Task-Oriented Based Computing Power Measurement for Task Scheduling in Computing Power Networks

Peilong Ding*, Nuocheng Yang*, Xinxin He*, Sihua Wang*, Changchuan Yin*

*Beijing Laboratory of Advanced Information Network,
Beijing University of Posts and Telecommunications, Beijing, China,
Emails: {peilongding,yangnuocheng,hxx_9000,sihuawang,ccyin}@bupt.edu.cn.

2025 IEEE/CIC International Conference on Communications in China (ICCC) | 979-8-3315-4444-7/25/$31.00 ©2025 IEEE | DOI: 10.1109/ICCC65529.2025.11149109

***Abstract*—Computing Power Network (CPN) has emerged as a promising paradigm that integrates communication networks with distributed computing resources to support new-generation applications, such as Artificial Intelligence (AI) and Virtual Reality (VR). While CPN enables efficient utilization of heterogeneous computing resources, its implementation faces several challenges, including accurate computing power measurement (CPM) of heterogeneous computing resources and efficient task scheduling. To adapt computing tasks with different attributes to heterogeneous computing resources, we propose a novel framework for CPM and computing task scheduling. First, to facilitate accurate CPM, a clustering-based algorithm categorizing tasks into distinct types based on their attributes is proposed. Then, a neural network model is designed to measure the computing power of heterogeneous computing resources after the task clustering. Second, we formulate a computing task scheduling problem to maximize the matching degree between tasks and computing resources based on the proposed CPM, whose environment is modeled as a Markov Decision Process (MDP). Then, a deep reinforcement learning-based task scheduling strategy with an adaptive exploration mechanism to accelerate the convergence is proposed. Simulation results show that the proposed framework can improve the rewards of task scheduling by up to 30.4% compared to baselines.**

***Keywords*—Computing Power Network, Computing Power Measurement, Task Scheduling.**

I. Introduction

With the rapid development of Artificial Intelligence (AI) applications, the demand for ubiquitous deployment of computing resources (e.g., cloud services, computing resource providers, and personal computing power) has grown significantly [1]–[3]. Meanwhile, computing resources in networks are gradually shifting from cloud computing centers to network edges [4]. Computing power network (CPN), integrating heterogeneous computing resources with communication networks to enable efficient task scheduling for delay-sensitive and computation-intensive applications (e.g., AR/VR rendering), has been proposed [5]. However, the implementation of CPN over wireless networks presents several challenges, including accurate computing power measurement (CPM) [6], which quantifies the heterogeneous computing resources in CPN, and efficient task scheduling [7].

Recently, a number of existing works such as [8]–[11] have studied the CPM methods. The authors in [8]–[11] proposed a weighted-sum paradigm to measure the computing power for heterogeneous resources by aggregating them through relative weights. In particular, the authors in [8], [9] employed manually configured weights in the paradigm. Compared to the static weights used in [8], [9], the authors in [10] employed the entropy of different hardware as the aggregation weights. In contrast, the authors in [11] determined the weights through an auction mechanism, which are dynamically adjusted based on the bids between computing resource demanders and providers. However, the authors in [8]–[11] ignored that the computing resources exhibit distinct capabilities when processing varying task types, thus leading to unreliable CPM performance.

Recently, a number of existing works such as [12]–[14] have studied task scheduling issues in CPN. The authors in [12] adopted homogeneous computing tasks and proposed a Lyapunov-optimized deep reinforcement learning algorithm to optimize the delay of computing tasks under load balancing constraints. The authors in [13] adopted green energy-driven tasks and proposed a PPO-based deep reinforcement learning algorithm to maximize green energy utilization. However, the authors in [12], [13] ignored the diverse types of tasks in CPN for differentiated task scheduling, resulting in incompatibility between tasks and computing resources. To fill this gap, the authors in [14] measured the compatibility between tasks and computing resources by calculating the Euclidean distance between their features and proposed a meta-reinforcement learning approach to maximize the compatibility in task scheduling. However, calculating the compatibility between each task and computing resources introduces significant costs.

To address the challenges in CPM and task scheduling, we propose a comprehensive framework for CPN that integrates CPM and efficient task scheduling strategy. Our key contributions are as follows:

- To facilitate accurate CPM, we first introduce a clustering-based task classification algorithm that categorizes computing tasks into distinct types based on their attributes. Then, we propose a novel CPM model to achieve accurate CPM, which can capture the association between computing resources and typical tasks.
- To enhance the task scheduling, we first introduce the matching degree to quantify the compatibility between tasks and computing resources, and formulate the considered task scheduling environment as an MDP. Then, we propose a deep reinforcement learning-based scheduling strategy to maximize the global matching degree with the delay constraints. Specifically, we calculate the computing power of each node following the task type,

This work was supported by the National Key Research and Development Program of China under Grant 2024YFE0200300.

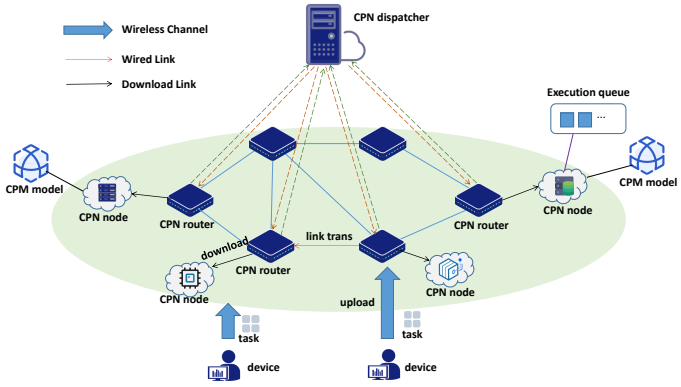

Fig. 1. Illustration of the considered CPN system.

thus reducing the cost of task scheduling. Besides, to accelerate the convergence rate of the proposed strategy, an adaptive exploration mechanism is proposed.

- Numerical evaluation results show that our proposed framework can significantly enhance the accuracy of CPM, task scheduling performance, and convergence rate compared with baselines.

The rest of the article is structured as follows. Section II outlines the system model. Section III details the proposed CPM model and scheduling algorithm. Simulation results are presented in Section IV, and Section V offers the conclusion.

II. System Model and Problem Formulation

As shown in Fig. 1, we consider a CPN system, which consists of a set $\mathcal{V}$ of V devices that generate computing tasks, a CPN dispatcher that schedules the tasks offloading to M CPN nodes in set $\mathcal{M}$ for computing. Moreover, a set $\mathcal{K}$ of K CPN routers are connected with devices and CPN nodes for data transmission. In each time slot t, each device v can generate task $a_{v,t} \in \mathcal{A}_t$ which can be divided into N types (AI training tasks, AI inference tasks, etc.) with the probability $\boldsymbol{\lambda}_v = [\lambda_{v,1}, \cdots, \lambda_{v,N}]$, where $\lambda_{v,n} \in (0,1)$ and $\sum_{n=1}^{N} \lambda_{v,n} = 1$. Then, the detail information of each task $a_{v,t}$ can be given by $\boldsymbol{b}(a_{v,t}) = [q_{v,t}, w_{v,t}, e_{v,t}, r_{v,t}, z_{v,t}]$, where $q_{v,t} \in [1, N]$ indicates the type of task, $w_{v,t}$ is the data volume, $e_{v,t}$ represents the computation amount, $r_{v,t}$ is the delay requirement, and $z_{v,t}$ is the algorithm complexity. To describe the computing capabilities of heterogeneous CPN nodes for various types of tasks, we introduce $\boldsymbol{F}_m = [f_{m,1}, \cdots, f_{m,N}]$ as the computing power based on device m' hardware information $\mathcal{H}_m \in \mathbb{R}^{Y \times 1}$ to represent the hardware information (e.g., CPU, GPU, NPU, and cache). The overall process of the task scheduling in the considered CPN system is given as follows:

1) The CPN node m transmits its computing power information $\boldsymbol{F}_m$ to the CPN dispatcher through the network.
2) Each device v generates a computing task $a_{v,t}$ with type probability $\boldsymbol{\lambda}_v$ and transmits $a_{v,t}$ and $\boldsymbol{b}(a_{v,t})$ to its connected CPN router.
3) The CPN router reports $\boldsymbol{b}(a_{v,t})$ to the CPN dispatcher, which then determines the task scheduling according to the $\boldsymbol{b}(a_{v,t})$ and $\boldsymbol{F}_m$ of each CPN node m.
4) The CPN routers transmit $a_{v,t}$ to the corresponding CPN node for computing.
5) The CPN node computes $a_{v,t}$ and returns the result to device.

Steps 2)-5) are performed until all tasks have been calculated.

A. Transmission Process

We utilize the orthogonal frequency division multiple access (OFDMA) transmission scheme for task transmission from device v to its related CPN router k through the Rayleigh fading wireless channel [15], the data rate is given by

$$R_{v,k} = B\log_2\left(1+\frac{p_v h(d_{v,k})}{\sigma^2}\right), \tag{1}$$

where B represents the bandwidth, with p_v being the transmit power. $h(d_{v,k}) = \rho d_{v,k}^{-2}$ is the channel gain between device v and CPN router k. ρ is the Rayleigh fading parameter, and $d_{v,k}$ is the distance between device v and the CPN router k. σ^2 is the additive white Gaussian noise. Thus, the wireless transmission delay of task $a_{v,t}$ is given by

$$T_{v,k}^{upload} = \frac{w_{v,t}}{R_{v,k}}. \tag{2}$$

We utilize R_l to represent the data rate of link l between CPN routers. Therefore, the transmission delay of task $a_{v,t}$ from CPN router k to CPN node m is given by

$$T_{k,m}^{link} = \sum_{l \in \mathcal{L}_{k,m}} \frac{w_{v,t}}{R_l}, \tag{3}$$

where $\mathcal{L}_{k,m}$ is the set of paths from the source CPN router k to the destination CPN node m. Since the size of task information $\boldsymbol{b}(a_{v,t})$ can be ignored compared with $w_{v,t}$, we disregard the delay for transmitting $\boldsymbol{b}(a_{v,t})$ from CPN router to the CPN dispatcher. Similarly, we don't consider the delay for receiving results from the CPN node.

Once the CPN dispatcher determines the scheduling location of task $a_{v,t}$, the total transmission delay of task $a_{v,t}$ is given by

$$T_{a_{v,t},m}^{trans} = T_{v,k}^{upload} + T_{k,m}^{link}. \tag{4}$$

B. Computation Process

We introduce the scheduling decisions $x_{a_{v,t},m} \in [0,1]$, where $x_{a_{v,t},m} = 1$ implies that task $a_{v,t}$ is offloaded to the CPN node m, and $x_{a_{v,t},m} = 0$, otherwise. In our work, each task is indivisible and can only be processed by one CPN node. Thus, we can have

$$\sum_{m \in \mathcal{M}} x_{a_{v,t},m} = 1, \ \forall a_{v,t} \in \mathcal{A}_t. \tag{5}$$

For the convenience of describing the computation process, we measure the computing power of CPN node m for processing task $a_{v,t}$, which is given by

$$f_{m,q_{v,t}} = G(\mathcal{H}_m, \boldsymbol{b}(a_{v,t}), \tau), \tag{6}$$

where $G(\cdot)$ represents the paradigm of transforming heterogeneous hardware information $\mathcal{H}_m$ and the task information

$\boldsymbol{b}(a_{v,t})$ into a unified computing power $f_{m,q_{v,t}}$, and τ is the parameters of the function $G(\cdot)$.

Then, we introduce $\boldsymbol{Y}_m(t) = [y_{m,1}(t), \ldots]$ to describe the execution queue of CPN node m at time slot t, and S is the maximum length (i.e., $|\boldsymbol{Y}_m(t)| \leqslant S$). The total process delay of task $a_{v,t}$ comprises both computing and queue waiting delay, which is given by

$$T^{pro}_{a_{v,t},m} = \sum_{j=1}^{|\boldsymbol{Y}_m(t)|-1} \frac{e_{m,j}}{f_{m,q_{m,j}}} + \frac{e_{v,t}}{f_{m,q_{v,t}}}, \tag{7}$$

where $e_{m,j}$ represents the computation amount for the j-th task in the execution queue.

C. Matching Degree Model

While the CPN nodes can handle different types of tasks, they exhibit distinct preferences and provide different computing capabilities due to the heterogeneous properties of hardware. To quantify this relationship, we define the computing preference degree $\rho_{a_{v,t},m} \in [0,1]$ between task type $q_{v,t}$ and CPN node m as follows

$$\rho_{a_{v,t},m} = \frac{f_{m,q_{v,t}}}{\hat{f}_m}, \tag{8}$$

where $\hat{f}_m$ is the maximum computing power of CPN node m, i.e., $\hat{f}_m = \max(\boldsymbol{F}_m)$.

To avoid most tasks being scheduled to high-capability CPN nodes, which results in extended queue delay and overloading, we introduce the utilization rate of computing resources, under the constraint that task $a_{v,t}$ can be completed by CPN node m, as follows

$$\beta_{a_{v,t},m} = \frac{e_{v,t}}{\left(r_{v,t} - T^{trans}_{a_{v,t},m}\right) f_{m,q_{v,t}}}. \tag{9}$$

Then, we introduce the matching degree between task $a_{v,t}$ and CPN node m as a weighted sum of preference degree and utilization rate, which is given by

$$\eta_{a_{v,t},m} = \mu_1 \rho_{a_{v,t},m} + \mu_2 \beta_{a_{v,t},m}, \tag{10}$$

where μ_1 and μ_2 represent the weighting values.

D. Optimization Formulation

Our goal is to maximize the global matching degree while satisfying the delay requirements of tasks. Accordingly, the optimization problem can be formulated as

$$\max_{\boldsymbol{X}_t, \tau} \sum_{t \in T} \sum_{m \in \mathcal{M}} \sum_{a_{v,t} \in \mathcal{A}_t} x_{a_{v,t},m} \eta_{a_{v,t},m}, \tag{11}$$

$$\text{s.t.} \sum_{m \in \mathcal{M}} x_{a_{v,t},m} = 1, \ \forall a_{v,t} \in \mathcal{A}_t, \tag{11a}$$

$$\sum_{m \in \mathcal{M}} x_{a_{v,t},m} (T^{pro}_{a_{v,t},m} + T^{trans}_{a_{v,t},m}) \leqslant r_{v,t}, \ \forall a_{v,t} \in \mathcal{A}_t, \tag{11b}$$

$$|\boldsymbol{Y}_m(t)| \leqslant S, \ \forall m \in \mathcal{M}, \tag{11c}$$

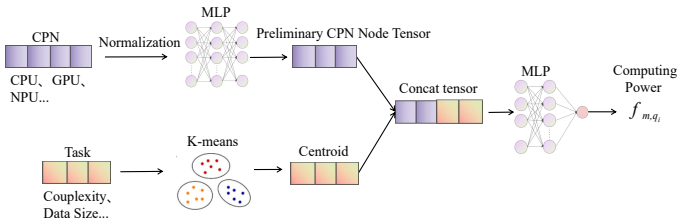

Fig. 2. Architecture of the CPM model.

where $\boldsymbol{X}_t = \left[x_{a_{v,t},m}, \cdots\right]$ represents the scheduling set. Constraint (11a) ensures that each task can be deployed only on one CPN node. The constraint (11b) guarantees that the CPN node can complete the task within the maximum delay requirement. The constraint (11c) ensures that the CPN node's execution queue length is prevented from overflowing.

The problem in (11) is difficult to solve due to the following reasons. First, task classification is challenging because the diverse requirements of tasks make it hard to categorize them accurately. Second, proposing an accurate CPM is challenging due to the heterogeneous nature of computing resources, as converting them into a unified computing power is hard. Third, the global task scheduling is an integer programming problem making it difficult to solve due to its NP-hard nature.

To address these challenges, we first introduce a K-means-based clustering algorithm to divide tasks into N types based on their attributes before task scheduling. Subsequently, we propose a CPM model based on a novel neural network that quantifies heterogeneous computing resources into a unified computing power based on task types. Finally, we introduced a deep reinforcement learning-based task scheduling strategy with an adaptive exploration mechanism to accelerate the convergence rate, which maximizes the matching degree under the delay requirement.

III. Proposed CPM Model and Algorithm

A. Task Classification

We employ the K-means algorithm for task classification since it can effectively address the challenge of classifying tasks with diverse requirements by grouping tasks with similar demands into clusters. The initial values of task centroids $\boldsymbol{c}' = [c'_1, \cdots, c'_N]$ are chosen randomly from task information set $\mathcal{B} = \{\boldsymbol{b}(a_{v,t}), \cdots\}$. Then, the task $a_{v,t}$ is declared to be the same cluster with the closest centroids which divide $\mathcal{A}$ into $\mathcal{C} = \{\mathcal{C}_1, \cdots, \mathcal{C}_N\}$. After each clustering, the parameters of new centroids are estimated as the mean of tasks that are members of the cluster. The procedures of partition assignment and updating centroid are performed repeatedly until stable centroids have been reached [16]. We assume that the set of stable centroids is $\boldsymbol{c} = [c_1, \cdots, c_N]$, and the tasks in the same cluster $\mathcal{C}_i$ are classified as a unified standard task c_i. Then, the computing power to handle the tasks $a_{v,t} \in \mathcal{C}_i$ for CPN node m is determined as follows

$$f_{m,q_{v,t}} = f_{m,c_i}, \ \forall a_{v,t} \in \mathcal{C}_i. \tag{12}$$

B. Components of CPM Model

From (6), the computing power is determined by both the information of hardware and task attributes. To capture

the association between computing power and task types simultaneously, we propose a CPM model based on a neural network, as illustrated in Fig. 2. The proposed CPM model consists of four parts: the hardware resources input model, the task attribute input model, the concatenate model, and the output model.

In the hardware resources input model, we employed the Z-score method to standardize the hardware resources information $\mathcal{H}_m$ to $\widehat{\mathcal{H}_m}$, which is given by

$$\widehat{\mathcal{H}_m} = \frac{x_j^i - \overline{X}_j}{\sigma_j},\ \forall i \in 1,2,...,M,\ \forall j \in 1,2,...,Y, \quad (13)$$

where $\overline{X}_j$ and σ_j denote the mean and standard deviation of the corresponding feature in the data set, respectively. Then, the preliminary CPN node tensor h'_m is given by

$$h'_m = \psi\left(\phi_1\left(\widehat{\mathcal{H}_m}\right)\right), \quad (14)$$

where $\psi(\cdot)$ indicates the relu function, and $\phi_1(\cdot)$ represents the multilayer perceptron layer (MLP).

In the task attribute input model, we assume the considered task $a_{v,t}$ belongs to the cluster $\mathcal{C}_i$, and the attributes of the cluster's centroid are $\boldsymbol{b}(c_i)$ which will be used together with node information tensor h'_m as the inputs of the concatenate model. Then, the concatenate model is designed to combine the node information h'_m with task attributes $\boldsymbol{b}(c_i)$, which is given by

$$\varphi_{n,m} = \Phi(\boldsymbol{b}(c_i), h'_m), \quad (15)$$

where $\varphi_{n,m}$ is the concatenated tensor, and $\Phi(\cdot)$ represents the concatenate function. Given the concatenated tensor $\varphi_{n,m}$, we utilize an MLP in the output model to transform the concatenated tensor into predicted computing power $f'_{m,n}$, which can be expressed as

$$f'_{m,n} = \psi(\phi_2(\varphi_{n,m})), \quad (16)$$

where $\phi_2(\cdot)$ represents the MLP in output model.

Here, we minimize the mean square error $\mathcal{G}$ which measures the gap between predicted and ground truth computing power, as follows

$$\mathcal{G} = \frac{1}{n}\sum_{i=1}^{n}(f'_{m,q_i} - f_{m,q_i})^2, \quad (17)$$

where n is the batch size and f_{m,q_i} represents the ground truth computing power.

C. Modeling of Markov Decision Process

Given the CPM, we apply Deep Q-Network (DQN) as the reinforcement learning framework [17], which combines Q-learning with deep neural networks to handle large state spaces and actions, making it particularly advantageous for CPN. Then, we propose a DQN-based task Scheduling strategy for Matching degree maximization (DQSM) to effectively address the problem in (11).

To apply deep reinforcement learning to the task scheduling in CPN, we need to first formulate our problem as a Markov Decision Process (MDP), which can be defined by the tuple $M = \langle \mathcal{S}_t, \boldsymbol{X}_t, r(\mathcal{S}_t, \boldsymbol{X}_t), \mathcal{P}\rangle$, where $\mathcal{S}_t$ is the state space, $\boldsymbol{X}_t$ is the action, $r(\mathcal{S}_t, \boldsymbol{X}_t)$ is the reward, $\mathcal{P}$ is the state transition function, and are detailed as follows:

- **State**: The system state $\mathcal{S}_t$ is defined as $\mathcal{S}_t = \{\boldsymbol{b}(a_{v,t}), \mathcal{F}, \mathcal{Y}(t)\}$, which is primarily composed of three parts including the current task information $\boldsymbol{b}(a_{v,t})$, total node information set with computing power $\mathcal{F}$ and queue length $\mathcal{Y}(t)$.
- **Action**: The scheduling actions for task is given by $\boldsymbol{X}_t = \left[x_{a_{v,t},1}, \cdots, x_{a_{v,t},m}\right]$.
- **Reward**: The reward function is designed based on the optimization goal in (11) while satisfying the delay constraints, which is given by

$$r(\mathcal{S}_t, \boldsymbol{X}_t) = \sum_{m\in\mathcal{M}}\sum_{a_{v,t}\in\mathcal{A}_t} x_{a_{v,t},m}\Big(\eta_{a_{v,t},m} - \xi\max(0, T^{pro}_{a_{v,t},m} + T^{trans}_{a_{v,t},m} - r_{v,t})\Big), \quad (18)$$

  where ξ is the penalty weight for delay violations.
- **State Transition Probability**: The state transition probability $P(\mathcal{S}_{t+1}|\mathcal{S}_t, \boldsymbol{X}_t)$ is the probability of transitioning from state $\mathcal{S}_t$ to state $\mathcal{S}'_t$ when taking action $\boldsymbol{X}_t$, which is

$$P(\mathcal{S}_{t+1}|\mathcal{S}_t, \boldsymbol{X}_t) = \Pr\{\mathcal{S}_{t+1} = \mathcal{S}'_t|\mathcal{S}_t, \boldsymbol{X}_t\}. \quad (19)$$

D. Proposed DQSM Algorithm

The DQN architecture consists of two primary modules: the main Q-Network and the target Q-Network. The main Q-Network parameterized by θ is a deep neural network responsible for approximating the Q-function $\mathcal{Q}(\mathcal{S}_t, \boldsymbol{X}_t; \theta)$, which represents the expected cumulative reward of taking action $\boldsymbol{X}_t$ in state $\mathcal{S}_t$ by combining both immediate and future rewards, thereby enabling the agent to evaluate the long-term value of actions in a given state. To ensure the training stability, the target Q-Network is introduced as a copy of the main Q-Network, with its parameters θ^- periodically synchronized with θ. The target Q-Network is used to compute stable target Q-values z_t through the Q-function, which is calculated as

$$z_t = r(\mathcal{S}_t, \boldsymbol{X}_t) + \gamma\max_{\boldsymbol{X}_{t+1}}\mathcal{Q}(\mathcal{S}_{t+1}, \boldsymbol{X}_{t+1}; \theta^-) \quad (20)$$

where the discount factor $\gamma \in (0, 1]$ indicates the importance of future rewards relative to current rewards.

Using the target Q-value z_t, the main Q-Network is trained to minimize the temporal difference error, which measures the difference between the predicted Q-value and the target Q-value, thus enhancing the agent's decision-making capabilities. The temporal difference error is defined as:

$$\mathcal{L}(\theta) = \mathbb{E}\left[\left(z_t - \mathcal{Q}(\mathcal{S}_t, \boldsymbol{X}_t; \theta)\right)^2\right]. \quad (21)$$

Considering the huge and complex optimization space caused by a large number of tasks and heterogeneous CPN nodes, we adopt the Boltzmann-based exploration policy $\pi(\mathcal{S}_t, \boldsymbol{X}_t) = \left[\pi(\mathcal{S}_t, x_{a_{v,t},1}), \cdots, \pi(\mathcal{S}_t, x_{a_{v,t},m})\right]$ with an adaptive exploration rate to accelerate the convergence rate

Algorithm 1 Procedure for DQSM

Input: exploration rate $\omega(t)$, discount factor γ, replay buffer size υ, mini-batch size κ, synchronization interval C.
Output: Task scheduling decisions $\boldsymbol{X}^*$.

Initialize main Q-Network with random parameters θ and target Q-Network with $\theta^- = \theta$.
Initialize replay buffer $\mathcal{D}$ with capacity υ.
for episode $= 0, 1, 2, ...$ **do**
  Initialize random state $\mathcal{S}_t$.
  for $t = 0, 1, 2, ...$ **do**
    Compute the exploration policy $\pi(\mathcal{S}_t, x_{a_{v,t},m})$ using (22)
    Execute action $\boldsymbol{X}_t$, and observe reward r and next state $\mathcal{S}_{t+1}$.
    Store transition $(\mathcal{S}_t, \boldsymbol{X}_t, \mathcal{R}, \mathcal{S}_{t+1})$ in $\mathcal{D}$.
    Sample a mini-batch of κ transitions from $\mathcal{D}$.
    Compute the target Q-values z_t.
    Update θ by minimizing the temporal difference error $\mathcal{L}(\theta)$.
    if $t \mod C == 0$ **then**
      Synchronize target Q-Network: $\theta^- = \theta$.
    end if
  end for
end for
Use the trained Q-Network to determine the optimal task scheduling decisions $\boldsymbol{X}^* = \arg\max_{\boldsymbol{X}_t} \mathcal{Q}(\mathcal{S}_t, \boldsymbol{X}_t; \theta)$.
return Task scheduling decisions $\boldsymbol{X}^*$.

TABLE I
TYPICAL PARAMETERS OF TASKS

Task Type	Computation Amount	Delay Requirement	Algorithm Complexity
AI Training	$[2, 20]$ PFLOPs	$[50, 100]$ s	$[O(n^2), O(n^3)]$
AI Inference	$[10, 70]$ GFLOPs	$[30, 150]$ ms	$[O(n), O(n \log n)]$
AR/VR	$[20, 50]$ EFLOPs	$[20, 50]$ ms	$[O(n \log n), O(n^2)]$
Video Rendering	$[100, 300]$ GFLOPs	$[20, 150]$ ms	$[O(n \log n), O(n^2)]$

[18]. The exploration probability $\pi(\mathcal{S}_t, x_{a_{v,t},m})$ can be given by

$$\pi(\mathcal{S}_t, x_{a_{v,t},m}) = \frac{e^{\frac{\mathcal{Q}(\mathcal{S}_t, x_{a_{v,t},m})}{\omega(t)}}}{\sum_{i \in \mathcal{M}} e^{\frac{\mathcal{Q}(\mathcal{S}_t, x_{a_{v,t},i})}{\omega(t)}}}, \tag{22}$$

where $\omega(t) = \omega_0 \cdot e^{-\epsilon t} + \omega_{\min}$ represents the exploration rate, with ω_0 represents initial exploration rate, λ is the decay weight, and $\omega_{\min}$ denotes the minimum exploration weight. The procedure for the proposed DQSM is shown in Algorithm 1.

IV. NUMERICAL EVALUATION

For our simulations, we consider a CPN system comprising 50 devices, generating computing tasks classified into 4 types with typical parameters listed in Table I [19], and 30 CPN nodes for computing, where the hardware information of each CPN node is categorized into three levels: large-scale, medium-scale, small-scale, and are randomly initialized from uniform distributions over the intervals $[1400, 1600]$, $[700, 900]$, and $[400, 600]$, respectively. The actual computing power, which acts as training labels, is generated by synthetic functions. The other parameters used in simulations are listed in Table II. For CPM comparison, we implement the following algorithms

TABLE II
SIMULATION PARAMETERS

Parameters	Value	Parameters	Value
B	20 MHz	R	100 Mbps
p_v	$[0.5, 1]$ W	σ^2	-100 dBm
ρ	0.9	$d_{v,k}$	$[50, 200]$ m
$w_{v,t}$	$[10, 100]$ MB	γ	0.95

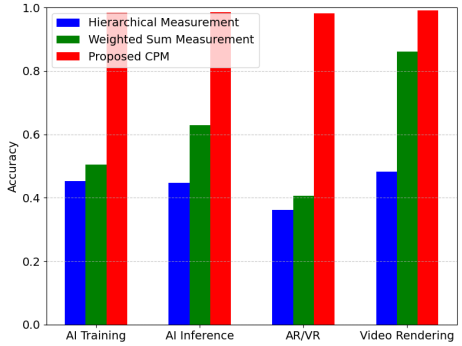

Fig. 3. CPM accuracy vs. the measurement methods.

- **Weighted Sum Measurement:** The weighted sum measurement employs fixed weights to aggregate hardware resources as computing power [9].
- **Hierarchical Measurement**: The hierarchical measurement employs fixed values to measure computing power for each layer.

Fig. 3 shows how measurement methods affect the measurement accuracy based on mean absolute percentage error. The measurement accuracy results are evaluated for four task types: AI Training, AI Inference, AR/VR, and video rendering. From Fig. 3, we can see that the proposed CPM outperforms the hierarchical measurement and weighted sum measurement in all scenarios, demonstrating its robustness and adaptability. Specifically, for AR/VR tasks, the proposed CPM maintains superior accuracy and stability, while the other methods exhibit poor performance. This is due to the fact that the proposed method can capture the relationship between hardware capabilities and task types, thus achieving higher accuracy.

Fig. 4 shows how measurement methods affect the reward under varying numbers of tasks. From Fig. 4, we can see that the proposed CPM measurement improves the reward by 30.4% over the weighted sum measurement and 128.4% over the hierarchical measurement. This is due to the fact that accurate measurement reduces uncertainty in task scheduling, which ensures more stable scheduling decisions.

Fig. 5 shows how measurement methods and exploration policies affect the training of the task scheduling process. From Fig. 5, we can see that the proposed DQN with Boltzmann-based exploration policy can improve cumulative reward by up to 9.3% compared with the ϵ-greedy policy. This is due to the fact that the Boltzmann policy provides targeted exploration by focusing on potentially optimal actions, in contrast to the random exploration in the ϵ-greedy policy, thereby accelerating the convergence rate and maximizing the cumulative reward. From Fig. 5, we can also see that the proposed CPM

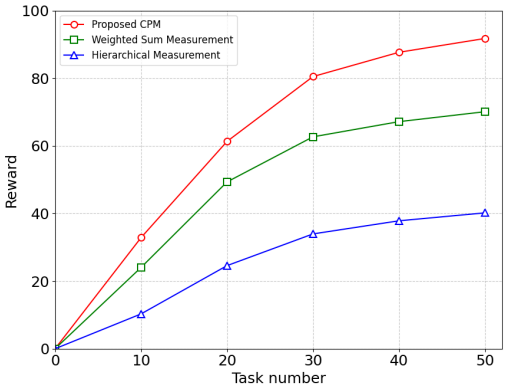

Fig. 4. Task scheduling reward vs. the measurement methods.

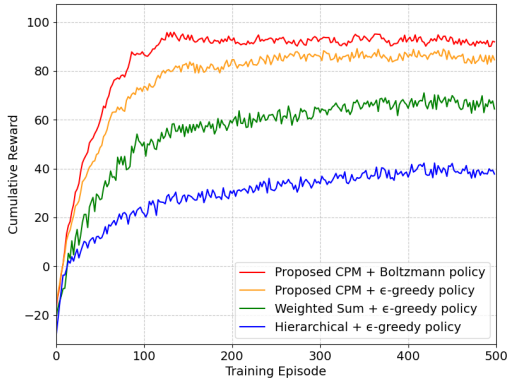

Fig. 5. Cumulative reward vs. the explore strategies.

method with the Boltzmann policy demonstrates a 50.3% and 73.2% improvement compared with the weighted sum and hierarchical-based method, respectively. This is due to the fact that accurate measurement ensures more efficient exploration in the training process, thus accelerating the convergence.

V. Conclusion

In this paper, we proposed a novel framework for CPM and task scheduling in heterogeneous CPN. The optimization problem was formulated to maximize the global matching degree between computing tasks and CPN nodes while satisfying the diverse constraints through joint CPM and task scheduling. To solve this problem, we first facilitate accurate CPM by developing a clustering-based algorithm that can classify computing tasks into different clusters. Then, we designed a neural network based CPM model to accurately measure the computing power of heterogeneous computing resources. Given the proposed CPM, we formulated the task scheduling as a Markov Decision Process whose goal is to maximize the matching degree between computing resources and tasks. Then, we proposed a deep reinforcement learning-based scheduling strategy with an adaptive exploration mechanism to optimize the convergence rate. Simulation results showed that the proposed framework can achieve significant improvements in measurement accuracy, scheduling performance, and convergence rate.

References

[1] M. Vaezi, A. Azari, S. R. Khosravirad, M. Shirvanimoghaddam, M. M. Azari, D. Chasaki, and P. Popovski, "Cellular, wide-area, and non-terrestrial IoT: A survey on 5G advances and the road toward 6G," *IEEE Communications Surveys & Tutorials*, vol. 24, no. 2, pp. 1117–1174, Feb. 2022.

[2] S. Wang, H. Guo, X. Zhu, C. Yin, and V. K. N. Lau, "Communication-efficient distributed bayesian federated learning over arbitrary graphs," *IEEE Transactions on Signal Processing*, vol. 73, pp. 1351–1366, Feb. 2025.

[3] G. Cui, Q. He, B. Li, X. Xia, F. Chen, H. Jin, Y. Xiang, and Y. Yang, "Efficient verification of edge data integrity in edge computing environment," *IEEE Transactions on Services Computing*, vol. 15, no. 6, pp. 3233–3244, June. 2022.

[4] J. Xie, S. Wang, and C. Yin, "Machine learning based task scheduling for wireless powered mobile edge computing IoT networks," in *Proc. of International Conference on Wireless Communications and Signal Processing (WCSP)*, Xi'an, China, Dec. 2019.

[5] X. Tang, C. Cao, Y. Wang, S. Zhang, Y. Liu, M. Li, and T. He, "Computing power network: The architecture of convergence of computing and networking towards 6G requirement," *China Communications*, vol. 18, no. 2, pp. 175–185, Feb. 2021.

[6] Y. Sun, B. Lei, J. Liu, H. Huang, X. Zhang, J. Peng, and W. Wang, "Computing power network: A survey," *China Communications*, vol. 21, no. 9, pp. 109–145, Apr. 2024.

[7] Z. Di, T. Luo, C. Qiu, C. Zhang, Z. Liu, X. Wang, and J. Jing, "In-network pooling: Contribution-aware allocation optimization for computing power network in B5G/6G era," *IEEE Transactions on Network Science and Engineering*, vol. 10, no. 3, pp. 1190–1202, Nov. 2023.

[8] J. Hao, M. Jin, H. Bai, J. Yang, S. Wang, Y. Ren, and F. Qi, "Heterogeneous resource scheduling in computing power networks: A method based on improved D3QN," in *Proc. of Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC)*, Chongqing, China, Jul. 2024.

[9] Y. Li, R. Xie, Q. Tang, and T. Huang, "Resource trading incentive mechanism based on stackelberg game and auction theory in computing power network," in *Proc. of International Conference on Computer and Communications (ICCC)*, Chengdu, China, Apr. 2023.

[10] R. Chai, S. Gao, J. Lan, and N. Liu, "Efficient computing resource metric method in computing-first network," *Journal of Computer Research and Development*, vol. 60, no. 4, pp. 763–771, Nov. 2023.

[11] L. Han, R. Xie, Y. Ren, F. R. Yu, and T. Huang, "An NFT-based distributed auction mechanism for multi-resource trading in computing power network," in *Proc. of International Conference on Computer and Communications (ICCC)*, Chengdu, China, Dec. 2022.

[12] L. Feng, R. Xie, Q. Tang, and T. Huang, "Delay-prioritized task scheduling with load balancing in computing power networks," in *Proc. of IEEE Wireless Communications and Networking Conference (WCNC)*, Dubai, United Arab Emirates, Jul. 2024.

[13] X. Huang, R. R. Liu, B. Lei, W. Xing, and X. Zhang, "Platform profit maximization for space-air-ground integrated computing power network supplied by green energy," in *Proc. of IEEE International Conference on Communications (ICC)*, Denver, USA, Aug. 2024.

[14] Z. Liu, C. Qiu, Y. Zhao, X. Wang, and J. Jiang, "Bat-FG: A broad attention based fine-grained offloading in green computing power networks," in *Proc. of IEEE International Conference on Communications (ICC)*, Rome, Italy, Oct. 2023.

[15] N. Yang, S. Wang, Y. Liu, C. G. Brinton, C. Yin, and M. Chen, "Graph neural networks for the optimization of collaborative federated learning energy efficiency," *IEEE Transactions on Mobile Computing*, pp. 1–12, June. 2025.

[16] A. Aslam, U. Qamar, R. A. Khan, and P. Saqib, "Improving K-Mean method by finding initial centroid points," in *Proc. of International Conference on Advanced Communication Technology (ICACT)*, Phoenix Park, Korea, Feb. 2020.

[17] H. Tong, M. Chen, J. Zhao, Y. Hu, Z. Yang, Y. Liu, and C. Yin, "Continual reinforcement learning for digital twin synchronization optimization," *IEEE Transactions on Mobile Computing*, pp. 1–15, Feb. 2025.

[18] S. Wang, M. Chen, X. Liu, C. Yin, S. Cui, and H. V. Poor, "A machine learning approach for task and resource allocation in mobile-edge computing-based networks," *IEEE Internet of Things Journal*, vol. 8, no. 3, pp. 1358–1372, Jul. 2021.

[19] J. Li, C. Cao, A. Li, and B. Pang, "Computing power modeling for business experience in computing power network," *ZTE Technology Journal*, vol. 26, no. 5, pp. 34–38, Oct. 2020.