Mixed-Precision Quantization for Federated Learning on Resource-Constrained Heterogeneous Devices Supplementary Material

Huancheng Chen
University of Texas at Austin
huanchengch@utexas.edu

Haris Vikalo
University of Texas at Austin
hvikalo@ece.utexas.edu

1. Illustration of Data Partitions

We used Dirichlet distribution to generate non-IID data partitions across the clients, following the strategy in [1]. Specifically, the proportion $\mathbf{p}^{(c)}$ of samples with label c among N clients is drawn as

$$\mathbf{p}^{(c)} = \{p_n^{(c)}, n \in [N]\} \sim \mathbf{Dir}_N(\alpha), \tag{1}$$

where α is the concentration parameter characterizing the level of data heterogeneity. A smaller α leads to higher level of data heterogeneity, while a larger α induces lower level of data heterogeneity. The number of samples with label c assigned to client n is given by

$$\mathcal{N}_n^{(c)} = \frac{p_n^{(c)}}{\sum_{k=1}^{N} p_k^{(c)}} \mathcal{N}^{(c)}, \tag{2}$$

where $\mathcal{N}^{(c)}$ denotes the number of samples with label c in the overall training dataset. Figures 1 and 2 illustrate the label distribution of clients by color-coding the number of samples: the darker the color, the larger the number of samples with the corresponding label. To illustrate label distributions corresponding to different levels of data heterogeneity, we give examples of three scenarios by setting $\alpha = 0.1$, 0.5 and 5. As shown in Figures 1 and 2, when $\alpha = 0.1$ the clients typically have only 2 or 3 classes present in their datasets, while $\alpha = 5$ leads to local datasets with almost all classes.

2. Examples of Bit-Width Allocation

After completing bit-level pruning via sparsity-promoting training, local models are compressed to average bit-widths that may be lower than their budgets. The server then implements the *pruning-growing* algorithm to restore the bit-width of local models. As illustrated in Figures 3 and 4, the clients allocated budgets of 2, 4 and 6 bits exhibit similar patterns during the training process. All local models, regardless of their budget, end up having the first and the last layer assigned 8 bits (the maximum bit-width in the considered setting). The layers selected to be compressed into low precision are in the proximity of the region close to the fully-connected layer (for instance, from "conv14" to "conv18" in Fig. 3 and from "30" to "42" in Fig. 4). Features of the input are extracted by the innermost layers with high-precision filters and passed to the outermost layers with low-precision filters, so the quantization error does not accumulate. The outermost layers have more parameters which makes their compression cost-effective as more bit-width space is made available for other layers. Moreover, the fully-connected layer has a relatively small number of parameters and plays a crucial role in the classification, so it is always assigned the maximum precision.

3. Computing Binary Representations

Given any weight w in the float-point format, we use *round* operation to obtain its binary representation as follows:
(1) **layer-wise scaling factor**:

$$s^{(l)} \leftarrow \max_{j,k}\{w \in \mathbf{W}^{(l)}\}$$

(2) **addition with zero point**:

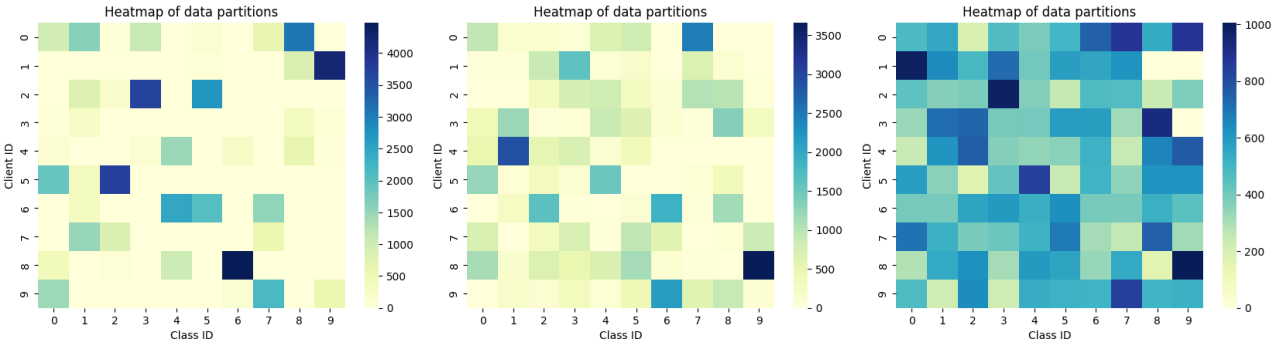
$$\textbf{step} \leftarrow 2^{\mathbf{b}^{(l)}} - 1$$
$$z^{(l)} \leftarrow 2^{\mathbf{b}^{(l)}-1}$$
$$w \leftarrow w + \frac{s^{(l)} \cdot z^{l}}{\textbf{step}}$$

(3) **round operation**:

$$w \leftarrow \textit{round}(w \cdot \textbf{step}/s^{(l)})$$

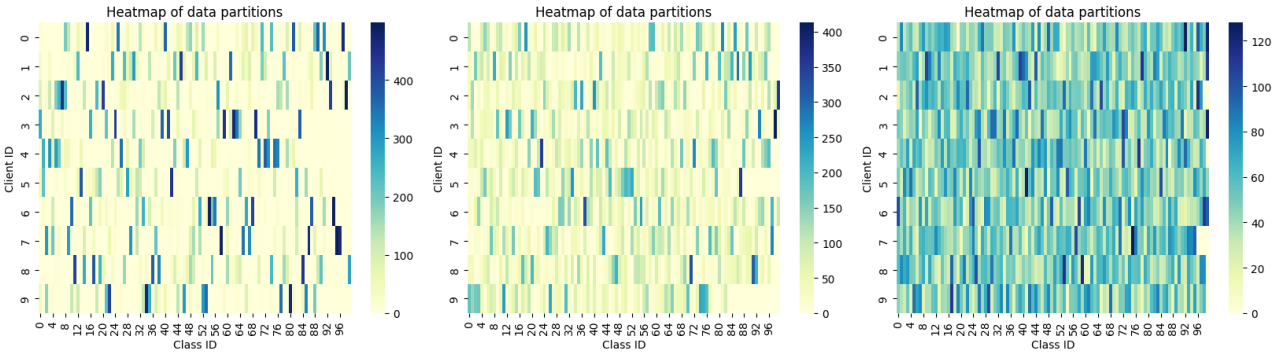
(4) **compute binary representation**:
let $\mathbf{B}$ denotes the binary weight; for $i \in \{0, \ldots, \mathbf{b}^{(l)} - 1\}$,

$$\textbf{ex} \leftarrow 2^{\mathbf{b}^{(l)}-i-1}$$
$$\mathbf{B}_i \leftarrow \textit{floor}(w/\textbf{ex})$$
$$w \leftarrow w - \textit{floor}(w/\textbf{ex}) \cdot \textbf{ex}$$
$$i \leftarrow i + 1$$



(a) $\alpha = 0.1$ (b) $\alpha = 0.5$ (c) $\alpha = 5$

Figure 1. Experiments on CIFAR10. Training data is split into 10 partitions according to a Dirichlet distribution. The concentration parameter is set as follows: (a) $\alpha = 0.1$; (b) $\alpha = 0.5$; (c) $\alpha = 5$.



(a) $\alpha = 0.1$ (b) $\alpha = 0.5$ (c) $\alpha = 5$

Figure 2. Experiments on CIFAR100. Training data is split into 10 partitions according to a Dirichlet distribution. The concentration parameter is set as follows: (a) $\alpha = 0.1$; (b) $\alpha = 0.5$; (c) $\alpha = 5$.

4. Additional Experimental Results

To provide insight in the improvement achieved by our methods, we study the test accuracy evaluated after a number of global rounds in the experiments in various settings. For each dataset, we run two groups of experiments with $\alpha = 0.5$ and $\alpha = 1.0$ where FedMPQ achieves performance similar to the FPQ8 baseline. As illustrated in Figures 5 and 6, FedMPQ converges to the final test accuracy much slower than FPQ8. As we demonstrated in the paper, the bit-width allocation of the local models in FedMPQ changes at each round and so does the search space of the optimization; meanwhile, FPQ8 has a larger, fixed search space so its training converges faster. Although the size of the search space is fixed due to the limited bit-width budget, FedMPQ enables local models to converge to the parameters that provide better performance.

The difference in convergence speed between FPQ8 and FedMPQ is smaller on CIFAR10 than CIFAR100. This happens because CIFAR10 data is simpler than CIFAR100 (fewer classes) so the networks can learn representations of the entire dataset even with a low capacity. On the other hand, Tiny-ImageNet is more complex than CIFAR100 so FedMPQ converges slower than FPQ8 and has performance that lags behind FPQ8 after 50 global training rounds. AQFL has fixed and small search space because of the constrained bit-width budgets so it converges fast but cannot achieve high accuracy, as illustrated in Figures 5, 6 and 7.

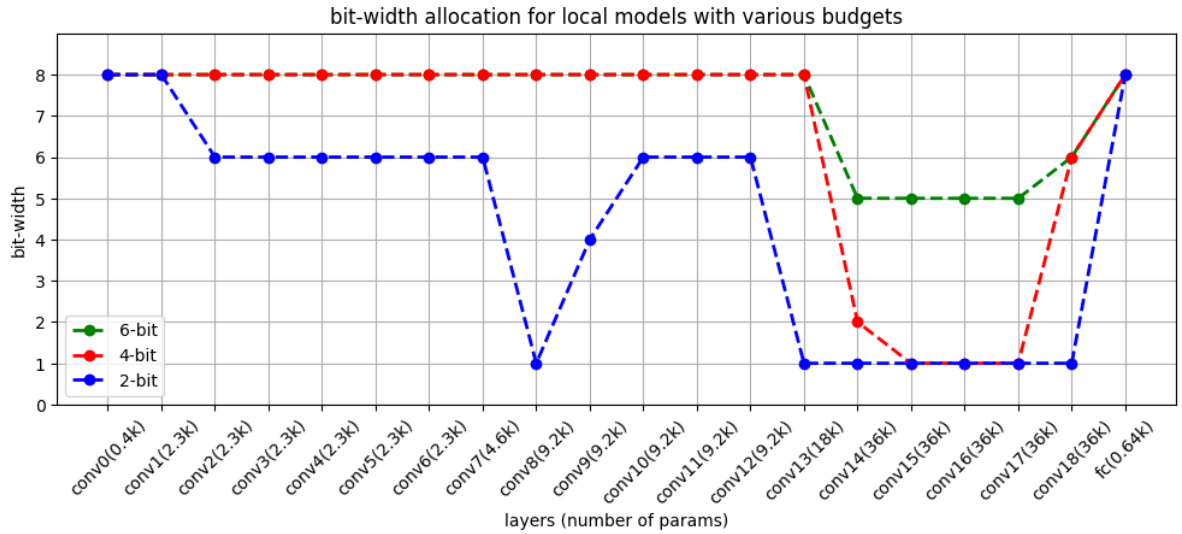


Figure 3. Bit-width allocation of local models (resnet20) assigned to three clients with various bit-width budget after 50 global rounds. The value of y axis denotes the bit-width assigned to a layer while the x axis indicates the layer's ID as well as the number of parameters in this layer.

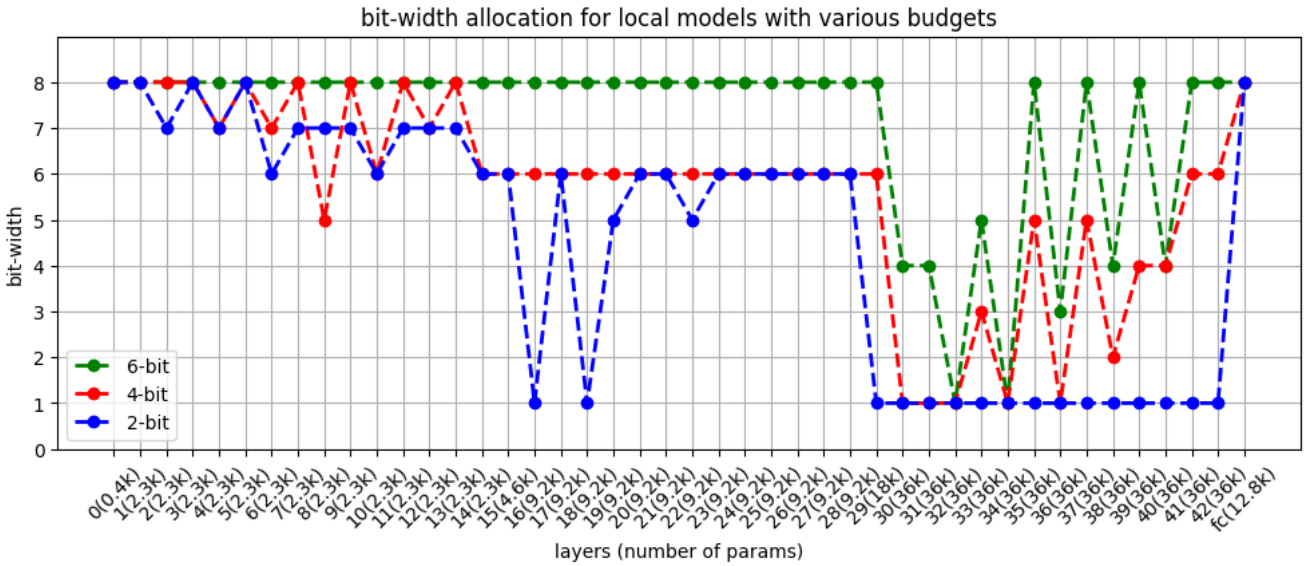
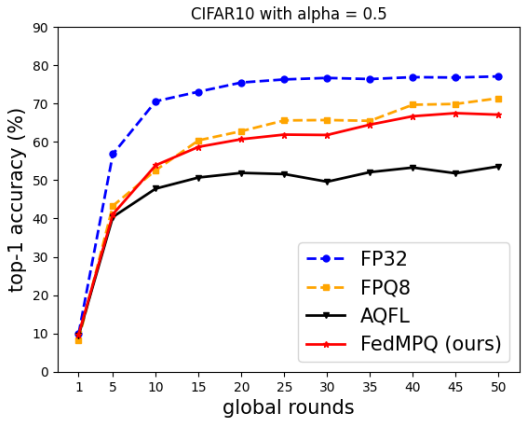
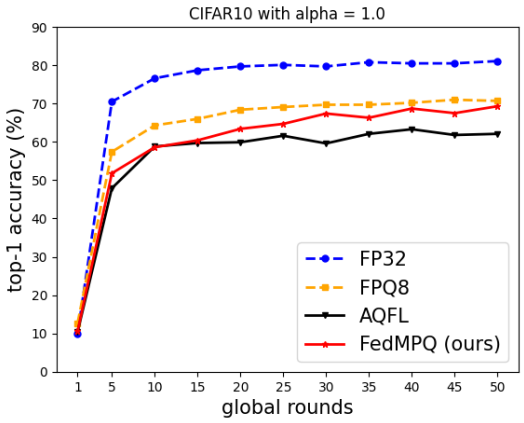


Figure 4. Bit-width allocation of local models (resnet44) assigned to three clients with various bit-width budget after 50 global rounds. The value of y axis denotes the bit-width assigned to a layer while the x axis indicates the layer's ID as well as the number of parameters in this layer. For clarity, we omit "conv" in the label of x axis.

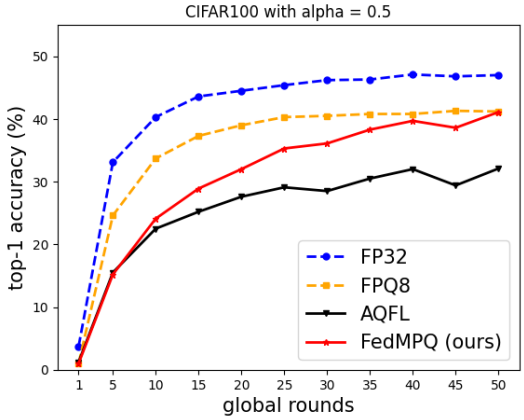


(a) $\alpha = 0.5$

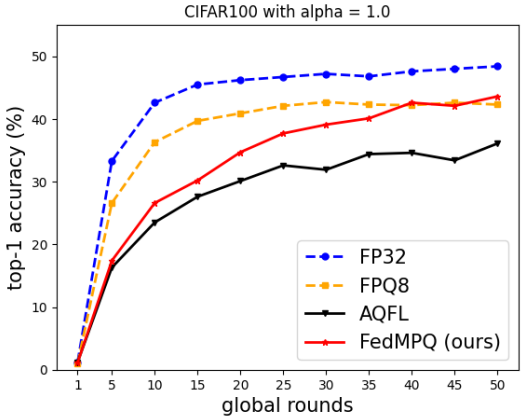


(b) $\alpha = 1$

Figure 5. Test accuracy vs. the number of global rounds. The number of clients is 10. The bit-width budget is as same as in the experiments in the main paper: {2,2,4,4,4,6,6,6,8,8}.

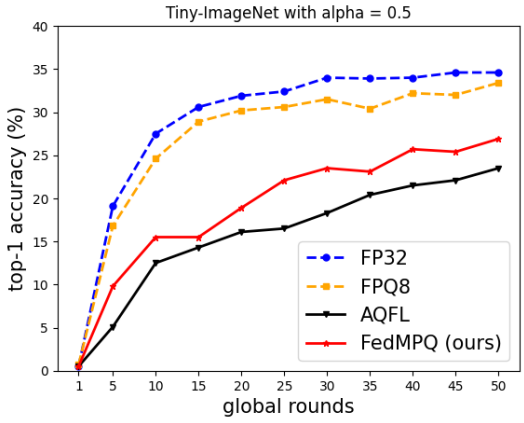


(a) $\alpha = 0.5$

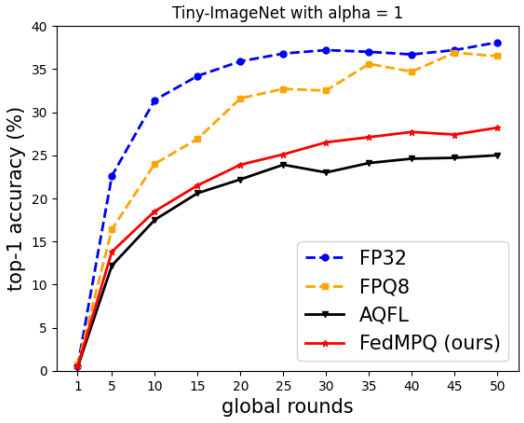


(b) $\alpha = 1$

Figure 6. Test accuracy vs. the number of global rounds. The number of clients is 10. The bit-width budget is as same as in the experiments in the main paper: {2,2,4,4,4,6,6,6,8,8}.



(a) $\alpha = 0.5$



(b) $\alpha = 1$

Figure 7. Test accuracy vs. the number of global rounds. The number of clients is 10. The bit-width budget is as same as in the experiments in the main paper: {2,2,4,4,4,6,6,6,8,8}.

References

[1] Mikhail Yurochkin, Mayank Agarwal, Soumya Ghosh, Kristjan Greenewald, Nghia Hoang, and Yasaman Khazaeni. Bayesian nonparametric federated learning of neural networks. In *International conference on machine learning*, pages 7252–7261. PMLR, 2019. 1