FedWSQ: Efficient Federated Learning with Weight Standardization and Distribution-Aware Non-Uniform Quantization
–Supplementary Document–

A. Technical Lemmas

This section introduces some technical lemmas that are useful to understand our main document.

Lemma 1. *Consider any vector $\mathbf{v} \in \mathbb{R}^d$. The mean subtraction of $\mathbf{v}$ is given by*

$$\begin{aligned}\bar{\mathbf{v}} &= \mathbf{v} - \left(\frac{1}{d}\mathbf{1}^T\mathbf{v}\right)\mathbf{1} \\ &= \left(\mathbf{I} - \frac{1}{d}\mathbf{1}\mathbf{1}^T\right)\mathbf{v} \\ &= \left(\mathbf{I} - \mathbf{P}_{\mathbf{1}}\right)\mathbf{v}\end{aligned} \tag{11}$$

where $\mathbf{I} \in \mathbb{R}^{d\times d}$ is the identity matrix, $\mathbf{1} \in \mathbb{R}^d$ is a vector whose elements are all ones, and $\mathbf{P}_{\mathbf{w}}$ represents the projection matrix onto the vector $\mathbf{w}$. Thus, mean subtraction is equivalent to projecting $\mathbf{v}$ onto $\mathrm{span}\{\mathbf{1}\}^{\perp}$. In other words, this projection removes the DC (constant) component from the given vector $\mathbf{v}$.

Lemma 2. *Consider any vector $\bar{\mathbf{v}} \in \mathbb{R}^d$ with zero mean. Normalization of $\bar{\mathbf{v}}$ using its standard deviation $\sigma(\bar{\mathbf{v}})$ is given by*

$$\begin{aligned}\tilde{\mathbf{v}} &= \frac{\rho}{\sigma(\bar{\mathbf{v}})}\bar{\mathbf{v}} \\ &= \frac{\rho\sqrt{d}}{\|\bar{\mathbf{v}}\|}\bar{\mathbf{v}}.\end{aligned} \tag{12}$$

Since $\bar{\mathbf{v}}$ is zero-centered, its standard deviation is given by $\sigma(\bar{\mathbf{v}}) = \sqrt{(\bar{\mathbf{v}}^T\bar{\mathbf{v}})/d}$.

Lemma 3. *Consider any vector $\mathbf{v} \in \mathbb{R}^d$. Let $\bar{\mathbf{v}}$ and $\tilde{\mathbf{v}}$ be its mean-subtracted and standardized versions, respectively. The derivative of $\tilde{\mathbf{v}}$ with respect to $\bar{\mathbf{v}}$ is then given by*

$$\begin{aligned}\frac{\partial\tilde{\mathbf{v}}}{\partial\bar{\mathbf{v}}} &= \frac{\rho}{\sigma(\bar{\mathbf{v}})}\left(\mathbf{I} - \frac{1}{d(\sigma(\bar{\mathbf{v}}))^2}\bar{\mathbf{v}}\bar{\mathbf{v}}^T\right) \\ &= \frac{\rho}{\sigma(\bar{\mathbf{v}})}\left(\mathbf{I} - \frac{1}{\|\bar{\mathbf{v}}\|^2}\bar{\mathbf{v}}\bar{\mathbf{v}}^T\right) \\ &= \frac{\rho}{\sigma(\bar{\mathbf{v}})}\left(\mathbf{I} - \mathbf{P}_{\bar{\mathbf{v}}}\right).\end{aligned} \tag{13}$$

Also, based on Lemma 1, the derivative of $\bar{\mathbf{v}}$ with respect to $\mathbf{v}$ is given by

$$\frac{\partial\bar{\mathbf{v}}}{\partial\mathbf{v}} = \left(\mathbf{I} - \mathbf{P}_{\mathbf{1}}\right). \tag{14}$$

Since $\sigma(\bar{\mathbf{v}}) = \sigma(\mathbf{v})$, by the chain rule, we can derive the gradient of a loss function $\mathcal{L}$ with respect to $\mathbf{v}$ as follows:

$$\frac{\partial\mathcal{L}}{\partial\mathbf{v}} = \frac{\rho}{\sigma(\bar{\mathbf{v}})}\left(\mathbf{I} - \mathbf{P}_{\mathbf{1}}\right)\left(\mathbf{I} - \mathbf{P}_{\bar{\mathbf{v}}}\right)\frac{\partial\mathcal{L}}{\partial\tilde{\mathbf{v}}}. \tag{15}$$

B. Derivation of Quantization Errors

In this section, we derive the expected quantization error, which measures the difference between the original LMPUs and their quantized values, where $p(x)$ represents a standard normal distribution. The error is formulated as

$$\mathbb{E}\left[(\Delta w - \Delta \bar{w})^2\right] = \sum_{r=0}^{R} \int_{u_r}^{u_{r+1}} (x - q_r)^2 p(x) dx \tag{16}$$

where q_r is the quantization level. To evaluate the integral, we expand the squared term as follows:

$$\begin{aligned}\int (x-q_r)^2 p(x) dx &= \frac{1}{\sqrt{2\pi}} \int (x-q_r)^2 e^{-\frac{x^2}{2}} dx \\ &= \frac{1}{\sqrt{2\pi}} \left(\underbrace{\int x^2 e^{-\frac{x^2}{2}} dx}_{P_1} \underbrace{-2q_r \int x e^{-\frac{x^2}{2}} dx}_{P_2} + \underbrace{q_r^2 \int e^{-\frac{x^2}{2}} dx}_{P_3} \right)\end{aligned} \tag{17}$$

We now calculate each term P_1, P_2, and P_3. Let $t = \frac{x}{\sqrt{2}}$, which transforms P_1 into

$$\begin{aligned}P_1 &= 2\sqrt{2} \int t^2 e^{-t^2} dt \\ &= -\sqrt{2} t e^{-t^2} + \sqrt{2} \int e^{-t^2} dt \quad \left(\because \int u dv = uv - \int v du \quad \text{where } u = t \text{ and } dv = t e^{-t^2} dt \right)\end{aligned} \tag{18}$$

The definite integral over the quantization boundaries is then given by

$$-\sqrt{2} \left[t e^{-t^2} \right]_{u_r/\sqrt{2}}^{u_{r+1}/\sqrt{2}} + \sqrt{2} \int_{u_r/\sqrt{2}}^{u_{r+1}/\sqrt{2}} e^{-t^2} dt = \left(u_r e^{-\frac{u_r^2}{2}} - u_{r+1} e^{-\frac{u_{r+1}^2}{2}} \right) + \sqrt{\frac{\pi}{2}} \left(\text{erf}\left(\frac{u_{r+1}}{\sqrt{2}} \right) - \text{erf}\left(\frac{u_r}{\sqrt{2}} \right) \right) \tag{19}$$

Also, we can evaluate the definite integral of P_2 over the qunatization boundaries as follows:

$$\begin{aligned}-2q_r \int_{u_r}^{u_{r+1}} x e^{-\frac{x^2}{2}} dx &= 2q_r \left[e^{-x^2} \right]_{u_r}^{u_{r+1}} \qquad \left(\because \int x e^{-\frac{x^2}{2}} dx = -e^{-\frac{x^2}{2}} \right) \\ &= 2q_r (e^{-\frac{u_{r+1}^2}{2}} - e^{-\frac{u_r^2}{2}})\end{aligned} \tag{20}$$

Finally, we can easily obtain the definite integral of P_3 over the qunatization boundaries by substituting $t = \frac{x}{\sqrt{2}}$, as follows:

$$\begin{aligned}q_r^2 \int_{u_r}^{u_{r+1}} e^{-\frac{x^2}{2}} dx &= \sqrt{2} q_r^2 \int_{u_r/\sqrt{2}}^{u_{r+1}/\sqrt{2}} e^{-t^2} dt \\ &= \sqrt{\frac{\pi}{2}} q_r^2 \left((\text{erf}\left(\frac{u_{r+1}}{\sqrt{2}} \right) - \text{erf}\left(\frac{u_r}{\sqrt{2}} \right) \right)\end{aligned} \tag{21}$$

Combining all the above derivations and unrolling the sum, we can obtain the final expression for the expected quantizaion error as follows:

$$\begin{aligned}\sum_{r=0}^{R} \int_{u_r}^{u_{r+1}} (x-q_r)^2 p(x) dx &= \sum_{r=0}^{R} \left\{ \frac{1}{\sqrt{2\pi}} (2q_r - u_{r+1}) e^{-\frac{u_{r+1}^2}{2}} - \frac{1}{\sqrt{2\pi}} (2q_r - u_r) e^{-\frac{u_r^2}{2}} \right. \\ &\qquad \left. + \frac{1}{2}(q_r^2 + 1) \left(\text{erf}\left(\frac{u_{r+1}}{\sqrt{2}} \right) - \text{erf}\left(\frac{u_r}{\sqrt{2}} \right) \right) \right\} \\ &= \frac{1}{2}(q_R^2 + 1) - \sqrt{\frac{2}{\pi}} q_1 e^{-\frac{u_1^2}{2}} - \frac{1}{2} q_1^2 \text{erf}\left(\frac{u_1}{\sqrt{2}} \right) \\ &\quad + \sqrt{\frac{2}{\pi}} \sum_{r=1}^{R-1} (q_r - q_{r+1}) e^{-\frac{u_{r+1}^2}{2}} + \frac{1}{2} \sum_{r=1}^{R-1} (q_r^2 - q_{r+1}^2) \text{erf}\left(\frac{u_{r+1}}{\sqrt{2}} \right).\end{aligned} \tag{22}$$

C. Experimental setup

Implementation details We follow most of the implementation setups and evaluation protocols in [1, 20, 38, 48]. The ResNet-18 architecture [11] is adopted as our backbone network. Consistent with [12] and common practice in FL, all BN layers in ResNet-18 are replaced with GN layers. Following the recommendation of Qiao *et al*. [35], WS is applied before each GN layer. All the models are trained from scratch by using the SGD optimizer with an initial learning rate of 0.1 and a weight decay of 0.001. For the proposed model, the learning rate is exponentially decayed at each communication round by a factor of 0.995. For the other models compared, we select the learning decay parameter from $\{0.995, 0.998, 1\}$ to attain the best accuracy. The global learning rate of FedAdam is set to 0.01, and that of the other methods is set to 1. Momentum is not used following the previous works [1, 20, 48], and gradient clipping is applied for learning stability. Unless otherwise noted, the number of local training epochs per round is set to 5, with the batch size adjusted so that each local epoch consists of 10 iterations. The hyper-parameter of WS is set to $\rho = 0.001$. The source code is implemented by using the PyTorch framework [34] on NVIDIA RTX 4090 GPUs. We set the number of local training epochs to 5. The batch size for local updates is adjusted so that each local epoch has 10 iterations (*i.e*., 50 iterations during a single communication round).

Hyper-parameter selection We adopt the hyper-parameter settings of the baseline methods suggested in [20, 38]. Table A summarizes the hyper-parameter settings we used, with the notations consistent with the original papers.

Table A. Summary of hyper-parameter selection

Method	Hyper-parameters
FedProx [27]	$\mu = 0.001$
FedAvgM [13]	$\beta = 0.4$
FedADAM [36]	$\tau = 0.001$, $\beta_1 = 0.9$, $\beta_2 = 0.99$
FedDyn [1]	$\alpha = 0.1$
FedMLB [21]	$\tau = 1$, $\lambda_1 = 1$, $\lambda_2 = 1$
FedLC [52]	$\tau = 1$
FedNTD [26]	$\tau = 1$, $\beta = 0.3$
FedDecorr [39]	$\beta = 0.01$
FedRCL [38]	$\tau = 0.05$, $\beta = 1$, $\lambda = 0.7$
FedACG [20]	$\beta = 0.001$, $\lambda = 0.85$

QLs of NUQ methods Figure A provides a comparative visualization of the QLs adopted by different NUQ methods. The histogram illustrates the empirical distribution of LMPUs with the standard normal distribution curve. As shown in the figure, the proposed DANUQ places QLs more adaptively based on the statistical structure of LMPUs, leading to improved quantization efficiency.

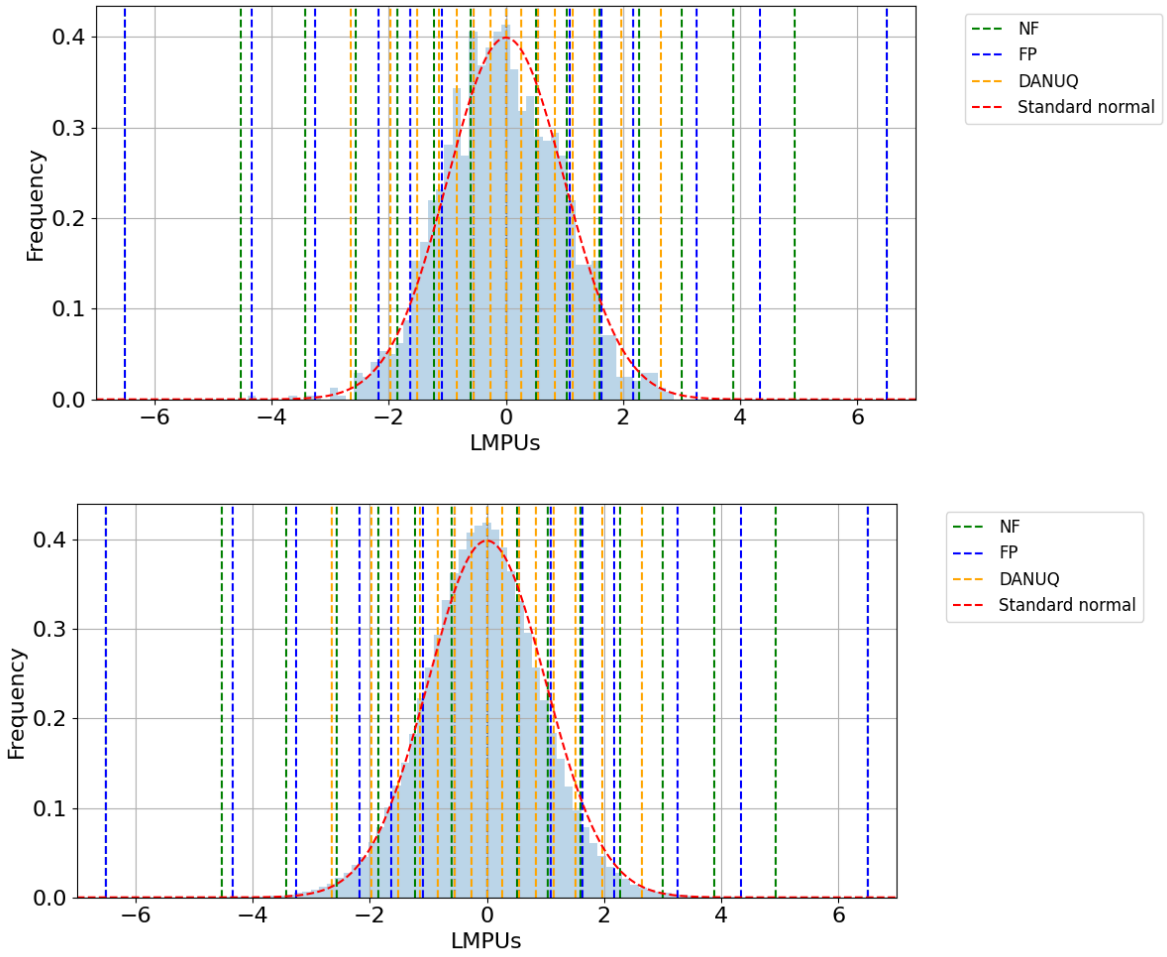
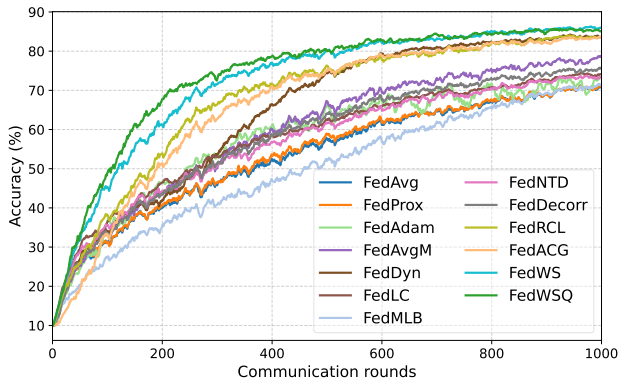


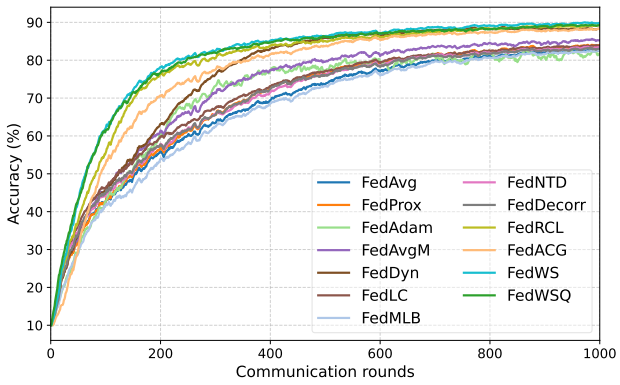
Figure A. Visualization of QLs used in different NUQ methods. The histogram represents the empirical distribution of LMPUs in the 1st and 3rd ResNet-18 blocks, and the red curve denotes the standard normal distribution. The vertical dashed lines indicate the QLs chosen by different methods, NF, FP, and the proposed DANUQ, where the 4-bit representation is used.

D. Convergence plot evaluated on various federated learning scenarios

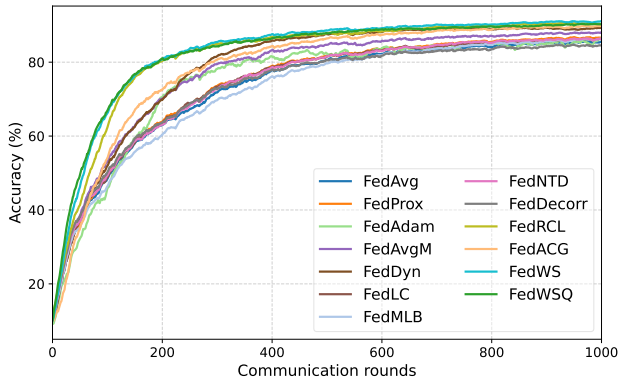
Figures B-D present the convergence plots of various FL methods on CIFAR-10, CIFAR-100, and Tiny-ImageNet, for *i.i.d* and non-*i.i.d.* data distributions with $\alpha \in \{0.05, 0.1, 0.3, 0.6\}$, using a participation rate of 5% over 100 distributed clients. As shown in the figures, FedWSQ consistently enhances the FL performance of conventional methods, outperforming those of state-of-the-art FL approaches.



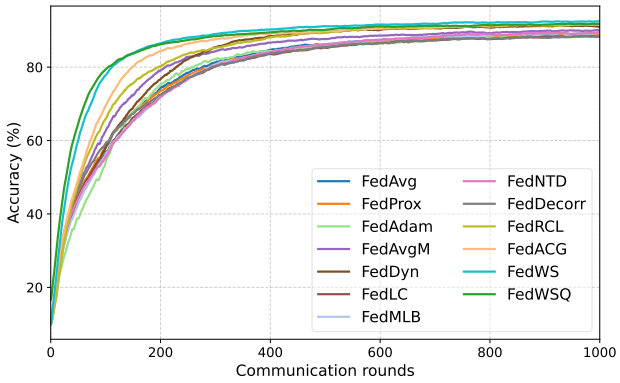
(a) $\alpha = 0.1$, 5% participation over 100 clients



(b) $\alpha = 0.3$, 5% participation over 100 clients

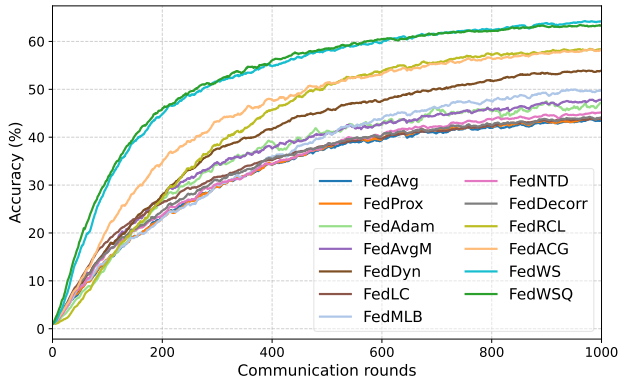


(c) $\alpha = 0.6$, 5% participation over 100 clients

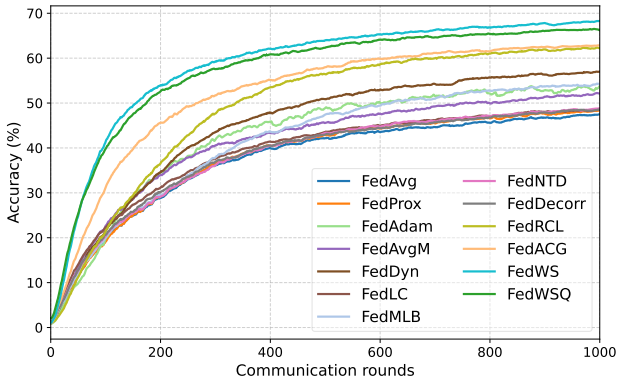


(d) *i.i.d.*, 5% participation over 100 clients

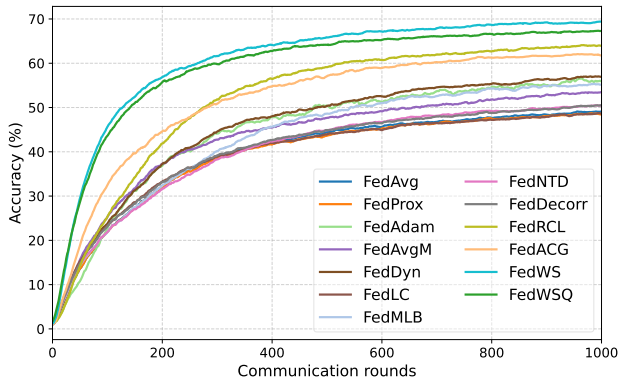
Figure B. Convergence plots of our FedWS and FedWSQ compared to conventional methods on CIFAR-10 with 5% participation over 100 clients under varying Dirichlet parameters.



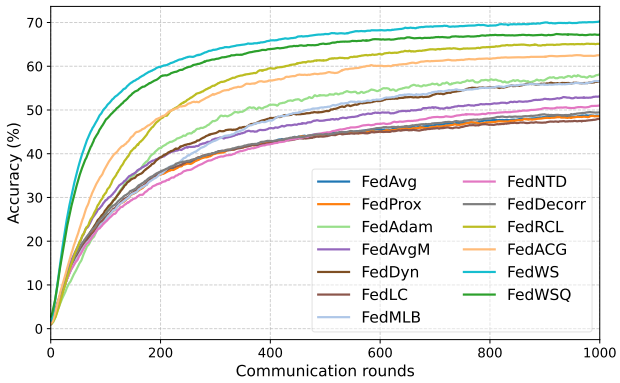
(a) $\alpha = 0.1$, 5% participation over 100 clients



(b) $\alpha = 0.3$, 5% participation over 100 clients

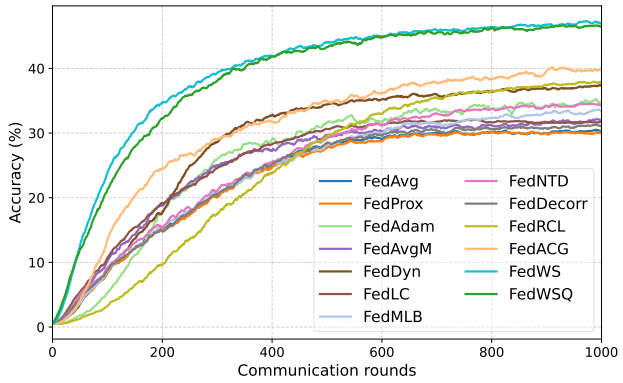


(c) $\alpha = 0.6$, 5% participation over 100 clients

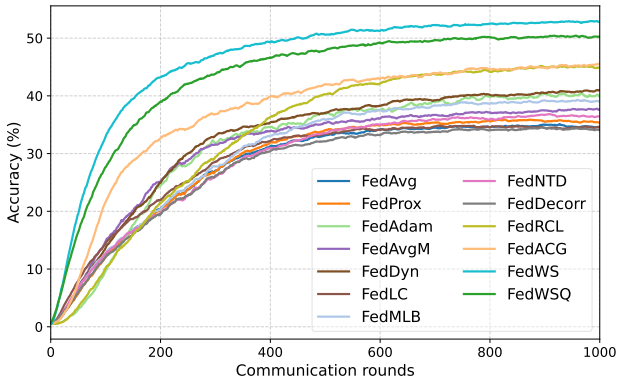


(d) *i.i.d.*, 5% participation over 100 clients

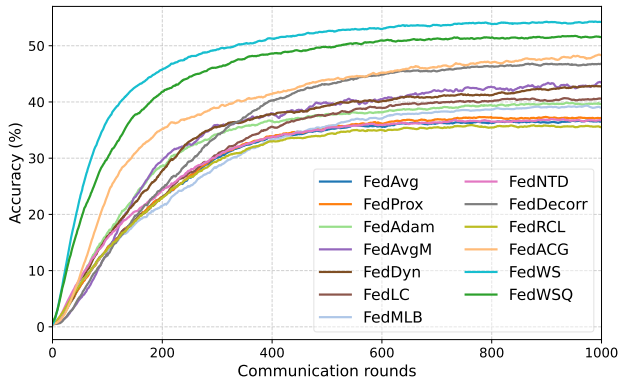
Figure C. Convergence plots of our FedWS and FedWSQ compared to conventional methods on CIFAR-100 with 5% participation over 100 clients under varying Dirichlet parameters.



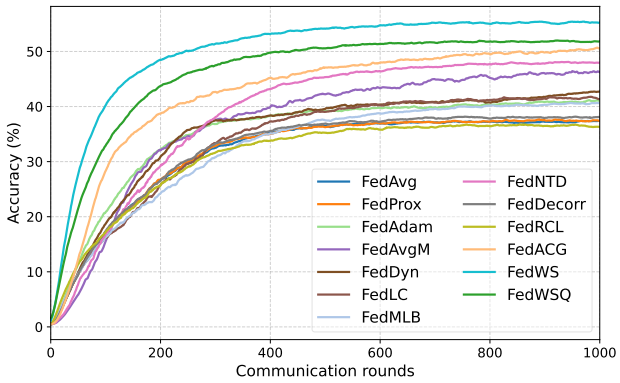
(a) $\alpha = 0.1$, 5% participation over 100 clients



(b) $\alpha = 0.3$, 5% participation over 100 clients



(c) $\alpha = 0.6$, 5% participation over 100 clients



(d) *i.i.d.*, 5% participation over 100 clients

Figure D. Convergence plots of our FedWS and FedWSQ compared to conventional methods on Tiny-ImageNet with 5% participation over 100 clients under varying Dirichlet parameters.