Artificial Intelligence and Computing Innovations

Intelligence Science Publishing

https://www.ispublishing.com

ISSN: 3070-4944

Vo.4, No.2, 2024

Hierarchical Communication and Computation Scheduling for Efficient Federated Learning in Cloud-Edge Collaborative Intelligent Systems

Ru Meng[1]*

[1]*Carnegie Mellon University, Pittsburgh, USA*

**Corresponding author: mengru.christine@gmail.com*

Abstract: As federated learning and cloud-edge collaborative training are increasingly applied in distributed intelligent computing, issues such as high communication overhead, strong heterogeneity of computing resources, and insufficient efficiency of multi-layer node collaboration have gradually become key factors restricting system performance improvement. To address these issues, this paper studies a joint scheduling optimization method for communication and computation, addressing the distributed training needs in cloud computing scenarios, and constructs a unified modeling framework suitable for collaborative training across terminals, edge nodes, and the cloud. In terms of method design, firstly, starting from the multi-layer collaborative structure, a systematic description of client access relationships, edge node activation states, and cloud coordination processes is provided, incorporating communication latency, local computation overhead, and regional forwarding costs during training into a unified latency model. Secondly, to improve the effectiveness of node participation and the rationality of resource allocation, a utility evaluation mechanism integrating data quality, training status, and resource consumption is introduced to achieve dynamic selection of participating nodes under constraints of limited bandwidth and computational budget. Subsequently, combining the characteristics of cloud-edge layered training, a joint optimization objective for resource-constrained environments is further constructed, considering latency control, training quality, and energy consumption costs collaboratively, and enhancing the organizational capacity and execution feasibility among multiple nodes through hierarchical scheduling constraints. Finally, an aggregation mechanism combining latency and data contribution is designed during the model update phase to mitigate the adverse effects of heterogeneous node differences on the global model evolution. Experimental results show that the proposed method performs well in terms of accuracy, overall performance, and system coordination capabilities, effectively improving resource utilization and overall training efficiency in federated learning and cloud-edge collaborative training scenarios.

Keywords: Federated optimization; cloud-edge collaboration; joint resource scheduling; distributed model aggregation

1. Introduction

With the continuous development of cloud computing, edge computing, and artificial intelligence technologies, the demand for collaborative training in distributed intelligent systems is constantly growing. Traditional centralized model training typically relies on uploading massive amounts of terminal data to the cloud for unified processing [1, 2]. While this approach facilitates centralized resource scheduling, it also faces challenges such as high data transmission pressure, high privacy risks, significant network congestion, and insufficient real-time performance on the edge. Against this backdrop, federated learning, by retaining local data storage and exchanging only model parameters or gradient information, provides a new implementation path for distributed intelligent modeling [3]. Cloud-edge collaborative training further combines the powerful global optimization capabilities of the cloud with the rapid response advantages of the edge, which is close to the data source, becoming a crucial technological foundation supporting scenarios such as the Internet of Things, the Industrial Internet, the Internet of Vehicles, and smart healthcare. Therefore, conducting systematic research on federated learning and cloud-edge collaborative training has significant theoretical and practical value [4].

However, in actual deployment, federated learning and cloud-edge collaborative training are not simply about distributing training tasks across different nodes; their core bottleneck often lies in the coupling constraints of communication and computing resources. On the one hand, significant differences exist in bandwidth, link quality, and latency levels between edge nodes, terminal devices, and cloud servers. Frequent parameter interactions during model updates can easily lead to accumulated communication latency, thus slowing down the global training process [5]. On the other hand, the computing power, energy consumption constraints, task load, and local data scale of participating nodes are also highly heterogeneous, resulting in significant differences in node completion times across training rounds and a tendency to create a tailing effect. The mutual influence and constraints between communication and computing resources mean that overall system performance no longer depends on single-dimensional optimization but on the global coordination capability under the combined effect of multiple factors.

Under these conditions, researching the joint scheduling optimization problem of communication and computing is of great significance. Focusing only on communication overhead compression while ignoring differences in node computing power may result in some nodes frequently participating but failing to complete local training promptly, thus reducing overall efficiency. Conversely, emphasizing only computational load balancing while ignoring network state fluctuations may cause long-term obstruction of the model synchronization process, affecting training stability and convergence efficiency. Especially in cloud-edge collaborative environments, issues such as task offloading, hierarchical model updates, node selection, bandwidth allocation, computational resource allocation, and synchronization mechanism design are intertwined, forming typical cross-level, multi-objective, and strongly constrained optimization scenarios. How to coordinate communication and computation processes in complex and dynamic environments to improve training efficiency, reduce resource consumption, and ensure stable system operation has become a key issue driving the large-scale application of federated intelligent training.

From a broader application perspective, joint scheduling optimization of communication and computation not only relates to the training speed and resource utilization of federated learning systems but also directly affects the implementation capability and service quality of cloud-edge intelligent collaborative systems. In real-world environments with complex network conditions, significant differences in device capabilities, and real-time changes in task requirements, efficient joint scheduling mechanisms help enhance the system's adaptability to dynamic scenarios, improve the continuity, reliability,

and economy of model training, and provide stronger technical support for privacy protection, low-latency services, and large-scale terminal collaboration. Research in this direction can not only enrich the theory of distributed intelligent optimization and resource collaboration methods but also provide important references for the construction of next-generation cloud-edge integrated artificial intelligence infrastructure.

2. BackGround

Federated learning is a collaborative modeling paradigm for distributed data environments. Its core idea is to update the global model by periodically exchanging model parameters, gradient information, or intermediate representations, while retaining the original data locally at each participating node. This training mode changes the traditional process of centralized data aggregation followed by unified modeling, shifting the model learning process from centralized data flow to distributed parameter collaboration. With the widespread deployment of terminal devices, edge servers, and cloud platforms in intelligent sensing, business analysis, and real-time decision-making, federated learning has gradually evolved from a simple privacy-preserving training framework into a crucial foundational mechanism supporting multi-level intelligent collaboration [6, 7]. In this process, the differences among different participants in data distribution, device performance, network conditions, and training requirements are constantly widening. This means that system design is no longer limited to the model update rules themselves, but needs to further consider node participation mechanisms, resource constraint expressions, and cross-level collaborative structures. Table 1 summarizes the key elements in federated learning and cloud-edge collaborative training scenarios, which together constitute the basic background of the joint scheduling problem of communication and computation.

Table 1. Key Background Elements in Federated Learning and Cloud-Edge Collaborative Training Scenarios

Key Element	Main Connotation	Impact on Joint Scheduling
Data distribution characteristics	Data are distributed across terminal devices or edge nodes and usually exhibit non-IID properties.	Affects local training effectiveness and the stability of global aggregation.
Node heterogeneity	Different nodes vary significantly in computing power, storage capacity, energy consumption, and available participation time.	Leads to inconsistent training completion times and more complex resource allocation.
Dynamic network environment	Link bandwidth, latency, packet loss rate, and connection stability fluctuate over time.	Influences model transmission efficiency and the selection of synchronization timing.
Cloud-edge hierarchical architecture	The cloud is responsible for global coordination, while the edge undertakes nearby aggregation and rapid response.	Determines task decomposition methods and multi-level resource coordination patterns.
Timeliness of training tasks	Different services impose different requirements on update frequency, response time, and service continuity.	Affects the setting of scheduling objectives and optimization priorities.

Cloud-edge collaborative training is a hierarchical intelligent training model formed under the deep integration of cloud computing and edge computing. Its goal is not merely to extend the training process from the central node to the edge node, but to achieve more efficient model building and service support through the synergy of global management capabilities in the cloud, near-source processing capabilities on the edge, and data perception capabilities on the terminal. Compared with traditional single-layer distributed training, cloud-edge collaborative training emphasizes the role division and task collaboration

among multiple layers of nodes. For example, the cloud handles global parameter integration and policy control, the edge performs regional aggregation, task caching, and local inference acceleration, and the terminal device is responsible for data collection and local updates. In this system, the training process exhibits significant hierarchical, dynamic, and coupled characteristics. Communication occurs not only between the terminal and the edge but also between the edge and the cloud; the computation process involves not only local model iteration but also task migration, model splitting, resource preemption, and service allocation. Therefore, the combination of federated learning and cloud-edge collaborative training expands intelligent model training from a single update mechanism problem to a comprehensive optimization problem encompassing network organization, resource coordination, and system regulation. This also provides a clear theoretical foundation and source of questions for subsequent research on joint scheduling of communication and computation.

3. Methodology

In the cloud-edge collaborative federated training scenario, communication delay and local computation latency are jointly modeled as a coupled scheduling problem rather than two isolated resource management tasks, because the overall convergence pace of distributed optimization is governed by the slowest effective update path across participating nodes. Considering a set of client devices, edge aggregators, and a cloud coordinator, the training system is represented by a three-layer collaboration structure in which each client first performs local updates, each edge server executes regional aggregation, and the cloud completes global fusion over selected edge domains. Such a formulation is introduced to explicitly preserve the hierarchical dependency between transmission and execution processes, thereby avoiding the oversimplified assumption that all nodes communicate directly with a single server under homogeneous network conditions. This paper also presents the overall model architecture, as shown in Figure 1.

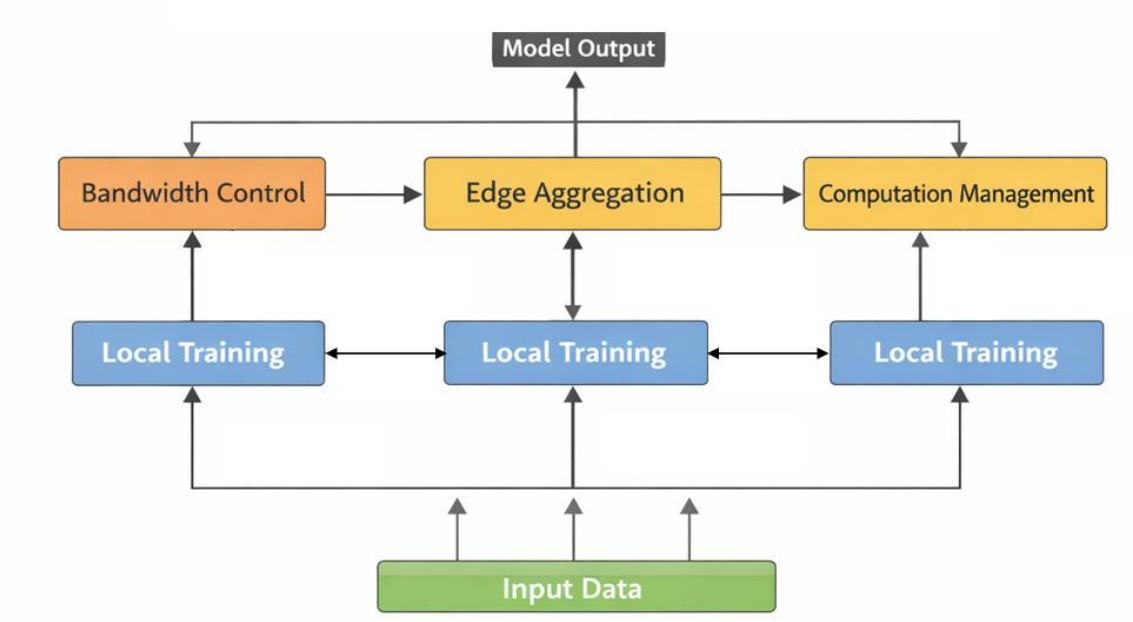

Figure 1. Overall model architecture

Let $\mathcal{C}$, $\mathcal{E}$, and $\mathcal{S}$ denote the client set, edge set, and cloud coordinator, respectively, while the decision variable $a_{i,j} \in \{0,1\}$ indicates whether client $i \in \mathcal{C}$ is associated with edge server $j \in \mathcal{E}$, and the variable $b_j \in \{0,1\}$ denotes whether edge server j is activated in the current round. The hierarchical scheduling topology is written as:

$$\sum_{j\in\mathcal{E}} a_{i,j} = 1, \quad \forall i \in \mathcal{C} \tag{1}$$

$$a_{i,j} \le b_j, \quad \forall i \in \mathcal{C}, \forall j \in \mathcal{E} \tag{2}$$

where the first constraint ensures unique client-edge assignment and the second one guarantees that only activated edge servers can receive local updates. To reflect the fact that communication and computation jointly determine the effective participation quality of a client, the round-wise latency of node i associated with edge server j is decomposed into uplink transmission time, local training time, and regional forwarding time as:

$$T_{i,j}^{(r)} = \frac{M_i^{(r)}}{R_{i,j}^{(r)}} + \frac{\tau_i D_i \kappa_i}{f_i^{(r)}} + \frac{\widetilde{M}_j^{(r)}}{\widetilde{R}_j^{(r)}} \tag{3}$$

in which $M_i^{(r)}$ denotes the uplink model size of client i in round r, $R_{i,j}^{(r)}$ is the client-edge transmission rate, τ_i is the number of local epochs, D_i is the local sample volume, κ_i represents the average computational cost per sample, $f_i^{(r)}$ is the allocated computing frequency, and $\widetilde{M}_j^{(r)}$ together with $\widetilde{R}_j^{(r)}$ characterizes the edge-cloud forwarding process. This design is important because a client with strong computation but poor connectivity, or a client with high bandwidth but insufficient compute resources, may both become bottlenecks under multi-round federated coordination.

To improve scheduling efficiency under heterogeneous resources, the optimization target is constructed from a weighted combination of end-to-end latency, energy expenditure, and statistical contribution, since a purely delay-driven strategy may repeatedly favor a small subset of strong nodes and thus weaken the diversity of distributed data participation. For this reason, each client is endowed with a utility score that jointly measures update timeliness and training value. Denoting by $q_i^{(r)}$ the data quality coefficient of client i in round r, by $\ell_i^{(r)}$ the local loss after training, and by $E_i^{(r)}$ the total energy consumed by communication and computation, the node utility is defined as:

$$U_i^{(r)} = \alpha \frac{q_i^{(r)}}{T_{i,j}^{(r)}} + \beta \frac{1}{\ell_i^{(r)} + \epsilon} - \gamma E_i^{(r)} \tag{4}$$

where α, β, and γ control the importance of timeliness, optimization benefit, and energy cost, while $\epsilon > 0$ avoids numerical instability when the local loss becomes very small. Instead of selecting all available clients, the scheduler forms a participation subset that maximizes the cumulative utility under limited bandwidth and compute budgets, which is expressed as:

$$\max_{\boldsymbol{a},\boldsymbol{b},\boldsymbol{f},\boldsymbol{w}} \sum_{r=1}^{R} \sum_{i\in\mathcal{C}} w_i^{(r)} U_i^{(r)} \tag{5}$$

with $w_i^{(r)} \in \{0,1\}$ indicating whether client i is admitted into round r. Such an objective has practical significance because it balances system efficiency and model evolution simultaneously, preventing the scheduling policy from collapsing into a myopic mechanism that only minimizes instantaneous delay while ignoring long-term learning quality.

Meanwhile, resource constraints are embedded into the joint scheduler to preserve feasibility under dynamic cloud-edge environments. Since edge servers typically operate with finite spectrum resources and shared processing capacity, the aggregate bandwidth assigned to clients connected to edge server j cannot exceed its available communication budget $B_j^{(r)}$, and the total computation frequency allocated to admitted clients cannot surpass the instantaneous edge-assisted coordination capacity $F_j^{(r)}$. Accordingly, the scheduling constraints are formulated as:

$$\sum_{i\in\mathcal{C}} a_{i,j}\, w_i^{(r)} \rho_{i,j}^{(r)} \le B_j^{(r)}, \quad \forall j \in \mathcal{E} \tag{6}$$

$$\sum_{i\in\mathcal{C}} a_{i,j}\, w_i^{(r)} f_i^{(r)} \le F_j^{(r)}, \quad \forall j \in \mathcal{E} \tag{7}$$

where $\rho_{i,j}^{(r)}$ denotes the allocated bandwidth share of client i on edge server j. Beyond feasibility, these constraints also enforce a realistic competition mechanism among nodes, which is necessary because cloud-edge training platforms often encounter transient congestion, unstable uplinks, and fluctuating device availability. A further requirement is introduced on the round completion time, because the global synchronization barrier is determined by the slowest selected update path rather than the average delay over all clients. Therefore, the effective completion time of round r is modeled as:

$$T_{\text{round}}^{(r)} = \max_{i\in\mathcal{C}, j\in\mathcal{E}} \left(a_{i,j} w_i^{(r)} T_{i,j}^{(r)} \right) \tag{8}$$

which explicitly captures the straggler effect and provides a direct optimization handle for suppressing synchronization inefficiency in hierarchical federated training.

Finally, the model aggregation process is coupled with the scheduling policy so that node selection and parameter fusion remain statistically consistent. Rather than assigning equal aggregation weights to all admitted participants, the proposed formulation uses a delay-aware and data-aware weighting mechanism to reduce the influence of stale or weak updates while preserving contributions from informative local distributions. Let $\theta_i^{(r+1)}$ be the updated local model of client i after round r, and let $\Theta^{(r+1)}$ denote the aggregated global model. The cloud-edge collaborative aggregation rule is written as:

$$\Theta^{(r+1)} = \sum_{i\in\mathcal{C}} \eta_i^{(r)}\, w_i^{(r)} \theta_i^{(r+1)} \tag{9}$$

where the aggregation coefficient $\eta_i^{(r)}$ is normalized over all selected clients and can be designed as a function of local data volume, latency penalty, and update quality. Through this mechanism, scheduling decisions do not merely determine which nodes participate but also influence how strongly their updates shape the next global model. Such a coupling is meaningful because communication-computation co-optimization should ultimately serve the learning objective rather than remain a purely infrastructural adjustment. By integrating hierarchical association, latency decomposition, utility-driven selection, resource-constrained scheduling, and delay-aware aggregation into a unified framework, the proposed method establishes a mathematically coherent basis for joint optimization in federated learning and cloud-edge collaborative training systems, while also providing a clear pathway for subsequent algorithmic realization under dynamic multi-node environments.

4. Experimental Results and Analysis

4.1 Dataset

This paper uses Microsoft Azure VM Trace as the experimental foundation and constructs a reproducible experimental mapping mechanism for federated learning and cloud-edge collaborative training on top of real virtual machine running trajectories. Specifically, the virtual machine records are first reordered according to timestamps, and continuous running segments are divided into segments with fixed-length time windows. Each virtual machine trajectory segment that meets the minimum active duration constraint is mapped to a client to represent a single heterogeneous terminal or local training node participating in training. The CPU utilization, memory usage, disk I/O intensity, and network activity statistics of each client within a certain time window are jointly encoded into its local state vector to characterize the available computing power and load pressure of the node in the current training cycle. Subsequently, multiple clients are clustered according to the physical region where the virtual machines

are located, load similarity, and time correlation. Each cluster is defined as an edge collaborative node to simulate the local convergence and resource coordination process of multiple terminal training tasks on the edge side. All edge nodes are further connected to the cloud resource pool to form a three-level collaborative training topology of "client-edge-cloud". Based on this mapping method, the dynamic resource changes in the original VM trajectory are transformed into node heterogeneity, edge grouping relationships, and cross-time resource competition states under the federated training scenario. This enables the dataset to provide a feasible state foundation for the communication-computation joint scheduling problem.

In terms of training process construction, this paper defines each discrete time window as a training round. In the t-th round, each client decides whether to participate in local updates based on its current state vector and scheduling policy, and estimates the local training duration based on available CPU and memory resources. Each edge node is responsible for aggregating the model updates of clients within the cluster and uploading the aggregation results to the cloud for global synchronization. Communication latency is generated by "data transmission volume / effective bandwidth," where the data transmission volume is given by the model parameter scale or gradient magnitude, and the effective bandwidth is normalized based on the network load level within the time window corresponding to the VM trajectory. High load periods correspond to lower available bandwidth, thus forming time-varying communication overhead. Energy consumption consists of two parts: computational energy consumption and communication energy consumption. Computational energy consumption is calculated based on CPU utilization, runtime, and node power coefficient, while communication energy consumption is estimated based on the amount of data transmitted, link duration, and unit transmission power consumption, ultimately yielding a round-level total energy consumption index. For constructing supervised task labels, this paper does not directly use existing categories in the original dataset. Instead, it generates binary labels based on the scheduling results: when a training round simultaneously meets the preset accuracy improvement threshold and latency and energy consumption constraints, it is recorded as a high-quality scheduling sample; if the training round is completed but exhibits significant insufficient accuracy, excessive latency, or excessive energy consumption, it is recorded as a low-quality scheduling sample. Furthermore, ACC and F1 are used to evaluate the model's ability to distinguish between good and bad categories of scheduled samples, while LAT and EC measure the average communication and computation latency and overall energy consumption level, respectively. Through this approach, Azure VM Trace is no longer merely general cloud load data, but is specifically mapped as federated learning cloud-edge collaborative experimental data with client definition, edge organization, training round construction, latency generation, energy consumption calculation, and supervised label support.

4.2 Experimental Results and Analysis

To enhance the relevance of the comparisons, seven representative studies closely related to federated learning, edge computing, cloud-edge collaboration, communication optimization, and computation scheduling were selected and cross-referenced from two perspectives: method design and performance. The research covers key issues such as hierarchical aggregation, asynchronous updates, participant scheduling, resource allocation, dynamic training round control, and edge-cloud task scheduling, providing a direct comparative reference for the communication-computing joint scheduling optimization framework discussed in this paper. Under a unified evaluation caliber, Table 2 presents the comparison results between the related methods and the proposed method using four indicators. ACC and F1 are used to characterize model effectiveness, while LAT and EC reflect training latency and overall energy consumption, respectively.

The proposed method demonstrates stable overall performance, indicating that the constructed joint communication and computation scheduling mechanism can effectively adapt to the complex resource constraints of federated learning and cloud-edge collaborative training scenarios. This method not only

improves the discrimination capability and overall scheduling quality during task execution but also exhibits strong advantages in communication latency and resource overhead control. This demonstrates the good synergy between the designed hierarchical collaborative framework, latency-aware scheduling strategy, and joint optimization mechanism in the aggregation phase, effectively mitigating common problems such as latency accumulation, resource imbalance, and limited training efficiency during heterogeneous node participation, thereby improving the overall system performance.

Table 2. Experimental Results Compared with Other Models

Method	ACC↑	F1↑	LAT↓	EC↓
Liu et al.[8]	87.96	87.41	26.48	20.31
Wang et al. [9]	89.12	88.73	24.31	18.94
Liu et al. [10]	88.41	87.95	25.74	19.68
Hu et al. [11]	89.67	89.21	23.88	18.45
Zhang et al. [12]	89.84	89.37	23.47	18.11
Xiang et al. [13]	90.26	89.91	22.56	17.64
Yu et al. [14]	88.95	88.41	24.86	19.27
Ours	91.34	90.88	20.73	16.19

The superior performance of the proposed method stems from its approach of not separating the communication and computation processes but rather modeling node selection, resource allocation, local training, and global aggregation holistically from a unified scheduling perspective. In a multi-layered cloud-edge collaborative environment, this joint optimization approach better aligns with the operational characteristics of real-world training systems, ensuring both learning effectiveness and balancing execution efficiency and energy consumption constraints. Therefore, the proposed method achieves a good balance between accuracy, stability, and system cost control, providing a valuable solution for efficient joint scheduling in federated learning and cloud-edge collaborative training scenarios.

The number of local training rounds directly affects the intensity of parameter updates in a single round and the rhythm of global model evolution. Therefore, it is necessary to analyze the variation of model performance under different local training depths. As the number of local iterations is adjusted, the coupling relationship between communication frequency, local fitting degree, and global collaborative efficiency will also change. Based on this consideration, a sensitivity analysis of the number of local training rounds is further conducted to characterize the role of this factor in the joint scheduling framework of communication and computation. The experimental results are shown in Figure 2.

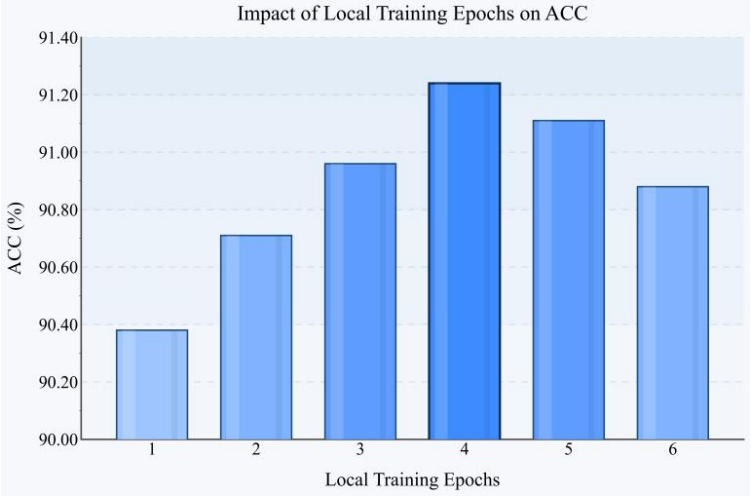

Figure 2. The impact of local training rounds on ACC

The results show that the proposed algorithm maintains good performance under the current conditions, indicating that the constructed joint scheduling mechanism for communication and computation has strong adaptability to the local training process. In cloud-edge collaborative training scenarios, this method effectively coordinates the relationship between local update intensity and global collaborative efficiency, enabling the model to maintain a stable and effective optimization state during training. Furthermore, this demonstrates that the designed joint scheduling strategy can provide reliable support for federated learning tasks, further showcasing the effectiveness and practical value of the proposed method in complex heterogeneous environments.

To further analyze the role of each key module in the overall framework, targeted ablation analysis was conducted on the proposed method. By gradually removing core components from the joint scheduling framework, the impact of different modules on model performance and system efficiency can be observed. The relevant results are shown in Table 3, which summarizes the differences in model performance under different ablation settings.

Table 3. Ablation Test Results

Ablation Setting	ACC↑	F1↑	LAT↓	EC↓
Without hierarchical association scheduling	89.74	89.58	21.11	16.92
Without utility-driven node selection	90.12	89.96	21.44	16.85
Without delay-aware aggregation	90.36	90.21	21.63	16.80
Ours	91.34	90.88	20.73	16.19

The proposed algorithm demonstrates good overall performance, indicating the effectiveness of the constructed joint scheduling framework for communication and computation. After unified modeling, all components work together in the federated learning and cloud-edge collaborative training processes, maintaining high levels of accuracy and comprehensive task performance. This shows that the proposed method not only achieves reasonable coordination of heterogeneous node resources but also enhances global synergy and scheduling stability during training.

Furthermore, ablation analysis demonstrates the practical significance of the key designs in the proposed method, and the complete framework can more fully leverage the advantages of the joint scheduling mechanism. Hierarchical associative scheduling helps improve the collaborative organization capabilities among multi-level nodes, utility-driven node selection enhances the rationality of participating node configuration, and latency-aware aggregation helps improve the overall coordination effect during model updates. Therefore, the proposed algorithm exhibits good comprehensive performance in federated learning and cloud-edge collaborative training scenarios, further demonstrating the application value of this method in joint optimization problems of communication and computation.

As shown in Figure 3, the proposed algorithm maintains good performance under the current learning rate setting, indicating strong consistency between the constructed communication-computing joint scheduling mechanism and the model optimization process. This method effectively coordinates the relationship between parameter updates, resource allocation, and global collaboration in federated learning and cloud-edge collaborative training scenarios, thus ensuring good stability and adaptability of the model training process. This demonstrates the strong usability of the proposed method in complex heterogeneous environments and further reflects the rationality of the overall framework design.

The learning rate, as a crucial factor influencing the model's optimization state, plays a significant role in the proposed method, indicating that the designed joint scheduling strategy not only focuses on communication and computing resources themselves but also effectively supports key optimization stages in the training process. The good performance of the proposed algorithm reflects the strong synergy among

the core modules, providing stable guarantees for global model updates. This also demonstrates the strong practical value and application potential of this method in federated collaborative training tasks for cloud computing scenarios.

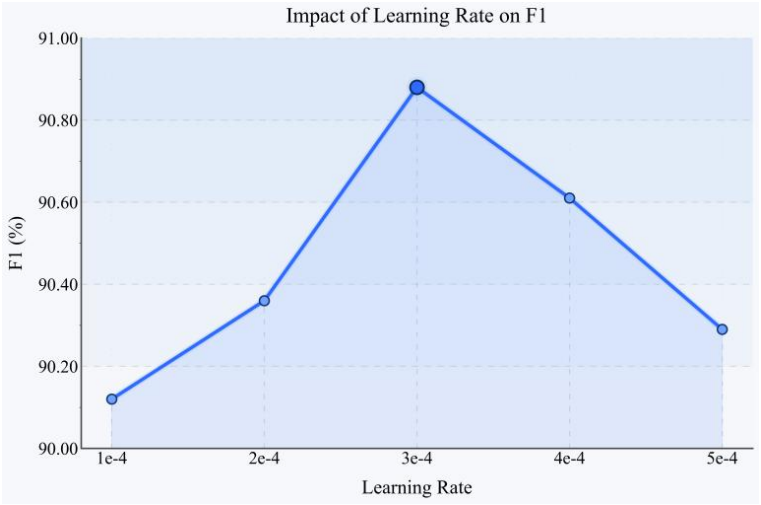

Figure 3. The effect of the learning rate on the F1 score of experimental results

5. Conclusion

This paper addresses the scheduling challenges arising from the deep coupling of communication and computing resources in federated learning and cloud-edge collaborative training scenarios. It systematically studies the joint scheduling optimization problem of communication and computing and constructs a unified framework for multi-level node collaborative training. To address the problems of high communication overhead, uneven computational load, limited synchronization efficiency, and insufficient global coordination capabilities inherent in traditional distributed training methods in heterogeneous node environments, this paper starts from the hierarchical relationship between cloud coordination, edge aggregation, and local terminal training, modeling node association, resource allocation, training participation, and model aggregation within a single optimization process. This research not only expands the analytical perspective of resource scheduling problems in federated learning scenarios but also provides a more holistic approach to system design in cloud-edge collaborative intelligent training, enabling the training process to move beyond single communication or single computation optimization and towards unified collaborative control for complex environments.

At the methodological level, the framework proposed in this paper emphasizes joint modeling and multi-factor collaboration. By introducing a hierarchical associative scheduling mechanism, it enhances the organizational capabilities among terminals, edge computing, and the cloud. By introducing a utility-driven node selection strategy, it improves resource utilization efficiency and node configuration rationality during training. Furthermore, by introducing a latency-aware aggregation mechanism, it mitigates the adverse effects of asynchronous differences among multiple nodes on global model updates. Overall, this method balances communication costs, computational constraints, and learning quality, enabling the scheduling process to better reflect the dynamic characteristics and heterogeneous attributes of real-world cloud computing environments. This joint optimization approach, simultaneously implemented at the system, resource, and training levels, helps improve the overall stability and scalability of the cloud-edge collaborative training system and lays a clear theoretical foundation for subsequent research on intelligent scheduling in complex network environments.

From an application perspective, this research has positive significance for promoting the implementation of federated learning and cloud-edge collaborative training in real-world scenarios. In

typical scenarios such as smart industry, connected vehicles, intelligent manufacturing, edge sensing, smart healthcare, and urban IoT, massive amounts of data are naturally dispersed across terminal devices and edge nodes. This demands both high real-time response capabilities for model training and adaptability to heterogeneous resources, dynamic links, and privacy constraints. The communication-computing joint scheduling method proposed in this paper provides more stable training support for these scenarios, enabling multi-node collaborative learning processes to maintain good organization, sustainability, and operational efficiency even under complex resource conditions. Furthermore, this research offers insights into building distributed learning paradigms for next-generation intelligent infrastructure, helping to improve the deployment efficiency, resource utilization, and overall service quality of intelligent services under cloud-edge converged architectures, thereby enhancing the practical operational capabilities and engineering potential of related application systems.

Future research can be further deepened in several directions. First, with the development of large-scale model training, cross-domain collaborative learning, and multi-task edge intelligence, the communication-computing joint scheduling problem will exhibit greater dynamism, hierarchy, and complexity. How to achieve efficient and robust online decision-making in a larger-scale node environment remains an important issue worthy of continued attention. Secondly, network conditions, device availability, data distribution, and business priorities in real-world scenarios are often constantly changing. Therefore, adaptive scheduling mechanisms for non-stationary environments, intelligent decision-making strategies for long-term gains, and trusted collaborative optimization methods for privacy and security constraints all have further research value. Furthermore, as the cloud-edge-device multi-layered collaborative architecture continues to evolve, more closely integrating joint scheduling with factors such as green computing, energy consumption constraints, model compression, asynchronous aggregation, and quality of service assurance will become an important direction for improving the overall capabilities of the system. In-depth research on these issues will not only further improve the theoretical framework of federated learning and cloud-edge collaborative training but also provide stronger technical support for the construction of intelligent collaborative computing platforms for large-scale real-world scenarios.

References

[1] C. Zhang, "Adaptive Multi-Tenant Resource Scheduling in Cloud Computing via Reinforcement Learning," 2024.

[2] R. H. Gau, D. C. Liang, T. Y. Wang, et al., "Edge-to-Cloud Latency Aware User Association in Wireless Hierarchical Federated Learning", Proceedings of the 2024 IEEE 99th Vehicular Technology Conference (VTC2024-Spring), pp. 1-6, 2024.

[3] G. Pang and X. Zhu, "Training Efficiency Optimization Algorithm of Wireless Federated Learning Based on Processor Performance and Network Condition Awareness", EURASIP Journal on Advances in Signal Processing, vol. 2024, no. 1, p. 98, 2024.

[4] S. Li, "Adaptive Scheduling for Multi-Model Collaborative Distributed Inference under Resource Heterogeneity and Dynamic Workloads," 2024.

[5] Y. Tang, "Research on Hierarchical Multi-Agent Reinforcement Learning Resource Orchestration for Large-Scale Heterogeneous Distributed Clusters with Dynamic Load Awareness," 2024.

[6] Z. Gao, Z. Zhang, Y. Zhang, et al., "Online Client Scheduling and Resource Allocation for Efficient Federated Edge Learning", arXiv preprint arXiv:2410.10833, 2024.

[7] C. Ma, X. Li, B. Huang, et al., "Personalized Client-Edge-Cloud Hierarchical Federated Learning in Mobile Edge Computing", Journal of Cloud Computing, vol. 13, no. 1, p. 161, 2024.

[8] L. Liu, J. Zhang, S. H. Song, et al., "Client-Edge-Cloud Hierarchical Federated Learning", Proceedings of the 2020 IEEE International Conference on Communications, pp. 1-6, 2020.

[9] Z. Wang, H. Xu, J. Liu, et al., "Resource-Efficient Federated Learning with Hierarchical Aggregation in Edge Computing", Proceedings of the 2021 IEEE Conference on Computer Communications, pp. 1-10, 2021.

[10] J. Liu, H. Xu, Y. Xu, et al., "Communication-Efficient Asynchronous Federated Learning in Resource-Constrained Edge Computing", Computer Networks, vol. 199, Art. no. 108429, 2021.

[11] 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.

[12] J. Zhang, S. Chen, X. Zhou, et al., "Joint Scheduling of Participants, Local Iterations, and Radio Resources for Fair Federated Learning over Mobile Edge Networks", IEEE Transactions on Mobile Computing, vol. 22, no. 7, pp. 3985-3999, 2022.

[13] T. Xiang, Y. Bi, X. Chen, et al., "Federated Learning with Dynamic Epoch Adjustment and Collaborative Training in Mobile Edge Computing", IEEE Transactions on Mobile Computing, vol. 23, no. 5, pp. 4092-4106, 2023.

[14] R. Yu and P. Li, "Toward Resource-Efficient Federated Learning in Mobile Edge Computing", IEEE Network, vol. 35, no. 1, pp. 148-155, 2021.