{W}}\right)^T U\Omega$. This means that the auxiliary multipliers ω_i also affect the effective learning rate of the corresponding column vectors u'_i and v'_i when first-order optimization methods like SGD are used. For the collective estimator it follows from Eq. (8) that the values of the multipliers are bounded by the size of the group, $1 \leq \omega_i \leq C$. However, for the unbiased estimators there is no such upper bound as $\omega_i = \pi_i^{-1}$. Moreover, for small values of the keep ratio r there necessarily should exist relatively large values of π_i^{-1}. This is because

$$\mathbb{E} \textstyle\sum_{j=1}^{n} \omega_{\mathcal{I}_j} = \mathbb{E} \sum_{j=1}^{n} \pi_{\mathcal{I}_j}^{-1} = \mathbb{E} \sum_{i=1}^{N} z_i \pi_i^{-1} = N, \tag{9}$$

and one needs to approximate N with $n \approx rN$ randomly chosen terms. This can lead to excessively high values of the effective learning rate. Based on that, we adjust the nominal learning rate for each vector or, equivalently, override the value of the gradient before conducting the optimization step, $\frac{\partial L}{\partial u'_i} \leftarrow \frac{\partial L}{\partial u'_i} \cdot \min\left(1, \frac{\tau}{\omega_i}\right), \quad \frac{\partial L}{\partial v'_i} \leftarrow \frac{\partial L}{\partial v'_i} \cdot \min\left(1, \frac{\tau}{\omega_i}\right)$ for some threshold $\tau \geq 1$. With this modification, the effective learning rate cannot be more than τ times higher than the nominal. In all reported experiments we use $\tau = 10$. Worth noting that equalizing the efficient learning rate with $\tau = 1$ turned out very detrimental for the performance, as we found early in our experiments. This coincides with the intuition behind some of adaptive optimization methods like AdaGrad [7] that perform larger updates for the parameters which are in charge of 'less frequent' features.

Frobenius decay and momentum. As proposed in other works [22, 33, 44] we use Frobenius weight decay (FD) during local training, i.e. we apply an additional loss function proportional to the $\left\|\hat{W}\right\|_F^2$ for each effective weight matrix. The weight of FD in the resulting loss function is set to 1×10^{-4}. Additionally, confirming findings of Niu et al. [33], we found it necessary to use plain SGD with momentum weight of 0.9 during local optimization of the sub-model weights.

4 Experiments

4.1 Experimental Setup

Datasets and models. To evaluate the proposed strategies, we conducted experiments on several datasets which are usually employed to test federated systems. In case of artificial simulation of clients, we follow [19] and sample the prior probabilities of classes for each client from the Dirichlet distribution $\text{Dir}(\alpha p)$ where p is the vector of class proportions in the original dataset and α is a hyperparameter that controls the amount of non-i.i.d.-ness in the data splits. Note that this notation is different from the one used in some of the recent papers, e.g. [44, 32, 33].

Since in our work we do not aim to reach new state-of-the-art in the field and instead focus on analyzing the *relative* performance of the proposed strategies of spectral sharding, we have chosen to test all the methods in the presence of high data non-i.i.d.-ness between clients.

For *CIFAR-10* [23] we split the data with $\alpha = 1$ and conduct experiments with a ResNet-18 model [15] with the kernel size of the first convolutional block equal to 3×3 and the normalization layer replaced with GroupNorm [18]. For *TinyImagenet* [31] we use $\alpha = 10$ and a compact transformer (CCT) model [13], namely CCT-6/3x2. For *CIFAR-100* [23] the two-staged Pachinko allocation method (PAM) [26] is used: for each client, at first parameters of the multinomial distribution over the twenty coarse labels are sampled with $\alpha = 1$, and afterwards the distribution over the coarse-to-fine labels with $\alpha = 10$. On this dataset we train both the ResNet and CCT models described above. We select *Shakespeare* [29] as an example of a dataset with a natural data split over clients. We train a tiny transformer model with three GPT-2 [35] blocks on the task of next character prediction and report the performance in terms of accuracy in accordance with prior works [36].

For all the architectures we do not decompose the very first and very last layers of the model [22, 33]. However, unlike PriSM, we decompose all the affine layers inside the attention blocks, not just the fully connected ones [43]. Also, we follow Niu et al. [33], Wang et al. [43] instead of Khodak et al. [22] and decompose a convolutional layer to a sequence of a regular convolution with n output channels followed by a 1×1 convolution. For image datasets we employ per-image pixel value standardization similarly to [36] but do not apply random cropping.

Training. We train all models from scratch with cosine annealing learning rate scheduler [28]. The initial value for learning rate is 0.1 for CIFAR-10, 0.05 for CIFAR-100 and TinyImagenet, and 0.1 for the Shakespeare data. The client's batch size equals 32, 64, 128, and 10 respectively. All experiments are run with three random seeds which also affect data splitting between clients, if applicable. Standard deviation is reported in all tables and plots based on those runs.

Federated setup. While Shakespeare dataset naturally contains 715 clients, we simulate 100 clients for all other datasets. We randomly sample 10 clients for each communication rounds which results in participation ratio of 10% on image datasets and about 1.4% for Shakespeare. In each communication round all participating clients train their sub-models for two local epochs. The total number of local epochs (e.g., number of local epochs per round × number of communication rounds) equals $2{,}000$ for CIFAR-10, $3{,}000$ for CIFAR-100, $5{,}000$ for TinyImagenet and $3{,}000$ for Shakespeare.

Baselines. As stated above, we focus on exploring different strategies of sampling sub-models when a particular approach of training on weak and, possibly, heterogeneous devices was chosen, namely the approach described in Sec. 3.1. Therefore, we compare our presented strategies, denoted as *Unbiased* and *Collective*, with the *Top-n* sampling proposed by [44] and the *PriSM* method introduced in [32, 33]. We copy the value of the hyperparameter k required for sampling in PriSM from the original implementation. In detail, k depends on the keep ratio r: $k = 4$ if $r \leq 0.2$ and $k = 2.5$ otherwise.

To decouple the sampling strategies themselves from other training details discussed in Sec. 3 we additionally introduce modifications to the PriSM strategy: motivated from our unbiased method, we train it with auxiliary multipliers to allow for (approximate) unbiasedness of the estimators, i.e. $\omega_i = \pi_i^{-1}$ (this strategy is dubbed as *PriSM + Wallenius*), and clip the effective learning rate (*PriSM + Wallenius + ClipLR*). Exact computation of the mean vector for the Wallenius' distribution occurring in these two strategies is very time-consuming for the ranks N of weight matrices used in practice. For that reason we use the approximate algorithm[3] from [9].

Finally, we explore the simplest option of compensating for the missing terms of the estimator, namely, introducing a scaling factor that keeps the Frobenius norm of the estimator equal to that of the full matrix, i.e. $\omega_i^{(c)} = \sqrt{\frac{\sum_{k=1}^N \lambda_k^2}{\sum_{k=1}^N z_k^{(c)} \lambda_k^2}}$ for all i. This modification is marked as +*Scaled*.

All the considered methods require equal communication costs and equal amount of computation on the client side per round, since the overhead caused by the vector of auxiliary weights introduced by novel strategies is negligible.

4.2 Results

4.2.1 Main Results

We report the accuracy of different strategies in all the datasets considered in Tab. 1. In our experiments we mostly explore weak clients with low keep ratio. The results are provided for two setups with homogeneous clients ($r = 0.1$ and 0.2) and one setup with two groups, namely 60% clients having a keep ratio of 0.2 and 40% of 0.4. Also, for reference purposes we train a conventional full model on all the clients without sharding, as well as an over-parameterized model with keep ratio $r = 1$. Similarly to what was reported by Khodak et al. [22], we observe that overparameterization can provide a better performance than usual training.

As the results demonstrate, the preservation of the Frobenius norm (+*Scaled*) is the least effective and stable among the considered modifications. It is often detrimental for the Top-n strategy, and although it sometimes improves the performance of PriSM, the PriSM modification based on unbiasedness (+*Wallenius*+*ClipLR*) is generally more successful. In certain cases, it even achieves the best accuracy among all models, e.g. see the results of ResNet on CIFAR-100. However, overall the 'unbiasedness' has an inconsistent effect on PriSM: it is usually beneficial in case of ResNet but is less helpful or even harmful for attention-based architectures. As has already been noted, deeper analysis of such

[3] `https://cran.r-project.org/package=BiasedUrn`

Table 1: **Accuracy achieved by different strategies under limited computational budget**. Our *Unbiased* and *Collective* strategies outperform the vanilla *Top-n* and *PriSM* baselines on all datasets except for Shakespeare, although the gap is not significant for that dataset. In addition, the modifications proposed for local training allow PriSM to significantly improve its performance for ResNet architecture, and sometimes even surpass other methods in the current setting.

Strategy	CIFAR-10	TinyImagenet	CIFAR-100		Shakespeare
	ResNet	CCT	ResNet	CCT	Transformer
Keep ratio $r = 0.1$					
Top-n	$67.44_{\pm 2.34}$	$34.29_{\pm 0.12}$	$21.13_{\pm 2.01}$	$47.34_{\pm 1.10}$	$\mathbf{48.83}_{\pm 0.45}$
Top-n + Scaled	$63.01_{\pm 0.65}$	$22.29_{\pm 0.76}$	$14.13_{\pm 0.76}$	$33.96_{\pm 0.77}$	$36.14_{\pm 0.77}$
PriSM	$71.15_{\pm 3.77}$	$33.61_{\pm 0.88}$	$18.45_{\pm 1.31}$	$44.60_{\pm 0.90}$	$46.91_{\pm 1.36}$
PriSM + Scaled	$76.03_{\pm 5.00}$	$35.10_{\pm 0.39}$	$28.65_{\pm 1.07}$	$42.14_{\pm 0.88}$	$43.34_{\pm 3.17}$
PriSM + Wallenius	$77.53_{\pm 1.25}$	$37.43_{\pm 0.87}$	$37.75_{\pm 1.04}$	$42.04_{\pm 1.11}$	$45.60_{\pm 0.66}$
PriSM + Wallenius + ClipLR	$82.42_{\pm 0.35}$	$35.43_{\pm 0.91}$	$\mathbf{40.36}_{\pm 0.49}$	$35.50_{\pm 1.24}$	$48.24_{\pm 0.73}$
Unbiased	$80.57_{\pm 1.39}$	$37.57_{\pm 0.17}$	$35.12_{\pm 1.51}$	$45.01_{\pm 1.07}$	$48.23_{\pm 0.35}$
Collective	$\mathbf{82.59}_{\pm 0.27}$	$\mathbf{38.72}_{\pm 0.86}$	$37.83_{\pm 1.24}$	$\mathbf{49.93}_{\pm 1.77}$	$48.64_{\pm 0.50}$
Keep ratio $r = 0.2$					
Top-n	$78.37_{\pm 0.30}$	$38.37_{\pm 0.42}$	$35.97_{\pm 3.19}$	$51.70_{\pm 1.24}$	$\mathbf{51.21}_{\pm 0.50}$
Top-n + Scaled	$80.75_{\pm 1.06}$	$34.99_{\pm 0.33}$	$35.29_{\pm 2.02}$	$47.68_{\pm 0.70}$	$45.82_{\pm 0.38}$
PriSM	$82.59_{\pm 1.09}$	$38.66_{\pm 0.28}$	$33.76_{\pm 0.76}$	$52.04_{\pm 1.62}$	$49.63_{\pm 0.50}$
PriSM + Scaled	$81.63_{\pm 3.52}$	$40.30_{\pm 0.48}$	$40.86_{\pm 1.08}$	$51.64_{\pm 1.30}$	$48.30_{\pm 0.97}$
PriSM + Wallenius	$82.15_{\pm 0.14}$	$40.47_{\pm 0.93}$	$45.11_{\pm 0.63}$	$45.22_{\pm 0.86}$	$45.35_{\pm 0.50}$
PriSM + Wallenius + ClipLR	$84.60_{\pm 1.24}$	$38.13_{\pm 0.62}$	$46.73_{\pm 1.30}$	$38.54_{\pm 1.48}$	$49.44_{\pm 0.39}$
Unbiased	$84.33_{\pm 1.14}$	$\mathbf{41.28}_{\pm 0.31}$	$46.34_{\pm 0.02}$	$53.39_{\pm 0.81}$	$50.12_{\pm 0.89}$
Collective	$\mathbf{85.00}_{\pm 0.34}$	$41.12_{\pm 0.11}$	$\mathbf{47.74}_{\pm 1.15}$	$\mathbf{54.31}_{\pm 0.27}$	$50.05_{\pm 0.44}$
60% clients have keep ratio $r = 0.2$ and 40% clients have $r = 0.4$					
Top-n	$84.43_{\pm 0.90}$	$36.89_{\pm 0.14}$	$41.34_{\pm 1.36}$	$53.69_{\pm 0.71}$	$\mathbf{51.35}_{\pm 0.89}$
Top-n + Scaled	$85.24_{\pm 0.19}$	$37.12_{\pm 0.30}$	$42.24_{\pm 0.80}$	$47.99_{\pm 1.26}$	$50.11_{\pm 0.70}$
PriSM	$82.37_{\pm 0.62}$	$31.25_{\pm 0.21}$	$23.04_{\pm 2.01}$	$37.47_{\pm 0.26}$	$49.75_{\pm 0.03}$
PriSM + Scaled	$82.75_{\pm 0.14}$	$34.86_{\pm 0.41}$	$28.41_{\pm 2.12}$	$42.45_{\pm 0.90}$	$49.69_{\pm 0.22}$
PriSM + Wallenius	$78.71_{\pm 1.18}$	$34.83_{\pm 1.09}$	$34.03_{\pm 2.02}$	$34.23_{\pm 1.00}$	$38.21_{\pm 2.02}$
PriSM + Wallenius + ClipLR	$82.95_{\pm 0.84}$	$39.31_{\pm 0.17}$	$32.88_{\pm 0.66}$	$40.07_{\pm 0.39}$	$49.45_{\pm 0.85}$
Unbiased	$85.09_{\pm 0.52}$	$\mathbf{40.78}_{\pm 0.18}$	$\mathbf{44.61}_{\pm 2.43}$	$\mathbf{54.16}_{\pm 0.59}$	$51.05_{\pm 0.63}$
Collective	$\mathbf{86.11}_{\pm 0.66}$	$39.90_{\pm 0.43}$	$43.74_{\pm 1.87}$	$52.52_{\pm 0.93}$	$50.74_{\pm 1.15}$
Keep ratio $r = 1$	$89.54_{\pm 0.49}$	$41.28_{\pm 0.65}$	$63.45_{\pm 0.34}$	$59.25_{\pm 1.05}$	$52.10_{\pm 1.22}$
No sharding	$87.46_{\pm 0.31}$	$38.68_{\pm 0.16}$	$63.35_{\pm 0.68}$	$60.05_{\pm 0.79}$	$52.89_{\pm 0.10}$

Table 2: **Longer training on CIFAR-100.** Despite our ‘unbiased’ modification of PriSM demonstrated the best accuracy in case of limited computation budget, the more explorative Collective strategy closes the gap in performance if the number of communication rounds is increased.

Strategy	# communication rounds	
	1,500	5,000
Top-n	$21.13_{\pm 2.01}$	$32.22_{\pm 1.74}$
Top-n + Scaled	$14.13_{\pm 0.76}$	$39.68_{\pm 0.32}$
PriSM	$18.45_{\pm 1.31}$	$51.85_{\pm 1.25}$
PriSM + Scaled	$28.65_{\pm 1.07}$	$57.08_{\pm 0.45}$
PriSM + Wallenius	$37.75_{\pm 1.04}$	$59.81_{\pm 0.37}$
PriSM + Wallenius + ClipLR	$\mathbf{40.36}_{\pm 0.49}$	$60.02_{\pm 0.50}$
Unbiased	$35.12_{\pm 1.51}$	$59.44_{\pm 0.08}$
Collective	$37.83_{\pm 1.24}$	$\mathbf{60.26}_{\pm 0.31}$

Table 3: **Influence of the clipped learning rate.** In our experiments, we find that clipping the effective learning rate is beneficial for the Unbiased strategy in case of all architectures and values of the keep ratio r. Without clipping the performance drops consistently.

Strategy	ResNet	CCT
Keep ratio $r = 0.1$		
Unbiased	$35.12_{\pm 1.51}$	$45.01_{\pm 1.07}$
Unbiased w/o ClipLR	$30.14_{\pm 1.58}$	$32.46_{\pm 0.71}$
Keep ratio $r = 0.2$		
Unbiased	$46.34_{\pm 0.02}$	$53.39_{\pm 0.81}$
Unbiased w/o ClipLR	$43.11_{\pm 1.02}$	$46.67_{\pm 0.85}$

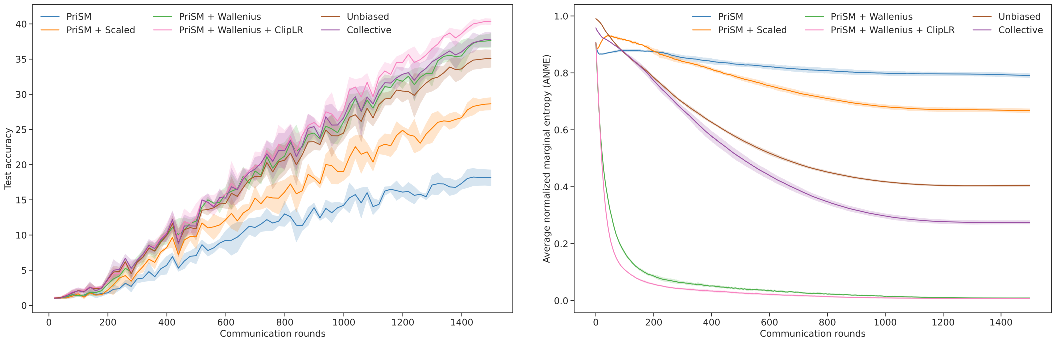
behavior is complicated due to unclear statistical properties of the PriSM sampling procedure and we leave this fo future work. One possible reason is that the approximation given by Wallenius' distribution may be less precise in some cases.

Our strategies presented in Sec. 3 in most cases outperform the baselines for all datasets but Shakespeare where Top-n demonstrates superior results, although the difference in performance is not significant. As the results imply, the Collective strategy is more suitable if the keep ratio is small, while for larger r the Unbiased strategy seems more preferable. This may be explained by the negative influence of too large auxiliary multipliers which also amplify the impact of less informative terms during the forward pass; see Sec. 3.3.

Interestingly, in all experiments, the novel sampling strategies significantly improve the results of baselines if those had a large accuracy gap compared to the full model. For the Shakespeare dataset, in contrast, all the low-rank methods achieve performance close to the full model even for small values of the keep ratio. Therefore, it seems that the performance on Shakespeare is determined more by the properties of the data split and the corresponding federated setup than by the training strategy.

4.2.2 Discussion

Explorative and exploitative strategies. To understand the behavior of the strategies better, we propose a tool named *average normalized marginal entropy* (ANME). For each decomposed layer with



(a) Accuracy of the global ResNet model on the test subset of CIFAR-100.

(b) Average normalized marginal entropy (ANME) shows how diverse each strategy is.

Figure 1: **Communication efficiency.** The original PriSM method is too explorative (high ANME), while our 'unbiased' modification (+*Wallenius*) makes it the most exploitative strategy and allows to achieve the best performance in some experiments with limited computational budget.

inclusion probabilities $(\pi_1, \ldots, \pi_N)$ we calculate the mean entropy of the corresponding marginal Bernoulli distributions, $\frac{1}{N}\sum_{i:\, 0<\pi_i<1}\left(-\pi_i \log \pi_i - (1-\pi_i)\log(1-\pi_i)\right)$. It is easy to show that the minimum value of this quantity equals 0 and is achieved by the deterministic Top-n strategy. The maximum value is achieved when all probabilities are equal, i.e. $\pi_i = \frac{n}{N}$. Therefore, we normalize the mean entropy from above by dividing it by the mean entropy of the uniform inclusion probabilities. To finish the computation of ANME, we average this normalized entropy across all sharded layers in the network.

Intuitively, low values of ANME mean that the server tends to send the same terms to all the clients, and we name such strategies *exploitative*. On the contrary, high value of ANME shows that the strategy is *explorative* and the selection of terms for each client is 'more random'. We observe the same qualitative behaviour in all our experiments (see Fig. 1b): PriSM is the most explorative strategy, while the modification based on unbiasedness turns it into the most exploitative among randomized methods. The Unbiased and Collective strategies are between these two extreme types of behaviour, and the latter is usually more exploitative than the former one. Based on this observation, we check if training with more communication rounds can help the explorative strategies to perform better and report the results in Tab. 2. Indeed, all methods improve with more training and the largest improvement is for PriSM, which is the most explorative. We also observe that our collective strategy overtakes our modified PriSM baseline and achieves the best performance.

Influence of learning rate clipping. In all the experiments we found that clipping the effective learning rate is necessary for our strategies. Tab. 3 illustrates this observation by using models trained on CIFAR-100. Notably, Tab. 1 demonstrates that, empirically, such clipping is not always beneficial for the modified PriSM method. This may be explained by the fact that 'unbiased' PriSM becomes extremely exploitative. In this case, terms which could produce too large auxiliary multipliers are very unlikely to be sampled in practice, resulting in the large variance of the left-hand side of Eq. (9).

More results. Please refer to the Appendix A for additional results on model convergence, post-client updates and influence of data heterogeneity.

5 Conclusion

We presented two novel sampling strategies for spectral model sharding, grounded as solutions to specific optimization problems. Alongside them, we presented techniques that facilitate local training with such methods. As shown, in a number of cases these techniques can also significantly improve the performance of the strategies proposed earlier in the literature. Nonetheless, our strategies demonstrate superior performance on a number of commonly used datasets in the presence of high data heterogeneity between clients. As a downside, in certain cases their learning curve is less steep than that of the baselines due to their more explorative nature. We leave the improvement of the convergence speed for future work.

References

[1] S. Alam, L. Liu, M. Yan, and M. Zhang. Fedrolex: Model-heterogeneous federated learning with rolling sub-model extraction. In *NeurIPS*, 2022. 1, 2

[2] S. Babakniya, A. Elkordy, Y. Ezzeldin, Q. Liu, K.-B. Song, M. EL-Khamy, and S. Avestimehr. SLoRA: Federated parameter efficient fine-tuning of language models. In *NeurIPS Workshop*, 2023. 2

[3] K. E. W. Brewer. A Simple Procedure for Sampling πpswor^1. *Australian Journal of Statistics*, 17(3):166–172, 1975. 14

[4] S. Caldas, J. Konečný, B. McMahan, and A. Talwalkar. Expanding the reach of federated learning by reducing client resource requirements, 2019. 2

[5] J. Chesson. A non-central multivariate hypergeometric distribution arising from biased sampling with application to selective predation. *Journal of Applied Probability*, 13(4):795–797, 1976. 3, 20

[6] E. Diao, J. Ding, and V. Tarokh. HeteroFL: Computation and communication efficient federated learning for heterogeneous clients. In *ICLR*, 2021. 1, 2

[7] J. Duchi, E. Hazan, and Y. Singer. Adaptive Subgradient Methods for Online Learning and Stochastic Optimization. *JMLR*, 12(61):2121–2159, 2011. 6

[8] C. Eckart and G. Young. The approximation of one matrix by another of lower rank. *Psychometrika*, 1(3):211–218, 1936. 1, 5

[9] A. Fog. Calculation Methods for Wallenius' Noncentral Hypergeometric Distribution. *Communications in Statistics - Simulation and Computation*, 37(2):258–273, 2008. 3, 7, 20

[10] A. Fog. Sampling Methods for Wallenius' and Fisher's Noncentral Hypergeometric Distributions. *Communications in Statistics - Simulation and Computation*, 37(2):241–257, 2008. 3, 20

[11] D. Gao, X. Yao, and Q. Yang. A survey on heterogeneous federated learning, 2022. 2

[12] C. R. Harris, K. J. Millman, S. J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith, R. Kern, M. Picus, S. Hoyer, M. H. van Kerkwijk, M. Brett, A. Haldane, J. F. del Río, M. Wiebe, P. Peterson, P. Gérard-Marchant, K. Sheppard, T. Reddy, W. Weckesser, H. Abbasi, C. Gohlke, and T. E. Oliphant. Array programming with NumPy. *Nature*, 585(7825):357–362, 2020. 3, 20

[13] A. Hassani, S. Walton, N. Shah, A. Abuduweili, J. Li, and H. Shi. Escaping the big data paradigm with compact transformers, 2022. 6

[14] C. He, M. Annavaram, and S. Avestimehr. Group knowledge transfer: federated learning of large CNNs at the edge. In *NeurIPS*, 2020. 2

[15] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 6

[16] D. G. Horvitz and D. J. Thompson. A generalization of sampling without replacement from a finite universe. *Journal of the American Statistical Association*, 47(260):663–685, 1952. 4

[17] S. Horváth, S. Laskaridis, M. Almeida, I. Leontiadis, S. Venieris, and N. D. Lane. FjORD: Fair and accurate federated learning under heterogeneous targets with ordered dropout. In *NeurIPS*, 2021. 1, 2

[18] K. Hsieh, A. Phanishayee, O. Mutlu, and P. Gibbons. The non-IID data quagmire of decentralized machine learning. In *ICML*, 2020. 6

[19] H. Hsu, H. Qi, and M. Brown. Measuring the effects of non-identical data distribution for federated visual classification. In *NeurIPS Workshop*, 2019. 2, 6

https://doi.org/10.52202/079017-1968

[20] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen. LoRA: Low-rank adaptation of large language models. In *ICLR*, 2022. 2

[21] S. R. Kamalakara, A. Locatelli, B. Venkitesh, J. Ba, Y. Gal, and A. N. Gomez. Exploring Low Rank Training of Deep Neural Networks, 2022. 2

[22] M. Khodak, N. A. Tenenholtz, L. Mackey, and N. Fusi. Initialization and regularization of factorized neural layers. In *ICLR*, 2021. 2, 5, 6, 7

[23] A. Krizhevsky. Learning multiple layers of features from tiny images, 2009. 6, 21

[24] F.-F. Li, J. Johnson, and S. Yeung. CS231n: Convolutional neural networks for visual recognition, 2017. URL `https://cs231n.github.io/neural-networks-2/`. 2, 4

[25] Q. Li, Z. Wen, Z. Wu, S. Hu, N. Wang, Y. Li, X. Liu, and B. He. A survey on federated learning systems: Vision, hype and reality for data privacy and protection. *IEEE Transactions on Knowledge & Data Engineering*, 35(04):3347–3366, 2023. 1

[26] W. Li and A. McCallum. Pachinko allocation: DAG-structured mixture models of topic correlations. In *ICML*, 2006. 6

[27] T. Lin, L. Kong, S. U. Stich, and M. Jaggi. Ensemble distillation for robust model fusion in federated learning. In *NeurIPS*, 2020. 2

[28] I. Loshchilov and F. Hutter. SGDR: Stochastic gradient descent with warm restarts. In *ICLR*, 2017. 7

[29] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y. Arcas. Communication-Efficient Learning of Deep Networks from Decentralized Data. In *AISTATS*, 2017. 6, 21

[30] Y. Mei, P. Guo, M. Zhou, and V. Patel. Resource-adaptive federated learning with all-in-one neural composition. In *NeurIPS*, 2022. 2, 3, 5

[31] M. A. Mnmoustafa. Tiny ImageNet, 2017. URL `https://kaggle.com/competitions/tiny-imagenet`. 6, 21

[32] Y. Niu, S. Prakash, S. Kundu, S. Lee, and S. Avestimehr. Federated learning of large models at the edge via principal sub-model training. In *NeurIPS Workshop*, 2022. 1, 2, 3, 6, 7

[33] Y. Niu, S. Prakash, S. Kundu, S. Lee, and S. Avestimehr. Overcoming Resource Constraints in Federated Learning: Large Models Can Be Trained with only Weak Clients. *TMLR*, 2023. 1, 2, 3, 5, 6, 7

[34] K. Pfeiffer, M. Rapp, R. Khalili, and J. Henkel. Federated learning for computationally constrained heterogeneous devices: A survey. *ACM Comput. Surv.*, 55(14s), 2023. 1, 2

[35] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever, et al. Language models are unsupervised multitask learners, 2019. 6

[36] S. J. Reddi, Z. Charles, M. Zaheer, Z. Garrett, K. Rush, J. Konečný, S. Kumar, and H. B. McMahan. Adaptive federated optimization. In *ICLR*, 2021. 6, 7

[37] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *JMLR*, 15(56):1929–1958, 2014. 2, 4

[38] Y. Sui, M. Yin, W. Yang, Y. Gong, J. Xiao, H. Phan, D. Ding, X. Xu, S. Liu, Z. Chen, and B. Yuan. ELRT: Towards efficient low-rank training for compact neural networks, 2023. 2

[39] Y. Tillé. *Sampling algorithms*. Springer, 2006. ISBN 978-0-387-30814-2. 2, 5, 14, 17

[40] Y. Tillé. Remarks on some misconceptions about unequal probability sampling without replacement. *Computer Science Review*, 47:100533, 2023. 3

[41] N. Timor, G. Vardi, and O. Shamir. Implicit Regularization Towards Rank Minimization in ReLU Networks. In *ALT*, 2023. 2

https://doi.org/10.52202/079017-1968

[42] K. T. Wallenius. *Biased Sampling: the Noncentral Hypergeometric Probability Distribution*. Stanford University, 1963. 3, 20

[43] H. Wang, S. Agarwal, P. U-chupala, Y. Tanaka, E. Xing, and D. Papailiopoulos. Cuttlefish: Low-Rank Model Training without All the Tuning. In *MLSys*, 2023. 2, 6

[44] D. Yao, W. Pan, M. J. O'Neill, Y. Dai, Y. Wan, H. Jin, and L. Sun. FedHM: Efficient federated learning for heterogeneous models via low-rank factorization, 2022. Version: 2. 1, 2, 3, 6, 7

[45] M. Ye, X. Fang, B. Du, P. C. Yuen, and D. Tao. Heterogeneous federated learning: State-of-the-art and research challenges. *ACM Comput. Surv.*, 56(3), 2023. 2

[46] Z. Zhang, Y. Yang, Y. Dai, Q. Wang, Y. Yu, L. Qu, and Z. Xu. FedPETuning: When federated learning meets the parameter-efficient tuning methods of pre-trained language models. In *ACL*, 2023. 2

https://doi.org/10.52202/079017-1968

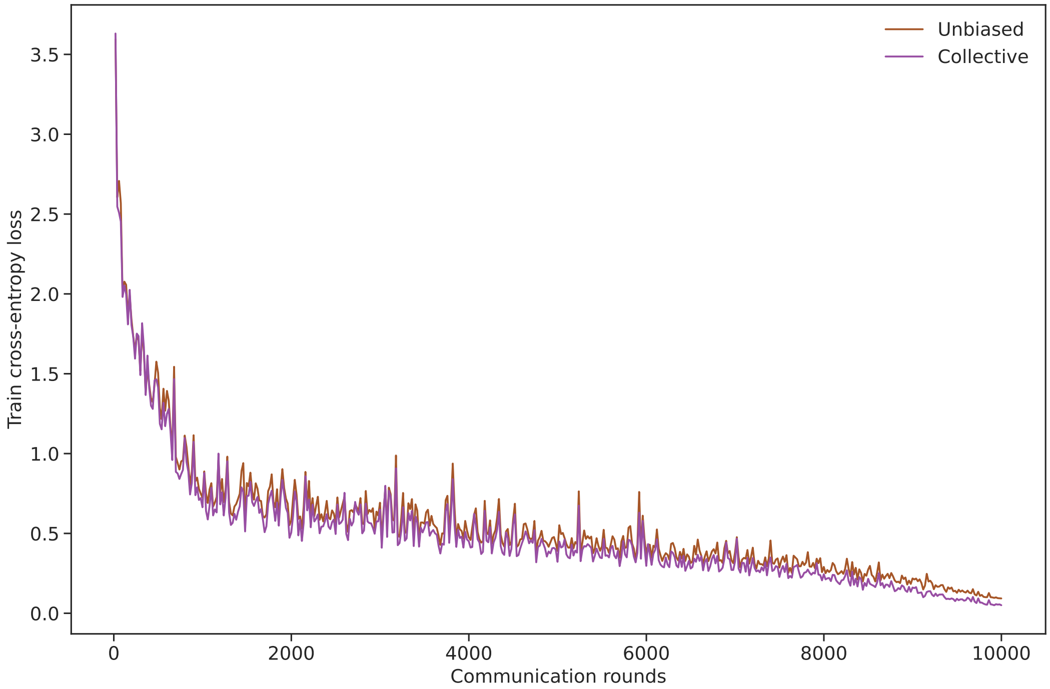
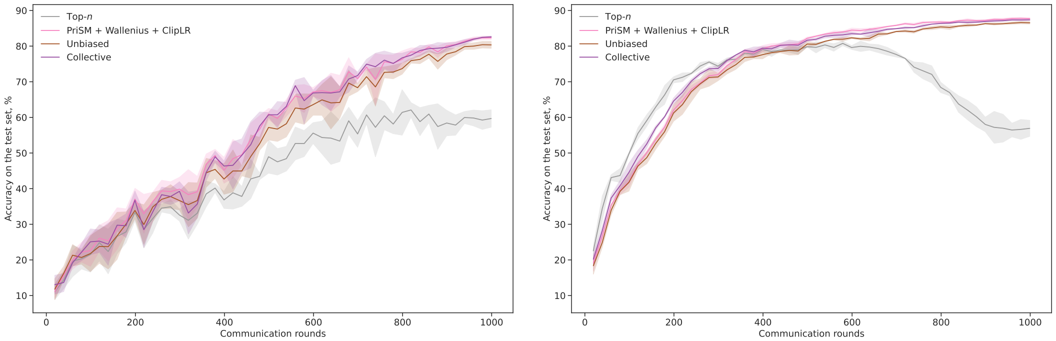


Figure 2: **Convergence analysis.** When being trained longer, the proposed strategies demonstrate the decrease of the cross-entropy loss of the global model on the training set. Unbiased strategy reaches the train accuracy of 97.0%, and Collective strategy achieves 98.4%. This serves as empirical evidence of convergence for our method.



(a) High data heterogeneity $\alpha = 1$.

(b) Low data heterogeneity $\alpha = 10$.

Figure 3: **Impact of data heterogeneity on communication efficiency.** When data distribution between clients is closer to i.i.d, the most exploitative Top-n strategy demonstrates the best training speed in the beginning, however it overfits soon. For more heterogeneous data, strategies with exploration achieve much better performance.

A Additional Discussion

Convergence analysis. It is rather involved to provide the theoretical proof of convergence for any of spectral sharding methods. In practice, we observe that the models trained in this manner successfully converge even in the presence of high data heterogeneity between clients. As an example, Fig. 2 demonstrates the cross-entropy loss on the train set for the ResNet model with keep ratio $r = 0.1$ trained on CIFAR-10 with non-i.i.d.-ness parameter $\alpha = 1$. The loss was computed at the end of each communication round, the number of rounds was intentionally increased up to 10,000 for this experiment. It is easy to see that the loss successfully decreases as the training progresses. Therefore, given these empirical observations, we did not focus on the formal convergence analysis. Having said that, it is an interesting direction for future work, as it can potentially uncover further improvements to our approach.

Impact of data heterogeneity. Throughout our paper, we evaluate all the strategies in the presence of a relatively high degree of data heterogeneity between clients. E.g., note that $\alpha = 1$ for CIFAR-10 roughly corresponds to only one or two distinct class labels present for each client, and $\alpha = 10$ for TinyImagenet means that each client only has access to 20-30% of all class labels.

We demonstrate the effect of data heterogeneity on the communication efficiency of the best performing strategies on CIFAR-10 in Fig. 3. When clients' data distributions are closer to i.i.d., i.e. $\alpha = 10$, Top-n strategy is the most efficient in the beginning, however it overfits soon. For more heterogeneous data, i.e. $\alpha = 1$, the proposed modification of PriSM and the novel Collective strategy are the most efficient and deliver the best accuracy.

Note that the novel strategies developed in our work do not use any knowledge of data distribution between clients. Neither they rely on the presence of a public dataset on the server side. Nevertheless, we believe that incorporating such information can be beneficial for the Collective strategy and leave this for future work.

Influence of the joint distribution. To test the sensitivity of the results to the choice of the joint distribution of the sampled indices while keeping the marginal distributions frozen, we compare the selected CPS sampling against two other methods, Brewer's [3] and Minimum support [39]. The advantage of those methods is that they allow for significantly faster sampling than CPS. Perhaps surprisingly, we do not notice any significant difference in our experiments, as reported in Tab. 4. We suggest that the effect of high entropy which is characteristic for CPS may be shown on more large-scale problems and leave this for future work.

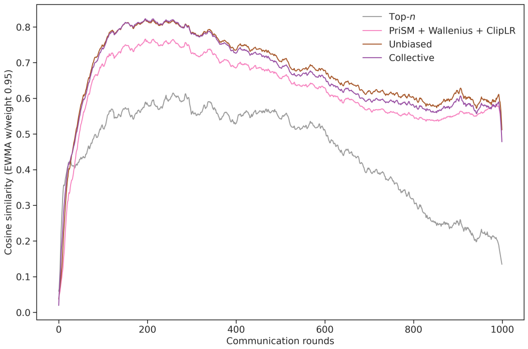


Figure 4: **Comparison with FedAvg weight updates.** For ResNet model trained on CIFAR-10, updates provided by the Top-n strategy significantly deviate from those of FedAvg method. This correlates with the worse performance in this experiment.

Post-client updates. To analyze the post-gradient updates, we followed the optimization path $\{\theta_t\}_{t=1}^{T}$ of the server model under each strategy with T communication rounds and calculated the cosine similarity between the server model update $\Delta_{sharding\_strategy}\theta_t$ under that strategy and $\Delta_{FedAvg}\theta_t$ under FedAvg method with full-size model on each client. Subsequently, the model weights were updated according to the selected strategy: $\theta_{t+1} = \theta_t + \Delta_{sharding\_strategy}\theta_t$. The results of these experiments are provided in Fig. 4. As shown, there is a notable connection between the relative performance of each strategy and its "alignment" with FedAvg updates. The deterministic Top-n strategy lags behind its randomized counterparts in terms of accuracy in the considered setting (see Tab. 1), and a similar observation is made for its update directions.

Schatten norm. Throughout this paper we used Frobenius norm to upper-bound the operator norm, see Eq. (2). While this choice allowed us to derive a closed-form solution which enables fast calculation on the server side, this upper bound is loose. Better approximation can be achieved by using Schatten p-norm with $p > 2$. Schatten norm is defined as $\|W\|_p = (\lambda_1^p + \dots + \lambda_N^p)^{\frac{1}{p}}$, where $\lambda_1, \dots, \lambda_N$ are singular values of the matrix W. Frobenius norm is an example of a Schatten p-norm with $p = 2$.

https://doi.org/10.52202/079017-1968

Table 4: **Influence of the joint distribution.** We have not found the evidence of the advantage of maximum entropy distribution (CPS) over other methods for our tasks. On CIFAR-100, all three sampling methods perform similarly provided that marginal inclusion probabilities are preserved.

Model	Keep ratio	Strategy	CPS	Brewer	MinSupport
ResNet	0.1	Unbiased	$35.12_{\pm 1.51}$	$35.67_{\pm 1.73}$	$35.19_{\pm 1.72}$
		Collective	$37.83_{\pm 1.24}$	$37.55_{\pm 1.92}$	$37.89_{\pm 1.91}$
	0.2	Unbiased	$46.34_{\pm 0.12}$	$46.94_{\pm 1.12}$	$45.71_{\pm 1.80}$
		Collective	$47.74_{\pm 1.15}$	$43.55_{\pm 1.06}$	$46.13_{\pm 1.23}$
CCT	0.1	Unbiased	$45.01_{\pm 1.07}$	$45.82_{\pm 0.70}$	$45.34_{\pm 1.05}$
		Collective	$49.93_{\pm 1.77}$	$49.51_{\pm 1.63}$	$50.65_{\pm 1.62}$
	0.2	Unbiased	$53.39_{\pm 0.81}$	$53.29_{\pm 1.46}$	$52.94_{\pm 1.01}$
		Collective	$54.31_{\pm 0.27}$	$53.74_{\pm 1.06}$	$52.98_{\pm 0.52}$

Table 5: **Unbiased estimator based on Schatten norm.** While Schatten norm approximates the largest singular value better than the Frobenius norm, it does not allow the closed-form derivation of sampling strategies. In our experiments with numerical constrained optimization, we have not seen any significant improvement over the methods that are based on the Frobenius norm.

Strategy	CIFAR-10	CIFAR-100	
	ResNet	ResNet	CCT
Unbiased	80.57 ± 1.39	35.12 ± 1.51	45.01 ± 1.07
Collective	82.59 ± 0.27	37.83 ± 1.24	49.93 ± 1.77
Schatten ($p = 4$)	80.72 ± 0.67	34.54 ± 0.62	45.75 ± 0.44

For the *unbiased* matrix estimators of the kind specified in Eq. (1), the average Schatten norm can be upper-bounded with the help of Jensen's inequality, since $(\cdot)^{\frac{1}{p}}$ is a concave function for $p > 1$

$$\begin{aligned}
\mathbb{E}_{p(z)} \left\| W - \hat{W} \right\|_p &= \mathbb{E}_{p(z)} \left(\sum_{i=1}^{N} \lambda_i^p \left| \frac{z_i}{\pi_i} - 1 \right|^p \right)^{\frac{1}{p}} \\
&\leq \left(\mathbb{E}_{p(z)} \sum_{i=1}^{N} \lambda_i^p \left| \frac{z_i}{\pi_i} - 1 \right|^p \right)^{\frac{1}{p}} \\
&= \left(\sum_{i=1}^{N} \lambda_i^p \left[1 - \pi_i + \pi_i \left(\pi_i^{-1} - 1 \right)^p \right] \right)^{\frac{1}{p}} .
\end{aligned} \tag{10}$$

Unfortunately, we were not able to derive a closed-form solution for large values of p in the same way as we proved Theorem 3.1. Therefore, we opted for numerical constrained optimization on the server side to find the optimal values of inclusion probabilities. As demonstrated in Tab. 5, for $p = 4$ we did not find any significant improvement over our strategies presented in Sec. 3. For larger p, the optimization procedure turned out unstable, and the results were unsatisfactory.

Computational resources. All our experiments were conducted with a single GPU and required not more than 10 Gb VRAM. We used a simulation of the federated learning environment with all participating clients being trained in a sequence on the same GPU.

B Optimal Sampling Distribution

In the main text we discussed estimators obtained from singular vector decomposition. However, we found that the obtained results are more general and apply to any orthogonal decomposition. Therefore, in this section we provide proofs in a slightly generalized form.

More specifically, we consider a vector space equipped with an inner product and the corresponding induced norm. Assume that some element of that space Y may be represented as a sum of mutually orthogonal terms $Y = \sum_{i=1}^{N} X_i$, and the terms are enumerated in descending order w.r.t. their norms, i.e. $\lambda_i = \|X_i\|$, and $\lambda_1 \geq \cdots \geq \lambda_N > 0$. In the particular case of SVD, Y is equal to the matrix W, $X_i = \lambda_i u_i v_i^T$, and the Frobenius inner product is considered.

B.1 Unbiased Estimator

Let $\hat{Y}$ be an estimator of the element Y of the following type,

$$\hat{Y} = \hat{Y}(\mathcal{I}) = \sum_{j=1}^{n} w_j(\mathcal{I}) X_{\mathcal{I}_j} = \sum_{i=1}^{N} z_i w_i(\mathcal{I}) X_i, \tag{11}$$

where $\mathcal{I} \subset \{1, \dots, N\}$ and the size of $\mathcal{I}$ equals n, z_i is an indicator variable $z_i = z_i(\mathcal{I}) = \mathbb{I}(i \in \mathcal{I})$, and $w_i \in \mathbb{R}$ are scalar values, potentially depending on the whole subset $\mathcal{I}$. Assuming there exists a distribution $p(\mathcal{I})$ over all subsets of size n, the expected value of the estimator can be calculated

$$\mathbb{E}_{p(\mathcal{I})}\hat{Y}(\mathcal{I}) = \sum_{\mathcal{I}} p(\mathcal{I})\hat{Y}(\mathcal{I}) = \sum_{i=1}^{N} X_i \sum_{\mathcal{I}} p(\mathcal{I}) z_i w_i(\mathcal{I}) = \sum_{i=1}^{N} X_i \mathbb{E}_{p(\mathcal{I})}\left[z_i w_i(\mathcal{I})\right]. \tag{12}$$

B.1.1 Horvitz-Thompson Estimator

The bias of the estimator above equals $\left\|\mathbb{E}_{p(\mathcal{I})}\hat{Y}(\mathcal{I}) - Y\right\| = \left\|\sum_{i=1}^{N} X_i \mathbb{E}_{p(\mathcal{I})}\left[z_i w_i(\mathcal{I}) - 1\right]\right\|$, and if (i) all the $\{X_i\}_i$ are mutually orthogonal and (ii) the estimator $\hat{Y}$ is unbiased, the following constraint is obtained

$$\mathbb{E}_{p(\mathcal{I})}\left[z_i w_i(\mathcal{I})\right] = 1 \quad \text{for } i = 1, \dots, N. \tag{13}$$

Taking into account that z_i is a binary random variable, we can use that $z_i^2 = z_i$ and derive the following

$$1 = \mathbb{E}\left[z_i w_i(\mathcal{I})\right] = \mathbb{E}\mathbb{E}\left[z_i w_i | z_i\right] = \mathbb{E}\left[z_i \mathbb{E}\left[w_i | z_i\right]\right] \tag{14}$$

$$= \Pr(z_i = 1)\mathbb{E}\left[w_i | z_i = 1\right] = \pi_i \mathbb{E}\left[w_i | z_i = 1\right], \tag{15}$$

leading to $\mathbb{E}\left[w_i | z_i = 1\right] = \pi_i^{-1}$.

Due to the mutual orthogonality of the $\{X_i\}_i$

$$\mathbb{E}_{p(\mathcal{I})}\left\|Y - \hat{Y}\right\|^2 = \sum_{i=1}^{N} \lambda_i^2 \mathbb{E}_{p(\mathcal{I})}\left[z_i w_i - 1\right]^2. \tag{16}$$

Now it is possible to state the following optimization problem to find the unbiased estimator with the lowest mean squared error

$$\begin{aligned} \min_{p(\mathcal{I}), \{w_i\}} \quad & \sum_{i=1}^{N} \lambda_i^2 \mathbb{E}_{p(\mathcal{I})}\left[z_i w_i - 1\right]^2 \\ \text{s.t.} \quad & \mathbb{E}_{p(\mathcal{I})}\left[z_i w_i\right] = 1, \quad i = 1, \dots, N, \\ & \sum_{i=1}^{N} \pi_i = n, \quad 0 \le \pi_i \le 1. \end{aligned} \tag{17}$$

The last constraint takes place since the size of the subset $\mathcal{I}$ is set equal to n which implies $\sum_i z_i = n$, and due to the linear property of expectation $\mathbb{E}\sum_i z_i = \sum_i \mathbb{E}z_i = \sum_i \pi_i$. Furthermore, the quadratic term under the expectation can be simplified thanks to the unbiasedness and lower-bounded with the help of Jensen's inequality

$$\begin{aligned} \mathbb{E}\left[z_i w_i - 1\right]^2 &= \mathbb{E}\left[z_i w_i^2\right] - 2\mathbb{E}\left[z_i w_i\right] + 1 \\ &= \mathbb{E}\left[z_i w_i^2\right] - 1 \\ &= \pi_i \mathbb{E}\left[w_i^2 | z_i = 1\right] - 1 \\ &\ge \pi_i \left(\mathbb{E}\left[w_i | z_i = 1\right]\right)^2 - 1 \\ &= \pi_i \cdot \left(\pi_i^{-1}\right)^2 - 1 \\ &= \pi_i^{-1} - 1, \end{aligned} \tag{18}$$

and this lower bound can be achieved by setting w_i to a non-random value $w_i = \pi_i^{-1}$.

B.1.2 Inclusion Probabilities

With the optimal values of auxiliary multipliers known, the optimization problem can be rewritten as

$$\begin{aligned} \min_{\{\pi_i\}} \quad & \sum_{i=1}^{N} \lambda_i^2 \left(\pi_i^{-1} - 1\right), \\ \text{s.t.} \quad & 0 \le \pi_i \le 1 \quad i = 1, \dots, N, \\ & \sum_{i=1}^{N} \pi_i = n. \end{aligned} \tag{19}$$

This is a typical contrained optimization problem which can be solved with the help of Lagrangian function $L(\{\pi_i\}, \alpha, \beta, \{\gamma_i\}, \{\delta_i\}) = \alpha \sum_i \lambda_i^2 \pi_i^{-1} + \beta \left(\sum_i \pi_i - n\right) + \sum_i \gamma_i \left(-\pi_i\right) + \sum_i \delta_i \left(\pi_i - 1\right)$, leading to the following equations for $i = 1, \dots, N$

$$\begin{aligned}
&\frac{\partial L}{\partial \pi_i} = -\frac{\alpha \lambda_i^2}{\pi_i^2} + \beta - \gamma_i + \delta_i = 0,\\
&\sum_i \pi_i = n,\\
&\gamma_i \pi_i = 0,\\
&\delta_i \left(\pi_i - 1\right) = 0,\\
&\alpha,\, \gamma_i,\, \delta_i \geq 0,\\
&\alpha^2 + \beta^2 + \sum_i \gamma_i^2 + \sum_i \delta_i^2 > 0.
\end{aligned} \tag{20}$$

Case $\alpha = 0$**.** In this case $\forall i\ \gamma_i - \delta_i = \beta = \text{const}$.

If $\exists k\ \delta_k > 0 \Rightarrow \pi_k = 1 \Rightarrow \gamma_k = 0 \Rightarrow \beta = -\delta_k < 0 \Rightarrow \forall i\ \beta = \gamma_i - \delta_i < 0 \Rightarrow \forall i\ \gamma_i < \delta_i$. But γ_i and δ_i cannot be both greater than zero simultaneously since this would imply that $\pi_i = 0$ and $\pi_i = 1$ at the same time. Therefore, the only way for the strict inequality to hold true is to set $\forall i\ \gamma_i = 0,\ \delta_i > 0 \Rightarrow \forall i\ \pi_i = 1$. The last equality leads to contradiction in case $n < N$, which is our main case of interest.

Therefore, $\forall i\ \delta_i = 0 \Rightarrow 0 \leq \gamma_i = \beta = \text{const}$. If $\beta = 0$, this breaks the constraint of existence of at least one non-zero coefficient. Therefore, $\forall i\ \gamma_i = \beta > 0 \Rightarrow \forall i\ \pi_i = 0$, and this leads to contradiction given that $n > 0$.

Case $\alpha = 1$**.** W.l.o.g. we can assume than $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_N > 0$. From our equations, $\forall i\ \pi_i^2 = \dfrac{\lambda_i^2}{\beta + \delta_i - \gamma_i}$.

If $\exists l\ \gamma_l > 0 \Rightarrow \pi_l = 0 \Rightarrow \lambda_l = 0$, and this contradicts the assumptions above. Therefore, $\forall i\ \gamma_i = 0$, and $\forall i\ \pi_i^2 = \dfrac{\lambda_i^2}{\beta + \delta_i}$.

Now let us assume that $\exists k\ \delta_k > 0 \Rightarrow \pi_k = 1 \Rightarrow \beta = \lambda_k^2 - \delta_k \Rightarrow \forall j \neq k\ 1 \geq \pi_j^2 = \dfrac{\lambda_j^2}{\lambda_k^2 - \delta_k + \delta_j} \Rightarrow$ $\forall j \neq k\ \lambda_j^2 - \lambda_k^2 \leq \delta_j - \delta_k$. Therefore, if $j < k$, then $\lambda_j > \lambda_k > 0$, and consequently $\delta_j > \delta_k > 0$ which leads to $\pi_j = 1$. Summarizing, if $\pi_k = 1$, then $\pi_1 = \dots = \pi_k = 1$.

Based on the previous observation, we define

$$t = \min_i \{1 \leq i \leq N :\ \pi_i < 1\} - 1, \tag{21}$$

which means that $\pi_1 = \dots \pi_t = 1$, and $\pi_{t+1}, \dots, \pi_N < 1 \Rightarrow \delta_l = 0$ for $l = t+1, \dots, N$, and $\pi_l^2 = \frac{\lambda_l^2}{\beta}$. After that, the sum of all the inclusion probabilities equals $\sum_{i=1}^N \pi_i = t + \sum_{l=t+1}^N \frac{\lambda_l}{\sqrt{\beta}} = n \Rightarrow \sqrt{\beta} = \frac{\sum_{l=t+1}^N \lambda_l}{n-t} \Rightarrow \pi_l = (n-t)\, \frac{\lambda_l}{\sum_{j=t+1}^N \lambda_j}$. Since all the probabilities π_l should be less than 1 for $l > t$, there is a natural constraint $\lambda_{t+1} < \frac{\sum_{j=t+1}^N \lambda_j}{n-t}$.

Thus, the search of the optimal set of inclusion probabilities can be done in time not worse than $O(Nn)$. See Algorithm 1 for details. In practice, we found that the computation speed can be significantly increased with vectorization. Note that the solution always exists in Algorithm 1 because for $t = n-1$ the condition $\lambda_{t+1} < \frac{\Lambda_t}{n-t}$ holds true as obviously $\lambda_n < \sum_{l=n}^N \lambda_l$.

Remark B.1. The optimal set of probabilities $(\pi_1, \dots, \pi_N)$ specified by Eq. (5) is *balanced* [39] w.r.t. the variables $(\lambda_1, \dots, \lambda_N)$, i.e. for any subset of indices $\mathcal{I}$ sampled according to the optimal distribution $p\left(\mathcal{I}\right)$ the following equality holds

$$\sum_{j=1}^n \frac{\lambda_{\mathcal{I}_j}}{\pi_{\mathcal{I}_j}} = \sum_{i=1}^N \lambda_i.$$

Therefore, in the case of spectral model sharding, before the local training starts, the sum of singular values of the estimator $\hat{W}$ is equal to the sum of singular values of the full matrix W.

B.2 Collective Estimator

Now assume a slightly modified version of the setting described above: There exist $C > 1$ clients each of which samples an i.i.d. estimator $\hat{Y}^{(c)},\ 1 \leq c \leq C$, of the target value Y using the same distribution $p\left(\mathcal{I}\right)$. In this section we consider a simpler case of weighting where coefficients w_i do not depend on the sampled subset of indices anymore, i.e. $\hat{Y}^{(c)} = \sum_{i=1}^{N} z_i^{(c)} w_i X_i$. Imagine another participant, a 'server', who tries to reconstruct the target value Y by averaging the clients' estimators, $\bar{Y} = \frac{1}{C}\sum_{c=1}^{C}\hat{Y}^{(c)} = \frac{1}{C}\sum_{i=1}^{N} w_i X_i \sum_{c=1}^{C} z_i^{(c)}$. We aim to minimize the average squared error between the server's estimator $\bar{Y}$ and ground-true value Y. As before, due to the mutual orthogonality of $\{X_i\}$, this error can be expressed solely in terms of the magnitudes,

$$\begin{aligned}
\mathbb{E}\left\|Y - \bar{Y}\right\|^2 &= \mathbb{E}\sum_{i=1}^{N}\left(\lambda_i - \lambda_i \frac{w_i}{C}\sum_{c=1}^{C} z_i^{(c)}\right)^2 \\
&= \sum_{i=1}^{N}\lambda_i^2 \mathbb{E}\left(1 - \frac{w_i}{C}\sum_{c=1}^{C} z_i^{(c)}\right)^2 \\
&= \sum_{i=1}^{N}\lambda_i^2\left(1 - \frac{2w_i}{C}\sum_{c=1}^{C}\mathbb{E}z_i^{(c)} + \frac{w_i^2}{C^2}\left[\sum_{c=1}^{C}\mathbb{E}z_i^{(c)} + 2\sum_{c'<c}\mathbb{E}\left\{z_i^{(c)} z_i^{(c')}\right\}\right]\right) \\
&= \sum_{i=1}^{N}\lambda_i^2\left(1 - \frac{2w_i}{C}C\pi_i + \frac{w_i^2}{C^2}\left[C\pi_i + 2\frac{C\left(C-1\right)}{2}\pi_i^2\right]\right) \\
&= \sum_{i=1}^{N}\lambda_i^2\left(1 - 2w_i\pi_i + \frac{w_i^2}{C}\pi_i + \frac{w_i^2\left(C-1\right)}{C}\pi_i^2\right) \\
&= \sum_{i=1}^{N}\lambda_i^2 + \sum_{i=1}^{N}\lambda_i^2 w_i\pi_i\left(-2 + \frac{w_i}{C} + \frac{w_i\pi_i\left(C-1\right)}{C}\right).
\end{aligned} \tag{22}$$

Now we can formulate a new constrained optimization problem,

$$\begin{aligned}
\min_{\{w_i,\pi_i\}_i} \quad & \sum_{i=1}^{N}\lambda_i^2 w_i\pi_i\left(-2 + \frac{w_i}{C} + \frac{w_i\pi_i\left(C-1\right)}{C}\right), \\
\text{s.t.} \quad & \sum_{i=1}^{N}\pi_i = n, \\
& 0 \leq \pi_i \leq 1,\ i = 1,\ldots,N.
\end{aligned} \tag{23}$$

The corresponding Lagrangian function equals

$$\begin{aligned}
L(\left\{\pi_i\right\}, \left\{w_i\right\}, \alpha, \beta, \left\{\gamma_i\right\}, \left\{\delta_i\right\}) = \alpha\sum_i \lambda_i^2 w_i\pi_i&\left(-2 + \frac{w_i}{C} + \frac{w_i\pi_i\left(C-1\right)}{C}\right) \\
&+ \beta\left(\sum_i \pi_i - n\right) \\
&+ \sum_i \gamma_i\left(-\pi_i\right) + \sum_i \delta_i\left(\pi_i - 1\right)
\end{aligned} \tag{24}$$

https://doi.org/10.52202/079017-1968

and leads to the following conditions for all $i = 1, \dots, N$

$$\begin{aligned}
&\frac{\partial L}{\partial \pi_i} = \alpha \lambda_i^2 w_i \left(-2 + \frac{w_i}{C} + \frac{2 w_i \pi_i \left(C-1\right)}{C}\right) + \beta - \gamma_i + \delta_i = 0,\\
&\frac{\partial L}{\partial w_i} = \alpha \lambda_i^2 \pi_i \left(-2 + \frac{2 w_i}{C} + \frac{2 w_i \pi_i \left(C-1\right)}{C}\right) = 0,\\
&\sum_i \pi_i = n,\\
&\gamma_i \pi_i = 0,\\
&\delta_i \left(\pi_i - 1\right) = 0,\\
&\alpha,\, \gamma_i,\, \delta_i \geq 0,\\
&\alpha^2 + \beta^2 + \sum_i \gamma_i^2 + \sum_i \delta_i^2 > 0.
\end{aligned} \tag{25}$$

Case $\alpha = 0$. This case does not differ from the one of the setting of the unbiased estimator.

Case $\alpha = 1$. Since $\sum_i \pi_i = n$, there exists such k that the corresponding inclusion probability π_k is greater than zero, $\exists k\ \pi_k > 0 \Rightarrow -2 + \frac{2w_k}{C} + \frac{2w_k \pi_k (C-1)}{C} = 0 \Rightarrow w_k = \frac{C}{1+\pi_k(C-1)}$. As we look for $0 < \pi_k \leq 1$, this implies $1 \leq w_k < C$. Also, since $\pi_k > 0$, then $\gamma_k = 0 \Rightarrow \lambda_k^2 w_k \left(-2 + \frac{w_k}{C} + \frac{2w_k \pi_k (C-1)}{C}\right) + \beta + \delta_k = 0 \Rightarrow \lambda_k^2 w_k \left(-\frac{w_k}{C}\right) + \beta + \delta_k = 0 \Rightarrow \beta = -\delta_k + \frac{\lambda_k^2 w_k^2}{C}$.

If $\pi_k < 1$, then $\delta_k = 0 \Rightarrow w_k = \frac{\sqrt{C\beta}}{\lambda_k} \Rightarrow \pi_k = \frac{1}{C-1}\left[\lambda_k \sqrt{\frac{C}{\beta}} - 1\right]$. Furthermore, as in this case $1 < w_k < C$, then $\frac{\lambda_k^2}{C} < \beta < \lambda_k^2 C \Rightarrow \max\limits_{k:0<\pi_k<1} \frac{\lambda_k^2}{C} < \beta < \min\limits_{k:0<\pi_k<1} \lambda_k^2 C$.

Otherwise, if $\pi_k = 1$, then $w_k = 1$, and $\beta = -\delta_k + \frac{\lambda_k^2}{C}$, and since $\delta_k \geq 0$, the following inequality holds true, $\beta \leq \min\limits_{k:\,\pi_k=1} \frac{\lambda_k^2}{C}$.

Combining these observations together, we obtain $\max\limits_{k:0<\pi_k<1} \frac{\lambda_k^2}{C} < \beta \leq \min\limits_{k:\,\pi_k=1} \frac{\lambda_k^2}{C}$. This inequality can hold true only if, similarly to the case of unbiased estimator, several largest magnitudes correspond to inclusion probabilities equal to 1, and all the rest inclusion probabilities are strictly less than 1.

Revisiting the optimization criterion $\sum_{i=1}^N F_i = \sum_{i=1}^N \lambda_i^2 w_i \pi_i \left(-2 + \frac{w_i}{C} + \frac{w_i \pi_i (C-1)}{C}\right)$, we can now consider each separate term of the sum. Starting with the product of the inclusion probability and the weighting coefficient

$$\pi_i w_i = \begin{cases} 1 & \text{if } \pi_i = 1,\\ \frac{1}{C-1}\left[\lambda_i \sqrt{\frac{C}{\beta}} - 1\right] \cdot \frac{\sqrt{C\beta}}{\lambda_i} = \frac{C}{C-1} - \frac{\sqrt{C\beta}}{\lambda_i (C-1)} & \text{if } 0 < \pi_i < 1,\\ 0 & \text{if } \pi_i = 0, \end{cases} \tag{26}$$

we can write down the equation of each term

$$F_i = \begin{cases} -\lambda_i^2 & \text{if } \pi_i = 1,\\ \lambda_i^2 \left(\frac{C}{C-1} - \frac{\sqrt{C\beta}}{\lambda_i (C-1)}\right)\left(-2 + \frac{\sqrt{\beta}}{\lambda_i \sqrt{C}} + 1 - \frac{\sqrt{\beta}}{\lambda_i \sqrt{C}}\right) = -\frac{C}{C-1}\lambda_i \left(\lambda_i - \sqrt{\frac{\beta}{C}}\right) & \text{if } 0 < \pi_i < 1,\\ 0 & \text{if } \pi_i = 0. \end{cases} \tag{27}$$

As was derived above, in case $0 < \pi_i < 1$ we have $\beta < \lambda_i^2 C \Rightarrow \lambda_i > \sqrt{\frac{\beta}{C}}$. Therefore, F_i is a monotonically decreasing function of λ_i with negative values if $0 < \pi_i < 1$. Since the goal is to minimize $\sum_i F_i$, this implies that it is beneficial to assign non-zero probabilities to greater values of λ_i. Therefore, the optimal solution $(\pi_1, \dots, \pi_N)$ has the following form for some $t \geq 0,\ u \geq 0$,

- $\pi_1 = \dots = \pi_t = 1$,

- $1 > \pi_{t+1} \geq \cdots \geq \pi_{t+u} > 0$,
- $\pi_{t+u+1} = \cdots = \pi_N = 0$.

Since the sum of all inclusion probabilities should be equal to n,

$$\sum_i \pi_i = t + \frac{\sqrt{C}}{\sqrt{\beta}\,(C-1)} \sum_{i=t+1}^{t+u} \lambda_i - \frac{u}{C-1} = n \Rightarrow \sqrt{\beta} = \frac{\sqrt{C}\sum_{i=t+1}^{t+u} \lambda_i}{(n-t)\,(C-1)+u}$$

The search for the best solution takes no longer then $O\big(N^2 n\big)$ time, see Algorithm 2 for details. As before, the calculations may be done in the vectorized form.

Remark B.2. Interestingly, for large values of C, the optimal values given by Eq. (8) are approaching *some* unbiased Horvitz-Thompson estimator of W, in accordance to the probability theory's law of large numbers. However, this estimator is not guaranteed to have the properties that are provided by Theorem 3.1. Therefore, for large values of C it is more beneficial to use the Unbiased estimator instead of the Collective one.

Remark B.3. It is straightforward to show that if the constraint of unbiasedness is applied to the collective estimator, i.e. $\mathbb{E}\bar{W} = W$, then it necessarily implies that $w_i = \pi_i^{-1}$, just as in the case of the unbiased estimator. Therefore, the unbiased collective estimator is the same as the unbiased estimator. Since Theorem 3.2 provides the estimator with the least Frobenius discrepancy among all the collective estimators, the achieved least discrepancy cannot be greater than the least possible discrepancy among unbiased collective estimators. This concludes the proof that the discrepancy of the collective estimator is not greater than the discrepancy of the unbiased estimator. However, as Remark B.2 implies, in the limit case $C \to \infty$ these discrepancies become equal.

C Wallenius' Distribution and `numpy.random.choice`

The official documentation[4] of NumPy [12] currently does not contain a description of how exactly sampling *without* replacement with unequal probabilities is done if the function `numpy.random.choice` is employed. Nevertheless, the source code[5] suggests that sampling is conducted in the following way:

1. The remaining number of items is sampled *with* replacement from the multinomial distribution with probabilities proportional to the current chances (in the beginning, those chances are equal to the probabilities provided by the user).
2. The unique items from the sampled subset are added to the output of the function, and the chances of those items are set equal to zero.
3. The steps above are repeated with the updated chances until the requested number of items is selected.

The analysis of the statistical properties of the output sample seems infeasible if the probabilities given by the user are far from being uniform. However, the case of non-uniform chances is exactly what we are interested in when working with singular values of weight matrices.

Interestingly, this algorithm from NumPy resembles the description of multivariate Wallenius' non-central hypergeometric distribution [42, 5, 9, 10] which is often defined through the urn problem. In detail, if one draws items one by one, instead of trying to get the maximum possible number at each step, this would provide a sample from a special case of Wallenius' law.

Clearly, the two procedures are different significantly enough to produce different distributions. Despite this, we found that in our simulations the mean vector of the Wallenius' distribution was quite a good approximation for the empirical inclusion probabilities of the output of NumPy function.

[4] `https://github.com/numpy/numpy/blob/0a4b2b83eaf3479f352a7fe5a4b378a169ab48f0/numpy/random/mtrand.pyx#L856-L954`

[5] `https://github.com/numpy/numpy/blob/0a4b2b83eaf3479f352a7fe5a4b378a169ab48f0/numpy/random/mtrand.pyx#L1026-L1045`

D Licenses for Assets

1. CIFAR-10 [23]: unknown
2. CIFAR-100 [23]: unknown
3. TinyImagenet [31]: unknown
4. Shakespeare [29]: Apache License, Version 2.0

Impact Statement

This paper presents work whose goal is to advance the field of federated learning. The potential societal consequences include the increased abilities of training more powerful models on a federation of user devices in privacy-friendly manner. At the moment, we do not see any direct negative consequences of the conducted research.

https://doi.org/10.52202/079017-1968

Algorithm 1 Inclusion Probabilities for the Unbiased Strategy

```
Require: Magnitudes $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_N > 0$, number of samples $0 < n < N$.
Ensure: Optimal inclusion probabilities $(\pi_1^*, \ldots, \pi_N^*)$
$E^* \leftarrow \infty$                # Best criterion value
for $t = 0$ to $n - 1$ do
    $\Lambda_t \leftarrow \sum_{l=t+1}^{N} \lambda_l$
    if $\lambda_{t+1} < \frac{\Lambda_t}{n-t}$ then
        for $l = 1$ to $t$ do
            $\pi_l \leftarrow 1$
        end for
        for $l = t + 1$ to $N$ do
            $\pi_l \leftarrow \frac{(n-t)\lambda_l}{\Lambda_t}$
        end for
        $E = \sum_{i=1}^{N} \lambda_i^2 \left(\pi_i^{-1} - 1\right)$
        if $E < E^*$ then
            $E^* \leftarrow E$
            $(\pi_1^*, \ldots, \pi_N^*) \leftarrow (\pi_1, \ldots, \pi_N)$
        end if
    end if
end for
```

Algorithm 2 Inclusion Probabilities and Auxiliary Multipliers for the Collective Strategy

```
Require: Magnitudes $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_N > 0$, number of samples $0 < n < N$, number of clients $C$.
Ensure: Optimal inclusion probabilities $(\pi_1^*, \ldots, \pi_N^*)$, optimal auxiliary multipliers $(w_1^*, \ldots, w_N^*)$.
    # Consider top-$n$ case first, i.e. $t = n,\ u = 0$

for $i = 1, \ldots, n$ do
    $\pi_i \leftarrow 1$
    $w_i \leftarrow 1$
end for
for $i = n + 1, \ldots, N$ do
    $\pi_i \leftarrow 0$
    $w_i \leftarrow 0$
end for
$E^* \leftarrow -\sum_{i=1}^{n} \lambda_i^2$
$(\pi_1^*, \ldots, \pi_N^*) \leftarrow (\pi_1, \ldots, \pi_N)$
$(w_1^*, \ldots, w_N^*) \leftarrow (w_1, \ldots, w_N)$
# Other cases
for $t = 0, \ldots, n - 1$ do
    for $i = 1, \ldots, t$ do
        $\pi_i \leftarrow 1$
        $w_i \leftarrow 1$
    end for
    $E_t \leftarrow -\sum_{i=1}^{t} \lambda_i^2$
    for $u = 1, \ldots, N - t$ do
        $\sqrt{\beta} \leftarrow \frac{\sqrt{C} \sum_{i=t+1}^{t+u} \lambda_i}{(n-t)(C-1)+u}$
        if $\frac{\lambda_{t+1}}{\sqrt{C}} < \sqrt{\beta} \leq \frac{\lambda_t}{\sqrt{C}}$ & $\sqrt{\beta} < \lambda_{t+u}\sqrt{C}$ then
            for $i = t + 1, \ldots, t + u$ do
                $\pi_i \leftarrow \frac{1}{C-1}\left[\frac{\lambda_i \sqrt{C}}{\sqrt{\beta}} - 1\right]$
                $w_i \leftarrow \frac{\sqrt{C}\sqrt{\beta}}{\lambda_i}$
            end for
            for $i = t + u + 1, \ldots, N$ do
                $\pi_i \leftarrow 0$
                $w_i \leftarrow 0$
            end for
            $E \leftarrow E_t - \frac{C}{C-1} \sum_{i=t+1}^{t+u} \lambda_i \left(\lambda_i - \frac{\sqrt{\beta}}{\sqrt{C}}\right)$
            if $E < E^*$ then
                $E^* \leftarrow E$
                $(\pi_1^*, \ldots, \pi_N^*) \leftarrow (\pi_1, \ldots, \pi_N)$
                $(w_1^*, \ldots, w_N^*) \leftarrow (w_1, \ldots, w_N)$
            end if
        end if
    end for
end for
```

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 strategy using unbiased estimators is introduced in Sec. 3.2.1. The strategy minimizing the squared approximation error is derived in Sec. 3.2.2. The practical considerations are discussed in Sec. 3.2.3. The quantitative evaluation is provided in Sec. 4.

   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: The used assumptions and their limitations are explicitly discussed in Secs. 3.2.1, 3.2.2 and 5 and Appendix A.

   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?

https://doi.org/10.52202/079017-1968

Answer: [Yes]

Justification: Due to the lack of space, the proof of Theorem 3.1 was put in Appendix B.1. The proof of Theorem 3.2 for the case of $C = 1$ is given in the main text, and the proof for $C > 1$ is provided in Appendix B.2.

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: The step-by-step algorithms for the proposed strategies are presented in Algorithms 1 and 2. All the training details are provided in Sec. 4.

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.

https://doi.org/10.52202/079017-1968

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] for data and [No] for code

   Justification: All the datasets used are public, see the provided references in Sec. 4. The code cannot be released at the moment due to copyright procedures.

   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: All these details are provided in Sec. 4.

   Guidelines:

   - 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: All the reported results in both tables and plots include error bars obtained from multiple runs with different random seeds, as explained in Sec. 4.

   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.

https://doi.org/10.52202/079017-1968

- 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: Due to the lack of space, the details are provided in Appendix A.

   Guidelines:

   - 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: Our research conforms with the NeurIPS Code of Ethics.

   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: Due to the lack of space, the impact statement was put to the supplementary material.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.

https://doi.org/10.52202/079017-1968

- 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: Since our work is mostly theoretical, we believe that our paper poses no such risk.

    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: All the models, algorithms and datasets are properly cited in the text. We re-implemented all the models and algorithms ourselves. The linceses for datasets are listed in the supplementary material.

    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.

https://doi.org/10.52202/079017-1968

- 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: [NA]

    Justification: No new assets released.

    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: Our 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: Our 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.

https://doi.org/10.52202/079017-1968