Journal of Intelligent Computing and Networking
https://www.ffspub.com/index.php/jicn/index
ISSN 3079-9228 (print)
ISSN 3107-2798 (online)
E-mail: jicn.office@ffspub.com

Article

Cross-Domain Collaborative Federated Intelligence for Wireless Computing Power Networks

Yuan Feng[1,2], Jiayao Yuan[1,2], Jiajia Liu[1,2] Yunlong Lu[1,2,†] Hao Wu[1,2]
[1] *School of Electronic Information Engineering, Beijing Jiaotong University, Beijing 100044, China*
[2] *State Key Laboratory of Advanced Rail Autonomous Operation, Beijing Jiaotong University, Beijing 100044, China*
[†] *E-mail: yllu@bjtu.edu.cn*

Received: February 4, 2026 / Revised: March 7, 2026 / Accepted: March 10, 2026 / Published online: March 25, 2026

Abstract: With the rapid development of wireless and intelligent computing technologies, emerging applications like autonomous driving and AI agents demand ubiquitous, low-latency, and efficient computing resources. Wireless computing power networks (WCPNs), which interconnect geographically distributed and heterogeneous computing nodes through wireless communication infrastructures, are regarded as a key paradigm for supporting such computing-intensive and latency-sensitive services. However, the current critical challenge is how to achieve efficient, low-latency joint optimisation across domains and multiple tasks by coordinating highly heterogeneous distributed computing and communication resources while protecting each node's data privacy. To address the challenge of privacy-preserving collaboration among heterogeneous computing nodes with non-IID data distributions in WCPNs, this paper introduces a federated intelligence–driven learning framework. By leveraging federated learning with a FedOpt-based aggregation mechanism, collaborative model training can be achieved without sharing raw data, while improving accuracy and convergence performance under data and node heterogeneity. To further address the cross-domain multi-task resource allocation problem under stringent real-time requirements, a joint optimisation model is formulated that incorporates wireless communication conditions, heterogeneous computing capabilities, task service demands, and system energy consumption. By minimising the total execution delay of multiple tasks subject to resource, energy, and scheduling constraints, a genetic algorithm–based solution is employed to derive near-optimal task orchestration and resource allocation strategies. Simulation results demonstrate that the proposed framework achieves lower task execution delay and higher energy efficiency than baseline schemes, validating its effectiveness in WCPNs.

Keywords: Cross-domain Collaboration; Federated Learning; Resource Orchestration; Wireless Computing Power Networks
https://doi.org/10.64509/jicn.21.75

1 Introduction

With the rapid evolution of sixth-generation (6G) [1] wireless communication and artificial intelligence (AI) technologies [2], intelligent applications are undergoing a radical transformation, moving towards autonomy, real-time interaction, and large-scale collaboration. Rising frameworks such as AI agents[3], autonomous driving[4], immersive extended reality (XR)[5], and intelligent industrial systems[6] increasingly depend on persistent perception, decision-making, and learning from vast volumes of data. These applications impose rigorous requirements on pervasive computing capability, low latency, and high reliability, which can no longer be met by conventional centralised computing architectures.

To sustain such computationally intensive and delay-sensitive services, computing resources are being extensively deployed across the network, from centralised clouds to edge servers and intelligent terminals. In this scenario, nearly every interconnected device can serve as a computing node, forming a highly distributed computing ecosystem characterised by heterogeneity in computing capacity, data distribution, and energy supply [7]. Moreover, these computing nodes are frequently managed by different service providers or fall under distinct administrative domains, resulting in a dynamic, cross-domain, heterogeneous computing environment. This structural transformation drives the rise of wireless computing power networks (WCPNs)[8], in which dispersed computing resources are linked via wireless communication frameworks to jointly support intelligent services.

† Corresponding author: Yunlong Lu
* Academic Editor: Chunxiao Jiang

However, the dynamic and cross-domain characteristics of WCPNs also pose crucial challenges that hinder efficient cross-domain collaboration. On the one hand, computing nodes are widely distributed across separate administrative domains, and the datasets generated and stored on these devices are subject to strict privacy constraints that prevent unrestricted sharing [9]. Concurrently, the substantial number of heterogeneous devices inherently yields highly non-independent and identically distributed (non-IID) data patterns across networked nodes [10]. Without targeted treatment, such data heterogeneity significantly undermines both the performance and stability of collaborative intelligence [11]. On the other hand, the massive number of heterogeneous computing nodes, coupled with dynamic wireless communication conditions, makes it extremely challenging to coordinate cross-domain nodes for efficient collaboration and computing resource allocation. Task execution performance is jointly affected by wireless transmission, computing capability, and energy constraints, and the lack of effective cross-domain coordination mechanisms can result in suboptimal task scheduling, increased execution delay, and inefficient resource utilisation.

Federated learning (FL) has emerged as an effective paradigm for enabling collaborative intelligence without sharing raw data [12]. In FL, each computing node locally trains a model on its private data and exchanges only model updates or gradients with a coordinating server, thereby keeping sensitive data within local domains and significantly reducing the risk of privacy leakage. This decentralised training mechanism is particularly suitable for cross-domain computing environments, where direct data aggregation is prohibited by privacy, security, or regulatory constraints. However, in cross-domain wireless environments, data heterogeneity arises naturally from differences in application scenarios, sensing modalities, user behaviours, and service objectives, leading to highly inconsistent local model updates. Under such conditions, conventional aggregation strategies that rely on simple weighted averaging, such as FedAvg [13], are unable to adequately capture heterogeneous update patterns, resulting in slow convergence, unstable training dynamics, and performance degradation. Therefore, how to effectively mitigate the non-IID data challenge has become a key research problem in the design of federated intelligence–enabled wireless computing power networks.

Existing resource allocation and task scheduling approaches are often designed based on centralised optimisation or static resource assumptions, under which global system information is assumed to be fully available, and network conditions are relatively stable [14–16]. Such methods typically rely on predefined models or fixed scheduling rules to allocate computing and communication resources, which are difficult to adapt to the highly dynamic and heterogeneous nature of wireless computing power networks. However, in wireless computing power networks, the heterogeneity of computing resources across cloud and edge nodes, combined with multi-domain deployment and complex wireless environments, makes it challenging to design efficient and secure cross-domain resource orchestration and collaborative computing mechanisms [17]. Furthermore, the tight coupling among task execution delay, computing capability, wireless transmission, and energy consumption renders traditional scheduling methods ineffective, often leading to suboptimal resource utilisation and increased task execution delay [18], [19]. It is necessary to develop an adaptive and intelligent resource orchestration framework that jointly considers multi-task demands, heterogeneous computing resources, and dynamic wireless environments, enabling efficient cross-domain collaboration and optimal resource allocation in wireless computing power networks.

To address the aforementioned challenges in WCPNs, the main contributions of this paper are as follows:

- To address the privacy and non-IID data challenges in cross-domain WCPNs, this paper adopts a FedOpt-based task-coupled aggregation scheme[20] within the proposed federated intelligence framework(FI-Opt). FI-Opt builds upon the standard FedOpt aggregation mechanism and integrates federated learning with task execution and resource allocation decisions in WCPNs.
- To tackle the challenge of dynamic and heterogeneous multi-task resource management in WCPNs, this paper formulates a joint multi-task optimization model that explicitly considers wireless communication conditions, heterogeneous computing capabilities, task service requirements, and system energy consumption. To minimize the total execution delay of latency-sensitive tasks under resource and energy constraints, the proposed model provides a unified formulation for multi-task orchestration and resource planning.
- Due to the non-convex and combinatorial nature of the formulated optimization problem, a genetic algorithm–based solution is developed to derive near-optimal task scheduling and resource allocation strategies. Simulation results demonstrate that the proposed solution significantly reduces task execution delay and energy consumption while improving resource utilization efficiency compared with baseline scheduling schemes.

The remainder of this paper is organised as follows. Section 2 reviews the related work and discusses the research background. Section 3 presents the system model, including the network architecture, federated intelligence framework FI-Opt, and multi-task model. The problem formulation and solution are described in Section 4. Experimental evaluations and performance analyses are reported in Section 5. Finally, Section 6 concludes the paper and outlines future research directions.

2 Related Work

2.1 Federated Learning for Privacy Preserving Collaborative Intelligence

FL has been widely studied as a privacy-preserving collaborative learning paradigm, enabling distributed model training without sharing raw data. Existing research mainly focuses on improving learning performance, communication efficiency, and robustness in distributed and heterogeneous environments.

Zhou [21] et al. demonstrated that FL can support collaborative intelligence in large-scale heterogeneous wireless systems by enabling decentralized model training under privacy constraints. Ma [22] et al. systematically analyzed the non-IID data problem in federated learning, showing that data heterogeneity severely degrades convergence speed and model performance. To mitigate non-IID data effects, Zhou [23] et al. proposed Federated CINN Clustering, which groups clients based on information-theoretic similarity and performs clustered aggregation, achieving improved accuracy under heterogeneous data distributions. Gao [24] et al. introduced FedPC, a prototype-based clustered federated learning framework that leverages shared feature representations to reduce distribution mismatch, leading to improved performance in highly heterogeneous medical imaging tasks. From the perspective of aggregation robustness, Pillutla [25] et al. proposed a geometric-median-based aggregation strategy to reduce the influence of abnormal or corrupted updates, improving robustness compared with simple averaging. Hu [26] et al. developed a lightweight privacy-preserving data aggregation scheme for federated learning, reducing communication overhead while ensuring secure transmission of model updates in resource-constrained environments. Despite these advances, existing FL solutions still face critical limitations when applied to WPCNs. Most approaches either focus solely on data heterogeneity or robustness, without jointly considering system heterogeneity, non-IID data distributions, and resource-aware collaboration. In particular, conventional aggregation strategies lack adaptability to highly heterogeneous update patterns induced by diverse tasks, computing capabilities, and dynamic wireless environments. This motivates the adoption of adaptive server optimization–based aggregation mechanisms, such as FedOpt [20], which can better regulate global model updates and enhance convergence stability under non-IID and heterogeneous conditions.

2.2 Resource Allocation and Multi-Task Scheduling

Resource allocation and multi-task scheduling have been extensively studied in distributed computing and edge networks to minimize execution latency, reduce energy consumption, and improve overall system efficiency under heterogeneous and dynamic environments. Existing research mainly focuses on centralized optimization and intelligent decision-making mechanisms for complex computing systems.

Tan [27] et al. addressed the joint task offloading and resource allocation problem in vehicular edge computing systems by formulating it as a decentralized convex optimization problem, achieving near-optimal performance while reducing coordination overhead among distributed nodes. To cope with dynamic network conditions and user mobility, several works have incorporated stochastic optimization and control theory. Chen [28] et al. investigated joint trajectory optimization and resource allocation in UAV-assisted MEC systems, proposing a Lyapunov-assisted deep reinforcement learning approach that effectively reduces system delay and energy consumption under time-varying task arrivals and mobility. Jalali [29] et al. studied placement, orientation, and resource allocation in cell-free optical wireless communication networks, formulating a joint optimization problem and solving it via successive convex approximation to significantly improve both spectral efficiency and energy efficiency. For large-scale distributed computing systems with concurrent task execution, Min [30] et al. proposed a joint task scheduling and computing resource allocation framework to handle heterogeneous task requirements, demonstrating improved system throughput while satisfying latency constraints. Similarly, Jie [31] et al. modelled multi-task resource competition among distributed nodes using game-theoretic approaches, enabling cooperative resource allocation and enhancing system stability and fairness. More recently, learning-based methods have been introduced to address the high complexity and uncertainty of dynamic computing environments. Machine learning has been widely applied to learn adaptive scheduling and resource allocation policies [32], enabling systems to respond to fluctuating workloads and network conditions without explicit system models.

In contrast, existing studies on resource allocation and multi-task scheduling primarily assume centralized control or single-domain coordination, which limits their applicability in practical wireless computing power networks. Moreover, cross-domain resource heterogeneity, distributed ownership, and privacy constraints are often overlooked, making it challenging to support scalable and flexible collaboration among heterogeneous computing nodes in dynamic environments.

3 System Model

This paper considers a federated intelligence-driven wireless computing power network(FIWCPN) with cross-domain collaborative capability. As illustrated in Figure 1, FIWCPN consists of a centralized computing network control center R, multiple geographically distributed computing power clusters, and a large number of heterogeneous computing nodes. The control center R is responsible for global orchestration, including task coordination, resource management, and global model aggregation. Each computing power cluster is managed by a local agent, which performs cluster-level coordination, node selection, and local aggregation. The computing nodes deployed in different clusters execute tasks and participate in federated learning in a distributed manner, ensuring the data privacy of the computing nodes.

The control center collects global node status information from all clusters, including computing capability, storage capacity, communication conditions, and trust levels. Based on the collected information, tasks are assigned to appropriate computing clusters and nodes, and the global learning model is broadcast to selected nodes. The selected nodes perform local task execution and model training using their private data. After local execution, task results and model updates are returned to the cluster agents and further aggregated at the control center. Finally, node status and trust information are updated based on observed performance, supporting adaptive task scheduling and trusted federated collaboration in subsequent rounds.

3.1 Network Architecture

The computing power network consists of one control center R, a task set, and a set of computing power clusters. The task

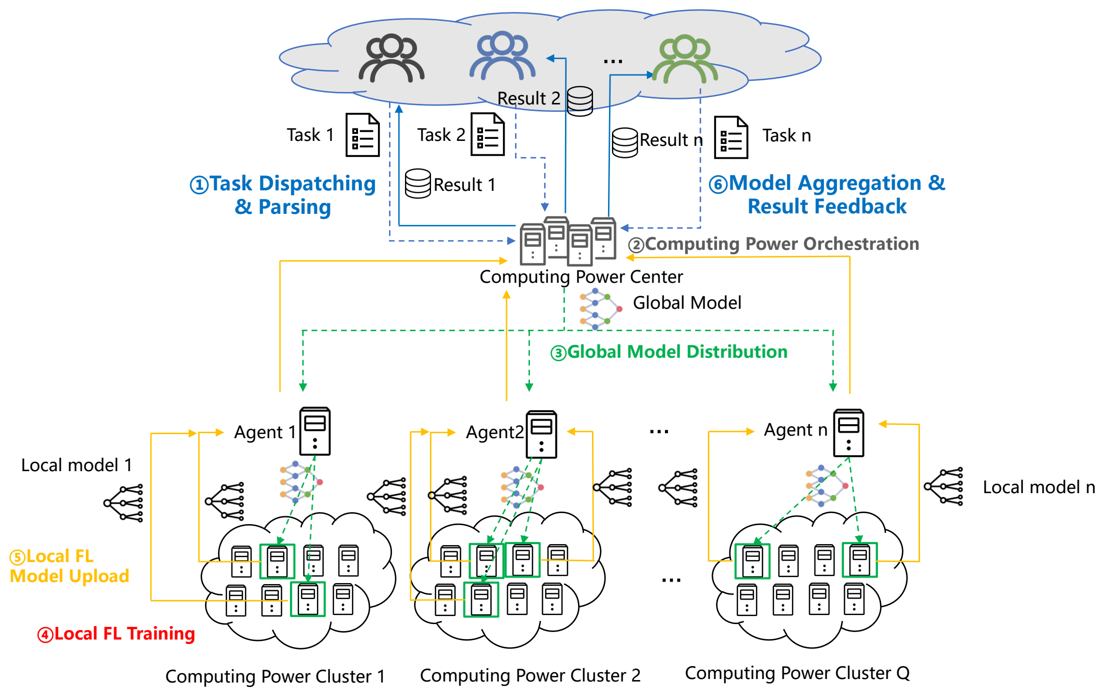

Figure 1: Federated intelligence-driven cross-domain collaborative model of wireless computing power networks.

set and the cluster set are denoted by $\mathscr{N} = \{1,\ldots,n,\ldots,N\}$ and $\mathscr{Q} = \{1,\ldots,q,\ldots,Q\}$, respectively.

Each computing power cluster $q \in \mathscr{Q}$ deploys M computing nodes, which are denoted by

$$\mathscr{C}_q = \{c_{q1},\ldots,c_{qm},\ldots,c_{qM}\}. \tag{1}$$

The computing nodes can be cloud computing centers, edge computing nodes, or user terminal nodes, reflecting the heterogeneity of the computing power network.

The computing resources of node c_{qm} are characterized by a tuple

$$\langle f_{qm}, s_{qm}\rangle, \tag{2}$$

where f_{qm} denotes the maximum CPU frequency that node m in cluster q can provide, and s_{qm} represents the available memory capacity, measured in bytes.

We assume that the computing power network is connected, i.e., there exists a communication link between the control center R and each computing cluster. The communication delay between the control center and cluster q is denoted by $t_{R\to q}$. Through cross-domain collaboration across clusters, coordinated allocation of computing resources and task scheduling can be achieved.

3.2 Multi-Task Model

Assume multiple task initiators simultaneously submit computational tasks, which are executed in a distributed manner across different computing nodes.The task set is indicated as $\mathscr{N} = \{1,2,\ldots,n,\ldots,N\}$, where the parameters for each task n are defined as:

$$T_n = \langle D_n, C_n, O_n, \alpha_n\rangle. \tag{3}$$

Here,D_n represents the input data volume (bits) for task n, C_n denotes the CPU cycles per bit required, O_n indicates the storage demand (bits), and α_n represents the maximum tolerable execution delay threshold. Thus, the total CPU computation required for task n can be expressed as:

$$C_n^{total} = C_n D_n. \tag{4}$$

Define the task assignment variable as $a_{qm}^n \in \{0,1\}$. When $a_{qm}^n = 1$, task n is assigned to compute node c_{qm} in cluster q for execution; otherwise, $a_{qm}^n = 0$. In a multi-task concurrent execution scenario, a compute node's resources can be allocated to multiple tasks. Define $\beta_{qm}^n \in (0,1]$ as the proportion of compute resources allocated to task n on node c_{qm}. Thus, the effective compute power for task n on this node is:

$$f_{qm}^n = f_{qm}\beta_{qm}^n. \tag{5}$$

3.3 Communication Model

Assuming computing task n is assigned to computing node c_{qm}, its computational latency is primarily determined by the CPU processing time. Node c_{qm} provides an effective computational capacity of f_{qm}^n for task n. For task n, its local computational latency can be expressed as:

$$t_n^{cal} = a_{qm}^n \frac{C_n^{total}}{f_{qm}^n}. \tag{6}$$

Before task execution, task data must be transferred from the computing network control center to the target computing cluster, and data forwarding among cluster nodes pleted. Therefore, the communication delay for node task execution primarily consists of two parts: link transmission delay and intra-cluster forwarding delay. Assuming the bandwidth allocated to computing node c_{qm} within cluster q for executing task n is b_{qm}^n , the task communication delay can be expressed as:

$$t_n^{comm} = t_{R\to q} + \frac{D_n}{b_{qm}^n}. \tag{7}$$

Therefore, the total execution delay for task n on node c_{qm} is:

$$t_n^{qm} = t_n^{comm} + t_n^{cal}. \tag{8}$$

The power consumption of computing resources consists of idle power consumption and dynamic power consumption related to computational load, expressed as:

$$E_{qm} = \eta P_{qm}^{max} + (1-\eta) P_{qm}^{max} \theta_{qm}, \tag{9}$$

where P_{qm}^{max} is the peak power consumption of node c_{qm} at full load, and η is the ratio of the average idle power consumption of node c_{qm} to its peak power consumption. This ratio can vary across heterogeneous computing resources in the network and is measured by the computing power sensing module. θ_{qm} is the current load factor of node c_{qm}, defined as the ratio of the total task volume processed by node c_{qm} to its processing capacity. Considering the local training round L in federated learning, when task n is deployed on node c_{qm}, its total computational energy consumption can be expressed as

$$E_{qm}^n = \sum_n (1-\eta) P_{qm}^{max} L \cdot t_n^{cal}. \tag{10}$$

Transmission energy consumption, which refers to the energy consumption incurred during the process of sending task data from the computing network center to specific nodes, can be expressed as:

$$E_{n\to qm}^{comm} = p_q \frac{D_n}{b_{qm}^n} + p_R t_{R\to q}, \tag{11}$$

where p_q is the transmission power within cluster q, and p_R is the power of the computing network center. The total system energy consumption can be expressed as:

$$E_n = \sum_{qm} a_{qm}^n \left(E_{n\to qm}^{comm} + E_{qm}^n \right). \tag{12}$$

3.4 FI-Opt Learning Model

This section presents a federated intelligence-based model deployed on top of the computing power network and task models described earlier. In the cross-domain computing power network considered, data are inherently distributed across heterogeneous computing nodes and cannot be centrally collected due to privacy, ownership, and communication constraints. FL is therefore adopted to enable collaborative model training while preserving data locality and reducing excessive data transmission.

The computing network control center R acts as the federated parameter server and maintains a global model $\mathbf{w} \in \mathbb{R}^d$, whereas distributed computing nodes c_{qm} perform local training using their private datasets. However, due to heterogeneity in task loads, computing capabilities, and data distributions across clusters, naive aggregation schemes such as FedAvg may suffer from slow convergence and unstable updates.

To improve robustness under heterogeneous and non-IID conditions, a task-coupled federated aggregation strategy based on FedOpt is employed at the control center, referred to as FI-Opt. Building upon the server-side adaptive optimization mechanism of FedOpt [20], the proposed scheme integrates federated aggregation with task execution and resource allocation processes in WCPNs. In particular, task-aware resource coupling is incorporated into client participation and aggregation weighting, allowing the global update process to better reflect heterogeneous computing capabilities and task loads across distributed nodes. Through this integration, the proposed framework provides more stable global model updates and improves training robustness in dynamic cross-domain computing environments.

(1) Local data and learning objectives. Each computing node c_{qm} maintains a local dataset $\mathscr{D}_{qm}$ of size $|\mathscr{D}_{qm}|$. The local loss function is defined as

$$F_{qm}(\mathbf{w}) = \frac{1}{|\mathscr{D}_{qm}|} \sum_{\xi \in \mathscr{D}_{qm}} \ell(\mathbf{w}; \xi), \tag{13}$$

where $\ell(\cdot)$ denotes the sample-wise loss function. Accordingly, the global federated learning objective is formulated as

$$\begin{aligned} \min_{\mathbf{w}} \; F(\mathbf{w}) &= \sum_{q\in\mathscr{Q}} \sum_{m=1}^{M} p_{qm} F_{qm}(\mathbf{w}), \\ p_{qm} &= \frac{|\mathscr{D}_{qm}|}{\sum_{q\in\mathscr{Q}} \sum_{m=1}^{M} |\mathscr{D}_{qm}|}, \end{aligned} \tag{14}$$

where p_{qm} represents the relative contribution of node c_{qm}.

(2) Round-based federated training. The federated learning process proceeds in synchronous communication rounds indexed by $t = 0, 1, \ldots$. At the beginning of round t, the control center R broadcasts the current global model $\mathbf{w}^{(t)}$ to all computing clusters. Within each cluster q, a subset of computing nodes $\mathscr{S}_q^{(t)} \subseteq \mathscr{C}_q$ is selected according to resource availability and trust constraints defined in the network model. Each selected node initializes its local model as

$$\mathbf{w}_{qm,0}^{(t)} = \mathbf{w}^{(t)}, \tag{15}$$

and performs local training for E iterations.

(3) Local update with proximal regularization. To mitigate client drift caused by data heterogeneity and system dynamics, a proximal regularization term is incorporated into the local update. Specifically, the local model is updated as

$$\begin{aligned} \mathbf{w}_{qm,e+1}^{(t)} = \mathbf{w}_{qm,e}^{(t)} - \eta \Big(&\nabla F_{qm}(\mathbf{w}_{qm,e}^{(t)}) \\ &+ \mu \left(\mathbf{w}_{qm,e}^{(t)} - \mathbf{w}^{(t)} \right) \Big), \end{aligned} \tag{16}$$

where η denotes the local learning rate and $\mu \geq 0$ is the proximal coefficient. After E local iterations, node c_{qm} obtains the updated model

$$\mathbf{w}_{qm}^{(t+1)} = \mathbf{w}_{qm,E}^{(t)}, \tag{17}$$

which is uploaded to the corresponding cluster agent.

(4) Coupling between task execution and local training. It is worth noting that local model training competes for computing resources with task execution. According to the task model, when $a_{qm}^n = 1$, a fraction β_{qm}^n of the computing capacity of node c_{qm} is allocated to task n. As a result, the effective computing resource available for federated learning at node c_{qm} can be expressed as

$$f_{qm}^{\mathrm{FL}} = f_{qm}\left(1 - \sum_{n\in\mathcal{N}} a_{qm}^{n}\beta_{qm}^{n}\right), \tag{18}$$

where f_{qm} denotes the maximum CPU frequency of node c_{qm}. This coupling allows task scheduling decisions to directly influence the local training efficiency and participation of nodes in federated learning.

(5) Hierarchical aggregation at cluster level. After receiving local models from nodes in $\mathscr{S}_q^{(t)}$, the cluster agent performs intra-cluster aggregation to obtain a cluster-level model

$$\begin{aligned} \mathbf{w}_q^{(t+1)} &= \sum_{c_{qm}\in\mathscr{S}_q^{(t)}} \alpha_{qm}^{(t)}\mathbf{w}_{qm}^{(t+1)}, \\ \text{s.t.} \quad & \sum_{c_{qm}\in\mathscr{S}_q^{(t)}} \alpha_{qm}^{(t)} = 1, \end{aligned} \tag{19}$$

where $\alpha_{qm}^{(t)}$ denotes the aggregation weight of node c_{qm}.

The cluster-level update relative to the global model is defined as

$$\Delta_q^{(t)} = \mathbf{w}_q^{(t+1)} - \mathbf{w}^{(t)}. \tag{20}$$

(6) FIOpt-based global update. Instead of directly averaging the cluster-level models, the control center applies a server-side optimization strategy. An aggregated update direction is first constructed as

$$\begin{aligned} \Delta^{(t)} &= \sum_{q\in\mathscr{Q}^{(t)}} \beta_q^{(t)}\Delta_q^{(t)}, \\ \beta_q^{(t)} &= \frac{\sum_{c_{qm}\in\mathscr{S}_q^{(t)}} |\mathscr{D}_{qm}|}{\sum_{q'\in\mathscr{Q}^{(t)}}\sum_{c_{q'm}\in\mathscr{S}_{q'}^{(t)}} |\mathscr{D}_{q'm}|}. \end{aligned} \tag{21}$$

Based on $\Delta^{(t)}$, the control center performs a FedOpt update. Taking FedAdam as a representative instance, the first- and second-order moment estimates are updated as

$$\begin{aligned} \mathbf{m}^{(t)} &= \rho_1\mathbf{m}^{(t-1)} + (1-\rho_1)\Delta^{(t)}, \\ \mathbf{v}^{(t)} &= \rho_2\mathbf{v}^{(t-1)} + (1-\rho_2)\big(\Delta^{(t)}\odot\Delta^{(t)}\big), \end{aligned} \tag{22}$$

where ρ_1 and ρ_2 denote the momentum coefficients. Finally, the global model is updated as

$$\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} + \gamma\frac{\mathbf{m}^{(t)}}{\sqrt{\mathbf{v}^{(t)}}+\varepsilon}, \tag{23}$$

where γ is the server learning rate and ε is a small constant to ensure numerical stability.

Algorithm 1 describes an FI-Opt hierarchical scheme with explicit task-aware resource coupling for WCPNs. In each training round, computing nodes perform local model updates on private data using limited resources that are jointly shared with task execution, thereby capturing realistic resource competition. Local updates are first aggregated at the cluster level to mitigate heterogeneity and reduce communication overhead, and then further optimized at the control center using a FedOpt-based server-side optimizer. By adaptively regulating global model updates under non-IID data distributions and heterogeneous system conditions, the proposed algorithm achieves stable convergence and efficient cross-domain collaborative learning while preserving data privacy.

Algorithm 1 FedOpt-Based Task-Coupled Aggregation Scheme

Input: Initial global model $\mathbf{w}^{(0)}$; local datasets $\{\mathscr{D}_{qm}\}$; task allocation variables $\{a_{qm}^{n}, \beta_{qm}^{n}\}$; FL parameters.
Output: Trained global model $\mathbf{w}^{(T)}$.
Initialization: Server initializes moment vectors $\mathbf{m}^{(0)} = \mathbf{0}$, $\mathbf{v}^{(0)} = \mathbf{0}$.
for $t = 0, 1, \ldots, T-1$ **do**
  Broadcast global model $\mathbf{w}^{(t)}$ to all clusters.
  for each cluster $q \in \mathscr{Q}$ **in parallel do**
    Select participating nodes $\mathscr{S}_q^{(t)}$.
    for each node $c_{qm} \in \mathscr{S}_q^{(t)}$ **in parallel do**
      Compute effective FL computing capacity

$$f_{qm}^{FL} = f_{qm}\Big(1 - \sum_{n} a_{qm}^{n}\beta_{qm}^{n}\Big)$$

      Perform local model update based on dataset $\mathscr{D}_{qm}$.
    end for
    Aggregate local updates to obtain cluster update $\Delta_q^{(t)}$.
  end for
  Aggregate cluster updates $\{\Delta_q^{(t)}\}$.
  Update global model via FedOpt

$$\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} + \gamma\frac{\mathbf{m}^{(t)}}{\sqrt{\mathbf{v}^{(t)}}+\varepsilon}$$

end for
return $\mathbf{w}^{(T)}$

4 Problem Formulation and Solution

In this section, we formulate a joint task orchestration and computing resource allocation model that integrates communication latency, multi-task heterogeneous computing demands, and federated learning model performance. The optimization objective is to minimize the system-wide maximum execution latency, thereby ensuring efficient concurrent task processing. Specifically, our model accounts for critical constraints including task-specific latency thresholds, system energy budgets, node capacity limits, and model accuracy requirements. Given that the formulated problem is a complex non-convex optimization problem with coupled variables, we utilize a Genetic Algorithm (GA)-based [33] decision-making approach to effectively seek the optimal strategy for task scheduling and resource planning.

4.1 Problem Formulation

We consider a computing network system consisting of a computation-network control center, multiple computing clusters, and heterogeneous computing nodes. The system executes multiple concurrent computational tasks, which must be scheduled to appropriate computing nodes while satisfying real-time and performance requirements. Due to the multi-round local training and global aggregation characteristics of federated learning, together with limited computing resources and energy constraints at computing nodes, the task scheduling and resource allocation problem is highly coupled and complex.

(1) Optimization Objective

This paper aims to minimize the *multi-task execution latency*. Since the overall system performance is typically dominated by the task with the longest completion time, the maximum task completion latency is adopted as the optimization objective. Let the task set be denoted by

$$\mathcal{N} = \{1, 2, \ldots, N\}. \tag{24}$$

The maximum system execution latency in the current scheme is defined as

$$T_n = \max_{n \in \mathcal{N}} t_n^{\text{tot}}, \tag{25}$$

where t_n^{tot} denotes the total execution latency of task n, including both communication and computation delays. Specifically, the execution latency of task n can be expressed as

$$t_n^{\text{tot}} = \sum_{q,m} a_{qmn} \left(t_{R \to q} + \frac{D_n}{b_{qmn}} + t_{nqm}^{\text{cal}} \right), \tag{26}$$

where a_{qmn} indicates whether the task n is assigned to the computing node c_{qm}.

- **Task orchestration variable can be described as:**

$$a_{qmn} \in \{0, 1\}, \tag{27}$$

which indicates whether task n is assigned to computing node c_{qm}.

- **Computing resource allocation variable can be described as:**

$$\beta_{qmn} \in (0, 1], \tag{28}$$

which represents the proportion of computing resources allocated by node c_{qm} to task n.

(2) Constraints

In practical computing networks, task scheduling and resource allocation are subject to multiple system constraints:

- **C1: Task latency constraint.** The execution latency of each task must not exceed its maximum tolerable delay:

$$\sum_{q,m} t_{nqm} \leq \alpha_n, \quad \forall n. \tag{29}$$

- **C2: System energy constraint.** The total system energy consumption should not exceed the energy budget:

$$E^{\text{tot}} \leq E^{\max}. \tag{30}$$

- **C3: Computing resource constraint.** The total computing resources allocated to all tasks at each computing node cannot exceed its available capacity:

$$\sum_{n} a_{qmn} \beta_{qmn} \leq 1, \quad \forall q, m. \tag{31}$$

- **C4: Task performance constraint.** The final performance of the federated learning model must satisfy a minimum accuracy requirement:

$$Acc_n \geq A_0, \quad \forall n \in \mathcal{N}. \tag{32}$$

This constraint guarantees that each participating cluster contributes a model of acceptable quality, avoiding low-accuracy local training results that could undermine system-level learning performance.

(3) Problem Formulation

The joint task orchestration and computing resource allocation problem is formulated as follows:

$$\begin{aligned} \min_{\mathbf{a}, \boldsymbol{\beta}} \quad & \max_{n \in \mathcal{N}} \sum_{q,m} a_{qmn} \left(t_{R \to q} + \frac{D_n}{b_{qmn}} + t_{nqm}^{\text{cal}} \right), \\ \text{s.t.} \quad & \text{C1} - \text{C4}, \end{aligned} \tag{33}$$

where our goal is to minimize the system-wide maximum execution latency under the stringent constraints of task-specific delay thresholds (C1), system energy budgets (C2), computing resource capacities (C3), and federated model accuracy requirements (C4). By jointly optimizing the task scheduling variables $\mathbf{a}$ and resource allocation ratios $\boldsymbol{\beta}$, we aim to achieve a balanced trade-off between task execution efficiency and communication energy consumption, thereby ensuring reliable and low-latency collaborative intelligence in wireless computing power networks.

4.2 Dynamic Adaptive Decision Algorithm Based on Federated Intelligence

This optimization problem jointly involves discrete variables a_{qmn} and continuous variables β_{qmn}, while both the objective function and constraints are highly nonlinear. As a result, the formulated problem is a typical mixed-integer nonlinear programming (MINLP) problem. When the number of tasks and computing nodes becomes large, the problem cannot be efficiently solved in polynomial time using conventional convex optimization or exhaustive search methods. Therefore, it is necessary to design efficient approximation or heuristic algorithms to obtain feasible suboptimal solutions. To seek the optimal strategy for cross-domain task orchestration and computing power scheduling, this paper employs a GA-based approach to design a federated intelligence-based dynamic adaptive decision algorithm. The algorithm design is illustrated in Algorithm 2.

To align the minimization objective of the joint orchestration problem with the maximization framework of the Genetic Algorithm (GA), we reformulate the fitness function as

$$\text{Fitness} = -T_n - \sum_{i=1}^{4} \lambda_i \mathcal{P}_i, \tag{34}$$

where T_n denotes the system latency objective defined in Equation 25, and λ_i is the penalty coefficient associated with constraint C_i.

The penalty term $\mathscr{P}_i$ corresponding to constraint C_i is given by

$$\mathscr{P}_i = \begin{cases} 0, & \text{if constraint } C_i \text{ is satisfied,} \\ \Delta_i, & \text{if constraint } C_i \text{ is violated,} \end{cases} \tag{35}$$

where Δ_i represents the violation magnitude or a predefined positive constant reflecting the severity of constraint violation.

The training latency under different resource allocation methods is the maximum training latency when executing tasks on the selected computing node, that is,

$$T = \max\left(t_{R\to q} + \frac{D_n}{b_{qm}^n} + t_n^{cal}\right). \tag{36}$$

If the execution delay exceeds the global limit α_{is} or the energy consumption exceeds the threshold, it is treated as a penalty function to impose a penalty on solutions that do not satisfy constraints, guiding the algorithm to converge faster to the optimal solution within the feasible search space.

Algorithm 2 Dynamic Adaptive Decision Algorithm Based on Federated Intelligence

Input: $\{D_n, C_n\}_{n\in\mathscr{N}}$, $\{f_{qm}\}$, and GA parameters (NP, N_G, P_c, P_m).
Output: Optimal task assignment $\mathbf{A}^* = [a_{qmn}]$ and resource allocation $\mathbf{B}^* = [\beta_{qmn}]$.
Initialization:
Proxy execution initializes global model parameters w_0;
Generate initial population $\mathscr{P}^{(0)}$ of size NP, where each chromosome encodes $(\mathbf{A}, \mathbf{B})$;
Set generation counter $g = 0$;
while $g < G_{max}$ **do**
  for each individual in $\mathscr{P}^{(g)}$ **parallel do**
    Fitness Evaluation with Penalty Design
    Perform FL local updates and global aggregation under the individual's resource scheme;
    Calculate system maximum latency T_n using Equation 25;
    $Fitness \leftarrow -T_n + \sum_l \lambda_l \mathscr{P}_l$;
  end for
  Selection: Apply Roulette Wheel selection based on $Fitness$;
  Crossover: Generate offspring via binary/arithmetic crossover with probability p_c;
  Mutation: Perform mutation with probability p_m to maintain population diversity;
  Update population $\mathscr{P}^{(g+1)}$ and increment $g \leftarrow g + 1$;
end while
return Best strategy $(\mathbf{A}^*, \mathbf{B}^*)$ with the highest fitness from the final population.

5 Experimental Evaluation and Performance Analyses

5.1 Simulation Settings

To evaluate the proposed task orchestration model and the corresponding GA-based solution, experiments were conducted in a Python-based numerical simulation environment that emulates a wireless computing power network with heterogeneous computing nodes, task arrivals, and wireless communication conditions. The simulation mainly focuses on the multi-task optimization problem described in Section 4, where the genetic algorithm is used to determine task assignment and resource allocation under latency, energy, and resource constraints.

The key system and algorithmic parameters are explicitly specified, as summarized in Table 1. The noise power spectral density is set to -174 dBm/Hz, corresponding to the standard thermal noise level used in wireless communication analysis. The number of computing tasks is varied from 5 to 20, and the number of computing nodes is varied from 8 to 40, so as to evaluate the scalability of the proposed scheme under different workload intensities and resource scales. The computing capacity of each node is randomly generated within the range of 20–60 GHz to reflect heterogeneous node capabilities in cross-domain environments. The task density is set within 100–200 to emulate different levels of task generation intensity.

For the GA configuration, the population size and the number of generations are both set to 100, which provides a balance between solution quality and computational cost. The crossover probability is set to 0.7 to enhance the global exploration capability of the population, while the mutation probability is set to 0.2 to avoid premature convergence and maintain population diversity. Under this setting, the proposed method is evaluated in terms of execution delay, energy consumption, task execution efficiency, and energy efficiency, thereby enabling a systematic assessment of its effectiveness under different task and node configurations.

(1) Compared schemes. To demonstrate the performance advantages of the proposed approach, the following baseline schemes are implemented for comparison:

- **Proposed scheme**: Joint cross-domain task orchestration and resource allocation with FedOpt-based federated learning.
- **Task-only scheduling**: Task orchestration without considering dynamic computing resource optimization.
- **Resource-only scheduling**: Resource allocation without task-aware coordination.
- **No scheduling**: Tasks are executed without any orchestration or optimization strategy.

All schemes share the same network topology, task generation process, and federated learning configuration to ensure a fair comparison.

(2) Performance Metrics. The performance of different schemes is evaluated using the following quantitative metrics:

- **Model accuracy**: Classification accuracy of the federated learning model.

Table 1: Simulation parameters used in the experiments.

Parameter	Value
Noise power spectral density (N_0)	-174 dBm/Hz
Number of generations (N_G)	100
Population size (NP)	100
Chromosome length (LC)	determined by task-node mapping
Mutation probability (P_M)	0.2
Crossover probability (P_C)	0.7
Number of tasks	5–20
Number of computing nodes	8–40
Computing capacity	20–60 GHz
Task density	100–200

- **Training loss**: Difference between predicted outputs and ground-truth labels during model training.
- **Task execution delay**: Time required to complete computing tasks.
- **Energy consumption**: Total system energy consumed for task execution and model training.
- **Task execution efficiency**: Number of tasks processed per unit time.
- **Energy efficiency**: Number of tasks processed per unit energy consumption.
- **Resource utilization**: Utilization ratios of computing and communication resources.

These metrics jointly characterize learning performance, system efficiency, and resource effectiveness under heterogeneous and cross-domain conditions. To assess robustness and scalability, experiments are conducted under different task workloads and computing node configurations. For each configuration, the experiments are repeated multiple times, and the averaged results are reported to mitigate randomness. Comparative results demonstrate that the proposed scheme consistently reduced task execution delay, and improved energy efficiency compared with baseline methods, validating its effectiveness in multi-task cross-domain wireless computing power networks.

5.2 Performance Evaluation

5.2.1 Evaluation of Task Quantity on System Performance

To investigate the performance of our proposed FI-Opt scheme under varying workloads, we evaluate the system performance across different numbers of computing tasks. The performance is examined from four aspects: task execution delay, energy consumption, task execution efficiency, and energy efficiency.

(1)Task execution delay. Figure 2 illustrates the variation of execution delay with increasing numbers of computing tasks under different scheduling strategies. As the task load increases, the execution delay of all schemes grows due to intensified competition for computing and communication resources. Nevertheless, the proposed FI-Opt scheme consistently achieves the lowest execution delay. When the number of tasks reaches 15, FI-Opt reduces the execution delay by 6.7%, 8.0%, and 19.8% compared with task-only scheduling, resource-only scheduling, and no scheduling, respectively. This demonstrates that FI-Opt maintains favourable scalability and delay performance under multi-task conditions.

(2)Energy consumption. Figure 3 presents the energy consumption comparison as the task number increases. As with the delay trend, overall energy consumption increases with heavier workloads across all schemes. However, the proposed FI-Opt scheme exhibits the lowest energy consumption across all task settings. At 15 tasks, FI-Opt improves energy consumption performance by 9.9%, 10.6%, and 54.1% relative to the three baseline schemes, indicating its effectiveness in jointly optimizing task execution and resource utilization.

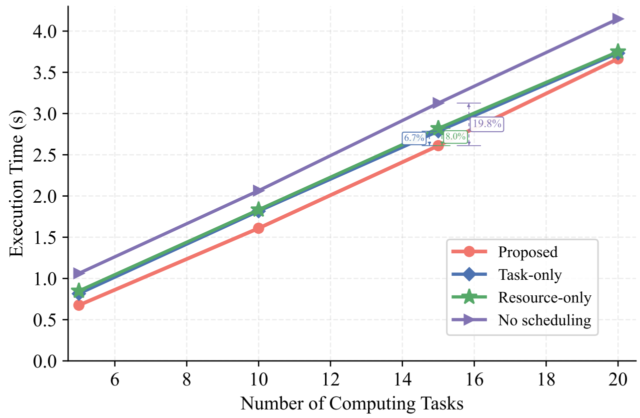

Figure 2: Time consumption varies with the number of computing tasks under different algorithms.

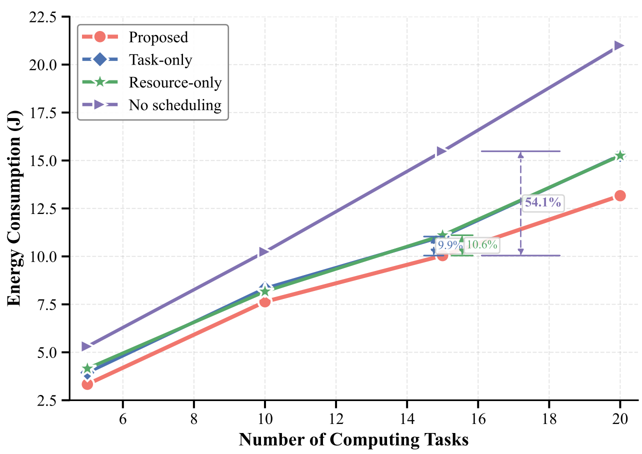

Figure 3: Energy consumption varies with the number of computing tasks under different algorithms.

(3)Task execution efficiency. The task execution efficiency under different task quantities is depicted in Figure 4. The proposed FI-Opt scheme consistently achieves the highest execution efficiency. In particular, when the number of tasks is 15, FI-Opt improves task execution efficiency by 6.7%, 8.0%, and 19.8% compared with task-only scheduling, resource-only scheduling, and no scheduling, respectively. Moreover, under light workloads (e.g., 5 tasks), FI-Opt exhibits significantly higher efficiency gains of 20.7%, 25.7%, and 57.2%, highlighting its strong adaptability to both low-load and high-load scenarios.

(4)Energy efficiency. Figure 5 further compares the energy efficiency of different schemes. The proposed FI-Opt scheme consistently outperforms the baseline methods in terms of energy efficiency. When the number of tasks is 15, FI-Opt achieves energy efficiency improvements of 9.9%, 10.6%, and 54.1% over task-only scheduling, resource-only scheduling, and no scheduling, respectively. These results indicate that FI-Opt effectively balances task performance and energy consumption under varying task workloads.

Overall, the experimental results demonstrate that the proposed FI-Opt scheme caachieve low delay, low energy consumption, and high execution efficiency across varying task counts, validating its scalability and robustness for multi-task cross-domain wireless computing power networks.

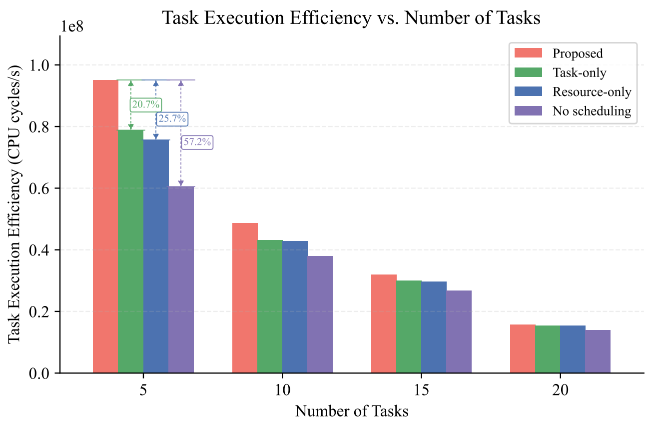

Figure 4: Task execution efficiency varies with the number of computing tasks under different algorithms.

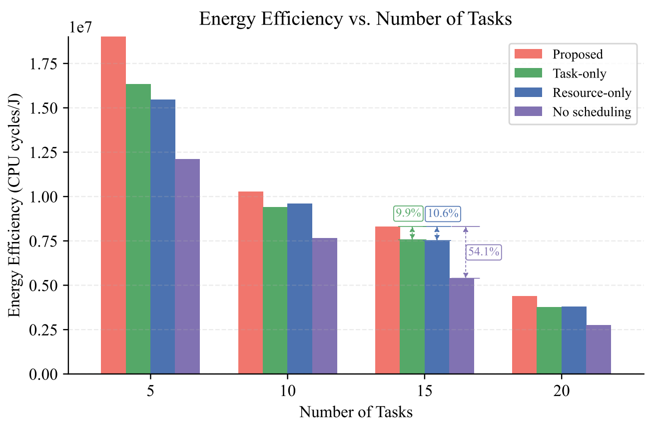

Figure 5: Energy efficiency varies with the number of computing tasks under different algorithms.

5.2.2 Performance Impact of Varying Numbers of Computing Nodes

In practical WCPNs, the number of available computing nodes may vary significantly due to dynamic deployment, mobility, and cross-domain participation. Therefore, it is essential to evaluate whether the proposed FI-Opt scheme can maintain stable performance as the network scale changes. To this end, we analyze the system performance under different numbers of computing nodes from the perspectives of execution delay, energy consumption, task execution efficiency, and energy efficiency.

The execution delay under different node configurations is depicted in Figure 6. As more computing nodes become available, the overall execution time decreases for all schemes, benefiting from increased parallel processing. However, the proposed FI-Opt scheme consistently achieves the lowest delay across all settings. When the number of computing nodes reaches 40, FI-Opt reduces the execution delay by 15.59%, 20.85%, and 35.36% compared with task-only scheduling, resource-only scheduling, and no scheduling, respectively. This significant reduction indicates that FI-Opt can effectively coordinate a large number of heterogeneous computing nodes and fullleverageof network scaling.

Figure 7 illustrates the corresponding energy consumption

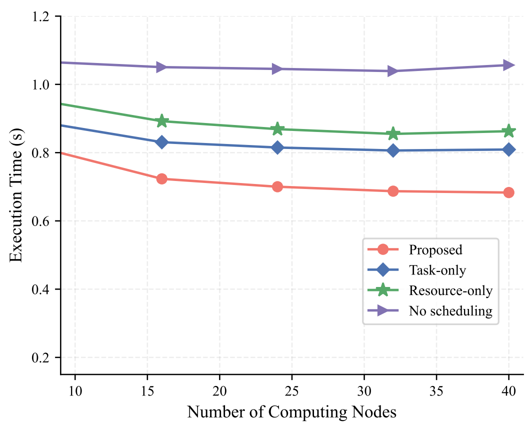

Figure 6: Time complexity varies with the number of computing nodes under different algorithms.

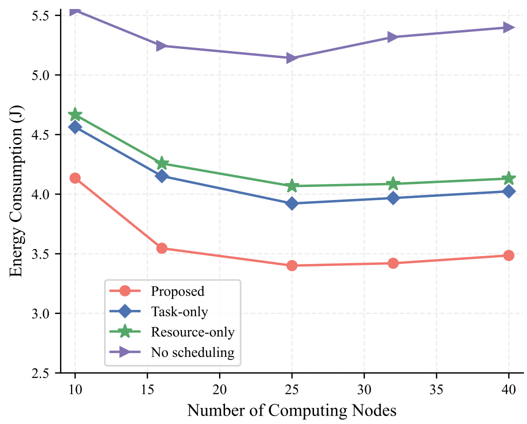

Figure 7: Energy consumption varies with the number of computing nodes under different algorithms.

results. Although increasing the number of computing nodes generally reduces energy consumption by improving workload distribution, the proposed FI-Opt scheme clearly outperformsbaseline methods. At 40 computing nodes, FI-Opt reduces energy consumption by 13.38%, 15.63%, and 35.45% compared with the three baseline schemes. These results demonstrate that FI-Opt not only improves computational performance but also enhances energy-aware resource utilization in large-scale networks.

The impact of node scaling on task execution efficiency is shown in Figure 8. Across all node configurations, the proposed FI-Opt scheme achieves the highest execution efficiency. In particular, when the number of computing nodes is 32, FI-Opt improves execution efficiency by 17.41%, 24.39%, and 50.71% over task-only scheduling, resource-only scheduling, and no scheduling. As the number of nodes increases to 40, the performance gains further increase to 19.04%, 25.35%, and 55.07%, respectively. These results highlight the strong capability of FI-Opt to leverage additional FI-Opt's strong ability to execute.

Energy efficiency under varying numbers of computing nodes is presented in Figure 9. The proposed FI-Opt scheme consistently outperforms the baseline methods. When the number of nodes is 32, FI-Opt achieves energy efficiency improvements of 16.31%, 20.75%, and 56.97% compared with task-only scheduling, resource-only scheduling, and no scheduling. When the number of nodes increases to 40, the corresponding improvements further rise to 16.52%, 19.48%, and 54.46%. These results confirm that FI-Opt remains highly energy-efficient as the network scale expands.

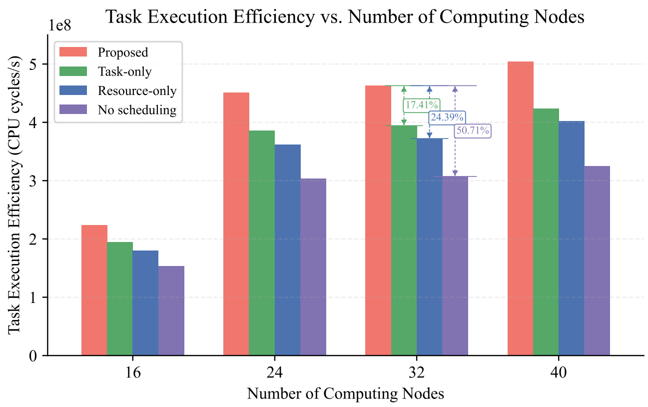

Figure 8: Task execution efficiency varies with the number of computing nodes under different algorithms.

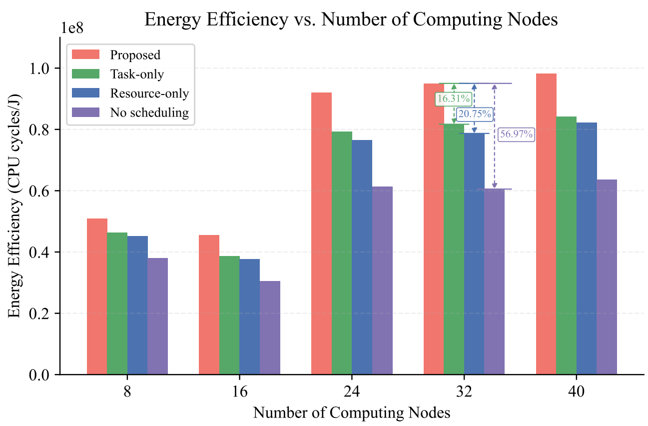

Figure 9: Energy efficiency varies with the number of computing nodes under different algorithms.

Overall, the above observations indicate that the proposed FI-Opt scheme exhibits strong scalability and robustness with respect to the number of computing nodes, and can effectively adapt to large-scale and dynamically changing wireless computing power networks.

5.2.3 Results Summary

For representative analysis, Table 2 summarizes the key performance indicators of the proposed approach with a task number of 5. Functionally, all major system components operate as expected; extensive testing validates the accuracy and completeness of data acquisition, transmission, storage, and processing, with correct recording of computing node configurations, consistent mapping between user requests and allocated resources, effective resource planning matched to task requirements, and accurate real-time updates of node operational states such as training loss and model accuracy, confirming the framework's functional correctness and reliability. Performance-wise, the proposed FI-Opt scheme outperforms all baseline methods across model accuracy, execution delay, energy consumption, energy efficiency, and task execution efficiency, maintaining superiority under diverse task loads and computing node scales and demonstrating strong effectiveness, scalability, and robustness as system size increases. Overall, the experimental results fully validate the effectiveness, reliability, and scalability of the proposed FI-Opt-based cross-domain collaborative computing framework.

6 Conclusion

This paper proposes a cross-domain collaboration framework in WCPNs under heterogeneous resources and privacy constraints. To enable collaborative intelligence without raw data sharing and to mitigate the impact of non-IID data, a federated intelligence–driven learning framework with a FedOpt-based aggregation mechanism was proposed, achieving improved convergence stability and model accuracy. Furthermore, to handle cross-domain resource allocation and multi-task execution under dynamic network conditions, a task-aware optimization model was formulated by jointly considering communication links, computing capabilities, task requirements, and energy constraints, and an efficient solution was obtained to derive near-optimal task orchestration strategies. Experimental results verified that the proposed approach consistently reduces execution latency and energy consumption while improving task execution efficiency and energy efficiency compared with baseline schemes, demonstrating its effectiveness for cross-domain collaboration in wireless computing power networks.

Funding

This work is supported by the Beijing Natural Science Foundation (L251039), the National Natural Science Foundation

Table 2: Overall experimental results under the baseline setting (number of tasks = 5).

Scheme	Latency (s)	Energy (J)	Efficiency metrics	
			Energy efficiency[1]	Execution rate[2]
Proposed	**0.68**	**3.33**	$\mathbf{1.927 \times 10^7}$	$\mathbf{9.512 \times 10^7}$
Task-only	0.82	3.94	1.632×10^7	7.881×10^7
Resource-only	0.85	4.15	1.547×10^7	7.566×10^7
No scheduling	1.06	5.30	1.211×10^7	6.050×10^7

[1] Energy efficiency = processed task volume / energy consumption. [2] Execution rate = processed task volume / time cost.

of China under Grant U2468201, and the Fundamental Research Funds for the Central Universities 2025JBZY013, 2025JBXT010.

Author Contributions

Writing and methodology: Yuan Feng and Jiayao Yuan; experiments: Jiajia Liu; conceptualization: Yunlong Lu; supervision: Yunlong Lu and Hao Wu. All authors have read and agreed to the published version of the manuscript.

Conflict of Interest

All the authors declare that they have no conflict of interest.

References

[1] Chowdhury, M.Z., Shahjalal, M., Ahmed, S., Jang, Y.M.: 6G Wireless Communication Systems: Applications, Requirements, Technologies, Challenges, and Research Directions. IEEE Open Journal of the Communications Society **1**, 957–975 (2020). https://doi.org/10.1109/OJCOMS.2020.3010270

[2] Yang, H., Alphones, A., Xiong, Z., Niyato, D., Zhao, J., Wu, K.: Artificial-Intelligence-Enabled Intelligent 6G Networks. IEEE Network **34**(6), 272–280 (2020). https://doi.org/10.1109/MNET.011.2000195

[3] Deng, Z., Guo, Y., Han, C., Ma, W., Xiong, J., Wen, S., Xiang, Y.: AI Agents Under Threat: A Survey of Key Security Challenges and Future Pathways. ACM Computing Surveys **57**(7), 1–36 (2025). https://doi.org/10.1145/3716628

[4] Sun, W., Lin, X., Shi, Y., Zhang, C., Wu, H., Zheng, S.: SparseDrive: End-to-End Autonomous Driving via Sparse Scene Representation. In 2025 IEEE International Conference on Robotics and Automation (ICRA), pp. 8795–8801 (2025). https://doi.org/10.1109/ICRA55743.2025.11128800

[5] Alhakamy, A.A.: Extended Reality (XR) Toward Building Immersive Solutions: The Key to Unlocking Industry 4.0. ACM Computing Surveys **56**(9), 1–38 (2024). https://doi.org/10.1145/3652595

[6] May, M.C., Glatter, D., Arnold, D., Pfeffer, D., Lanza, G.: IIoT System Canvas—From architecture patterns towards an IIoT development framework. Journal of Manufacturing Systems **72**, 437–459 (2024). https://doi.org/10.1016/j.jmsy.2023.12.001

[7] Liu, J., Lu, Y., Wu, H., Ai, B., Jamalipour, A., Zhang, Y.: Joint Task Coding and Transfer Optimization for Edge Computing Power Networks. IEEE Transactions on Network Science and Engineering **12**(4), 2783–27962 (2025). https://doi.org/10.1109/TNSE.2025.3554100

[8] Shen, H., Lu, Y., Li, H., Zhao, W., Feng, Y., Yuan, Y.: Reliable Federated Learning-Based Wireless Computing Power Scheduling for Efficient Edge Computing Networks. In 2024 IEEE 24th International Conference on Communication Technology (ICCT), pp. 168–173 (2024). https://doi.org/10.1109/ICCT62411.2024.10946361

[9] Teng, F., Ban, Z., Li, T., Sun, Q., Li, Y.: A privacy-preserving distributed economic dispatch method for integrated port microgrid and computing power network. IEEE Transactions on Industrial Informatics **20**(8), 10103–10112 (2024). https://doi.org/10.1109/TII.2024.3393569

[10] Lu, Z., Pan, H., Dai, Y., Si, X., Zhang, Y.: Federated learning with non-iid data: A survey. IEEE Internet of Things Journal **11**(11), 19188–19209 (2024). https://doi.org/10.1109/JIOT.2024.3376548

[11] Qin, K., Gu, S., Zhang, Z., Zhang, Q., Xiang, W.: Relay Selection and Load Allocation for LT Coded Distributed Computing in Two-Hop Heterogeneous Computation Network. In 2024 IEEE Wireless Communications and Networking Conference (WCNC), pp. 1–6 (2024). https://doi.org/10.1109/WCNC57260.2024.10571074

[12] Gecer, M., Garbinato, B.: Federated learning for mobility applications. ACM Computing Surveys **56**(5), 1–28 (2024). https://doi.org/10.1145/3637868

[13] Zhou, Y., Ye, Q., Lv, J.: Communication-efficient federated learning with compensated overlap-fedavg. IEEE Transactions on Parallel and Distributed Systems **33**(1), 192–205 (2021). https://doi.org/10.1109/TPDS.2021.3090331

[14] Lee, H., Lee, S.H., Quek, T.Q.: Deep learning for distributed optimization: Applications to wireless resource management. IEEE Journal on Selected Areas in Communications, **37**(10), 2251–2266 (2019). https://doi.org/10.1109/JSAC.2019.2933890

[15] Liu, Y., Zhang, W., Zhang, Q., Norouzi, M.: An optimized human resource management model for cloud-edge computing in the internet of things. Cluster Computing **25**(4), 2527–2539 (2022). https://doi.org/10.1007/s10586-021-03319-y

[16] Hussain, F., Hassan, S.A., Hussain, R., Hossain, E.: Machine learning for resource management in cellular and IoT networks: Potentials, current solutions, and open challenges. IEEE communications surveys & tutorials **22**(2), 1251–1275 (2020). https://doi.org/10.1109/COMST.2020.2964534

[17] Shen, Y., Shi, Y., Zhang, J., Letaief, K.B.: Graph neural networks for scalable radio resource management: Architecture design and theoretical analysis. IEEE Journal on Selected Areas in Communications **39**(1), 101–115 (2020). https://doi.org/10.1109/JSAC.2020.3036965

[18] Xiang, X., Li, Q., Khan, S., Khalaf, O.I.: Urban water resource management for sustainable environment planning using artificial intelligence techniques. Environmental impact assessment review **86**, 106515 (2021). https://doi.org/10.1016/j.eiar.2020.106515

[19] Xia, S., Yao, Z., Li, Y., Mao, S.: Online distributed offloading and computing resource management with energy harvesting for heterogeneous MEC-enabled IoT. IEEE Transactions on Wireless Communications **20**(10), 6743–6757 (2021). https://doi.org/10.1109/TWC.2021.3076201

[20] Asad, M., Moustafa, A., Ito, T.: Fedopt: Towards communication efficiency and privacy preservation in federated learning. Applied Sciences **10**(8), 2864 (2020). https://doi.org/10.3390/app10082864

[21] Li, L., Fan, Y., Tse, M., Lin, K. Y.: A review of applications in federated learning. Computers & Industrial Engineering **149**, 106854 (2020). https://doi.org/10.1016/j.cie.2020.106854

[22] Ma, X., Zhu, J., Lin, Z., Chen, S., Qin, Y.: A state-of-the-art survey on solving non-iid data in federated learning. Future Generation Computer Systems **135**, 244–258 (2022). https://doi.org/10.1016/j.future.2022.05.003

[23] Zhou, Y., Shi, M., Tian, Y., Li, Y., Ye, Q., Lv, J.: Federated CINN clustering for accurate clustered federated learning. In ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 5590–5594 (2024). https://doi.org/10.1109/ICASSP48485.2024.10447282

[24] Gao, T., Liu, K., Yang, Y., Liu, X., Zhang, P., Wang, G.: FedPC: An Efficient Prototype-Based Clustered Federated Learning on Medical Imaging. IEEE Journal of Biomedical and Health Informatics **29**(10), 7396–7408 (2025). https://doi.org/10.1109/JBHI.2025.3567055

[25] Pillutla, K., Kakade, S. M., Harchaoui, Z.: Robust aggregation for federated learning. IEEE Transactions on Signal Processing **70**, 1142–1154 (2022). https://doi.org/10.1109/TSP.2022.3153135

[26] Hu, C., Wang, S., Liu, C., Zhang, T.: Efficient privacy-preserving data aggregation for lightweight secure model training in federated learning. In 2023 7th International Conference on Cryptography, Security and Privacy (CSP), pp. 119–123 (2023). https://doi.org/10.1109/CSP58884.2023.00026

[27] Tan, K., Feng, L., Dán, G., Törngren, M.: Decentralized convex optimization for joint task offloading and resource allocation of vehicular edge computing systems. IEEE Transactions on Vehicular Technology **71**(12), 13226–13241 (2022). https://doi.org/10.1109/TVT.2022.3197627

[28] Chen, Y., Yang, Y., Wu, Y., Huang, J., Zhao, L.: Joint trajectory optimization and resource allocation in UAV-MEC systems: A Lyapunov-assisted DRL approach. IEEE Transactions on Services Computing **18**(2), 854–867 (2025). https://doi.org/10.1109/TSC.2025.3544124

[29] Jalali, J., Tabassum, H., Famaey, J., Saad, W., Uysal, M.: Placement, Orientation, and Resource Allocation Optimization for Cell-Free OIRS-Aided OWC Network. IEEE Transactions on Vehicular Technology **74**(6), 10011–10016 (2025). https://doi.org/10.1109/TVT.2025.3538334

[30] Min, H., Rahmani, A.M., Ghaderkourehpaz, P., Moghaddasi, K., Hosseinzadeh, M.: A joint optimization of resource allocation management and multi-task offloading in high-mobility vehicular multi-access edge computing networks. Ad Hoc Networks **166**, 103656 (2025). https://doi.org/10.1016/j.adhoc.2024.103656

[31] Jie, Y., Guo, C., Choo, K.K.R., Liu, C.Z., Li, M.: Game-theoretic resource allocation for fog-based industrial internet of things environment. IEEE Internet of Things Journal **7**(4), 3041–3052 (2020). https://doi.org/10.1109/JIOT.2020.2964590

[32] Wang, Y., Yang, X.: Intelligent resource allocation optimization for cloud computing via machine learning. arXiv preprint arXiv:2504.03682 (2025). https://doi.org/10.48550/arXiv.2504.03682

[33] Lambora, A., Gupta, K., Chopra, K.: Genetic algorithm-A literature review. In 2019 international conference on machine learning, big data, cloud and parallel computing (COMITCon), pp. 380–384 (2019). https://doi.org/10.1109/COMITCon.2019.8862255