

## A. Experiment Configurations

**Datasets.** We perform experiments on three popular vision datasets (CIFAR-10 [17], Fashion-MNIST [35] and SVHN [26]), and a natural language dataset (Shakespeare [6]). We use the standard train/test split on these datasets and report accuracy on test dataset with a global model on server-side. To simulate different data heterogeneity levels, for CV tasks we split the datasets by class label with the distribution  $Dir_{|\mathcal{C}|}(\alpha)$  [14, 32], and adjust the  $\alpha$  constant for varying the degree of data heterogeneity. We chose  $\alpha = 0.5$  unless specified for all experiments. We employ simple data augmentation for the training images, including random crop and normalization, and an extra random flip for CIFAR-10. For the Shakespeare NLP dataset, we follow LEAF [6] for next-character prediction, which assumes each role in each play to be an individual client. The resulting splits are thus inherently heterogeneous since each local client only contains the corresponding roles sentences.

**Models.** All models listed for computer vision (CV) tasks (*i.e.*, Fashion-MNIST, SVHN, CIFAR-10) are standard CNNs equipped with channel dropouts, and Tables 6 to 8 respectively show their layer configurations.

**Competing sparse FL methods.** In our experiments, we compare Flado against SOTA sparse FL methods, namely HeteroFL [9], FjORD [13] and eFD (an extension of federated dropout [5] by [13]). As we employ FLOPs budget constraints, we define

$$p_c = \inf_p \{g_c(r_c, p) \geq 0\}, \quad (6)$$

*i.e.*, the shared density of all layers in the client  $c$  that satisfies the FLOPs budget. The implementation details of the methods are as follows:

- FjORD samples a  $p_{c,k}$  from a conditional distribution of model widths  $D_p | D_p \leq p_{\max}^c$  for each client  $c$  at the  $k^{\text{th}}$  local training step. Here,  $p_{\max}^c$  denotes the maximum permissible model for the client. Following [13], we let  $D_p$  be a uniform distribution  $\mathcal{U}[\underline{p}, \bar{p}]$ , where  $(\underline{p}, \bar{p}) = (\min_{c \in \mathcal{C}} p_c, \max_{c \in \mathcal{C}} p_c)$ . To further ensure clients respect their respective FLOPs budgets, we let  $p_{\max}^c = \inf_p \{\mathbb{E}_{q \sim \mathcal{U}[\underline{p}, \bar{p}]} [g_c(r_c, q)] \geq 0\}$ .
- eFD samples sub-model for each client in each training round. Following the definition of eFD in [13], at the start of each round, for each client  $c \in \mathcal{C}$  we choose to enable and transmit its neurons with a shared probability  $p_c$ . This reduces the full model to a sub-model satisfying the  $r_c$  FLOPs budget.
- HeteroFL selects the first  $\lceil p_c C^{[l]} \rceil$  channels of each layer  $l$  for each client  $c$  to form a  $p_c$ -reduced sub-model, where  $C^{[l]}$  denotes the number of channels in layer  $l$ . The clients then train the derived sub-models for a round.

Table 6. Layout of the model used for Fashion-MNIST training.

	Layer	Kernel	Stride	Feature shape	#Params	#FLOPs
1	Conv+ReLU	$5 \times 5$	1	$32 \times 28 \times 28$	832	652 k
2	Max Pool	$2 \times 2$	2	$32 \times 14 \times 14$	—	25.1 k
3	Conv+ReLU	$5 \times 5$	1	$64 \times 14 \times 14$	51.3 k	10.0 M
4	Max Pool	$2 \times 2$	2	$64 \times 7 \times 7$	—	125 k
5	Conv+ReLU	$3 \times 3$	1	$64 \times 5 \times 5$	36.9 k	923 k
6	Avg Pool	$2 \times 2$	1	$64 \times 2 \times 2$	—	1.6 k
7	FC	—	—	512	132 k	132 k
8	FC	—	—	10	5.13 k	5.13 k
<b>Total</b>					226 k	11.8 M

Table 7. Layout of the model used for SVHN training.

	Layer	Kernel	Stride	Feature shape	#Params	#FLOPs
1	Conv+ReLU	$5 \times 5$	1	$32 \times 28 \times 28$	2.43 k	1.91 M
2	Max Pool	$2 \times 2$	2	$32 \times 14 \times 14$	—	25.1 k
3	Conv+ReLU	$5 \times 5$	1	$64 \times 10 \times 10$	51.3 k	5.13 M
4	Max Pool	$2 \times 2$	2	$64 \times 5 \times 5$	—	6.4 k
5	Conv+ReLU	$3 \times 3$	1	$64 \times 3 \times 3$	36.9 k	333 k
6	Avg Pool	$2 \times 2$	1	$64 \times 2 \times 2$	—	576
7	FC	—	—	512	132 k	132 k
8	FC	—	—	10	5.13 k	5.13 k
<b>Total</b>					227 k	7.54 M

Table 8. Layout of the VGG-9 model used for CIFAR-10 training.

	Layer	Kernel	Stride	Feature shape	#Params	#FLOPs
1	Conv+ReLU	$3 \times 3$	1	$32 \times 32 \times 32$	896	918 k
2	Conv+ReLU	$3 \times 3$	1	$64 \times 32 \times 32$	18.5 k	18.9 M
3	Max Pool	$2 \times 2$	2	$64 \times 16 \times 16$	—	65.5 k
4	Conv+ReLU	$3 \times 3$	1	$128 \times 16 \times 16$	73.9 k	18.9 M
5	Conv+ReLU	$3 \times 3$	1	$128 \times 16 \times 16$	148 k	37.8 M
6	Max Pool	$2 \times 2$	2	$128 \times 8 \times 8$	—	32.8 k
7	Conv+ReLU	$3 \times 3$	1	$256 \times 8 \times 8$	295 k	18.9 M
8	Conv+ReLU	$3 \times 3$	1	$256 \times 8 \times 8$	590 k	37.8 M
9	Avg Pool	$8 \times 8$	—	$256 \times 1 \times 1$	—	16.4 k
10	FC	—	—	512	132 k	132 k
11	FC	—	—	512	263 k	263 k
12	FC	—	—	10	5.13 k	5.13 k
<b>Total</b>					1.53 M	134 M

After each round of training, the parameters of each neuron are aggregated over clients that updated this neuron in the current training round. Namely, for each channel neuron  $n$ , we perform the following aggregation for its parameters  $\theta^n$ :

$$\theta^n = \sum_{c \in \text{trained}(n)} \frac{\lambda_c}{\lambda} \theta_c^n, \quad (7)$$

where  $\lambda = \sum_{c \in \text{trained}(n)} \lambda_c$ , and the function  $\text{trained}(n)$  returns the set of clients that trained neuron  $n$  in the current round. This process is identical to the ones employed in both FjORD [13] and HeteroFL [9].

## B. Computing the Number of FLOPs

In the  $l^{\text{th}}$  sparse layer (denoted by  $\hat{h}^{[l]}$ ), the expected number of FLOPs per image per step is the sum of the FLOPs

required by both the convolution and ReLU activation:

$$\text{flops}(\hat{h}^{[l]}) = 2(\hat{C}^{[l]}\hat{C}^{[l-1]}K^{[l]2}H^{[l]}W^{[l]} + \hat{C}^{[l]}H^{[l]}W^{[l]}), \quad (8)$$

where  $K^{[l]2}$  is the 2-dimensional kernel size, and  $H^{[l]} \times W^{[l]}$  is the output feature map size. Moreover,  $\hat{C}^{[l-1]}$  and  $\hat{C}^{[l]}$  are the number of active input and output channels respectively; we assume  $\hat{C}^{[l]}$ , i.e., the average number of remaining output channels for layer  $l$ , to be  $C^{[l]}$  mean( $\mathbf{p}_c^{[l]}$ ), and mean( $\mathbf{z}$ ) computes the mean of elements in  $\mathbf{z}$  and  $C^{[l]}$  is the number of total output channels of layer  $l$ .

To generalize, we can rewrite the total number of FLOPs required by the overall sparse model  $\hat{h}$  with  $L$  layers to be:

$$\text{flops}(\hat{h}, \mathbf{p}_c) = 2\sum_{l=1}^L C^{[l]} \text{mean}(\mathbf{p}_c^{[l]}) (K^{[l]2}C^{[l-1]} \text{mean}(\mathbf{p}_c^{[l-1]}) + 1) H^{[l]} W^{[l]}, \quad (9)$$

and assume  $\mathbf{p}_c^{[0]} = \mathbf{1}$  and  $\mathbf{p}_c^{[L]} = \mathbf{1}$ , since both the input and output of the model must be dense. Finally, by evaluating  $\text{flops}(\hat{h}, \mathbf{1})$ , we can get the number of FLOPs consumed by the fully dense model.

## C. Fast Johnson-Lindenstrauss Transform

**Lemma C.1** (The FJLT Lemma [1]). *Given a set  $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n\}$  of  $n$  vectors in  $\mathbb{R}^d$ , and let  $\Phi = \mathbf{PHD} \sim \text{FJLT}(n, d, k, q, \epsilon)$  be the sampled FJLT transform. Here,*

- $\mathbf{P}$  is a  $k \times d$  sparse matrix, where its entries  $p_{ij} \sim \mathcal{N}(0, q^{-1})$  with probability  $q$ , and  $p_{ij} = 0$  otherwise. We let  $q = \min\{\Theta(1/d \log^2 n), 1\}$ .
- $\mathbf{H}$  is a  $d \times d$  Hadamard matrix.
- $\mathbf{D}$  is a diagonal matrix entries drawn from  $\{-1, 1\}$  uniformly.

With high probability ( $> 2/3$ ), the following events hold:

- For all  $\mathbf{x}_i \in \mathbf{X}$ ,

$$(1 - \epsilon)k\|\mathbf{x}_i\|_2 \leq \|\Phi\mathbf{x}_i\|_2 \leq (1 + \epsilon)k\|\mathbf{x}_i\|_2. \quad (10)$$

- The projection  $\Phi : \mathbb{R}^m \rightarrow \mathbb{R}^k$  requires  $O(d \log d + \min\{d\epsilon^{-2} \log n, \epsilon^{-2} \log^3 n\})$  operations.

It is easy to prove that a tight bound hold for cosine-similarity with high probability.

**Lemma C.2** (The FJLT lemma adapted for cosine similarity). *With the same assumptions in Lemma C.1, we have*

$$\frac{1-\epsilon}{1+\epsilon} \text{cossim}(\mathbf{x}_i, \mathbf{x}_j) \leq \text{cossim}(\Phi\mathbf{x}_i, \Phi\mathbf{x}_j), \text{ and} \quad (11)$$

$$\text{cossim}(\Phi\mathbf{x}_i, \Phi\mathbf{x}_j) \leq \frac{1+\epsilon}{1-\epsilon} \text{cossim}(\mathbf{x}_i, \mathbf{x}_j).$$

*Proof.* We begin with the cosine similarity of the FJLT of both vectors  $\mathbf{x}_i$  and  $\mathbf{x}_j$ :

$$\begin{aligned} \text{cossim}(\Phi\mathbf{x}_i, \Phi\mathbf{x}_j) &= \left\langle \frac{\Phi\mathbf{x}_i}{\|\Phi\mathbf{x}_i\|_2}, \frac{\Phi\mathbf{x}_j}{\|\Phi\mathbf{x}_j\|_2} \right\rangle \\ &= 1 - \frac{1}{2} \left\| \frac{\Phi\mathbf{x}_i}{\|\Phi\mathbf{x}_i\|_2} - \frac{\Phi\mathbf{x}_j}{\|\Phi\mathbf{x}_j\|_2} \right\|_2^2. \end{aligned}$$

Applying (10) of Lemma C.1 on the norms in the denominators, using interval arithmetic:

$$\in 1 - \frac{1}{2} \left\| \Phi \frac{1}{(1 \pm \epsilon)k} \left( \frac{\mathbf{x}_i}{\|\mathbf{x}_i\|_2} - \frac{\mathbf{x}_j}{\|\mathbf{x}_j\|_2} \right) \right\|_2^2,$$

and also on the outer norm, we have:

$$\begin{aligned} &\subseteq 1 - \frac{1}{2} (1 \pm \epsilon)^2 k^2 \left\| \frac{1}{(1 \pm \epsilon)k} \left( \frac{\mathbf{x}_i}{\|\mathbf{x}_i\|_2} - \frac{\mathbf{x}_j}{\|\mathbf{x}_j\|_2} \right) \right\|_2^2 \\ &= 1 - \frac{1}{2} \left( \frac{1+\epsilon}{1-\epsilon} \right)^{\pm 1} \left\| \frac{\mathbf{x}_i}{\|\mathbf{x}_i\|_2} - \frac{\mathbf{x}_j}{\|\mathbf{x}_j\|_2} \right\|_2^2 \\ &= 1 - \frac{1}{2} \left( \frac{1+\epsilon}{1-\epsilon} \right)^{\pm 1} (2 - 2 \text{cossim}(\mathbf{x}_i, \mathbf{x}_j)) \\ &\subseteq \left( \frac{1+\epsilon}{1-\epsilon} \right)^{\pm 1} \text{cossim}(\mathbf{x}_i, \mathbf{x}_j). \end{aligned}$$

□

In each training round, we sample the same FJLT matrices from  $\text{FJLT}(n^{[l]}, d^{[l]}, k^{[l]}, q^{[l]}, \epsilon)$  that are shared across all clients. One can simply use the current round number as the shared random seed to guarantee identical sampling. We let  $d$  be the number of clients, set  $k = O(\epsilon^{-2} \log n)$  and finally fix  $q$  according to Lemma C.1. For example, for  $\epsilon \leq 0.02$ , which provides a good trade-off between the size of projected embeddings and the approximation bounds, satisfying this bound, would require  $k = 4/(\epsilon^2/2 - \epsilon^3/3) \log n$  [8]. For CIFAR-10 with 20 clients, this evaluates to 60,724 values after performing the FJLT on the full model parameters.

In practice, the observed error bound is much smaller than the theoretical bounds, and we can thus afford to reduce  $k$  further. Figure 7 provides the empirical error bounds  $\mathcal{E} = \|\text{cossim}(J(\mathbf{a}), J(\mathbf{b})) - \text{cossim}(\mathbf{a}, \mathbf{b})\|_\infty$  between the approximated cosine similarity (CS) of FJLT randomly-projected vectors and the true CS before projection, where  $\mathbf{a} = \Delta\theta^{(t)}$  and  $\mathbf{b} = \nabla_{\theta^{(t)}} \ell_c(\mathbf{b}_c \circ \theta^{(t)})$ . We gather the results after the first round of training 20 CIFAR-10 clients, following our default data and system heterogeneity settings. Here, we vary the size of  $k$  to trade off the size of the projected embeddings and the error bounds  $\mathcal{E}$  on CS approximation. We highlight that empirically, a small  $k$  still preserves the CS after approximation very well. This means a very small amount of additional communicated parameters for each client downloads before its training round is necessary for an accurate representation of the model update trajectories. We thus chose  $k = \lceil \frac{M}{10^3} \rceil$ , where  $M$  is the number of

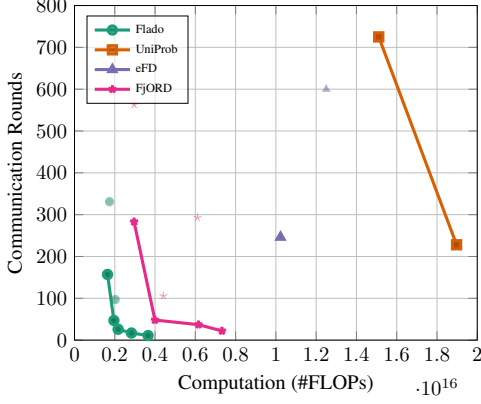


Figure 6. Comparing the FLOPs vs. communicated parameters trade-off across different FL methods reaching a target accuracy under both data and system heterogeneity. For detailed explanations, see the caption of Figure 5.

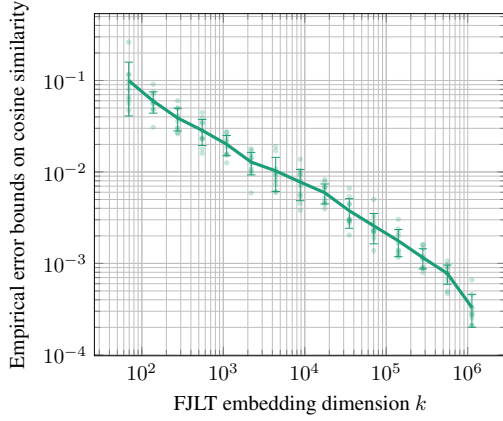


Figure 7. The maximum empirical error bounds on the approximation of cosine similarity values for 20 clients trained on CIFAR-10 under the default system and data heterogeneity. We repeat the experiments 10 times with different seeds for statistical bounds.

model parameters. This translates to only requiring an additional  $< 0.1\%$  downloaded parameters per round for each client. Under this setting, the number of FLOPs required for computing FJTL transform is 22.5 M additional FLOPs for every 1% of all local steps (or  $< 0.02\%$  additional FLOPs by a client per round).

## D. Additional Results

In addition to Figure 5 which compares the FLOPs-parameters trade-off Pareto frontiers, we also include a comparison on the numbers of FLOPs vs. communication rounds in Figure 6.

Figure 8 provides ablation and sensitivity analyses of Flado. Here, we adjust the proportion of local updates used to adapt channel selection probabilities. We find that 1% is

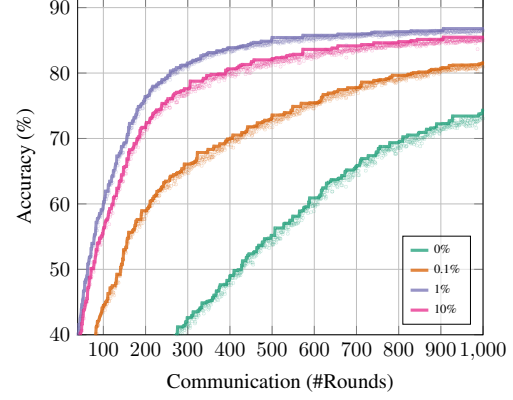


Figure 8. Ablation and sensitivity analyses of the proportion of local steps used to optimize channel selection probabilities on CIFAR-10. Here, 0% corresponds to no optimizations. We found 1% is typically the optimal proportion.

typically the optimal proportion, and this is used throughout our experiments. Note that with 0%, it disables the probability optimization process.