Joint Optimization of Adaptive Gradient Compression and Aggregation for Efficient Asynchronous Federated Learning

Yingwei Hou[1], Danyang Xiao[1], Linlin You[2], Weigang Wu[1,*]
[1]School of Computer Science and Engineering, Sun Yat-Sen University, Guangzhou, China
[2]School of Intelligent Systems Engineering, Sun Yat-Sen University, Shenzhen, China

***Abstract*—Asynchronous Federated Learning (AFL) enables distributed model training without client synchronization, offering enhanced scalability in heterogeneous and bandwidth-limited environments. However, in many AFL approaches, the use of gradient compression is coupled with fixed aggregation intervals and compression rates, which often leads to suboptimal performance caused by excessive model staleness and communication cost. We propose FedJACA, a novel AFL framework that jointly and adaptively configures the aggregation interval and gradient compression rate, so as to maximize gradient utility while minimizing staleness and communication cost. We model their intertwined impact and formulate a joint optimization problem under training time and bandwidth constraints. And we design a customized solution algorithm based on the Trust Region Newton method that yields high-quality solutions for achieving a balanced trade-off between efficiency and cost. Extensive experiments on four benchmarks show that FedJACA improves accuracy, reduces training time by 40%, and lowers communication cost by 35%, outperforming state-of-the-art baselines.**

***Index terms*—** Asynchronous Federated Learning, Joint Optimization, Adaptive Gradient Compression, Aggregation Interval, Model Staleness

I. Introduction

Asynchronous Federated Learning (AFL) [1] enables clients to update a global model without synchronization, improving scalability in heterogeneous networks. However, it suffers from model staleness, as outdated gradients may be incorporated. The aggregation interval, which defines how frequently global aggregation occurs, significantly affects model staleness and training efficiency. Short aggregation intervals can amplify staleness and reduce the diversity of aggregated gradients, while long intervals may trigger the straggler effect, reducing aggregation frequency and slowing convergence [2].

To reduce communication cost, AFL often applies gradient compression such as sparsification and quantization [3], [4]. However, fixed compression rates are suboptimal: aggressive compression harms convergence by discarding critical information, while conservative compression increases delay and aggravates staleness. Recent adaptive compression methods [5] improve efficiency or energy-awareness, but they optimize compression and aggregation separately, without considering their interdependence. Similarly, existing aggregation schemes, from fully asynchronous [1] to buffered or periodic designs [6], [7], often rely on static or heuristic intervals, lacking adaptability to dynamic client conditions.

Critically, aggregation interval and compression rate are interdependent, and their joint impact on training efficiency and communication cost is far more complex than when considered separately. Gradient compression can help reduce communication time, but its effectiveness depends on how well it is aligned with the aggregation interval. For instance, longer aggregation intervals may require higher compression rates to transmit more gradient information within budget, whereas shorter intervals benefit from lower compression rates to reduce staleness. Striking a balance between these competing factors is essential for optimizing performance.

In this paper, we address a critical yet underexplored challenge in AFL: the intertwined impact of aggregation interval and gradient compression rate on both training efficiency and communication cost. We propose FedJACA, the first AFL framework that jointly and adaptively optimizes the aggregation interval and gradient compression rate to achieve a balanced trade-off between training efficiency and communication cost. Our main contributions are:

- We model their combined influence on training dynamics and formulate a joint optimization problem that captures the trade-off between gradient utility, staleness, and communication cost under the dual constraints.
- We design a customized solution algorithm based on the Trust Region Newton method to efficiently derive high-quality aggregation and compression policies.
- Extensive experiments on four benchmarks show that FedJACA outperforms state-of-the-art baselines, reducing training time by 40% and communication cost by 35%, while improving accuracy.

* The dataset used in this study is publicly available. However, the source code cannot be released due to privacy considerations.

II. Related Work

Aggregation Optimization. A substantial body of work has investigated aggregation optimization and model staleness in asynchronous federated learning (AFL). Early studies such as FedAsync [1] enabled fully asynchronous aggregation without synchronization constraints, while subsequent works introduced partial synchronization to balance efficiency and

*Corresponding author: Weigang Wu (email: wuweig@mail.sysu.edu.cn)

stability, e.g., periodic aggregation in FedPer [6] and buffered aggregation in BuffAVFL [7].

To mitigate staleness more directly, several methods designed staleness-aware aggregation strategies, including temporal weight fading [8], long-term adaptive aggregation [9], and selective aggregation of minimally stale updates [10]. However, most existing approaches rely on static or heuristic aggregation intervals, limiting their ability to adapt to dynamic system heterogeneity. In contrast, our work adaptively optimizes aggregation intervals in a principled manner and further integrates this optimization with gradient compression.

Gradient Compression. Gradient compression is a core technique for improving communication efficiency in bandwidth-constrained FL systems. Prior works proposed various compression schemes for AFL, including joint sparsification and quantization [5], channel state information compression [11], and quantization-oriented asynchronous compression [12].

More recent studies explored adaptive compression strategies to accommodate heterogeneous client conditions, such as bandwidth-aware quantization [13], energy-aware sparsification [14], data-dependent compression [15], and learning-regime-aware compression [16]. Although these methods improve compression adaptivity, they typically optimize compression rates independently of aggregation frequency.

Different from existing literature, which treats aggregation interval and compression rate as separate design dimensions, our proposed FedJACA framework jointly and adaptively optimizes these two interdependent factors, enabling a more effective balance between model staleness, communication cost, and training efficiency in AFL.

III. Preliminaries and Problem Formulation

In this section, we first present the fundamental concepts of Federated Learning (FL) and AFL, including aggregation interval and gradient compression. Then we formulate a joint optimization problem that captures the complex interdependence between aggregation intervals and gradient compression rates in AFL under resource constraints.

A. FL and Asynchronous Aggregation

We consider a standard FL setting with a centralized PS and N distributed clients $\mathcal{N}$. The PS maintains the global model $\mathbf{w} \in \mathbb{R}^d$, while each client $i \in \mathcal{N}$ trains a local model $\mathbf{w}_i$ on dataset $D_i = [\xi_{i,1}, \ldots, \xi_{i,|D_i|}]$. The local loss function is:

$$F_i(\mathbf{w}_i) := \frac{1}{|D_i|} \sum_{j=1}^{|D_i|} f(\mathbf{w}_i, \xi_{i,j}), \tag{1}$$

and the global objective is:

$$F(\mathbf{w}) = \frac{1}{N} \sum_{i \in \mathcal{N}} F_i(\mathbf{w}), \quad \mathbf{w}^* = \arg\min_{\mathbf{w} \in \mathbb{R}^d} F(\mathbf{w}). \tag{2}$$

At each round, the PS distributes $\mathbf{w}^t$, and client i initializes $\mathbf{w}_i^{t,0} = \mathbf{w}^t$. After α_i iterations of local training,

$$\mathbf{w}_i^{t,j+1} = \mathbf{w}_i^{t,j} - \eta_l \nabla F_i(\mathbf{w}_i^{t,j}, \xi_i^{i,j}), \tag{3}$$

yielding the local gradient:

$$\mathbf{g}_i^t = \mathbf{w}_i^{t,0} - \mathbf{w}_i^{t,\alpha_i}. \tag{4}$$

In AFL, clients upload gradients asynchronously. The PS aggregates a subset $\mathbf{S}^t$ after aggregation interval $\tilde{T}_t$ as:

$$\mathbf{w}^{t+1} = \mathbf{w}^t - \frac{\eta_g}{|\mathbf{S}^t|} \sum_{i \in \mathbf{S}^t} \tilde{\mathbf{g}}_i^{t-\tau_i^t}, \tag{5}$$

where τ_i^t is the staleness:

$$\tau_i^t = t - \max_{t'<t} \{t' | i \in \mathbf{S}^{t'}\}. \tag{6}$$

To reduce communication cost, we adopt top_k sparsification [3] method, with compression rate $\delta = \frac{k}{d}$ and

$$\tilde{\mathbf{g}}_i^t = C_{\delta_i^t}(\mathbf{g}_i^t). \tag{7}$$

The client execution time is then

$$l_i^t = l_i^{train} + \delta_i^t l_i^{comm}, \tag{8}$$

where l_i^{train} is the computation time for local training and l_i^{comm} the transmission time for a full gradient. Since upload bandwidth is typically the bottleneck [17], communication time dominates total execution time.

B. Joint Optimization Problem Formulation

We formulate a joint optimization problem for AFL to capture the interdependence of aggregation intervals and compression rates under limited training time and bandwidth. The objective is to maximize the utility of aggregated gradients while satisfying the resource constraints, particularly in edge scenarios where communication dominates execution time.

We define $\rho_i(t)$ as the index of the most recent global aggregation round completed by client i before round t:

$$\rho_i(t) = \max_{t'<t} \{t' | i \in \mathbf{S}^{t'}\}, \;\; \rho_i(1) = 1. \tag{9}$$

Due to asynchronous updates, gradients at the PS may have different staleness levels. Following [1], the aggregation weight β_i^t is defined as:

$$\text{if } l_i^t \leq \sum_{j=\rho_i(t)}^{t} \tilde{T}_j, \; \beta_i^t = (t - \rho_i(t) + 1)^{-0.5}; \text{else } \beta_i^t = 0. \tag{10}$$

Here, $\tilde{T}_t$ is the aggregation interval between the $(t-1)$-th and t-th global rounds. $\beta_i^t = 0$ indicates that some gradients transmitted from the client i are incomplete, thus the gradient $\tilde{\mathbf{g}}_i^t$ cannot participate in the aggregation. When $\beta_i^t > 0$, $\rho_i(t)$ is updated as $\rho_i(t) = t$.

Since gradients with lower staleness lead to more substantial improvements in model performance [18], we define the effective gradient information aggregated at round t as:

$$GI_t = \sum\nolimits_{i \in \mathbf{S}^t} \beta_i^t \, \delta_i^t \, d, \tag{11}$$

where d is the total number of original gradients.

We define b as the bandwidth cost of transmitting one full model, T as the total training time budget, and B as the bandwidth budget. The joint optimization problem is formulated:

$$\max \sum\nolimits_{t=1}^{t_{max}} GI_t$$
$$s.t. \begin{cases} \sum_{t=1}^{t_{max}} \tilde{T}_t \leq T \\ \sum_{t=1}^{t_{max}} \sum_{i \in \mathbf{S}^t} sign(\beta_i^t) \, \delta_i^t \, b \leq B \\ l_i^t = l_i^{train} + \delta_i^t \, l_i^{comm} \\ \rho_i(t) = \max_{t' < t} \{t' | i \in \mathbf{S}^{t'}\}, \ \rho_i(1) = 1 \\ \text{if } l_i^t \leq \sum_{j=\rho_i(t)}^{t} \tilde{T}_j, \ \beta_i^t = (t - \rho_i(t) + 1)^{-0.5}; \\ \text{else } \beta_i^t = 0 \\ \text{if } \beta_i^t > 0, \ \rho_i(t) = t \\ \tilde{T}_t \in [\tilde{T}_{min}, \tilde{T}_{max}], \ \delta_i^t \in (0, 1) \end{cases} \tag{12}$$

The last constraints specify the feasible ranges for $\tilde{T}_t$ and δ_i^t.

IV. Design of FedJACA

In this section, we design the TRON-based solution algorithm that solves the joint optimization problem to obtain the optimal aggregation intervals and gradient compression rates. Then, we describe the framework design of FedJACA.

A. TRON-based Solution Algorithm

The proposed joint optimization problem (Eq. 12) poses unique challenges due to its high-dimensional continuous variables, nonlinear and non-differentiable constraints, and the asynchronous nature of AFL. Standard off-the-shelf solution algorithms are insufficient to handle these features effectively. Therefore, to efficiently solve the problem, we design a customized solution algorithm based on the Trust Region Newton (TRON) method [19], tailored specifically for the AFL scenario. To address the specific challenges of AFL, we incorporate three key strategies into our solution algorithm.

Smooth Approximation Strategy. The original optimization problem involves non-smooth elements such as the sign(·) function in the bandwidth constraint and the piecewise-defined staleness-aware aggregation weight. To enable effective gradient-based updates within TRON, we introduce a smooth approximation strategy that replaces such terms with smooth surrogates:

$$sign(x) \approx tanh(\gamma x), \ \gamma \gg 1, \tag{13}$$

This approximation enables continuous gradient flow while preserving the decision semantics of whether a client's gradient is included in aggregation.

Multi-Timescale Decomposition Strategy. The optimization variables, aggregation intervals $\tilde{T}_t$ and per-client compression rates δ_i^t, operate at different temporal granularities. Aggregation intervals typically remain effective across longer time windows, while compression rates should react to short-term client conditions. Hence, we adopt a hierarchical decomposition: 1) Outer optimization: Aggregation intervals $\tilde{T}_t$ are updated less frequently; 2) Inner optimization: Compression rates δ_i^t are dynamically refined based on recent staleness and bandwidth feedback. This design reduces convergence instability caused by co-optimizing fast- and slow-changing variables, and improves solution quality under heterogeneous dynamics.

Compression Clustering Strategy. Directly optimizing δ_i^t for all clients in all rounds incurs a large number of variables, hindering scalability. We propose a compression clustering mechanism, where clients with similar bandwidth and computation profiles are grouped and share a common

Algorithm 1 FedJACA

Parameter Server:
  Receive l_i^{train} from clients
  Measure l_i^{comm} with clients
  Obtain the optimal set $\{\tilde{T}_t\}$ and $\{\delta_i^t\}$ according to Eq. 12
  Deliver $\{\delta_i^t\}$ to client i
  Distribute w^0 to all clients and start them
  Initialize wall-clock timer $\Psi \leftarrow 0$, and aggregation timer $\Psi_{agg} \leftarrow 0$
while $\Psi \leq T$ **do**
    Continuously receive $\widetilde{\mathbf{g}}_i^{t-\tau_i^t}$ from clients set S^t
    if $(\Psi - \Psi_{agg}) \in \{\tilde{T}_t\}$ **then**
      $\mathbf{w}^{t+1} \leftarrow \mathbf{w}^t - \frac{\eta_g}{|\mathbf{S}^t|} \sum_{i \in \mathbf{S}^t} \widetilde{\mathbf{g}}_i^{t-\tau_i^t}$
      for $i \in \mathbf{S}^t$ **do**
        Send new global model $\mathbf{w}^{t+1}$ to client i
      end for
      $\Psi_{agg} \leftarrow \Psi$
      $t \leftarrow t + 1$
    end if
    $\Psi \leftarrow \Psi + 1$
end while
  Notify all clients to **STOP**

Client:
  Measure l_i^{train} and report it to the PS
  Measure l_i^{comm} with the PS
  Receive $\{\delta_i^t\}$ from the PS
while not **STOP do**
    Receive $\mathbf{w}^t$ from the PS
    Set $\mathbf{w}_i^{t,0} \leftarrow \mathbf{w}^t$
  for each local iteration $j \in \{0, 1, \ldots, \alpha_i - 1\}$ **do**
      $\mathbf{w}_i^{t,j+1} \leftarrow \mathbf{w}_i^{t,j} - \eta_i \nabla F_i(\mathbf{w}_i^{t,j}; \xi_i^{t,j})$
  end for
    Compute gradient $\mathbf{g}_i^t \leftarrow \mathbf{w}_i^{t,0} - \mathbf{w}_i^{t,\alpha_i}$
    Compute compressed gradient $\widetilde{\mathbf{g}}_i^t \leftarrow C_{\delta_i^t}(\mathbf{g}_i^t)$
    Send $\widetilde{\mathbf{g}}_i^t$ to the PS
end while

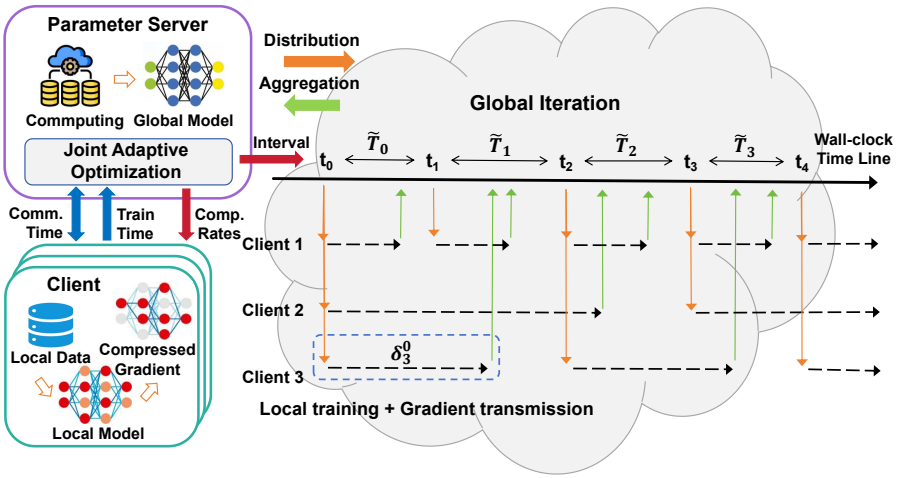

Fig. 1. Work procedure of the proposed FedJACA.

compression policy. Specifically, a lightweight k-means clustering is performed before training based on $\{l_i^{train}, l_i^{comm}\}$, and compression rates are optimized at the cluster level instead of per client. This significantly reduces the variable dimension and speeds up solution.

In summary, unlike directly adopting existing solvers, our solution algorithm incorporates smooth approximations, multi-timescale variable control, and compression clustering strategies to efficiently handle non-convexity and efficiency, enabling high-quality solutions under complex constraints.

B. Framework Design

The pseudo-code of the complete FedJACA framework is presented in Alg. 1. The implementation comprises two main phases: 1) Pre-Training Optimization Phase: The PS obtains the optimal aggregation interval set $\{\tilde{T}_t\}$ and the client-specific compression rate set $\{\delta_i^t\}$ by solving the joint optimization problem, subject to training time and bandwidth constraints; 2) Collaborative Training Phase: The PS and all clients collaboratively apply the optimized hyperparameter settings to perform the global training of AFL.

During the pre-training phase, each client performs a series of preliminary tests to measure two key metrics: the average local training time l_i^{train}, and the communication time l_i^{comm} for full gradient uploads to the PS. These measurements reflect the heterogeneity in computation and network conditions across clients. Based on these metrics and predefined resource constraints (e.g., total training time budget and bandwidth limits), the PS solves the optimization problem in Eq. 12 to determine the optimal values of $\tilde{T}_t$ and δ_i^t.

Once the optimal hyperparameters are determined, they are distributed to the clients, initiating the collaborative training process. The overall workflow of FedJACA is illustrated in Fig. 1. After receiving the initial global model from the PS, all participating clients independently perform their training process using their local data. Once a client has completed its local training for a predetermined number of iterations, it compresses the computed gradients using the optimized compression rate and transmit them asynchronously to the PS. Unlike traditional synchronous FL, the PS does not wait for all clients. Instead, it performs aggregation after the optimized aggregation interval elapses and update the global model accordingly. After that, a new global model is distributed to participating clients to update their local models.

This procedure has negligible extra overhead compared to existing AFL methods with gradient compression. Only a single local computation at the PS is required to determine the optimal hyperparameter settings, along with one bidirectional communication round to collect training and communication time metrics and distribute the optimal compression rates to clients. Both operations are completed prior to training. FedJACA does not require external nodes in the PS architecture, the server handles the calculations, while in decentralized training, any client can temporarily act as the server. A larger number of clients does not induce extra cost, thus FedJACA has good scalability.

V. Experiments

A. Experiment Settings

System simulation. We simulate 30 clients with *non-IID* data partitions generated via Dirichlet distribution ($\alpha = 0.4$) [6], [15]. Client total execution time is the sum of local training time and communication time. Local training workload is evaluated in terms of FLOPs using `thop` (available at https://pypi.org/project/thop/), with device-specific processing capabilities yielding training time of 2–31s. Bandwidth is varied between 0.15–1 Mb/s to reflect diverse network conditions.

Datasets and models. Experiments are conducted on MNIST, FMNIST, CIFAR-10, and CIFAR-100. We use a CNN (3.8M parameters) for MNIST/FMNIST, AlexNet (3.9M) for CIFAR-10, and VGG11 (9.3M) for CIFAR-100. Training uses SGD (lr=0.01) for CNN/VGG11 and Adam (lr=0.0001) for AlexNet, with batch sizes of 64 (MNIST/FMIST/CIFAR-100) and 32 (CIFAR-10).

Baselines. We compare FedJACA with: (1) **FedPer** [6], periodic aggregation with fixed compression; (2) **FedLuck**

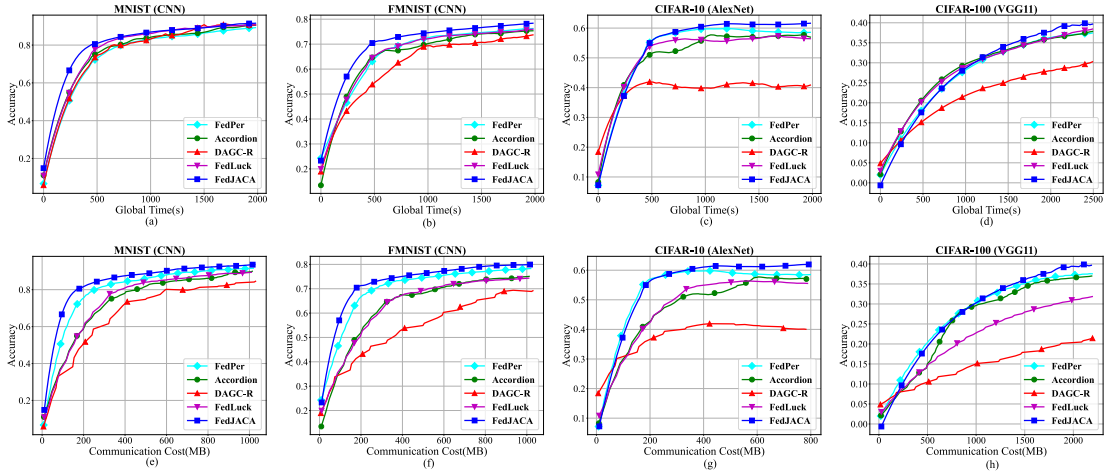

Fig. 2. The accuracy curve of FedJACA and the four baselines with the global time and communication cost.

TABLE I
THE EXPERIMENTAL METRIC SCORES FOR REACHING THE TARGET ACCURACY FOR FEDJACA AND FOUR BASELINES. (BOLD INDICATE THE MINIMUM VALUE OF THE COLUMN, WHICH IS THE OPTIMAL VALUE)

Method	Completion Time (s)				Communication Cost (MB)			
	MNIST (80%)	FMNIST (75%)	CIFAR-10 (57%)	CIFAR-100 (35%)	MNIST (80%)	FMNIST (75%)	CIFAR-10 (57%)	CIFAR-100 (35%)
FedPer	751.05	1622.30	571.51	1770.00	262.13	553.96	**207.21**	1496.99
Accordion	720.17	1830.44	1020.24	1770.42	458.14	1005.95	587.01	1584.69
DAGC-R	660.16	–	–	–	597.71	–	–	–
FedLuck	570.14	1740.42	1590.38	–	392.85	1163.10	1102.60	–
FedJACA	**450.11**	**1110.27**	**570.14**	**1590.38**	**168.87**	**401.07**	217.14	**1405.87**

[20], adaptive local updates and compression; (3) **DAGC-R** [15], data-volume–based adaptive compression; (4) **Accordion** [16], regime-aware adaptive compression.

Metrics. Performance is evaluated by: (1) **Test accuracy** of the global model at each round; (2) **Completion time**, i.e., total training time to reach a target accuracy; (3) **Communication cost**, measured as cumulative gradient traffic.

Compression settings. We adopt top_k sparsification [3]. For FedPer, compression rates are set as 0.03. Given the total time, as well as the per-client training and communication time, we calculate the overall bandwidth consumption of FedPer. This value is then used as the bandwidth constraint of adaptive compression algorithms for fair comparison. For ACCORDION, compression rates are set as $\delta_{\min} = 0.02$ and $\delta_{\max} = 0.06$.

B. Comparative Experiments

Training Performance. Fig. 2 shows that FedJACA consistently achieves the highest accuracy under equal training time and communication cost across all four datasets. For MNIST and FMNIST, it outperforms all baselines throughout the entire process. Overall, FedJACA achieves optimal performance across all datasets, validating both the effectiveness and generalization capability of our approach.

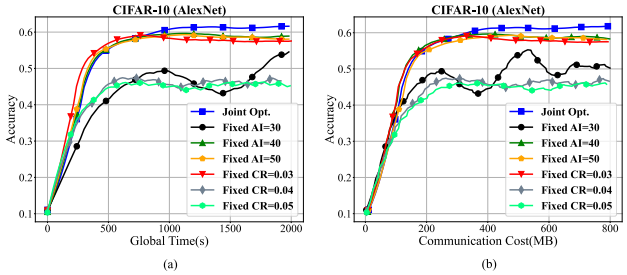

Fig. 3. The accuracy of FedJACA under different optimization settings with the global time and communication cost.

Resource Efficiency. Table I reports the time and cost to reach target accuracy. For instance, on MNIST, FedJACA attains 80% accuracy with 40.0% less training time and 35.5% lower communication cost than FedPer. Notably, FedJACA demonstrates the most efficient resource utilization in the majority of test scenarios, underscoring its advantage in constrained environments.

Experiment Analysis. These gains stem from FedJACA's joint optimization strategy, which adaptively tunes the aggregation interval and compression rate to achieve an optimal balance between efficiency and cost. This joint adaptation ensures that each compressed gradient is aggregated at its

most appropriate timing, thereby minimizing staleness and maximizing the utility of gradient information. Joint adaptive optimization maximizes the efficiency of resource utilization and global model performance improvement.

TABLE II
ACCURACY OF FEDJACA UNDER DIFFERENT OPTIMIZATION SETTINGS OVER 2,000S GLOBAL TIME.

Model@Dataset	Opt. setting	Hyperparameter	Accuracy
AlexNet @CIFAR-10	Joint Opt.	Optimized	**61.61%**
	Fixed AI	30	49.74%
		40	58.90%
		50	58.07%
	Fixed CR	0.03	57.44%
		0.04	46.08%
		0.05	45.48%

C. Ablation Studies

The Necessity of Joint Optimization. To investigate the importance of joint optimization, we compare FedJACA with two variants: (i) *Fixed CR*, which fixes compression rates $\delta = \{0.03, 0.04, 0.05\}$ while optimizing aggregation interval, and (ii) *Fixed AI*, which fixes aggregation intervals $\tilde{T} = \{30, 40, 50\}$ while optimizing compression rate. As shown in Fig.3 and TableII, FedJACA consistently outperforms both variants in accuracy, training time, and communication cost, confirming the necessity of joint optimization.

Experiment Analysis. The superior performance of joint optimization arises from its ability to achieve a more balanced trade-off between training efficiency and communication cost. In contrast, optimizing only one parameter yields suboptimal outcomes: fixed compression ignores staleness effects, while fixed aggregation overlooks communication efficiency. Moreover, our results reveal that conservative compression does not guarantee higher accuracy, as aggregating stale gradients can mislead convergence.

VI. CONCLUSION

This paper focuses on the intertwined impact of aggregation interval and gradient compression in AFL. We propose FedJACA, a framework that jointly and adaptively optimizes both factors via a TRON-based solution, effectively balancing training performance and communication cost. Experiments show that FedJACA reduces communication cost and training time, outperforming state-of-the-art baselines. To the best of our knowledge, this is the first work to unify aggregation and compression optimization in AFL, offering a more efficient and practical solution for resource-constrained scenarios.

ACKNOWLEDGMENTS

This research is partially supported by National Natural Science Foundation of China (62372487) and Guangdong Natural Science Foundation of China (No. 2024A1515010378).

This paper only employs AI tools for the polishing and refinement of textual expression.

REFERENCES

[1] C. Xie, S. Koyejo, and I. Gupta, "Asynchronous federated optimization," *arXiv preprint arXiv:1903.03934*, 2019.

[2] C. Chen, Y. Zhao, Z. Zhang, W. Li, and J. Wu, "Towards efficient and scalable asynchronous federated learning via stragglers version control," *IEEE Transactions on Mobile Computing*, 2025.

[3] S. U. Stich, J.-B. Cordonnier, and M. Jaggi, "Sparsified sgd with memory," in *Advances in neural information processing systems*, 2018, pp. 4452–4463.

[4] H. Cui, Z. Qu, B. Ye, B. Tang, T. Zhuang, X. Wang, and Y. Zeng, "Lightweight adaptive quantization algorithms for federated learning with heterogeneous clients," *IEEE Transactions on Mobile Computing*, 2025.

[5] J. Jia, J. Liu, C. Zhou, H. Tian, M. Dong, and D. Dou, "Efficient asynchronous federated learning with sparsification and quantization," *Concurrency and Computation: Practice and Experience*, vol. 36, no. 9, p. e8002, 2024.

[6] C.-H. Hu, Z. Chen, and E. G. Larsson, "Scheduling and aggregation design for asynchronous federated learning over wireless networks," *IEEE Journal on Selected Areas in Communications*, vol. 41, no. 4, pp. 874–886, 2023.

[7] L. Hu, J. Jia, and S. Cheng, "Buffavfl: Buffered asynchronous vertical federated learning," *IEEE Internet of Things Journal*, vol. 12, no. 19, pp. 40 944–40 954, 2025.

[8] L. You, S. Liu, Y. Chang, and C. Yuen, "A triple-step asynchronous federated learning mechanism for client activation, interaction optimization, and aggregation enhancement," *IEEE Internet of Things Journal*, vol. 9, no. 23, pp. 24 199–24 211, 2022.

[9] S. Sun, Z. Zhang, Q. Pan, M. Liu, Y. Wang, T. He, Y. Chen, and Z. Wu, "Staleness-controlled asynchronous federated learning: Accuracy and efficiency tradeoff," *IEEE Transactions on Mobile Computing*, vol. 23, no. 12, pp. 12 621–12 634, 2024.

[10] Z. Zhou, Y. Li, X. Ren, and S. Yang, "Towards efficient and stable k-asynchronous federated learning with unbounded stale gradients on non-iid data," *IEEE Transactions on Parallel and Distributed Systems*, vol. 33, no. 12, pp. 3291–3305, 2022.

[11] Y. Dong, H. Zhang, G. Chen, X. Fan, V. C. Leung, and X. Hu, "Model splitting enhanced communication-efficient federated learning for csi feedback," *IEEE Transactions on Vehicular Technology*, vol. 74, no. 12, pp. 19 766–19 771, 2025.

[12] Y. Liu, P. Huang, F. Yang, K. Huang, and L. Shu, "Quasyncfl: Asynchronous federated learning with quantization for cloud–edge–terminal collaboration enabled aiot," *IEEE Internet of Things Journal*, vol. 11, no. 1, pp. 59–69, 2023.

[13] S. Wang, Y. Xu, C. You, M. Shao, and T. Q. Quek, "Communication-efficient federated learning by quantized variance reduction for heterogeneous wireless edge networks," *IEEE Transactions on Mobile Computing*, vol. 24, no. 8, pp. 7632–7647, 2025.

[14] A. Asheralieva and D. Niyato, "Effective uav-aided asynchronous decentralized federated learning with distributed, adaptive and energy-aware gradient sparsification," *IEEE Internet of Things Journal*, vol. 12, no. 14, pp. 27 461–27 480, 2025.

[15] R. Lu, Y. Jiang, Y. Mao, C. Tang, B. Chen, L. Cui, and Z. Wang, "Data-aware gradient compression for fl in communication-constrained mobile computing," *IEEE Transactions on Mobile Computing*, vol. 24, no. 4, pp. 2755–2768, 2024.

[16] S. Agarwal, H. Wang, K. Lee, S. Venkataraman, and D. Papailiopoulos, "Adaptive gradient communication via critical learning regime identification," *Proceedings of Machine Learning and Systems*, vol. 3, pp. 55–80, 2021.

[17] G. Liu, W. Lin, T. Huang, F. Shi, W. Wu, and L. Shen, "Adaptivefl: Communication-adaptive federated learning under dynamic bandwidth," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 36, no. 9, pp. 17 199–17 211, 2025.

[18] W. Zhang, S. Gupta, X. Lian, and J. Liu, "Staleness-aware async-sgd for distributed deep learning," *arXiv preprint arXiv:1511.05950*, 2015.

[19] C.-J. Lin, R. C. Weng, and S. S. Keerthi, "Trust region newton method for large-scale logistic regression." *Journal of Machine Learning Research*, vol. 9, no. 4, pp. 627–650, 2008.

[20] J. Song, J. Luo, R. Lu, S. Xie, B. Chen, and Z. Wang, "A joint approach to local updating and gradient compression for efficient asynchronous federated learning," in *European Conference on Parallel Processing*, 2024, pp. 196–211.