

AnycostFL: Efficient On-Demand Federated Learning over Heterogeneous Edge Devices

Peichun Li^{*,†}, Guoliang Cheng^{*}, Xumin Huang^{*,†}, Jiawen Kang^{*}, Rong Yu^{*}, Yuan Wu[†], and Miao Pan[‡]

^{*}School of Automation, Guangdong University of Technology, Guangzhou, China

[†]State Key Laboratory of Internet of Things for Smart City, University of Macau, Macau, China

[‡]Department of Electrical and Computer Engineering, University of Houston, Houston, USA

Email: peichun@mail2.gdut.edu.cn, guoliang_cheng@126.com, huangxu_min@163.com,

{kavinkang, yurong}@gdut.edu.cn, yuanwu@um.edu.mo, mpan2@uh.edu

Abstract—In this work, we investigate the challenging problem of on-demand federated learning (FL) over heterogeneous edge devices with diverse resource constraints. We propose a cost-adjustable FL framework, named AnycostFL, that enables diverse edge devices to efficiently perform local updates under a wide range of efficiency constraints. To this end, we design the model shrinking to support local model training with elastic computation cost, and the gradient compression to allow parameter transmission with dynamic communication overhead. An enhanced parameter aggregation is conducted in an element-wise manner to improve the model performance. Focusing on AnycostFL, we further propose an optimization design to minimize the global training loss with personalized latency and energy constraints. By revealing the theoretical insights of the convergence analysis, personalized training strategies are deduced for different devices to match their locally available resources. Experiment results indicate that, when compared to the state-of-the-art efficient FL algorithms, our learning framework can reduce up to 1.9 times of the training latency and energy consumption for realizing a reasonable global testing accuracy. Moreover, the results also demonstrate that, our approach significantly improves the converged global accuracy.

Index Terms—Federated learning, edge intelligence, mobile computing, resource management.

I. INTRODUCTION

Federated learning (FL) is an emerging distributed learning paradigm that enables multiple edge devices to train a common global model without sharing individual data [1]. This privacy-friendly data analytics technique over massive devices is envisioned as a promising solution to realize pervasive intelligence [2]. However, in many real-world application areas, mobile devices are often equipped with different local resources, which raises the emerging challenges for locally on-demand training [3]. Given different local resources status (e.g., computing capability and communication channel state) and personalized efficiency constraints (e.g., latency and energy), it is crucial to customize training strategies for heterogeneous edge devices.

We perform an in-depth analysis on the time delay and the energy consumption for performing the local model updates at edge devices. Specifically, we evaluate and record the cost of local training on three different NVIDIA Jetson family platforms (i.e., Nano, NX AGX, and Xavier AGX) under different channel states (i.e., good, medium, and poor). On the one hand, we observe that the learning efficiency differs significantly

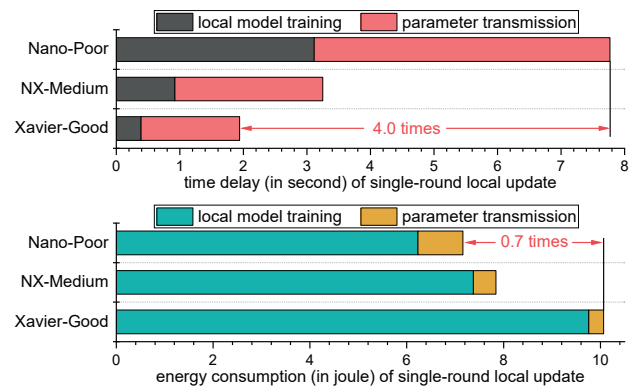


Fig. 1. The time delay (top) and energy consumption (bottom) of single-round local update on different hardware platforms with varying communication conditions.

with diverse learning scenarios. As shown in Fig. 1, the single-epoch training on Nano with poor communication condition consumes about 4.0 times training latency than that of Xavier AGX with good communication condition, while its energy consumption is about 0.7 times less than the latter one's. On the other hand, we observe that the bottlenecks of latency and energy are induced by parameter transmission and local model training, respectively.

The above observations provide insights for a proper design of the on-demand FL system. To handle the resource heterogeneity, it is suggested to alleviate the energy and the latency cost of the local device. More importantly, the computation and communication costs should be jointly reduced to achieve efficient local training. In the literature, most existing studies either employ resource allocation and device scheduling to mitigate the system cost [4]–[10], or design gradient compression to accelerate the parameter transmission procedure [11]–[17]. The former method inherits the ideas of traditional design for mobile edge systems and takes no account of the optimization for neural networks, while the latter overlooks the computation cost of local model training.

In this paper, we propose “anycost” FL, named AnycostFL, to break the latency and energy bottlenecks for on-demand distributed training over heterogeneous edge devices. Our goal is to develop a cost-adjustable FL framework that enables edge devices to perform local updates under diverse learning scenarios. To this end, we first design the model shrinking and

gradient compression to enable adaptive local updates with different computation and communication costs. Meanwhile, an enhanced parameter aggregation scheme is proposed to fuse the knowledge of the local updates. Following that, we investigate the on-demand learning of AnycostFL by regulating the local model structure, gradient compression policy and computing frequency under personalized latency and energy constraints. However, customizing training strategy for different learning scenarios is a non-trivial task, since how the global accuracy is affected by the local model structure and compression rate is still unknown. To address this issue, we theoretically reveal the convergence insights of our framework, which are further leveraged to guide optimization analysis. Finally, the optimal training strategy is derived for each device according to its locally available resource.

Our main contributions are summarized as follows.

- We propose a novel FL framework, named AnycostFL, that enables the local updates with elastic computation cost and communication overhead.
- We theoretically present the optimal aggregation scheme and convergence analysis for AnycostFL.
- We investigate the on-demand training problem of AnycostFL, and the optimal training strategy is devised to adapt the locally available resource.
- Extensive experiments indicate that the proposed AnycostFL outperforms the state-of-the-art efficient FL methods in terms of resource utilization and learning accuracy.

The remainder of this paper is organized as follows. Section II describes related studies. In Section III, we detail the main operations of AnycostFL to fulfill the single-round training. The problem formulation, theoretical analysis and the corresponding solution are provided in Section IV. The experiment evaluations are presented in Section V, and we finally conclude the paper in Section VI and discuss the future directions.

II. RELATED WORK

Resource Management Methods. Resource management methods aim to reduce the FL system cost by arranging the local and system resources. Resource allocation methods employ frequency scheduling [18], transmission power control [19], and bandwidth allocation [20] to balance the cost of local training. Recent device selection methods directly exclude those weak devices with poor computation or communication capabilities to accelerate the convergence time [21]–[23]. Besides, topology-aware management is another very effective method to mitigate the network throughput [18], [24], [25]. However, these methods inherit the ideas of the efficient design for traditional mobile systems and overlook the optimization of neural networks.

Neuron-aware Techniques. Neuron-aware techniques focus on revealing the black box of neural networks to improve the training efficiency of the FL system. Early gradient compression utilizes sparsification [11], [26], and quantization [14], [27], [28] to reduce the transmission cost of FL system. In addition, feature maps fusion and knowledge distillation can be carried out to improve the information aggregation [29], [30].

Besides, FedMask proposes to train a personalized mask for each device to improve the test accuracy on the local dataset [31]. Recently, model structure pruning enables multiple devices with different model architectures to train a shared global model [32], [33]. Such methods can reduce the cost of local training, but how to customize optimal training strategies (e.g., gradient compression and model pruning policy) for different learning scenarios is still unknown.

III. TRAINING WITH ANYCOSTFL

In this section, we first outline the overall design of AnycostFL. Next, we detail the key techniques of our framework, including elastic model shrinking (EMS), flexible gradient compression (FGC), and all-in-one aggregation (AIO).

A. Outline of AnycostFL

We consider a generic application scenario of FL with a set of I edge devices $\mathcal{I} = \{1, 2, \dots, I\}$. We use \mathcal{D}_i to denote the local training data of the device i , and $\mathcal{D} = \cup_{i=1}^I \mathcal{D}_i$ indicates the global data. Let $F_i(\mathbf{w}) = \ell(\mathbf{w}, \mathcal{D}_i)$ represent the local training loss of device i with respect to model weight \mathbf{w} , where $\ell(\cdot, \cdot)$ is the predetermined loss function. The objective of the FL system is to minimize the following global loss function

$$F(\mathbf{w}) \triangleq \sum_{i=1}^I \frac{|\mathcal{D}_i|}{|\mathcal{D}|} F_i(\mathbf{w}), \quad (1)$$

where $|\mathcal{D}_i|$ is the size of \mathcal{D}_i . Given the specified learning task, the original training workload of single sample W and the data size of uncompressed gradient S can be empirically measured.

As shown in Fig. 2(a), to reduce the computational complexity of the local model training and the communication cost of gradient update transmission, we propose AnycostFL with two device-side techniques, i.e., model shrinking and gradient compression. At the t -th global iteration of AnycostFL, the device i is enabled to adjust its training workload and gradient size as $W_{t,i} = \alpha_{t,i}W$ and $S_{t,i} = \beta_{t,i}S$, respectively. Here, $\alpha_{t,i} \in (0, 1]$ and $\beta_{t,i} \in (0, 1]$ are defined as the model shrinking factor and the gradient compression rate, respectively. The training procedure of AnycostFL is summarized as follows.

1) *Elastic local training:* At the t -th global round, the device i downloads the latest global model \mathbf{w}_t from the parameter server. With the pre-calculated model shrinking factor $\alpha_{t,i}$, the specialized sub-model $\mathbf{w}_{t,i}^\alpha = \text{shrink}(\mathbf{w}_t, \alpha_{t,i})$ can be efficiently derived, where function $\text{shrink}(\cdot, \cdot)$ indicates the operations for model shrinking. Then, the local training is conducted with sub-model $\mathbf{w}_{t,i}^\alpha$ and local data \mathcal{D}_i , and the updated local sub-model $\mathbf{w}_{t+1,i}^\alpha$ is obtained. Furthermore, the local gradient update can be acquired as $\mathbf{u}_{t,i} = \mathbf{w}_{t,i}^\alpha - \mathbf{w}_{t+1,i}^\alpha$.

2) *Flexible gradient upload:* To further reduce the uplink traffic, the local device i is motivated to compress the gradient update $\mathbf{u}_{t,i}$ before the parameter transmission. With the given compression rate $\beta_{t,i}$, the compressed gradient update $\tilde{\mathbf{u}}_{t,i} = \text{cmprrs}(\mathbf{u}_{t,i}, \beta_{t,i})$ is uploaded to the server, where $\text{cmprrs}(\cdot, \cdot)$ is the function for gradient compression.

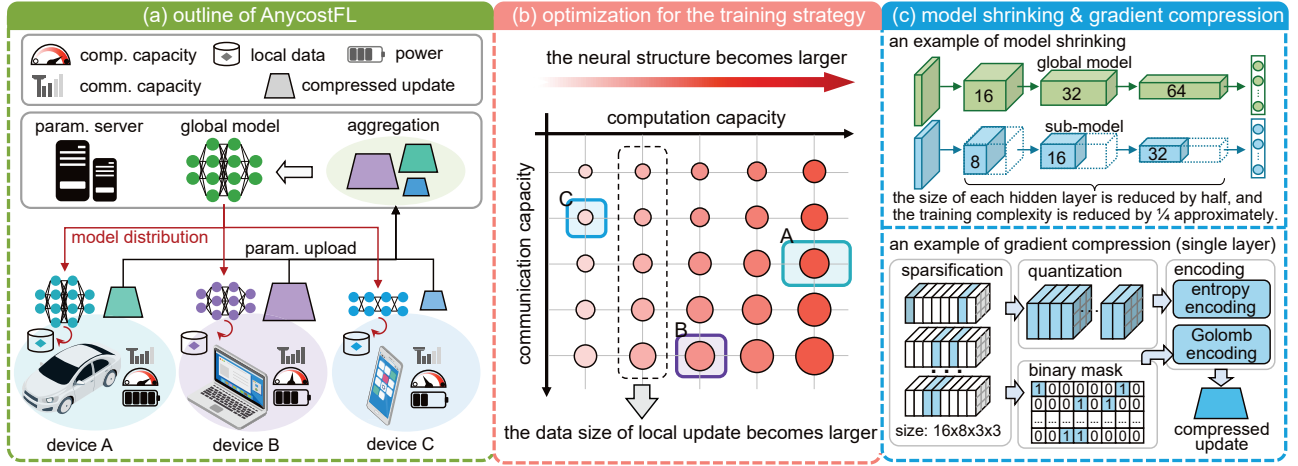


Fig. 2. **left:** AnycostFL over heterogeneous edge devices. **middle:** the neural structure and gradient compression strategies are customized for diverse devices according to their locally available resources; the darker color indicates the higher computing complexity for training and the larger marker size denotes the larger data size of the local update. **right:** illustrations of the model shrinking for the local model and the gradient compression for the local update.

3) *Parameter aggregation:* The server collects the compressed local updates $\{\tilde{\mathbf{u}}_{t,i}\}_{\forall i}$ with different shrinking factors $\{\alpha_{t,i}\}_{\forall i}$ and compression rates $\{\beta_{t,i}\}_{\forall i}$. After that, the global update is calculated by $\tilde{\mathbf{u}}_t = \text{aioagg}(\{\tilde{\mathbf{u}}_{t,i}\}_{\forall i})$, where $\text{aioagg}(\cdot)$ is the server-side all-in-one aggregation. Then, the updated global model is computed as $\mathbf{w}_{t+1} = \mathbf{w}_t - \tilde{\mathbf{u}}_t$.

After the T -round training of the above three-step iterations, the final global model \mathbf{w}_T is obtained. Before introducing how to customize the values of $\{\alpha_{t,i}\}_{\forall i}$ and $\{\beta_{t,i}\}_{\forall i}$ in Section IV, we illustrate the details of model shrinking, gradient compression and update aggregation in the rest of this section.

B. Elastic Model Shrinking

We aim to derive the sub-model $\mathbf{w}_{t,i}^\alpha$ with training complexity of $\alpha_{t,i}W$ from global model \mathbf{w}_t by reducing the width of the global model. The shrinking operations work as follows.

1) *Server-side channel sorting:* To avoid incurring extra memory cost for the edge devices, the server first sorts the channels of the latest global model before the model distribution. Given one layer of the weight of the global model, the server sorts the output channels in the current layer in descending order according to their values of L2 norm, and meanwhile, the input channels of the next layer should be sorted accordingly in the same order to maintain the permutation invariance of the whole model [34].

2) *Layer-wise uniform shrinking:* Next, the server broadcasts the weight of each layer of the global model in a channel-by-channel manner. Instead of downloading the full global model, each device only receives those important parameters from the global model to assemble the local sub-model. Here, we utilize the fixed shrinking ratio for each layer in the same sub-model. Empirically, given model shrinking factor $\alpha_{t,i}$, we can reduce the size of the hidden layer by $\sqrt{\alpha_{t,i}}$ to acquire the sub-model. For example, as shown in Fig. 2(c), when shrinking a global model with hidden sizes of $\{16, 32, 64\}$ under $\alpha_{t,i} = \frac{1}{4}$, we approximately reduce the size of each hidden layer by half as $\{8, 16, 32\}$ to form the sub-model.

At the beginning of the t -th global round, all device initialize their local sub-models $\{\mathbf{w}_{t,i}^\alpha\}_{\forall i}$ by choosing the most important channels from the global model \mathbf{w}_t . In this way, the training complexity is significantly reduced while maintaining the performance of local sub-models. After that, the local training of device k is conducted with sub-model $\mathbf{w}_{t,i}^\alpha$, which produces the local gradient $\mathbf{u}_{t,i}$ with data size of $\alpha_{t,i}S$.

C. Flexible Gradient Compression

Given the local update $\mathbf{u}_{t,i}$ with the desired compression rate $\beta_{t,i}$, we aim to obtain the compressed update $\tilde{\mathbf{u}}_{t,i}$ with data size of $\alpha_{t,i}\beta_{t,i}S$. Let $\rho_{t,i}$ and $L_{t,i}$ denote the sparsity rate and the number of quantization levels, respectively. The gradient compression scheme works as follows.

1) *Kernel-wise sparsification:* Without loss of generality, we take the convolution neural network (CNN) as an example to illustrate the sparsification procedure. We aim to acquire the sparse update $\hat{\mathbf{u}}_{t,i}$ from $\mathbf{u}_{t,i}$. Let $\mathbf{u}_{t,i}[k]$ denote the k -th kernel of $\mathbf{u}_{t,i}$, and $\mathbf{u}_{t,i} = \{\mathbf{u}_{t,i}[k]\}_{\forall k}$. We measure the importance of each kernel and obtain $\mathcal{N} = \{\|\mathbf{u}_{t,i}[k]\|_2\}_{\forall k}$, where $\|\cdot\|_2$ denotes the L2 norm operation. Next, by selecting the $\lceil \rho_{t,i}K \rceil$ -th largest value in \mathcal{N} as the threshold Π , the kernel-wise sparsification is expressed as

$$\hat{\mathbf{u}}_{t,i}[k] = \begin{cases} \mathbf{0} & \text{if } \|\mathbf{u}_{t,i}[k]\|_2 < \Pi, \\ \mathbf{u}_{t,i}[k] & \text{otherwise.} \end{cases} \quad (2)$$

Meanwhile, the binary mask of $\hat{\mathbf{u}}_{t,i}$ is denoted as $\mathbf{m}_{t,i}$.

2) *Probabilistic quantization:* Motivated by the studies in [35], [36], we aim to obtain the quantized update $\tilde{\mathbf{u}}_{t,i}$ with the given sparse $\hat{\mathbf{u}}_{t,i}$ and the quantization level $L_{t,i}$. Let $u \in \hat{\mathbf{u}}_{t,i}$ be a scalar value. To begin with, we first calculate the magnitude range of the non-zero elements of $\hat{\mathbf{u}}_{t,i}$, denoted as $[u_{\min}, u_{\max}]$, where $u_{\min} = \min\{|u|\}_{\forall u \neq 0}$, and $u_{\max} = \max\{|u|\}_{\forall u \neq 0}$. Next, let $\mathcal{Q} = \{Q_l\}_{l=1}^{L_{t,i}}$ denote the set of quantization points, where Q_l is computed by

$$Q_l = \frac{l(u_{\max} - u_{\min})}{L_{t,i}} + u_{\min}. \quad (3)$$

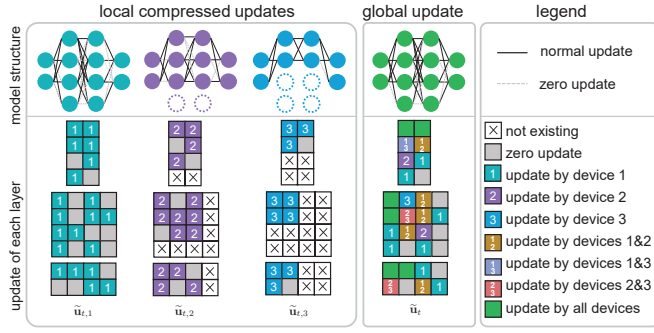


Fig. 3. An illustration of the all-in-one aggregation.

For any $u \in \hat{\mathbf{u}}_{t,i}$ and $u \neq 0$, we can always find a quantization interval $[Q_l, Q_{l+1}]$ such that $Q_l \leq |u| \leq Q_{l+1}$, and its corresponding quantized value \tilde{u} is further computed by

$$\tilde{u} = \begin{cases} \text{sgn}(u) \cdot Q_l & \text{with probability } \frac{Q_{l+1}-|u|}{Q_{l+1}-Q_l}, \\ \text{sgn}(u) \cdot Q_{l+1} & \text{otherwise,} \end{cases} \quad (4)$$

where $\text{sgn}(\cdot)$ calculates the sign of the given scalar. Furthermore, the set of the quantization indices of all $\tilde{u} \in \hat{\mathbf{u}}_{t,i}$ is denoted as $\mathcal{L}_{t,i} = \{l, Q_l = \tilde{u}\}_{\forall \tilde{u} \neq 0}$. Now, $\hat{\mathbf{u}}_{t,i}$ can be represented by a tuple of $\{u_{\min}, u_{\max}, L_{t,i}, \mathbf{m}_{t,i}, \mathcal{L}_{t,i}\}$.

3) *Lossless encoding*: Due to the distribution characteristics of $L_{t,i}$ that smaller indices may occur more frequently, we apply entropy coding to reduce the data size [14], [37]. Besides, the sparse binary matrix $\mathbf{m}_{t,i}$ can be compressed by Golomb encoding [11], [38].

After determining the compression scheme, we can vary the combinations of $\{\rho_{t,i}, L_{t,i}\}$ and record the corresponding compression rates. Based on the results, we can build a piecewise linear function to predict the compression strategy $\{\rho_{t,i}, L_{t,i}\}$ with the given $\beta_{t,i}$. Notably, this function can be efficiently fitted by the server with a rather small amount of public training data (e.g., 16 samples) in an offline manner.

D. All-in-One Aggregation

After all the devices upload their encoded updates, the server receives, decodes and then reconstructs the compressed local updates $\{\hat{\mathbf{u}}_{t,i}\}_{\forall i}$. Our goal is to obtain the global update $\hat{\mathbf{u}}_t$ by aggregating $\{\hat{\mathbf{u}}_{t,i}\}_{\forall i}$. However, the aggregation of local updates in our framework cannot be supported by conventional FedAvg [1], since the local updates are produced by different model structures with different levels of precision (i.e., different quantization levels and sparsity).

To tackle the above challenge, we propose an all-in-one aggregation scheme that fuses the local updates in an element-wise manner. Let the set $\{1, 2, \dots, J\}$ index elements of the global update $\hat{\mathbf{u}}_t$, and $\tilde{\mathbf{u}}_t^{[j]}$ denote the j -th element of $\hat{\mathbf{u}}_t$. To accomplish the aggregation for $\tilde{\mathbf{u}}_t^{[j]}$, we first determine the subset of devices $\mathcal{I}_j \subseteq \mathcal{I}$ whose local model structure also contains the j -th element. Then, we have

$$\tilde{\mathbf{u}}_t^{[j]} = \begin{cases} 0 & \text{if } \sum_{i \in \mathcal{I}_j} \mathbf{m}_{t,i}^{[j]} = 0, \\ \frac{1}{\sum_{i \in \mathcal{I}_j} p_{t,i} \mathbf{m}_{t,i}^{[j]}} \sum_{i \in \mathcal{I}_j} p_{t,i} \mathbf{m}_{t,i}^{[j]} \mathbf{u}_{t,i}^{[j]} & \text{otherwise,} \end{cases} \quad (5)$$

where $p_{t,i}$ is the aggregation coefficient for the j -th device at the t -th global round. The optimal values of $\{p_{t,i}\}_{\forall i}$ will be further analyzed in Section IV. Fig. 3 gives an example to illustrate the aggregation details. Specifically, different elements in the global update are updated by different subsets of devices, and more important elements will “absorb” knowledge from more devices. When the j -th element is zeroed out by all the devices in \mathcal{I}_j , we have $\tilde{\mathbf{u}}_t^{[j]} = 0$.

IV. THEORETICAL ANALYSIS AND OPTIMIZATION

In this section, we focus on the optimization of our framework by customizing the training strategies for diverse devices. We first formulate the on-demand training problem of AnycostFL. Then, we derive the upper bound of the convergence rate and reveal the key insights to improve the performance of AnycostFL. Based on the analysis, the optimization problem is transformed into a tractable form, and the closed-form solution is derived.

A. AnycostFL over Wireless Networks

In this subsection, we formulate the computation and communication models for our framework. After that, we build up an on-demand learning problem that minimizes the global training loss with given delay and energy constraints.

1) *Computation model*: For the device i at the t -th global round, given the model shrinking factor $\alpha_{t,i}$ and computing frequency $f_{t,i}$, the time consumption of local model training can be measured by

$$T_{t,i}^{\text{cmp}} = \frac{\tau |\mathcal{D}_i| \alpha_{t,i} W}{f_{t,i}}, \quad (6)$$

where τ denotes the number of local epochs. Meanwhile, the corresponding energy consumption can be given by

$$E_{t,i}^{\text{cmp}} = \epsilon_i f_{t,i}^2 \tau |\mathcal{D}_i| \alpha_{t,i} W, \quad (7)$$

where ϵ_i is the hardware energy coefficient of the device i .

2) *Communication model*: We consider the frequency division multiple access (FDMA) scheme for the transmission of the local gradient update. For the device i at the t -th global round, the achievable transmitting rate can be estimated by

$$r_{t,i} = b_i \log_2 \left(1 + \frac{|h_{t,i}| P_{t,i}^{\text{com}}}{N_0 b_i} \right), \quad (8)$$

where $P_{t,i}^{\text{com}}$ is the transmitting power; b_i is the achievable bandwidth; $|h_{t,i}|$ denotes the path loss of wireless channel; N_0 is the power spectral density of the additive white Gaussian noise. For the device i at t -th global round, given the update $\hat{\mathbf{u}}_{t,i}$ generated by the local model with a shrinking factor of $\alpha_{t,i}$ and compression rate of $\beta_{t,i}$, the required time $T_{t,i}^{\text{com}}$ and energy consumption $E_{t,i}^{\text{com}}$ of uplink transmission can be respectively measured by

$$T_{t,i}^{\text{com}} = \frac{\alpha_{t,i} \beta_{t,i} S}{r_{t,i}}, \text{ and } E_{t,i}^{\text{com}} = T_{t,i}^{\text{com}} P_{t,i}^{\text{com}}. \quad (9)$$

With the above computation and communication models, we next focus on the optimization problem of AnycostFL.

3) *Problem formulation*: To optimize AnycostFL, we study an on-demand training problem. Specifically, the shared maximal latency for each round T^{\max} is determined by the server. The local energy consumption budget for each round $E_{t,i}^{\max}$ is customized by the device itself. Given multiple devices with diverse local resources (e.g., computation, communication and data), our goal is to customize the training strategy for each device to minimize the global training loss with personalized constraints (e.g., latency and energy). To sum up, at the t -th global round, we aim to optimize the following problem.

$$\begin{aligned}
\text{(P1)} \quad & \min F(\mathbf{w}_t; \{\alpha_{t,i}\}_{\forall i}, \{\beta_{t,i}\}_{\forall i}) & (10) \\
\text{subject to:} \quad & T_{t,i}^{\text{cmp}} + T_{t,i}^{\text{com}} \leq T^{\max}, \forall i, & (10a) \\
& E_{t,i}^{\text{cmp}} + E_{t,i}^{\text{com}} \leq E_{t,i}^{\max}, \forall i, & (10b) \\
& \alpha^{\min} \leq \alpha_{t,i} \leq 1, \forall i, & (10c) \\
& 0 \leq \beta_{t,i} \leq \beta^{\max}, \forall i, & (10d) \\
& f_i^{\min} \leq f_{t,i} \leq f_i^{\max}, \forall i, & (10e) \\
\text{variables:} \quad & \{\alpha_{t,i}, \beta_{t,i}, f_{t,i}\}_{\forall i},
\end{aligned}$$

where $F(\mathbf{w}_t; \{\alpha_{t,i}\}_{\forall i}, \{\beta_{t,i}\}_{\forall i})$ denotes the global loss of the t -th round with given the global model weight \mathbf{w}_t under the training strategies of $\{\alpha_{t,i}\}_{\forall i}$ and $\{\beta_{t,i}\}_{\forall i}$. In the rest of this section, we analyze the relationship between training loss and training strategies. After that, Problem (P1) is further solved based on the theoretical insights.

B. Assumptions and Key Lemmas

Being in line with the studies in [5], [39], we make the following assumptions for the local loss function $F_i, \forall i$.

Assumption 1. F_i is λ -Lipschitz: $\|F_i(\mathbf{w}) - F_i(\mathbf{w}')\| \leq \lambda \|\mathbf{w} - \mathbf{w}'\|$, where $\lambda > 0$.

Assumption 2. F_i is ν -strongly convex: $F_i(\mathbf{w}) \geq F_i(\mathbf{w}') + (\mathbf{w} - \mathbf{w}')^\top \nabla F_i(\mathbf{w}') + \frac{\nu}{2} \|\mathbf{w} - \mathbf{w}'\|^2$.

Assumption 3. F_i is twice-continuously differentiable. Based on Assumptions 1 and 2, we have $\nu \mathbf{I} \preceq \nabla^2 F_i(\mathbf{w}) \preceq \lambda \mathbf{I}$.

Assumption 4. The ratios between the norms of $\nabla F_i(\mathbf{w})$ and $\nabla F(\mathbf{w})$ are bounded: $\|\nabla F_i(\mathbf{w})\|^2 \leq \varepsilon \|\nabla F(\mathbf{w})\|^2$, where $\varepsilon \geq 0$ is a positive constant.

Assumption 5. For the moderate shrinking factor $\alpha \geq \alpha^{\min}$, the first-shrinking-then-training can be approximated as first-training-then-shrinking: $\nabla F_i(\mathbf{w}^\alpha) = [\nabla F_i(\mathbf{w})]^\alpha$. Here, we use $[\nabla F_i(\mathbf{w})]^\alpha$ to denote the shrinking operation for $\nabla F_i(\mathbf{w})$.

Next, we give the following two definitions.

Definition 1 (Local gradient divergence). The local gradient divergence $\delta_{t,i}$ is defined as the difference between $\mathbf{u}_{t,i}$ and $\tilde{\mathbf{u}}_{t,i}$, which is given by $\delta_{t,i} = \|\mathbf{u}_{t,i} - \tilde{\mathbf{u}}_{t,i}\|$.

Definition 2 (Global gradient divergence). The global gradient divergence Δ_t is defined as the difference between \mathbf{u}_t and $\tilde{\mathbf{u}}_t$, which is measured by $\Delta_t = \|\mathbf{u}_t - \tilde{\mathbf{u}}_t\| = \left\| \sum_{i=1}^I p_{t,i} \mathbf{u}_{t,i} - \sum_{i=1}^I p_{t,i} \tilde{\mathbf{u}}_{t,i} \right\|$.

Notably, in Definition 1, $\mathbf{u}_{t,i}$ and $\tilde{\mathbf{u}}_{t,i}$ may have different dimensions. We pad the missing elements in $\tilde{\mathbf{u}}_{t,i}$ with zeros before the arithmetic operation. Next, we are interested in how

the training strategies $\{\alpha_{t,i}, \beta_{t,i}\}_{\forall i}$ affect $\{\delta_{t,i}\}_{\forall i}$ and Δ_t . We derive the following two lemmas.

Lemma 1. For the local training with the model shrinking factor $\alpha_{t,i}$ and compression rate $\beta_{t,i}$. The square of the local gradient divergence is bounded by

$$\mathbb{E} \|\delta_{t,i}\|^2 \leq (1 - \alpha_{t,i}(2 - \alpha_{t,i})\sqrt{\beta_{t,i}})^2 \mathbb{E} \|\mathbf{u}_{t,i}\|^2. \quad (11)$$

Proof. See Appendix A. \square

Lemma 2. For the local update $\{\tilde{\mathbf{u}}_{t,i}, \forall i\}$ with the corresponding training strategies $\{\alpha_{t,i}, \beta_{t,i}\}_{\forall i}$ and aggregation coefficients $\{p_{t,i}\}_{\forall i}$, the square of the global gradient divergence is bounded by

$$\mathbb{E} \|\Delta_t\|^2 \leq I \varepsilon \eta^2 \sum_{i=1}^I p_{t,i}^2 (1 - \alpha_{t,i}(2 - \alpha_{t,i})\sqrt{\beta_{t,i}})^2 \mathbb{E} \|\nabla F(\mathbf{w}_t)\|^2. \quad (12)$$

Proof. See Appendix B. \square

C. Optimal Aggregation Scheme and Convergence Analysis

Intuitively, the local update $\mathbf{u}_{t,i}$ generated with larger $\{\alpha_{t,i}, \beta_{t,i}\}$ may carry more accurate information, and thus a larger $p_{t,i}$ should be assigned during the aggregation. Based on Lemma 2, we deduce the following theorem.

Theorem 1 (Optimal aggregation scheme). Given the local updates $\{\tilde{\mathbf{u}}_{t,i}\}_{\forall i}$ with corresponding training strategies $\{\alpha_{t,i}, \beta_{t,i}\}_{\forall i}$, the optimal aggregation coefficients are

$$p_{t,i}^* = \frac{1}{\sum_i \frac{1}{(1 - \alpha_{t,i}(2 - \alpha_{t,i})\sqrt{\beta_{t,i}})^2}}, \forall i. \quad (13)$$

Proof. Based on Lemma 2, we study the following optimization problem to minimize the global gradient divergence.

$$\text{(P2)} \quad \min_{\{p_{t,i}\}_{\forall i}} \sum_{i=1}^I p_{t,i}^2 (1 - \alpha_{t,i}(2 - \alpha_{t,i})\sqrt{\beta_{t,i}})^2 \quad (14)$$

$$\text{subject to:} \quad p_{t,i} \geq 0, \forall i, \quad (14a)$$

$$\sum_{i=1}^I p_{t,i} = 1. \quad (14b)$$

It can be verified that Problem (P2) is a convex optimization problem. We further solve the problem by the Karush–Kuhn–Tucker (KKT) conditions. Let $\{\varpi\}_{\forall i}$ and θ be the Lagrange multipliers for Constraints (14a) and (14b), respectively. Then, we obtain

$$\varpi_i \geq 0, \quad \varpi_i p_{t,i} = 0, \quad p_{t,i} \geq 0, \quad \sum_{i=1}^I p_{t,i} = 1, \quad (15)$$

$$2p_{t,i}(1 - \alpha_{t,i}(2 - \alpha_{t,i})\sqrt{\beta_{t,i}})^2 - \varpi_i + \theta = 0, \quad \forall i.$$

Being in line with the study in [40], we can obtain

$$p_{t,i} = -\frac{\theta}{2(1 - \alpha_{t,i}(2 - \alpha_{t,i})\sqrt{\beta_{t,i}})^2}. \quad (16)$$

By putting Eqn. (16) into Eqn. (14b), we obtain

$$\theta = -\frac{2}{\sum_k \frac{1}{(1 - \alpha_{t,i}(2 - \alpha_{t,i})\sqrt{\beta_{t,i}})^2}}. \quad (17)$$

Putting Eqn. (17) into Eqn. (16) completes the proof. \square

With the optimal aggregation scheme, we investigate the upper bound of the convergence rate of AnycostFL.

Definition 3 (Local and global learning gains). The local and global learning gains are defined as $g_{t,i} = \alpha_{t,i}^4 \beta_{t,i}$ and $g_t = \sum_i g_{t,i}/I$, respectively. Specifically, the local and global learning gains (i.e., $g_{t,i} \in [0, 1]$ and $g_t \in [0, 1]$) measure the amount of effective information carried in the local and global updates, respectively.

Theorem 2 (Convergence rate of AnycostFL). Let $g^{\min} = \min\{g_t\}_{\forall t}$ be the minimal global learning gain over the T -round training. The upper bound of the convergence rate of AnycostFL satisfies

$$\mathbb{E}(F(\mathbf{w}_T) - F(\mathbf{w}^*)) \leq Z^{T-1} \mathbb{E}(F(\mathbf{w}_0) - F(\mathbf{w}^*)), \quad (18)$$

where $Z = 1 - \frac{\nu}{\lambda} (1 - \varepsilon(1 - g^{\min}))$. Recall that parameters ν, λ and ε are defined in Assumptions 1 to 4 before.

Proof. See Appendix C. \square

Based on Definition 3 and Theorem 2, we derive the following proposition.

Proposition 1. The key to minimizing the training loss of AnycostFL is to maximize the learning gain g_t for each global round. If $g_t = 1 \forall t$, AnycostFL degrades to conventional FL without model shrinking and gradient compression.

D. Solution for Problem (P1)

Based on Theorem 2 and Proposition 1, Problem (P1) can be transformed into the following problem.

$$(P3) \quad \max \frac{1}{I} \sum_{t=1}^I \alpha_{t,i}^4 \beta_{t,i} \quad (19)$$

subject to: Constrains (10a) to (10e),
variables: $\{\alpha_{t,i}, \beta_{t,i}, f_{t,i}\}_{\forall i}$.

Based on Constraints (10a) and (10b) for the training latency and energy, we obtain the following lemma.

Lemma 3. The equality will always hold for Constraints (10a) and (10b) when confirming the optimal training strategy $\{\alpha_{t,i}^*, \beta_{t,i}^*, f_{t,i}^*\}_{\forall i}$, and thus $T_{t,i}^* = T^{\max}$ and $E_{t,i}^* = E_{t,i}^{\max} \forall i$.

Proof. The lemma can be proved by showing the contradiction. Suppose that there exists i_0 such that $T_{t,i_0}^* < T^{\max}$. We can find a new solution $\{\alpha'_{t,i_0}, \beta'_{t,i_0}, f'_{t,i_0}\}$ for device i_0 and $\alpha'_{t,i_0} > \alpha_{t,i_0}^*$, $f'_{t,i_0} < f_{t,i_0}^*$, such that $T'_{t,i_0} = T^{\max}$ and $E'_{t,i_0} = E_{t,i_0}^{\max}$. Since the global learning gain increases with the increase of α_{t,i_0} , we have $g'_t > g_t^*$. Likewise, the contradiction also appears when $E_{t,i_0}^* < E_{t,i_0}^{\max}$, and thus we complete the proof. \square

Based on Lemma 3, we employ two intermediate variables (i.e., $\phi_{t,i}$ and $\varphi_{t,i}$) for each device to reparameterize Problem (P3). Specifically, $\phi_{t,i} \in [0, 1]$ and $\varphi_{t,i} \in [0, 1]$ are the splitting factors for latency and energy, respectively, such that

$$\begin{aligned} T_{t,i}^{\text{cmp}} &= \phi_{t,i} T^{\max}, \quad T_{t,i}^{\text{com}} = (1 - \phi_{t,i}) T^{\max}, \\ E_{t,i}^{\text{cmp}} &= \varphi_{t,i} E_{t,i}^{\max}, \quad E_{t,i}^{\text{com}} = (1 - \varphi_{t,i}) E_{t,i}^{\max}, \quad \forall i. \end{aligned} \quad (20)$$

By combining Eqns (6) and (20), the local learning gain of the device i at the t -th round can be rewritten as

$$g_{t,i}(\phi_{t,i}) = \kappa_{t,i} (E_{t,i}^{\max} - (1 - \phi_{t,i}) T^{\max} P_{t,i}^{\text{com}}) (\phi_{t,i}^2 - \phi_{t,i}^3), \quad (21)$$

where $\kappa_{t,i} = \frac{r_{t,i}}{S \varepsilon_i} \left(\frac{T^{\max}}{\tau |D_i| W} \right)^3$.

Note that Problem (P3) can be transformed into I sub-problems because the decision-making procedure of each device is independent. Based on Eqn. (21), the i -th sub-problem can be expressed as a single-variable optimization problem with respect to $\phi_{t,i}$ as follows.

$$(P4) \quad \max_{\phi_{t,i}} g_{t,i}(\phi_{t,i}) \quad (22)$$

subject to: $\phi_{t,i}^{\min} \leq \phi_{t,i} \leq \phi_{t,i}^{\max}$,

where the lower and upper limits of $\phi_{t,i}$ can be acquired by

$$\begin{aligned} \phi_{t,i}^{\min} &= \max \left\{ \frac{\alpha^{\min} \tau |D_i| W}{f_i^{\max} T^{\max}}, 1 - \frac{\beta^{\max} S}{r_{t,i} T^{\max}} \right\}, \\ \phi_{t,i}^{\max} &= \min \left\{ \frac{\tau |D_i| W}{f_i^{\min} T^{\max}}, 1 - \frac{\alpha^{\min} \beta^{\min} S}{r_{t,i} T^{\max}} \right\}. \end{aligned} \quad (23)$$

Based on the first-order optimality condition $\partial g_{t,i} / \partial \phi_{t,i} = 0$, we obtain the stationary points as

$$\phi_{t,i}^{s1} = \frac{\sqrt{\psi_{t,i}} - 3E_{t,i}^{\max}}{8P_{t,i}^{\text{com}} T^{\max}} + \frac{3}{4}, \quad \phi_{t,i}^{s2} = -\frac{\sqrt{\psi_{t,i}} + 3E_{t,i}^{\max}}{8P_{t,i}^{\text{com}} T^{\max}} - \frac{3}{4}, \quad (24)$$

where $\psi_{t,i} = 4(P_{t,i}^{\text{com}} T^{\max})^2 - 4E_{t,i}^{\max} P_{t,i}^{\text{com}} T^{\max} + 9(E_{t,i}^{\max})^2$. Let $\mathcal{S}_{t,i} = \{\phi_{t,i}^{\min}, \phi_{t,i}^{\max}, \phi_{t,i}^{s1}, \phi_{t,i}^{s2}\}$ denote the union of the stationary points and the boundary points for Problem (P4). Then, $\mathcal{S}'_{t,i} = \{\phi_{t,i} | \phi_{t,i} \in [\phi_{t,i}^{\min}, \phi_{t,i}^{\max}], \phi_{t,i} \in \mathcal{S}_{t,i}\}$ is the set of the feasible solutions of $\mathcal{S}_{t,i}$. The optimal solution for Problem (P4) can be acquired by

$$\phi_{t,i}^* = \arg \max_{\phi_{t,i} \in \mathcal{S}'_{t,i}} g_{t,i}(\phi_{t,i}). \quad (25)$$

Furthermore, we obtain the optimal solution for device i at the t -th global round by putting $\phi_{t,i}^*$ into the following equations.

$$\begin{aligned} \varphi_{t,i}^* &= 1 - \frac{(1 - \phi_{t,i}^*) T^{\max} P_{t,i}^{\text{com}}}{E_{t,i}^{\max}}, \quad \alpha_{t,i}^* = \sqrt[3]{\frac{(\phi_{t,i}^* T^{\max})^2 \varphi_{t,i}^* E_{t,i}^{\max}}{\varepsilon_i (\tau |D_i| W)^3}}, \\ \beta_{t,i}^* &= \frac{r_{t,i} (1 - \phi_{t,i}^*) T^{\max}}{\alpha_{t,i}^* S}, \quad f_{t,i}^* = \frac{\alpha_{t,i}^* \tau |D_i| W}{\phi_{t,i}^* T^{\max}}. \end{aligned} \quad (26)$$

Notably, the decision-making process of each device does not involve the auxiliary information of the resource status from other devices. At the beginning of each global round, each device can determine its training strategy locally.

V. EXPERIMENT EVALUATIONS

A. Experiment Settings

1) *Setup for FL training:* We consider the FL application with image classification on Fashion-MNIST and CIFAR-10 datasets [41], [42]. For Fashion-MNIST, we use a small convolutional neural network (CNN) with data size of model update as 53.22Mb [1]. For the CIFAR-10 dataset, we employ VGG-9 with data size of model update as 111.7Mb [43]. For IID and non-IID data settings, we follow the dataset partition strategy in [34]. For the learning hyper-parameters, the learning rate, batch size and local epoch are set as $\{0.01, 32, 1\}$ for Fashion-MNIST and $\{0.08, 64, 1\}$ for CIFAR-10 dataset. The maximal latency is set as $T^{\max} = 10$ seconds

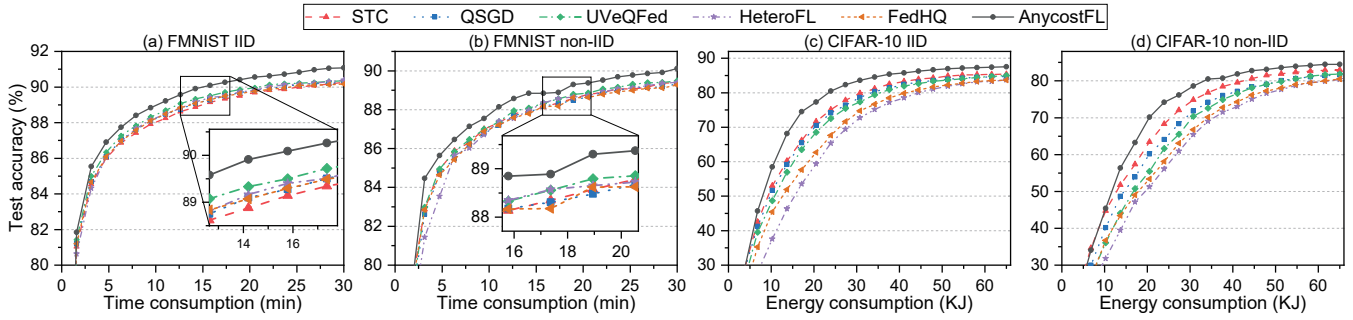


Fig. 4. Performance on various network architectures and datasets. ((a-b): global accuracy vs. time consumption with Fashion MNIST on 2-layer CNN; (c-d): global accuracy vs. energy consumption with CIFAR-10 on VGG-9.)

TABLE I
PERFORMANCE COMPARISON BETWEEN ANYCOSTFL AND OTHER METHODS ON FASHION-MNIST AND CIFAR-10 DATASETS.

Dataset	Method	IID						non-IID					
		#Round	Energy (KJ)	Latency (min)	Comp. (TFLOPs)	Comm. (GB)	Best Acc. (%)	#Round	Energy (KJ)	Latency (min)	Comp. (TFLOPs)	Comm. (GB)	Best Acc. (%)
FMNIST {90%, 89%}*	STC	305 (1.7×)	10.94 (1.4×)	25.42 (1.7×)	152.71	0.71	90.28±0.18	283 (1.3×)	10.17 (1.1×)	23.56 (1.3×)	141.53	0.66	89.47±0.16
	QSGD	283 (1.6×)	11.40 (1.4×)	23.56 (1.6×)	141.53	0.80	90.39±0.04	279 (1.3×)	11.27 (1.2×)	23.28 (1.3×)	139.86	0.79	89.49±0.07
	UVeQFed	247 (1.4×)	11.36 (1.4×)	20.58 (1.4×)	123.67	0.72	90.44±0.10	266 (1.2×)	12.21 (1.3×)	22.14 (1.2×)	133.01	0.77	89.64±0.16
	HeteroFL	233 (1.3×)	12.03 (1.5×)	21.78 (1.5×)	92.21	0.57	90.43±0.13	242 (1.1×)	12.51 (1.3×)	22.62 (1.3×)	95.77	0.59	89.42±0.10
	FedHQ	288 (1.6×)	13.89 (1.7×)	24.03 (1.6×)	144.36	0.86	90.21±0.07	313 (1.5×)	14.96 (1.6×)	26.06 (1.5×)	156.55	0.93	89.27±0.19
	AnycostFL	179 (1.0×)	8.07 (1.0×)	14.94 (1.0×)	67.49	0.35	91.20±0.09	214 (1.0×)	9.63 (1.0×)	17.83 (1.0×)	80.51	0.42	90.32±0.14
CIFAR-10 {82%, 80%}*	STC	341 (1.2×)	35.39 (1.3×)	56.83 (1.2×)	4160.56	1.78	85.38±0.29	412 (1.1×)	42.39 (1.3×)	68.67 (1.1×)	5026.84	2.15	83.09±0.53
	QSGD	337 (1.2×)	39.82 (1.5×)	56.17 (1.1×)	4111.76	2.14	84.83±0.54	430 (1.2×)	50.29 (1.5×)	71.61 (1.2×)	5242.39	2.73	81.94±0.13
	UVeQFed	296 (1.0×)	40.77 (1.5×)	49.28 (1.0×)	3607.45	2.12	85.09±0.16	377 (1.0×)	51.59 (1.5×)	62.89 (1.0×)	4603.87	2.71	82.30±0.28
	HeteroFL	332 (1.1×)	50.07 (1.9×)	69.14 (1.4×)	3222.26	1.65	83.75±0.55	413 (1.1×)	62.88 (1.9×)	85.78 (1.4×)	3990.49	2.05	80.68±0.45
	FedHQ	340 (1.2×)	48.95 (1.9×)	56.67 (1.2×)	4148.36	2.32	84.02±0.22	435 (1.2×)	61.99 (1.9×)	72.44 (1.2×)	5303.40	2.96	81.00±0.41
	AnycostFL	294 (1.0×)	26.43 (1.0×)	48.94 (1.0×)	2459.92	1.56	87.72±0.23	372 (1.0×)	33.51 (1.0×)	62.06 (1.0×)	3118.60	1.98	84.91±0.51

*{ x, y }: x and y denote the target global model accuracy under IID and non-IID data settings, respectively.

and the energy budget is set as $E_{t,i}^{\max} \sim U[3, 9]$ joules for the CIFAR-10 dataset, and the corresponding hyper-parameters for the FMNIST dataset are halved by default. Additionally, we set $\alpha^{\min} = 1/4$ and $\beta^{\max} = 1/15$.

2) *Setup for mobile system*: We investigate a mobile system with $I = 60$ devices located within a circle cell with a radius of 550 meters, and a base station is situated at the center. To simulate the mobility, the position of each device is refreshed randomly at the beginning of each round [44]. For the computation, the energy coefficient is set as $\epsilon_i \sim U[5 \times 10^{-27}, 1 \times 10^{-26}]$. For communication, the bandwidth is set as 1MHz equally for each device, and the path loss exponent is 3.76. The transmission power is set as 0.1W, and N_0 is set as -114dBm/MHz .

B. Performance Comparisons

We compare the proposed AnycostFL with the following efficient FL algorithms with three different random seeds.

- **STC**. The sparse ternary compression (STC) is adapted to reduce the cost of uplink parameter transmission [11].
- **QSGD**. The TopK sparsification and probabilistic quantization are combined to compress the local gradient [36].
- **UVeQFed**. The TopK sparsification and universal vector quantization are used to compress the local gradient [14].
- **HeteroFL**. Each device trains the local sub-model in different widths to match its computation capacity [32].
- **FedHQ**. Each device uses different quantization levels to compress the gradient according to its channel state [40].

Fig. 4 shows the performance of the global model over time consumption and energy consumption under the IID and the non-IID data setting. With the same training efficiency (i.e., time and energy consumption), the proposed AnycostFL consistently outperforms the baseline schemes to improve the test accuracy of the global model. Meanwhile, Table I provides the best accuracy and required system cost for achieving the specified test accuracy. Particularly, when compared with HeteroFL and FedHQ, AnycostFL can reduce up to 1.9 times the energy consumption to reach the test accuracy of 82% on CIFAR-10 dataset under the IID setting. When compared with STC, AnycostFL can reduce up to 1.7 times the time consumption to reach the test accuracy of 90% on FMNIST dataset under the IID setting. Moreover, our framework can significantly improve the best accuracy of the global model by 2.33% and 1.82% on CIFAR-10 dataset under the IID and the non-IID settings, respectively.

C. Impact of Key Mechanisms and Hyper-parameters

Fig. 5(a) verifies the advantages of the main techniques of AnycostFL. We gradually remove the elastic model shrinking (w/o EMS), the flexible gradient compression (w/o FGC) and the all-in-one aggregation (w/o AIO), and record the required system cost to achieve 80% test accuracy with CIFAR-10 dataset under the IID setting. We observe that the proposed EMS and FGC can significantly save the energy consumption and training time, respectively. Besides, AIO contributes to saving both energy and time.

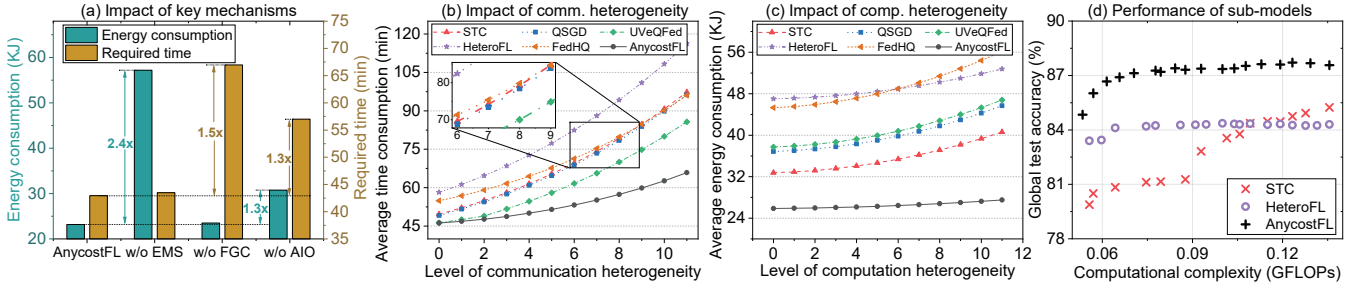


Fig. 5. The main advantages of AnycostFL. ((a): the impact of key mechanisms; (b-c): the impact of system heterogeneity; (d): the performance of sub-models.)

We next evaluate the impact of resource heterogeneity on the training efficiency in Fig. 5(b-c). We set the average energy coefficient ϵ_i as 7.5×10^{-27} and the average distance between the base station and edge devices as 400 meters, and then change their variances to simulate the computation and communication heterogeneity, respectively. The larger variance indicates a higher level of system heterogeneity. As we expect, the proposed AnycostFL shows more resilience than other baselines to tackle the high level of system heterogeneity.

We also evaluate the performance of sub-models in different widths in Fig. 5(d). Specifically, We compare AnycostFL with HeteroFL (i.e., local training with different widths) and STC (i.e., the best-performing compression-only method). The sub-models are derived from the well-trained global model without further re-training. Surprisingly, the sub-models of the global model trained by AnycostFL can still maintain satisfactory test accuracy, which provides dynamic inference for diverse edge devices after the training time.

VI. CONCLUSION

In this paper, we proposed AnycostFL, a joint computation and communication efficient framework for FL, that enables edge devices with diverse resources to train a shared global model. We aimed to minimize the global training loss under given personalized latency and energy constraints. By leveraging the theoretical insight of AnycostFL, we decomposed the optimization problem into multiple sub-problems. Following that, the optimal training strategy is derived for each device according to its locally available resource. Experiments demonstrate the advantage of our framework in improving the system efficiency and model performance compared to the state-of-the-art methods.

ACKNOWLEDGMENT

Rong Yu and Yuan Wu are the corresponding authors. This work was supported in part by National Key R&D Program of China under Grant 2020YFB1807802, in part by National Natural Science Foundation of China under Grants 61971148, 62102099, U22A2054 and 62001125, in part by Science and Technology Development Fund of Macau SAR under Grant 0162/2019/A3, in part by FDCT-MOST Joint Project under Grant 0066/2019/AMJ, in part by the Guangdong Basic and Applied Basic Research Foundation (2022A1515011287), and in part by US National Science Foundation under grant CNS-2107057.

APPENDIX A PROOF OF LEMMA 1

Proof. For the given local gradient $\tilde{\mathbf{u}}_{t,i}$ with shrinking factor $\alpha_{t,i}$ and gradient compression rate $\beta_{t,i}$, we aim to capture the divergence between $\tilde{\mathbf{u}}_{t,i}$ and $\mathbf{u}_{t,i}$. Suppose that the absolute value of the element in $\mathbf{u}_{t,i}$ follows uniform distribution $|u| \sim U(0, u_{\max})$, and $u_{\max} = \max\{|u|\}_{\forall u \in \mathbf{u}_{t,i}}$.

For clear notation, we sort the element-wise absolute value of $\mathbf{u}_{t,i}$ in ascending order. Then, we obtain $\mathbf{u}_{t,i} = [\mathbf{u}_{t,i}^{[1]}, \dots, \mathbf{u}_{t,i}^{[j]}, \dots, \mathbf{u}_{t,i}^{[J]}]^\top$ and $|\mathbf{u}_{t,i}^{[j]}| \leq |\mathbf{u}_{t,i}^{[j+1]}|$. Thus, we have

$$\mathbb{E}\|\mathbf{u}_{t,i}\|^2 = \mathbb{E} \sum_{j=1}^J |\mathbf{u}_{t,i}^{[j]}|^2 = J \mathbb{E}|\mathbf{u}_{t,i}^{[j]}|^2 = \frac{J u_{\max}^2}{3}. \quad (27)$$

Based on Assumption 5, the update generated from local training with $\mathbf{w}_{t,i}^\alpha$ is equal to $\text{shrink}(\mathbf{u}_{t,i}, \alpha_{t,i})$. The operation of model shrinking on $\mathbf{u}_{t,i}$ with $\alpha_{t,i}$ can be viewed as removing $(1 - \alpha_{t,i})J$ elements with the least value from $\mathbf{u}_{t,i}$. Then, we obtain $\text{shrink}(\mathbf{u}_{t,i}, \alpha_{t,i}) = [0, \dots, 0, \mathbf{u}_{t,i}^{[(1-\alpha_{t,i})J+1]}, \dots, \mathbf{u}_{t,i}^{[J]}]^\top$. Thus, we have

$$\begin{aligned} \mathbb{E}\|\mathbf{u}_{t,i} - \text{shrink}(\mathbf{u}_{t,i}, \alpha_{t,i})\|^2 &= \mathbb{E} \sum_{j=1}^{(1-\alpha_{t,i})J} |\mathbf{u}_{t,i}^{[j]}|^2 \\ &= J(1 - \alpha_{t,i})^3 u_{\max}^2 / 3 = (1 - \alpha_{t,i})^3 \mathbb{E}\|\mathbf{u}_{t,i}\|^2. \end{aligned} \quad (28)$$

We next focus on the gradient compression. The operation of gradient sparsification on $\mathbf{u}_{t,i}$ with sparsity of $\rho_{t,i}$ can be viewed as removing $\rho_{t,i}J$ elements with the least value from $\mathbf{u}_{t,i}$. Then, the quantization is conducted on the non-zero elements of $\hat{\mathbf{u}}_{t,i}$, and we obtain $\text{cmpres}(\mathbf{u}_{t,i}, \beta_{t,i}) = [0, \dots, 0, \tilde{\mathbf{u}}_{t,i}^{[\rho_{t,i}J+1]}, \dots, \tilde{\mathbf{u}}_{t,i}^{[J]}]^\top$. Furthermore, we have

$$\begin{aligned} \mathbb{E}\|\mathbf{u}_{t,i} - \text{cmpres}(\mathbf{u}_{t,i}, \beta_{t,i})\|^2 &= \mathbb{E} \underbrace{\sum_{j=1}^{\rho_{t,i}J} |\mathbf{u}_{t,i}^{[j]}|^2}_{(A)} + \mathbb{E} \underbrace{\sum_{j=\rho_{t,i}J+1}^J |\mathbf{u}_{t,i}^{[j]} - \tilde{\mathbf{u}}_{t,i}^{[j]}|^2}_{(B)}. \end{aligned} \quad (29)$$

Likewise to Eqn. (28), we have (A) = $\rho_{t,i}^3 \mathbb{E}\|\mathbf{u}_{t,i}\|^2$. Based on Eqn. (4) and the statistical feature of $\mathbf{u}_{t,i}$, we obtain (B) = $(1 - \rho_{t,i})^3 \mathbb{E}\|\mathbf{u}_{t,i}\|^2 / (2L_{t,i}^2)$.

Given plain update $\mathbf{u}_{t,i}$ in 32-bit floating point and the desired compression rate $\beta_{t,i}$, we can set $\rho_{t,i} = 1 - \sqrt{\beta_{t,i}}$ and $L_{t,i} = 2^{32} \sqrt{\beta_{t,i}}$ for the analysis. In this way, the operations

of sparsification and quantization contribute equally to the gradient compression. Furthermore, we have

$$\mathbb{E}\|\mathbf{u}_{t,i} - \text{cmprsr}(\mathbf{u}_{t,i}, \beta_{t,i})\|^2 \leq (1 - \beta_{t,i})^2 \mathbb{E}\|\mathbf{u}_{t,i}\|^2. \quad (30)$$

Next, we focus on the local divergence $\delta_{t,i}$ with respect to $\alpha_{t,i}$ and $\beta_{t,i}$. According to the Definition 1, we have

$$\begin{aligned} \mathbb{E}\|\delta_{t,i}\|^2 &= \mathbb{E}\|\mathbf{u}_{t,i} - \text{cmprsr}([\mathbf{u}_{t,i}]^\alpha, \beta_{t,i})\|^2 \\ &= \mathbb{E}\|\mathbf{u}_{t,i} - [\mathbf{u}_{t,i}]^\alpha\|^2 + \mathbb{E}\|[\mathbf{u}_{t,i}]^\alpha - \text{cmprsr}([\mathbf{u}_{t,i}]^\alpha, \beta_{t,i})\|^2 \\ &\quad + 2 \underbrace{\langle \mathbf{u}_{t,i} - [\mathbf{u}_{t,i}]^\alpha, [\mathbf{u}_{t,i}]^\alpha - \text{cmprsr}([\mathbf{u}_{t,i}]^\alpha, \beta_{t,i}) \rangle}_{(C)}. \end{aligned} \quad (31)$$

It can be verified that the two vectors in term (C) are orthogonal, and we obtain (C) = 0. According to Eqns (28) and (30), we further obtain

$$\begin{aligned} \mathbb{E}\|\delta_{t,i}\|^2 &\leq (1 - \alpha_{t,i})^3 \mathbb{E}\|\mathbf{u}_{t,i}\|^2 + (1 - \sqrt{\beta_{t,i}})^2 \mathbb{E}\|[\mathbf{u}_{t,i}]^\alpha\|^2 \\ &\stackrel{(a)}{\leq} (1 - \alpha_{t,i})^3 \mathbb{E}\|\mathbf{u}_{t,i}\|^2 \\ &\quad + (1 - \sqrt{\beta_{t,i}})^2 \alpha_{t,i} (\alpha_{t,i}^2 - 3\alpha_{t,i} + 3) \mathbb{E}\|\mathbf{u}_{t,i}\|^2 \\ &\stackrel{(b)}{\leq} (1 - \alpha_{t,i}(2 - \alpha_{t,i})\sqrt{\beta_{t,i}})^2 \mathbb{E}\|\mathbf{u}_{t,i}\|^2. \end{aligned} \quad (32)$$

Likewise to Eqn. (28), inequality (a) stems from the fact that $\mathbb{E}\|[\mathbf{u}_{t,i}]^\alpha\|^2 = \alpha_{t,i}(\alpha_{t,i}^2 - 3\alpha_{t,i} + 3)\mathbb{E}\|\mathbf{u}_{t,i}\|^2$. Besides, inequality (b) holds for all $\alpha_{t,i} \in [\alpha^{\min}, 1]$ and $\beta_{t,i} \in [0, \beta^{\max}]$. Thus, we complete the proof. \square

APPENDIX B PROOF OF LEMMA 2

Proof. Based on Definition 2 and Lemma 1, we have

$$\begin{aligned} \mathbb{E}\|\Delta_t\|^2 &= \mathbb{E}\left\| \sum_{i=1}^I p_{t,i} \mathbf{u}_{t,i} - \sum_{i=1}^I p_{t,i} \tilde{\mathbf{u}}_{t,i} \right\|^2 \\ &\leq \mathbb{E}\left(\sum_{i=1}^I p_{t,i} (1 - \alpha_{t,i}(2 - \alpha_{t,i})\sqrt{\beta_{t,i}}) \|\mathbf{u}_{t,i}\| \right)^2. \end{aligned} \quad (33)$$

We use η to denote the learning rate, and $\mathbf{u}_{t,i} = \eta \nabla F_i(\mathbf{w}_t)$. Based on Assumption 4, we obtain

$$\mathbb{E}\|\Delta_t\|^2 \leq \varepsilon \eta^2 \left(\sum_{i=1}^I p_{t,i} (1 - \alpha_{t,i}(2 - \alpha_{t,i})\sqrt{\beta_{t,i}}) \right)^2 \mathbb{E}\|\nabla F(\mathbf{w}_t)\|^2. \quad (34)$$

According to Cauchy–Schwarz inequality, we obtain

$$\mathbb{E}\|\Delta_t\|^2 \leq I \varepsilon \eta^2 \sum_{i=1}^I p_{t,i}^2 (1 - \alpha_{t,i}(2 - \alpha_{t,i})\sqrt{\beta_{t,i}})^2 \mathbb{E}\|\nabla F(\mathbf{w}_t)\|^2. \quad (35)$$

Thus, we complete the proof. \square

APPENDIX C ON THE CONVERGENCE OF ANYCOSTFL

Proof. Inspired by the studies in [5], [39], we deduce the convergence analysis of AnycostFL. According to Taylor expansion and Assumption 3, we have

$$\begin{aligned} F(\mathbf{w}_{t+1}) &\leq F(\mathbf{w}_t) + (\mathbf{w}_{t+1} - \mathbf{w}_t)^\top \nabla F(\mathbf{w}_t) + \frac{\lambda}{2} \|\mathbf{w}_{t+1} - \mathbf{w}_t\|^2 \\ &= F(\mathbf{w}_t) - \tilde{\mathbf{u}}_t^\top \nabla F(\mathbf{w}_t) + \frac{\lambda}{2} \|\tilde{\mathbf{u}}_t\|^2. \end{aligned} \quad (36)$$

By using learning rate $\eta = \frac{1}{\lambda}$, we obtain

$$\begin{aligned} \mathbb{E}(F(\mathbf{w}_{t+1})) &\leq \mathbb{E}\left(F(\mathbf{w}_t) - \lambda (\mathbf{w}_t - \Delta_t)^\top \mathbf{u}_t + \frac{\lambda}{2} \|\mathbf{w}_t - \Delta_t\|^2 \right) \\ &= \mathbb{E}\left(F(\mathbf{w}_t) - \frac{1}{2\lambda} \|\nabla F(\mathbf{w}_t)\|^2 + \frac{\lambda}{2} \|\Delta_t\|^2 \right). \end{aligned} \quad (37)$$

We now pay attention to the upper bound of $\|\Delta_t\|^2$. Based on Jensen's inequality and Eqn. (34), we obtain

$$\mathbb{E}\|\Delta_t\|^2 \leq \varepsilon \eta^2 \underbrace{\sum_{i=1}^I p_{t,i} (1 - \alpha_{t,i}(2 - \alpha_{t,i})\sqrt{\beta_{t,i}})^2}_{(D)} \mathbb{E}\|\nabla F(\mathbf{w}_t)\|^2. \quad (38)$$

By putting Eqn. (13) into (A), we have

$$\mathbb{E}\|\mathbf{D}\| \leq \mathbb{E} \left\| \frac{I}{\sum_{i=1}^I \frac{1}{(1 - \alpha_{t,i}(2 - \alpha_{t,i})\sqrt{\beta_{t,i}})^2}} \right\| \stackrel{(c)}{\leq} \mathbb{E} \left\| \frac{I}{\sum_{i=1}^I \frac{1}{1 - \alpha_{t,i}^4 \beta_{t,i}}} \right\|, \quad (39)$$

where (c) always holds for $\alpha_{t,i} \in [0, 1]$ and $\beta_{t,i} \in [0, 1]$. According to Definition 3, we have $g_{t,i} = \alpha_{t,i}^4 \beta_{t,i}$ and $g_t = \sum_i g_{t,i} / I$. Since $1 / (\sum_i \frac{1}{1 - g_{t,i}})$ is a concave function with respect to $g_{t,i}$, based on Jensen's inequality, we obtain

$$\mathbb{E}\|\mathbf{A}\| \leq \frac{I}{\sum_i \frac{1}{1 - \mathbb{E}(\alpha_{t,i}^4 \beta_{t,i})}} = 1 - g_t. \quad (40)$$

Since the training strategies of each device and the norm of the gradient of global data $\|\nabla F(\mathbf{w}_t)\|$ are independent, by putting Eqn. (40) back to Eqn. (38), we obtain

$$\mathbb{E}\|\Delta_t\|^2 \leq \mathbb{E}\left(\varepsilon \eta^2 (1 - g_t) \|\nabla F(\mathbf{w}_t)\|^2 \right). \quad (41)$$

Next, by putting Eqn. (41) back to Eqn. (37), we have

$$\mathbb{E}(F(\mathbf{w}_{t+1})) \leq \mathbb{E}\left(F(\mathbf{w}_t) - \frac{1 + \varepsilon(g_t - 1)}{2\lambda} \|\nabla F(\mathbf{w}_t)\|^2 \right). \quad (42)$$

Subtracting $F(\mathbf{w}^*)$ in both sides of Eqn. (42) yields

$$\begin{aligned} \mathbb{E}(F(\mathbf{w}_{t+1}) - F(\mathbf{w}^*)) &\leq \mathbb{E}\left(F(\mathbf{w}_t) - \frac{1 + \varepsilon(g_t - 1)}{2\lambda} \|\nabla F(\mathbf{w}_t)\|^2 - F(\mathbf{w}^*) \right). \end{aligned} \quad (43)$$

Based on Assumptions 2 and 3, we have [5], [45]

$$\|\nabla F(\mathbf{w}_t)\|^2 \geq 2\nu (F(\mathbf{w}_t) - F(\mathbf{w}^*)). \quad (44)$$

Plugging Eqn. (44) into Eqn. (43), we have

$$\mathbb{E}(F(\mathbf{w}_{t+1}) - F(\mathbf{w}^*)) \leq Z_t \mathbb{E}(F(\mathbf{w}_t) - F(\mathbf{w}^*)), \quad (45)$$

where $Z_t = 1 - \frac{\varepsilon}{\lambda} (1 - \varepsilon(1 - g_t))$.

Let $g^{\min} = \min\{g_t\}_{\forall t}$ be the minimal global learning gain over T global rounds. By recursively applying the above inequality from iteration round 0 to T , we can obtain

$$\mathbb{E}(F(\mathbf{w}_T) - F(\mathbf{w}^*)) \leq Z^{T-1} \mathbb{E}(F(\mathbf{w}_0) - F(\mathbf{w}^*)), \quad (46)$$

where $Z = 1 - \frac{\varepsilon}{\lambda} (1 - \varepsilon(1 - g^{\min}))$. Thus, we complete the proof. \square

REFERENCES

- [1] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y. Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proc. of AISTATS 2017*, 20–22 Apr. 2017, pp. 1273–1282.
- [2] Y. Qu, C. Dong, J. Zheng, H. Dai, F. Wu, S. Guo, and A. Anpalagan, "Empowering edge intelligence by air-ground integrated federated learning," *IEEE Network*, vol. 35, no. 5, pp. 34–41, 2021.
- [3] R. Yu and P. Li, "Toward resource-efficient federated learning in mobile edge computing," *IEEE Network*, vol. 35, no. 1, pp. 148–155, 2021.
- [4] T. Nishio and R. Yonetani, "Client selection for federated learning with heterogeneous resources in mobile edge," in *ICC*. IEEE, 2019, pp. 1–7.
- [5] M. Chen, Z. Yang, W. Saad, C. Yin, H. V. Poor, and S. Cui, "A joint learning and communications framework for federated learning over wireless networks," *IEEE Transactions on Wireless Communications*, vol. 20, no. 1, pp. 269–283, 2020.
- [6] Z. Yang, M. Chen, W. Saad, C. S. Hong, and M. Shikh-Bahaei, "Energy efficient federated learning over wireless communication networks," *IEEE Transactions on Wireless Communications*, vol. 20, no. 3, pp. 1935–1949, 2020.
- [7] C. T. Dinh, N. H. Tran, M. N. Nguyen, C. S. Hong, W. Bao, A. Y. Zomaya, and V. Gramoli, "Federated learning over wireless networks: Convergence analysis and resource allocation," *IEEE/ACM Transactions on Networking*, vol. 29, no. 1, pp. 398–409, 2020.
- [8] J. Yao and N. Ansari, "Enhancing federated learning in fog-aided iot by cpu frequency and wireless power control," *IEEE Internet of Things Journal*, vol. 8, no. 5, pp. 3438–3445, 2020.
- [9] N. H. Tran, W. Bao, A. Zomaya, M. N. Nguyen, and C. S. Hong, "Federated learning over wireless networks: Optimization model design and analysis," in *INFOCOM*. IEEE, 2019, pp. 1387–1395.
- [10] Y. Wu, Y. Song, T. Wang, L. Qian, and T. Q. Quek, "Non-orthogonal multiple access assisted federated learning via wireless power transfer: A cost-efficient approach," *IEEE Transactions on Communications*, vol. 70, no. 4, pp. 2853–2869, 2022.
- [11] F. Sattler, S. Wiedemann, K.-R. Müller, and W. Samek, "Robust and communication-efficient federated learning from non-iid data," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 31, no. 9, pp. 3400–3413, 2019.
- [12] L. Li, D. Shi, R. Hou, H. Li, M. Pan, and Z. Han, "To talk or to work: Flexible communication compression for energy efficient federated learning over heterogeneous mobile edge devices," in *INFOCOM*. IEEE, 2021, pp. 1–10.
- [13] P. Li, X. Huang, M. Pan, and R. Yu, "Fedgreen: Federated learning with fine-grained gradient compression for green mobile edge computing," in *GLOBECOM*, 2021, pp. 1–6.
- [14] N. Shlezinger, M. Chen, Y. C. Eldar, H. V. Poor, and S. Cui, "Uveqfed: Universal vector quantization for federated learning," *IEEE Transactions on Signal Processing*, vol. 69, pp. 500–514, 2021.
- [15] L. Cui, X. Su, Y. Zhou, and J. Liu, "Optimal rate adaption in federated learning with compressed communications," in *INFOCOM*. IEEE, 2022, pp. 1459–1468.
- [16] J. Kang, X. Li, J. Nie, Y. Liu, M. Xu, Z. Xiong, D. Niyato, and Q. Yan, "Communication-efficient and cross-chain empowered federated learning for artificial intelligence of things," *IEEE Transactions on Network Science and Engineering*, 2022.
- [17] P. Prakash, J. Ding, M. Shu, J. Wang, W. Xu, and M. Pan, "Squaff: Sketch-quantization inspired communication efficient federated learning," in *2021 IEEE/ACM Symposium on Edge Computing (SEC)*. IEEE, 2021, pp. 350–354.
- [18] P. Li, Y. Zhong, C. Zhang, Y. Wu, and R. Yu, "Fedrelay: Federated relay learning for 6g mobile edge intelligence," *IEEE Transactions on Vehicular Technology*, pp. 1–14, 2022.
- [19] D. Liu and O. Simeone, "Privacy for free: Wireless federated learning via uncoded transmission with adaptive power control," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 1, pp. 170–185, 2020.
- [20] J. Xu, H. Wang, and L. Chen, "Bandwidth allocation for multiple federated learning services in wireless edge networks," *IEEE Transactions on Wireless Communications*, vol. 21, no. 4, pp. 2534–2546, 2021.
- [21] S. Wang, M. Lee, S. Hosseinalipour, R. Morabito, M. Chiang, and C. G. Brinton, "Device sampling for heterogeneous federated learning: Theory, algorithms, and implementation," in *INFOCOM*. IEEE, 2021, pp. 1–10.
- [22] W. Shi, S. Zhou, Z. Niu, M. Jiang, and L. Geng, "Joint device scheduling and resource allocation for latency constrained wireless federated learning," *IEEE Transactions on Wireless Communications*, vol. 20, no. 1, pp. 453–467, 2020.
- [23] Y. Deng, F. Lyu, J. Ren, H. Wu, Y. Zhou, Y. Zhang, and X. Shen, "Auction: Automated and quality-aware client selection framework for efficient federated learning," *IEEE Transactions on Parallel and Distributed Systems*, vol. 33, no. 8, pp. 1996–2009, 2021.
- [24] L. Liu, J. Zhang, S. Song, and K. B. Letiaief, "Client-edge-cloud hierarchical federated learning," in *ICC*. IEEE, 2020, pp. 1–6.
- [25] Z. Wang, H. Xu, J. Liu, H. Huang, C. Qiao, and Y. Zhao, "Resource-efficient federated learning with hierarchical aggregation in edge computing," in *INFOCOM*. IEEE, 2021, pp. 1–10.
- [26] P. Han, S. Wang, and K. K. Leung, "Adaptive gradient sparsification for efficient federated learning: An online learning approach," in *ICDCS*. IEEE, 2020, pp. 300–310.
- [27] A. Reiszadeh, A. Mokhtari, H. Hassani, A. Jadbabaie, and R. Pedarsani, "Fedpaq: A communication-efficient federated learning method with periodic averaging and quantization," in *International Conference on Artificial Intelligence and Statistics*. PMLR, 2020, pp. 2021–2031.
- [28] N. Shlezinger, M. Chen, Y. C. Eldar, H. V. Poor, and S. Cui, "Federated learning with quantization constraints," in *ICASSP*. IEEE, 2020, pp. 8851–8855.
- [29] T. Lin, L. Kong, S. U. Stich, and M. Jaggi, "Ensemble distillation for robust model fusion in federated learning," in *NeurIPS*, vol. 33, 2020, pp. 2351–2363.
- [30] Z. Zhu, J. Hong, and J. Zhou, "Data-free knowledge distillation for heterogeneous federated learning," in *ICML*. PMLR, 2021, pp. 12 878–12 889.
- [31] A. Li, J. Sun, X. Zeng, M. Zhang, H. Li, and Y. Chen, "Fedmask: Joint computation and communication-efficient personalized federated learning via heterogeneous masking," in *Proceedings of the 19th ACM Conference on Embedded Networked Sensor Systems*, 2021, pp. 42–55.
- [32] E. Diao, J. Ding, and V. Tarokh, "Heterofl: Computation and communication efficient federated learning for heterogeneous clients," in *ICLR*, 2021.
- [33] H. Baek, W. J. Yun, Y. Kwak, S. Jung, M. Ji, M. Bennis, J. Park, and J. Kim, "Joint superposition coding and training for federated learning over multi-width neural networks," in *INFOCOM*. IEEE, 2022, pp. 1729–1738.
- [34] H. Wang, M. Yurochkin, Y. Sun, D. Papailiopoulos, and Y. Khazaeni, "Federated learning with matched averaging," in *ICLR*, 2020.
- [35] J. Konečný, H. B. McMahan, F. X. Yu, P. Richtárik, A. T. Suresh, and D. Bacon, "Federated learning: Strategies for improving communication efficiency," *arXiv preprint arXiv:1610.05492*, 2016.
- [36] D. Alistarh, D. Grubic, J. Li, R. Tomioka, and M. Vojnovic, "Qsgd: Communication-efficient sgd via gradient quantization and encoding," in *NeurIPS*, 2017.
- [37] S. Han, H. Mao, and W. J. Dally, "Deep compression: Compressing deep neural networks with pruning, trained quantization and Huffman coding," in *ICLR*, 2016.
- [38] S. Golomb, "Run-length encodings (corresp.)," *IEEE Transactions on Information Theory*, vol. 12, no. 3, pp. 399–401, 1966.
- [39] M. Chen, H. V. Poor, W. Saad, and S. Cui, "Convergence time optimization for federated learning over wireless networks," *IEEE Transactions on Wireless Communications*, vol. 20, no. 4, pp. 2457–2471, 2020.
- [40] S. Chen, C. Shen, L. Zhang, and Y. Tang, "Dynamic aggregation for heterogeneous quantization in federated learning," *IEEE Transactions on Wireless Communications*, vol. 20, no. 10, pp. 6804–6819, 2021.
- [41] H. Xiao, K. Rasul, and R. Vollgraf, "Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms," *arXiv preprint arXiv:1708.07747*, 2017.
- [42] A. Krizhevsky, G. Hinton *et al.*, "Learning multiple layers of features from tiny images," 2009.
- [43] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *ICLR*, 2015.
- [44] W. Shi, S. Zhou, Z. Niu, M. Jiang, and L. Geng, "Joint device scheduling and resource allocation for latency constrained wireless federated learning," *IEEE Transactions on Wireless Communications*, vol. 20, no. 1, pp. 453–467, 2021.
- [45] S. Boyd, S. P. Boyd, and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.