Bitwidth Heterogeneous Federated Learning with Progressive Weight Dequantization

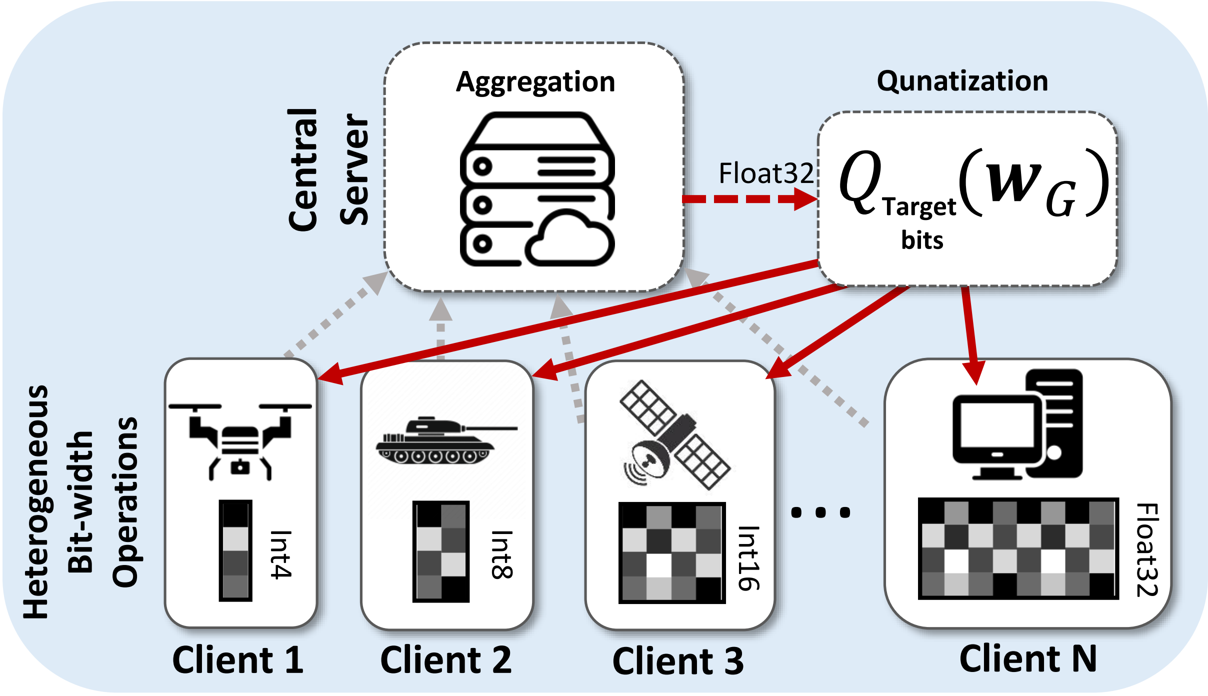
Jaehong Yoon*, Geon Park*, Wonyong Jeong, Sung Ju Hwang

(*: equal contribution)

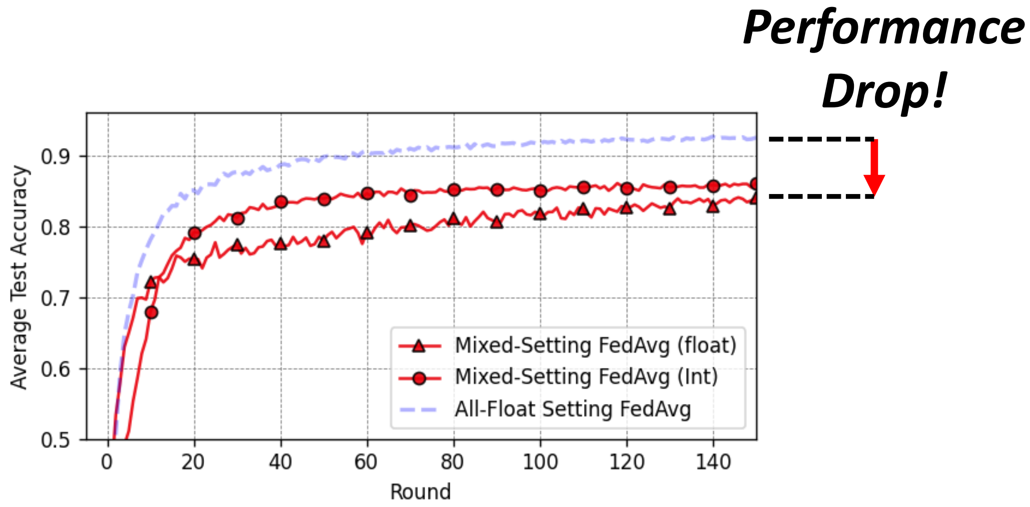
KAIST, South Korea

Bitwidth Heterogeneous Federated Learning

Embedded devices participating in federated learning may have to operate using **different bit-widths** for computation and storage during **training and inference**.



Bitwidth Heterogeneous Federated Learning (BHFL)



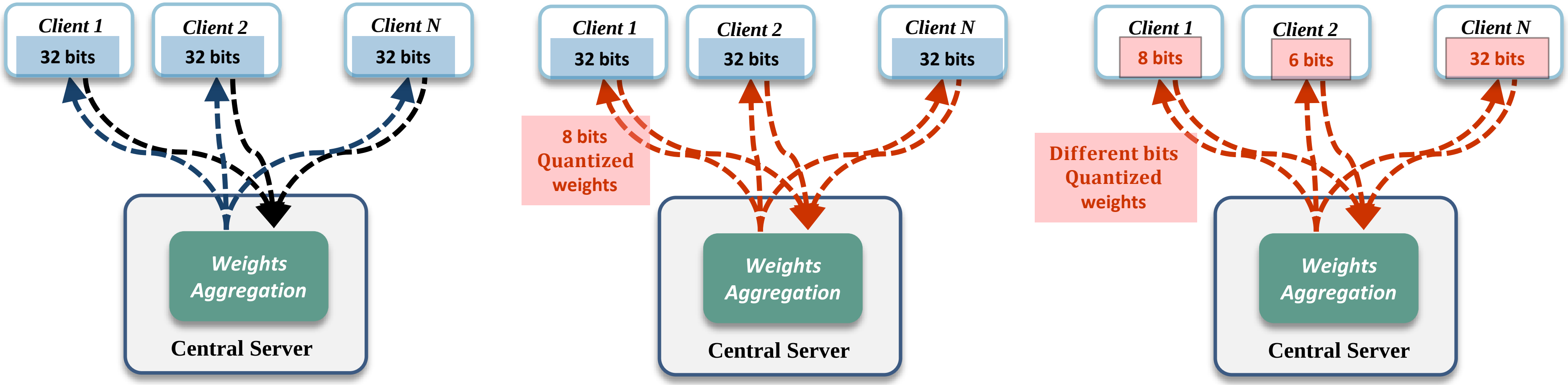
*FedAvg Performance Using Float32 * 5 & Int8 * 5 clients*

Related Works

Embedded devices participating in federated learning may have to operate using **different bit-widths** for computation and storage during **training and inference**.

METHODS	FL Type	$\text{Bits}_{\text{Server}}$	$\text{Bits}_{\text{Clients}}$	$\text{Bits}_{\text{Uplink}}$	$\text{Bits}_{\text{Downlink}}$	Communication
FEDAVG (McMahan et al., 2017)	FL	Float32	Float32	Float32	Float32	Weights
FEDPROX (Li et al., 2018)	FL	Float32	Float32	Float32	Float32	Weights
FEDPAQ (Reisizadeh et al., 2020)	QPC[1]	Float32	Float32	Target bits	Float32	$Q_{\text{Target\_bits}}(\text{Diffs})$[2]
FEDCOM (Haddadpour et al., 2021)	QPC	Float32	Float32	Target bits	Float32	$Q_{\text{Target\_bits}}(\text{Diffs})$
FEDCOMGATE (Haddadpour et al., 2021)	QPC	Float32	Float32	Target bits	Float32×2	$Q_{\text{Target\_bits}}(\text{Diffs})$
PROWD (OURS)	BHFL	Float32	Client-specific	Client-specific	Client-specific	W_Q[3]

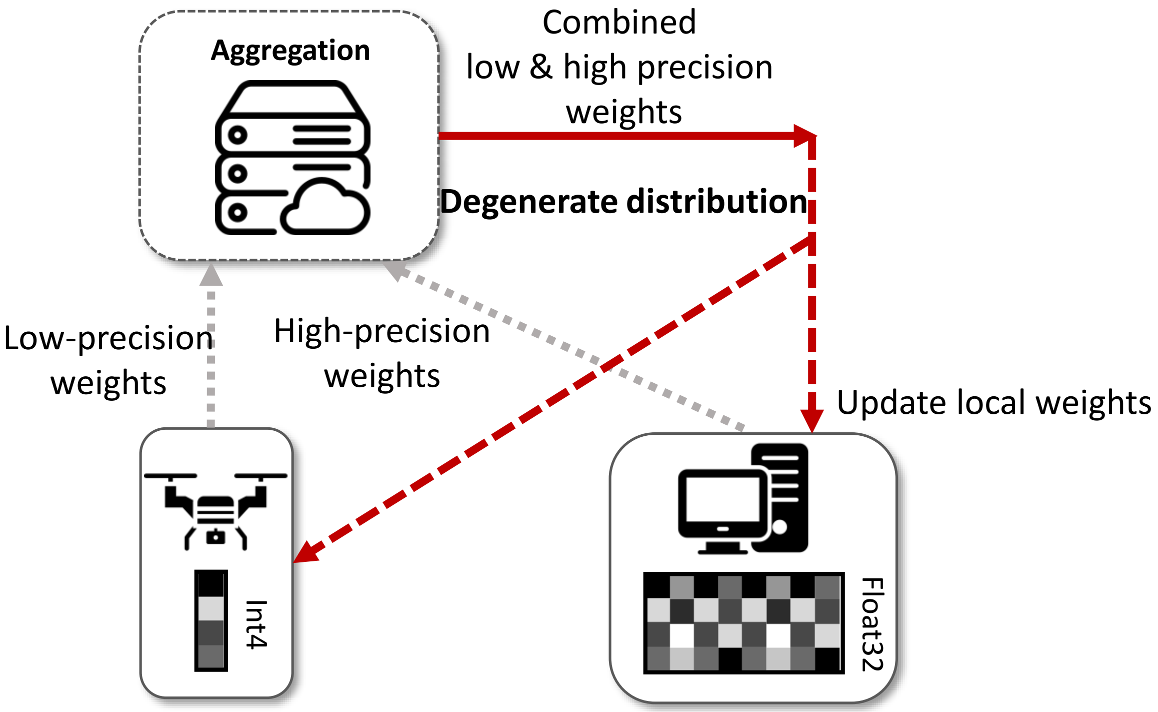
[1] Quantized Parameter Communication (QPC)
[2] $Q_{\text{Target\_bits}}(w)$: A function quantizes input w to the target bitwidth
[3] Weights obtained from clients' bit-dependent training

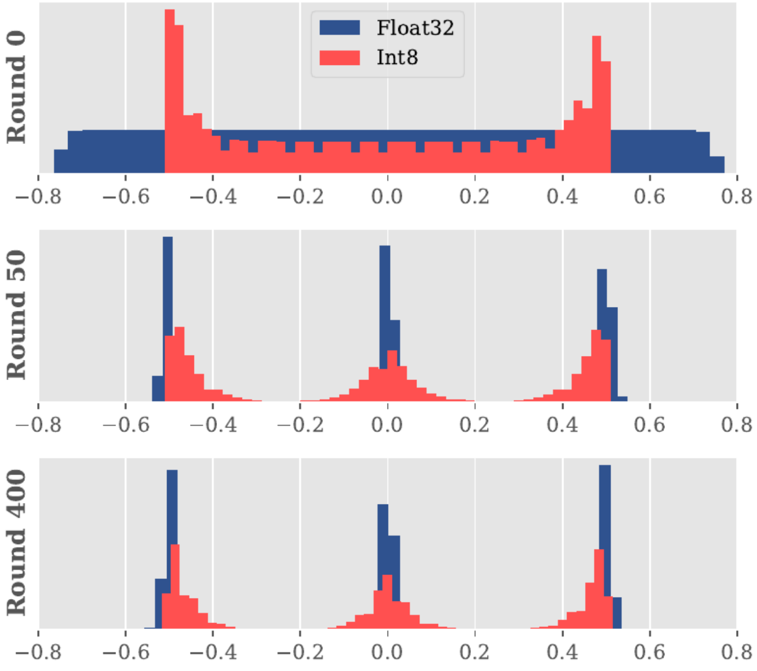


Federated Learning **Quantized Parameter Communication (QFC)** **Bitwidth Heterogeneous Federated Learning (BHFL)**

Challenges

We observed that the weights from low-bitwidth clients **cannot easily be combined** with high-bitwidth weights due to **distributional incompatibility**.





Degeneration of high-precision weights over naïve BHFL rounds

Methodology Overview

The server **dequantizes** the received mixed-bitwidth models into float32, before aggregating them and broadcasting to each client.

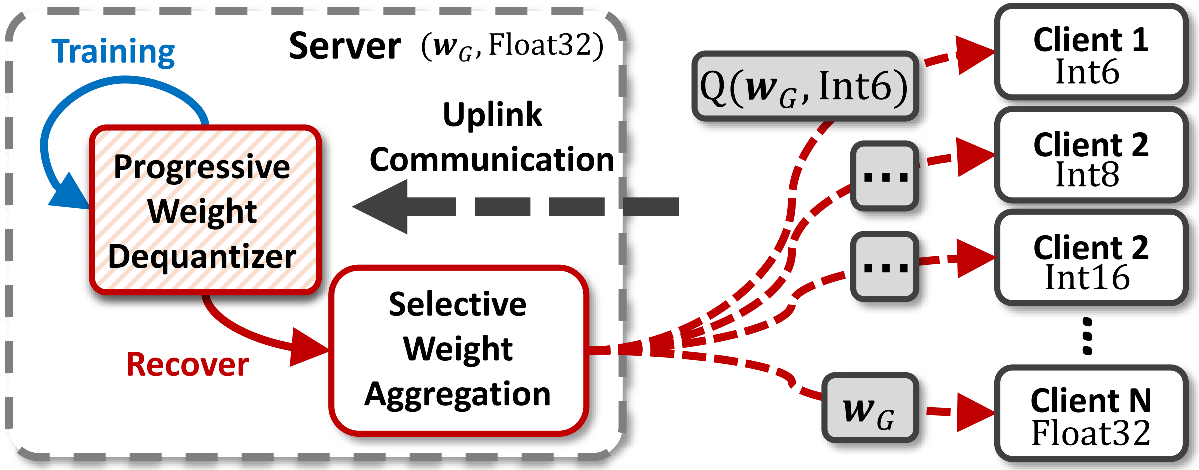
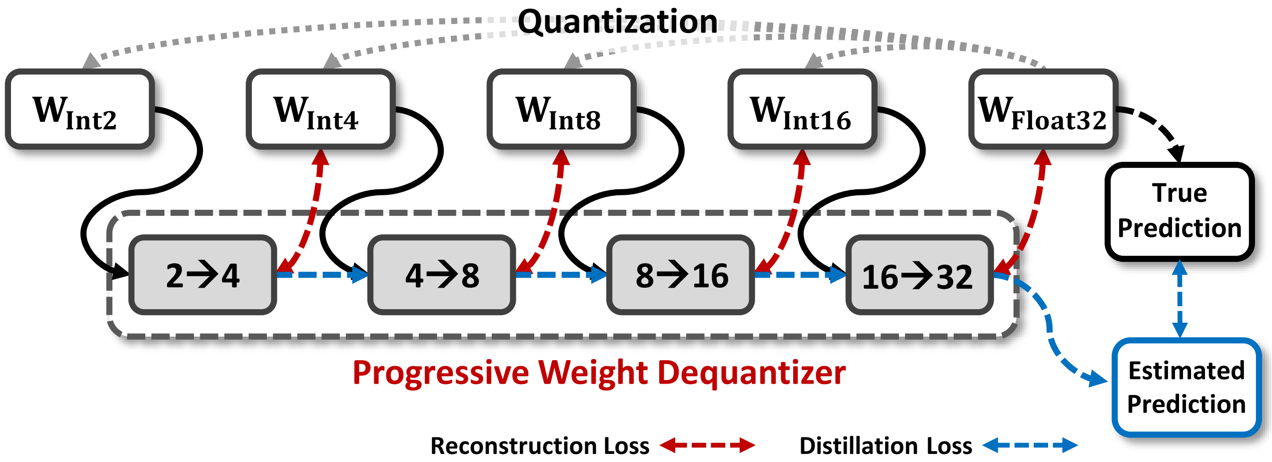


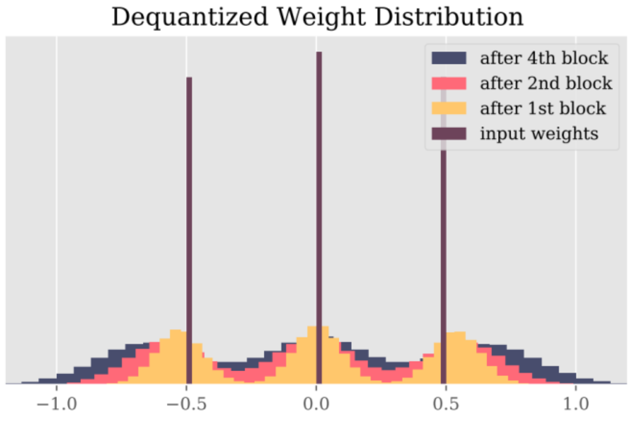
Illustration of our ProWD framework

The **Progressive Weight Dequantizer** and **Selective Weight Aggregation** can alleviate the distribution disparity problem.

Progressive Dequantizer

Each block in our dequantizer **progressively converts** a set of weights into a higher bitwidth.





Reconstruction of Low-bitwidth Weights

$$L_{recon} = \sum_{j=0}^{K-1} ||\overset{\text{Higher bits}}{q_{\pi_{j+1}}} - \phi^{\pi_j \to \pi_{j+1}}\left(\overset{\text{Lower bits}}{q_{\pi_j}}; \theta_j\right)||_1, \quad L_{distill} = -SIM\left(\overset{\text{Source}}{f(u; w)}, \overset{\text{Target}}{f\left(u; \phi^{0\to k}(q_{\pi_0}; \Theta)\right)}\right)$$

ProWD Objective

$$\mathcal{L} = \mathcal{L}_{recon} + \lambda \mathcal{L}_{distill}$$

Reconstruction error Distillation loss

The dequantizer is **fine-tuned periodically** during the BHFL process using the weights sent from the clients, and a tiny server-side data buffer.

Selective Weight Aggregation

In order to improve the stability of the training process, before aggregating, we **impose a mask** on the clients' weights based on the **relevancy** among the weights.

$$\boldsymbol{c}^{*} = \underset{\boldsymbol{c}}{\operatorname{argmax}} \frac{(\boldsymbol{c} \odot \Delta\overline{\boldsymbol{w}}_{\text{Low}})^{\top} \Delta\overline{\boldsymbol{w}}_{\text{High}}}{\|\boldsymbol{c} \odot \Delta\overline{\boldsymbol{w}}_{\text{Low}}\| \|\Delta\overline{\boldsymbol{w}}_{\text{High}}\|}, \text{ s.t. } |\boldsymbol{c}^{*}| \leq \tau$$

Mask Computation

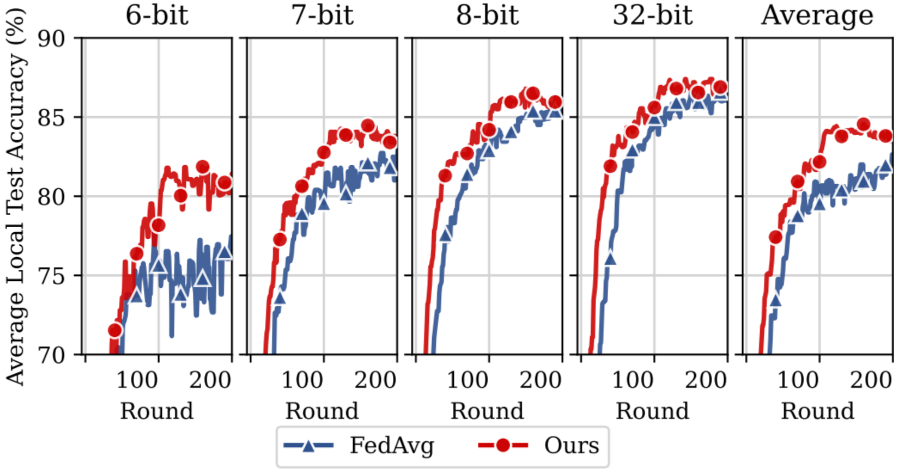
$$\boldsymbol{w}_G \leftarrow \frac{1}{N} \sum_{n=1}^{N} \boldsymbol{c}_n \odot \boldsymbol{w}_n$$

Selective weight aggregation

Experiments

ProWD consistently **outperforms** strong baselines such as FedAvg, FedProx, FedCOM, and FedGroupedAvg.

	Int8 50%	Float32 50%		Int8 80%	Float32 20%	
	8bits Accuracy	32bits Accuracy	**Averaged accuracy**	8bits Accuracy	32bits Accuracy	**Averaged accuracy**
Local Training	69.59	75.82	72.71	69.39	76.41	70.79
FedAvg	**76.88**	76.23	**76.56**	**77.43**	74.64	**76.87**
FedProx	71.16	69.28	70.22	69.60	66.28	68.94
FedCOM	75.37	77.69	76.53	73.60	**80.73**	75.03
GroupedAvg	61.85	**85.08**	73.46	71.76	78.07	73.02
ProWD (Ours)	**82.87**	**85.99**	**84.43**	**79.23**	**81.26**	**79.63**



Conclusion

- We propose a **novel yet practical scenario** for FL where the participating devices operate using different bit-widths, called BHFL.
- We propose a novel framework, *ProWD,* for bit-heterogeneous FL, which can tackle these issues based on progressive weight dequantization and selective weight aggregation.
- We evaluate ProWD on realistic BHFL scenarios and show the effectiveness of our framework.

Thank you

Contact Information:

Jaehong Yoon jaehong.yoon@kaist.ac.kr
Geon Park geon.park@kaist.ac.kr
Sung Ju Hwang sjhwang82@kaist.ac.kr

KAIST

MLAI
Machine Learning & Artificial Intelligence