Distributed Model Training Task Migration for Hotspot Management in Intelligent Computing Center Interconnection with Tidal Characteristics

DOI: 10.1109/TNSM.2025.3590011

Distributed Model Training Task Migration for Hotspot Management in Intelligent Computing Center Interconnection with Tidal Characteristics

Yingbo Fan, Yajie Li, Carlos Natalino, Yahui Wang, Jiaxing Guo, Wanping Wu, Rongrong Ruan, Wei Wang, Yongli Zhao, and Jie Zhang, *Member, IEEE,*

***Abstract*—Intelligent computing center (ICC) is a new type of data center constructed with intelligent computing power, such as graphic processing units (GPUs) and artificial intelligence acceleration cards. With billions of parameters, the emergence of large models (e.g., ChatGPT) presents a significant demand of computing power. It may be challenging for a single ICC to provide the required computing power during large model training. Thus, ICC interconnections (ICCI) will become a typical and effective solution to provide intensive computing power. Due to human activities, traditional computing tasks (e.g., transaction processing and online entertainment) exhibit a tidal effect of computing demand, which leads to the tidal variation of remaining computing resources. Moreover, distributed model training (DMT) tasks are likely to cover peaks and valleys of the tidal effect in computing power. In this case, it is easy for DMT tasks to cause an ICC to become a hotspot (i.e., computing load in an ICC exceeds a desired threshold), which significantly degrades the reliability and performance of the ICC. This paper proposes DeepHM, a deep reinforcement learning-based hotspot management strategy through task migration in ICCI networks. To comprehensively consider the bandwidth metrics of the ICCI network, we further propose a dynamic wavelength allocation strategy, i.e., DeepHM-DWA. Simulation results show that the DeepHM and DeepHM-DWA reduce the hotspot compute unit time blocks by 19% and 18% with fewer number of migrated workers while balancing the computing load among multiple ICCs. DeepHM and DeepHM-DWA reduce the average completion time ratio of the DMT tasks by 2% and 5%, respectively.**

***Index Terms*—Intelligent computing center interconnections, tidal effect, distributed model training, hotspot management, task migration.**

I. Introduction

An intelligent computing center (ICC) is a specialized data center (DC) for intelligent computing provisioning (e.g., GPU and artificial intelligence acceleration cards) in the application intelligence age [1]. The ICC is based on integrated architecture computing systems, driven by rich data, and utilizes intelligent computing power capabilities to perform complex processing of data. The overall goal is to realize intelligent applications through model training using large-scale datasets.

With billions of parameters, large models present a significant demand on computing power. It is challenging for a single ICC to train very large models. For example, large models such as GPT-4 require a massive infrastructure of 25,000 NVIDIA A100 cards [2]. Even on a single-node 1000P ICC, the training period extends to 249 days. Therefore, ICC interconnections (ICCI) will become a typical and effective solution to provide extensive computing power. The ICCI network connects various ICCs through an optical network, which interconnects the computing resources of distributed ICCs to achieve distributed collaboration of computing power in the network [3, 4].

Due to human activities, traditional computing tasks (e.g., transaction processing and online entertainment) have different computing demands at different times, which leads ICCs to show a periodic variation in utilization [5, 6]. The computing load of an ICC varies significantly between busy and idle times, an effect known as the computing tidal characteristics. For daytime working hours, ICC nodes in residential areas usually receive fewer requests, while those in business and industrial areas have a relatively higher number of requests. It is critical to mitigate these computing tidal characteristics for resource allocation, energy efficiency, and efficient task scheduling in ICCI networks.

Distributed model training (DMT) is a critical way to train large-scale models. The ring-all-reduce (RAR) architecture is an important communication method in DMT, where each training node is called a worker [7, 8]. RAR achieves efficient parameter synchronization and aggregation by communicating between workers in a ring structure. Large DMT tasks take a long time to train, which are likely to cover peaks and valleys of the tidal effect. In this case, it is easy for DMT tasks to cause an ICC to become a hotspot (i.e., when the computing load in an ICC exceeds a pre-defined threshold), which significantly degrades the reliability and performance of the ICC, and even decreases the task completion time [9–12]. Excessive demand for computing resources can lead to processing bottlenecks, increased latency, and slower task execution. In addition, the hotspot ICCs may result in increased energy consumption, reduced reliability, hardware fatigue, system failures, and increased downtime [10, 13]. It is crucial to investigate

This work is supported in part by NSFC (62471063, 62021005), the Beijing Natural Science Foundation (4232011), and BUPT Excellent Ph.D. Students Foundation (CX20241019). (*Corresponding author: Yajie Li.*)

Yajie Li is with the Beijing University of Posts and Telecommunications, Beijing and School of Information Science and Technology, Tibet University, Lhasa (e-mail: yajieli@bupt.edu.cn)

Yingbo Fan, Yahui Wang, Jiaxing Guo, Wanping Wu, Rongrong Ruan, Wei Wang, Yongli Zhao, and Jie Zhang are with the Beijing University of Posts and Telecommunications, Beijing 100876, China (e-mail: yingbo@bupt.edu.cn).

Carlos Natalino is with the Department of Electrical Engineering, Chalmers University of Technology, Gothenburg, Sweden (e-mail: carlos.natalino@chalmers.se).

how to implement hotspot management for RAR-DMT under computing tidal effects.

The contributions of this paper are summarized as follows:

- The paper investigates hotspot management by migrating DMT tasks in ICCI networks with tidal traffic.
- This work proposes DeepHM, a deep reinforcement learning (DRL)-based task migration strategy for hotspot management in ICCI network. The DRL model inputs the current network parameters, node states, and task characteristics into the DRL agent. The method intelligently and adaptively determines whether the task should be migrated and formulates the migration strategy.
- To comprehensively consider the bandwidth metrics of ICCI networks, we further propose a dynamic wavelength allocation strategy, DeepHM-DWA. In this way, the bandwidth allocation is dynamically adjusted during the task migration process, which further optimizes the task migration strategy and improves the performance and efficiency of the network.
- In addition, we adopt node-proximity-aware migration, full node migration, and random node migration as comparison algorithms. Simulation results show that the DeepHM and DeepHM-DWA algorithms reduce the hotspot computing unit (CU)-Time blocks by 19% and 18% with fewer number of migrated workers. Meanwhile, the algorithms decrease the variance of the remaining computing resources of the ICC nodes, which balances the computing load among multiple ICCs. Moreover, DeepHM and DeepHM-DWA reduce the average completion time ratio of DMT tasks by 2% and 5%, respectively.

The rest of the paper is organized as follows. Section 2 presents related work. Section 3 describes the network model, service model, and problem description for hotspot management. In Section 4, we detail the operation principle and design of DeepHM and DeepHM-DWA, including the modeling and training mechanisms. Then, in Section 5, we show the performance evaluation and illustrative numerical results. Finally, Section 6 concludes the paper.

II. Related Work

This section summarizes the key aspects of DMT training in tidal networks: hotspot management, the tidal characteristics of computing resources, and DMT task migration. Previous research has proposed various approaches to address these challenges.

A. Hotspot Management

Previous studies have discussed the challenges related to hotspot management, in terms of task allocation across DC nodes and within a DC. In terms of task allocation across different DCs, various methods have been proposed to mitigate the adverse effects of hotspots. For example, a location priority-based efficient mapping algorithm was proposed in [10] for the virtual network embedding problem, where the algorithm improved the request acceptance rate and the revenue-overhead ratio of the network in the presence of hotspot nodes. The authors in [13] proposed that a hotspot node scenario may be generated in the network when more traffic was destined to or originated from a DC node and resolved by optimizing the location of the DCs. In terms of task allocation within DCs, the work in [14] introduced a temperature-aware virtual DC embedding scheme that minimized the hot air drawn to each rack to prevent hotspots. In addition, some work performed hotspot management by changing DC locations and scheduling workloads. A scheduling algorithm was proposed in [15] to dynamically consolidate virtual machines to minimize overall energy consumption while proactively preventing hotspots. However, these works did not address the hotspot management for long-lasting DMT tasks, focusing on pre-scheduling tasks to manage hotspots rather than using task migration. There is no research on migrating long-lasting DMT tasks in ICCs for hotspot management.

B. Tidal Characteristics

The computing tidal scenario where the demand for computing and/or networking resources showed periodic fluctuations has attracted attention. In [5], the authors proposed that due to the real-time fluctuations in the load of computing nodes, which are nodes within a DC, some computing nodes exhibited significant periodic characteristics, thereby forming tidal computing nodes. Furthermore, a routing algorithm with a weighted wakeup routing penalty was proposed for tidal computing power nodes in the sleep state. Similarly, the authors in [6] showed that the demand for computing resources within a DC varied based on human behavior. The work predicted future computing resource demands of virtual machines and computed the optimal hybrid optical/electrical DC network configuration based on the prediction. The work in [16] used a recurrent neural network model to predict future traffic demands based on past demand profiles for each area, showing that the computing demand in urban DCs showed significant tidal changes over time. The authors of [17, 18] also pointed out that the DC computing demand was time varying. The study in [19] explores DNN training in centralized clusters exhibiting significant tidal patterns, which proves scenarios where training tasks coexist with tidal traffic. However, these studies focus on optimizing a single type of time-varying task and did not analyze the simultaneous impact of tidal effects and long-lasting DMT tasks, which is a realistic scenario in ICCI networks [19–21].

C. DMT Migration

Recently, DMT task migration has become a critical research focus on the field of improving training efficiency. For example, a cluster scheduling framework was presented in [22], which utilized intra-job predictability to time-slice GPUs across multiple jobs to provide low latency. The work assessed the interplay among different co-located DMT tasks. Moreover, the jobs were dynamically migrated to more suitable GPUs to improve the cluster efficiency and better utilization through migration and time-slicing jobs for job-to-resource fit. The authors in [23] proposed a network-aware dynamic model tracking to migrate models with resource efficiency tradeoffs and analyzed the concepts of cold models, preheated models, and model inertia in DMT over wireless networks. In [24], an adaptive machine-learning-based model was proposed to predict key characteristics of live migrations with high accuracy. The model based its prediction on migration algorithms and workloads running within a virtual machine. The authors in [25] proposed a two-dimensional spatio-temporal task migration mechanism to allocate task locations across geographically distributed DCs. The mechanism used migration strategies to maximize the utilization of renewable energy, balancing the emission-cutting effect of task scheduling. However, it only focused on reducing carbon pollution caused by high energy consumption, but failed to address the prevention of hotspot node generation. However, our work addresses the impact on ICCs when DMT tasks cover peaks and valleys of tidal effects through task migration, which has not been analyzed in the DMT migration literature.

In summary, most of the previous research on hotspot management and tidal characteristics primarily focuses on end-to-end task scheduling or optimization. However, DMT tasks require coordination across multiple nodes, making end-to-end task optimization unsuitable for managing hotspots in DMT tasks. Furthermore, when tidal characteristics are introduced in research on DMT task migration, resource utilization tends to become unbalanced, which may result in inefficient task migration.

III. Problem Description

A. Network Model

Our study focuses on the ICCI network using RAR architecture to train large-scale DMT tasks. Each ICC is internally constructed with a fat-tree topology. ICCs are interconnected by an optical transport network (OTN). The ICCs have different geographic locations, where some are located in residential areas and some in business areas, as shown in Fig. 1. In addition, the computing resource usage within each ICC periodically changes over time due to fluctuations in human activities, which affects the workload of the ICCs. For example, the ICCs located in residential areas receive fewer requests during daytime working hours, which results in relatively low computing demand. On the contrary, ICCs located in business areas have more traffic and computing tasks, which leads to variations in the level of remaining computing resources in different ICCs over time.

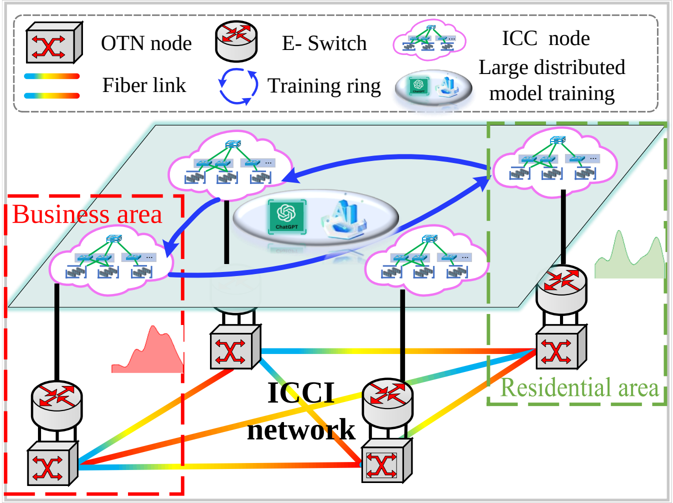


Fig. 1. DMT in ICCI network with tidal characteristics.

We model the ICCI network as a directed graph *G(V, E, N, type)*, where V denotes the set of ICC nodes, while E and N represent the set of fiber links and the set of GPUs in each ICC node, respectively. Furthermore, we consider a GPU as the smallest computing unit within an ICC node, meaning that one CU corresponds to one GPU. We focus on the hotspot problem caused by the utilization of computing resources exceeding a pre-defined threshold. Therefore, we pay attention to the usage of computing resources in ICC nodes. It is assumed that when the computing resources in an ICC are sufficient to train DMT tasks, the GPU memory within the ICC can also accommodate the task. The *type* parameter denotes the type of ICC node, which is categorized according to the location of the ICC: ICC in residential areas, business areas, and other areas. We represent each DMT task as $M_i(D_i, P_i, \theta_i, T_i, S_i, It_i)$, where D_i and P_i denote the size of the original training data and the number of parameters in the DMT task, respectively. θ_i and T_i represent the model accuracy requirement and deadline. S_i is the node originating the task. It_i indicates the number of iterations of the model. DMT tasks that cannot be completed within the deadline are considered blocked. Eq. 1 ensures that the DMT task is completed before its deadline. T_{de} is the time when the task is deployed. We use Eq. 2 to determine the number of CUs required to complete a DMT task in a single iteration [26]. λ is the computing required per unit of training data. ρ is the computing power per CU. N_i^{worker} is the number of workers (computing nodes) involved in the task. T_i^{com} represents the time required for a single CU to compute a subset of training data in one iteration. Eq. 3 represents the number of bandwidth units (BUs) used in a single iteration. χ is the capacity per wavelength. T_i^{trans} represents the time required for a single BU to transmit the gradient in one iteration. All the parameters used in the paper are listed in Table I.

$$(T_i^{com} + T_i^{trans}) \times It_i = T_i - T_{de} \tag{1}$$

$$N_i^{cu} = \frac{D_i \times \lambda}{\rho \times N_i^{worker} \times T_i^{com}} \tag{2}$$

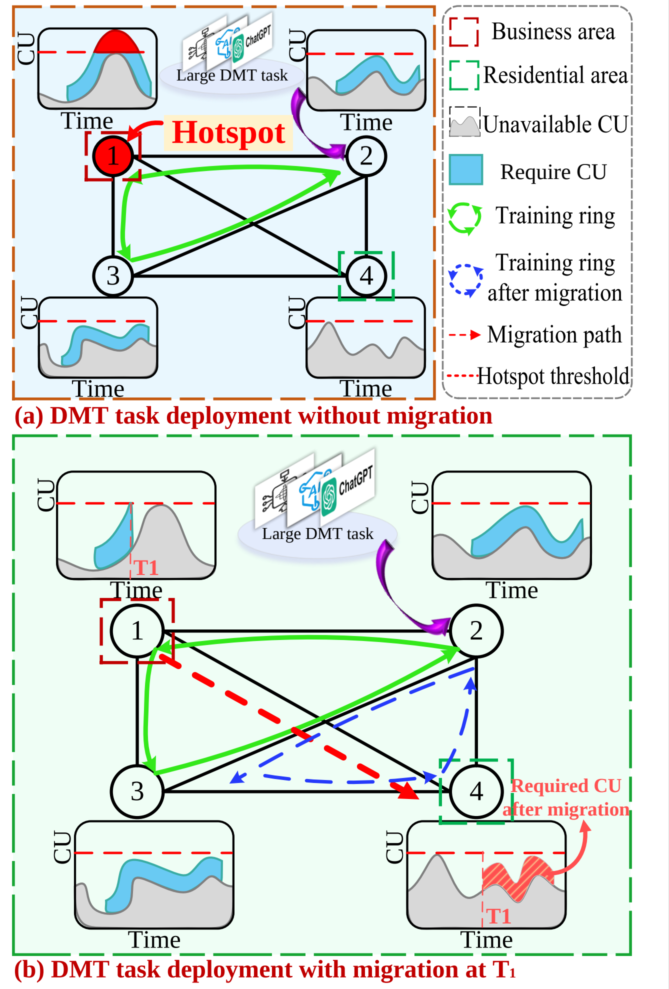
$$N_i^{bu} = \frac{2(N_i^{worker} - 1) \times P_i}{N_i^{worker} \times \chi \times T_i^{trans}} \tag{3}$$

B. Problem Description

Fig. 2 shows an example of migration of large DMT tasks in the ICCI network with tidal characteristics. In this example, the objective is to manage hotspots. In the network topology, node 1 is located in a business area with high activity during working hours, while node 4 is located in a residential area with high activity during early morning and evening hours. Assume that task M_i is generated at node 2. At the time of M_i arrival, node 1 and node 3 have more idle CU, therefore the DMT tasks are deployed on the ring formed by nodes

TABLE I
PARAMETERS

Category	Parameters	Description
ICCI network	V	ICC nodes
	E	set of fiber links
	N	set of GPUs
	$type$	type of ICC node
	λ	computing required per unit
	χ	wavelength capacity
	ε	hotspot threshold
DMT task	D_i	training data size
	P_i	update gradient size
	θ_i	model accuracy
	T_i	deadline
	It_i	number of iterations
	N_i^{cu}	number of CUs
	N_i^{bu}	number of BUs
	N_i^{worker}	number of workers
	T_i^{com}	time for a CU to compute data
	T_i^{trans}	time for a BU to transmit data
reward parameters	α	hotspot coefficients
	β	CU coefficients
	γ	hops coefficients
	η	migrated worker coefficients
	φ	BU coefficients
DRL parameters	ω	BU coefficients
	π	discount factor
	Ψ	probability distribution



(a) DMT task deployment without migration

(b) DMT task deployment with migration at T₁

Fig. 2. Task migration for hotspot management in the ICCI network.

1, 2, and 3. However, node 1 will experience a peak period over time. Node 1 will become a hotspot if it continues to train this task, which leads to problems such as performance, reliability degradation, and more energy consumption of node 1. However, there are still a lot of idle computing resources in node 4 at this time and the network load is not balanced.

To solve this imbalance and avoid node 1 becoming a hotspot, we trigger the DMT task migration process. We migrate DMT tasks that have been utilizing ICC computing resources for a long period at time T1. The training data and parameters of the worker at node 1 are migrated to node 4 by high-capacity optical fiber for the next training iteration, which can avoid the generation of hotspots, thus balancing the load in the network.

Next, we formulate the hotspot management optimization problem by specifying its objective function and constraints. The objective is to minimize the number of hotspot nodes in the network while minimizing the number of migrated workers as defined in Eq. 4. τ and $\tau*$ denote the node computing resource state before and after the migration, respectively. $hot(\cdot)$ returns the number of hotspot nodes. M_c denotes the number of workers migrated. The migration process is triggered when the utilization of a node violates a pre-defined threshold ε as Eq. 5. $act(v)$ returns the maximum computing resources used of node v. N_v is the computing resource capacity of node v. First, we need to identify the candidate rings and candidate links for the DMT tasks. A candidate ring consists of a set of interconnected nodes, excluding the current hotspot nodes, with each pair of nodes connected by the shortest path. A candidate link refers to the communication links between nodes within the candidate ring, excluding the shortest paths. During the migration, we must consider node computing resource constraints (Eq. 6), bandwidth resource constraints (Eq. 7), and the specific requirements of the DMT tasks. Specifically, these constraints include the ability of nodes to accommodate workers after migration and the capability of the candidate ring to transfer the communication traffic required for gradient updates between workers. Furthermore, the migration must reduce the number of hotspot nodes (Eq. 8) while meeting the demands of the DMT tasks, such as the need for multiple workers to execute the task, the requirement for a ring structure among workers (Eq. 9), and the availability of sufficient computing resources to ensure the DMT task is completed before the deadline.

$$\text{maximize}:[hot(\tau)-hot(\tau*)]-\eta\times M_c. \tag{4}$$

$$\frac{act(v)}{N_v} \geq \varepsilon. \tag{5}$$

$$To^{v*} \geq \mathrm{Re}_i. \tag{6}$$

$$C^{l,m} \geq O_i. \tag{7}$$

$$hot(\tau) - hot(\tau*) > 0. \tag{8}$$

$$\exists\Theta : v_i^* \rightarrow \{v_1^*...v_{n_i}^*\}, (v_k^*, v_{k+1}^*) \in E, \forall k \bmod n_i. \tag{9}$$

Re_i and To^{v*} represent the number of CUs required for task i and the number of CUs available on the migration destination node $v*$, respectively. O_i is the number of bandwidth resources used by the task i before migration. $C^{l,m}$ is the maximum number of continuity bandwidth resources after migration time m for link l. Let $v_i^* \rightarrow \{v_1^*...v_{n_i}^*\}$ denote the target node set for deployment after migration. This ensures that the selected worker set v_i^* forms a logical ring in the underlying network graph $G(V,E)$, which is essential for DMT training. Θ denotes a permutation of the given nodes. An edge $(v_k^*, v_{k+1}^*) \in E$ indicates that there exists a link between the two nodes. The notation $k \bmod n_i$ denotes a cyclic connection, meaning that the last node $v_{n_i}^*$ is also connected to the first node v_1^*.

Therefore, when tidal characteristics are present, it is necessary to migrate DMT tasks in the ICCI network. The candidate tasks which occupy a large amount of computing resources in ICCs for a long period are candidates. The network can be balanced to prevent the generation of hotspots and ensure the efficiency and reliability of the entire distributed training infrastructure by migrating DMT tasks at an appropriate time.

IV. DeepHM and DeepHM-DWA Approach

To effectively manage hotspots in the ICCI network through task migration, we introduce a DRL-based approach that automates the migration process. The method enables intelligent decision-making for managing hotspots and migrating tasks within the ICCI network. The DRL model collects state s_t including current network parameters, node states, and task characteristics, using an immediate reward function r_t to guide the agent in maximizing hotspot reduction. By intelligently and adaptively determining whether tasks should be migrated, as well as selecting the appropriate migration nodes and time, the method prevents nodes from becoming hotspots. In addition, the DeepHM-DWA can adjust the number of wavelengths allocated to a task based on the remaining wavelength resources in the network after task migration. In the following, we detail the design of the DRL model in the DeepHM and DeepHM-DWA methods.

A. Overall Structure

The principles and details of the DeepHM and DeepHM-DWA algorithms are illustrated in Fig. 3(a). Given an ICCI with tidal characteristics or varying computing resources, the algorithms aim to determine whether DMT tasks should be migrated and select migration workers and times to maximize long-term cumulative rewards. Firstly, the Software-Defined Network (SDN) controller obtains information about each node, link, and DMT task within the ICCI network (**step 1**). In **step 2**, the controller invokes the feature engineering module to organize specific state data s_t according to the current DMT task parameters and remaining network resources. Then, the state data s_t is input into the DNNs, which consist of a policy network and a value network in **step 3**. The DNNs process the input state parameters and give a probability distribution over all the policies using the policy network and the current value of s_t using the value network. The agent iterates through all provided candidate sets (i.e., migration decisions and strategies), take actions to migrate DMT tasks, and evaluates the current migration strategy through a reward system after migration (**steps 4, 5**). Once a round is complete, the training signals (state s_t, action a_t, reward r_t) for this round are stored in an experience buffer (**step 6**). In **step 7**, after reaching a certain sample capacity, the DNN is updated to maximize long-term cumulative rewards [27], which is represented as Eq. 10. It is critical to introduce a discount factor, denoted as γ, which modulates the impact of rewards occurring subsequent time steps. The role of the discount factor is to regulate the influence of future rewards in the calculation of the cumulative reward promoting a balance between consideration of immediate and long-term rewards. The discount factor ω is constrained to the interval [0, 1]. Each iteration of a DMT task consists of computing time and gradient update transmission time, as shown in Fig. 3(b). Migration of the DMT task will only occur after the completion of the current iteration when the computing resource utilization of the node reaches the hotspot threshold.

$$\Phi_t = \sum_{t}^{\infty} \omega^{t'-t} \times r_t \tag{10}$$

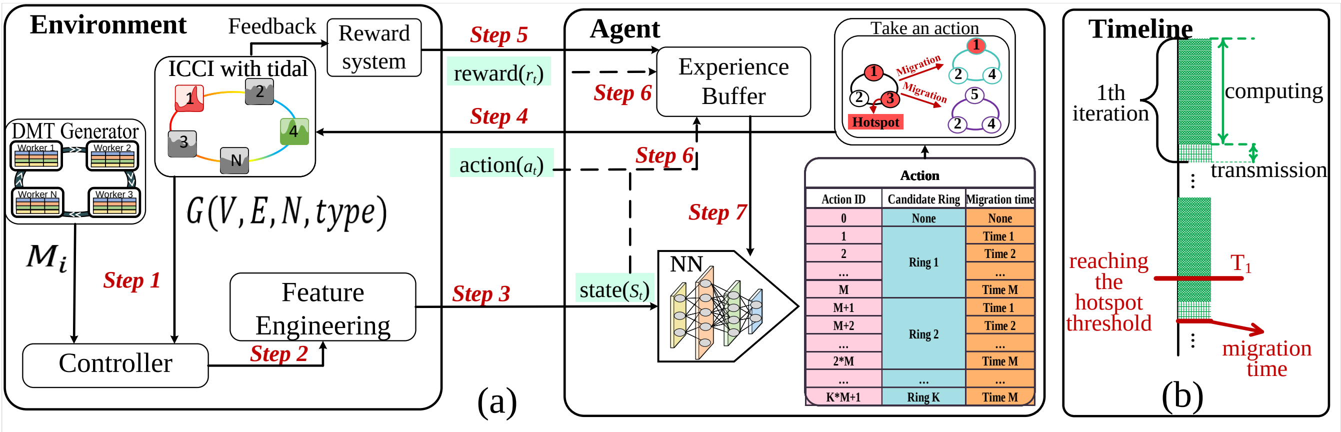
B. DRL Model Design of DeepHM

The DRL model of DeepHM consists of five modules: environment, state, action, agent, and reward.

- Environment: The environment includes an abstracted ICCI network, a DMT task generator, a controller, a feature engineering module, and a reward system. The controller integrates network and task resources, providing node and network resource utilization, node types, hotspots nodes, deployed DMT tasks, and topology abstraction to the feature engineering module. The feature engineering module generates the customized state s_t. The reward system provides immediate feedback on the action a_t to the agent.
- State: The state module includes the current state of the ICCI network with tidal characteristics, DMT task parameters, network resources, and other characteristics. Given K candidate rings and M time slots, the state contains $2+4\times M\times K$ elements. The information is organized into specific state data in Eq. 11.

$$s_t = \{G, Ns, Sp_k^m, Mt_k^m, Re_k^m, To_k^m, |_{m\in M,k\in K}\} \tag{11}$$

where G and Ns denote task generating node and task ring nodes (excluding task generating node), respectively. For each task, we have selected K candidate rings and M time slots to migrate. $Sp_k{}^m$ and $Mt_k{}^m$ are the migration start time and migration total time for the m^{th} migration slots in the k^{th}



Action

Action ID	Candidate Ring	Migration time
0	None	None
1	Ring 1	Time 1
2		Time 2
...		...
M		Time M
M+1	Ring 2	Time 1
M+2		Time 2
...		...
2*M		Time M
...	...	...
K*M+1	Ring K	Time M

Fig. 3. (a) The overall structure of DeepHM and DeepHM-DWA. (b)migration timeline of DeepHM and DeepHM-DWA.

candidate ring. Similarly, $Re_k{}^m$ and $To_k{}^m$ are the number of required and the total CUs, respectively.

- Action: The action module consists of two parts, the first part is to determine whether to migrate this task or not. The first action corresponds to the decision of not performing the migration for this task. On the contrary, M migration slots among K candidate ring nodes are selected for this task, so the action space is $K \times M+1$.
- Agent: The agent is responsible for scheduling the decision-making process, which synchronizes DNNs and parameters and interacts with the environment in the SDN controller. The agent observes the candidate set, performs actions, and evaluates migration policies to guide the dynamic decision process.
- Reward: We define the immediate reward of DeepHM as:

$$\begin{aligned} rt =& \alpha \times [\mathrm{hot}(\tau) - \mathrm{hot}\,(\tau^*)] \\ &+ \beta \times [\mathrm{act}(\tau) - \mathrm{act}\,(\tau^*)] - \eta \times M_c \end{aligned} \tag{12}$$

where M_C denotes the number of workers migrated, which is determined by comparing the candidate nodes selected by the action with the original task training rings in the state. α , β, and η are normalized positive coefficients. τ and τ^* denote the node computing resource state before and after the migration, respectively. hot$(\cdot)$ returns the number of hotspot nodes, act$(\cdot)$ returns the maximum computing resources used in the network, which is represented as the number of activated CUs in all nodes. The more CUs that are activated, the higher the network cost. Therefore, the reduction in CUs is a positive factor because it indicates that the task has been more efficiently allocated to resources, reducing the load on overburdened nodes and improving overall network performance. The immediate reward function combines the hotspots reduced by migrations, the reduced use of computing resources, and the number of workers migrated of each task. Therefore, the reward function reduces hotspot nodes and computing resource usage with the least number of migrations.

C. DRL Model Design of DeepHM-DWA

The environment and agent of the DRL model for DeepHM-DWA are the same as DeepHM, therefore state, action and reward are presented in this section.

The state space of DeepHM focuses on the parameters of the network nodes, while disregarding network resources. The bandwidth is a very important metric in ICCI networks. Therefore, the bandwidth resources of the network are considered in the state of the DRL of DeepHM-DWA. Given K candidate rings, L candidate links, and M time slots, the state contains $3+4\times M\times K+L\times K+L\times M$ elements. The specific state information is as follows:

$$s_t = \left\{ \begin{array}{l} G, Ns, B_k^l, Sp_k{}^m, Mt_k{}^m, \\ \mathrm{Re}_k{}^m, \mathrm{To}_k{}^m, O_i, C^{l,m}\big|_{m\in M, k\in K, l\in L} \end{array} \right\} \tag{13}$$

where B_l^l denotes the l^{th} link of the k^{th} ring. O_i is the number of bandwidth resources used by the task before migration. $C^{l,m}$ is the maximum number of continuity bandwidth resources after time m for link l.

Compared to DeepHM, the action space first adds the selected links within the candidate rings. Secondly, the action module selects the number of wavelengths used after migration. Note that the more wavelengths used after migration, the shorter the communication time. Therefore, the action space is $K \times M \times L + 2$.

We additionally consider wavelength and link in the immediate reward of DeepHM-DWA:

$$rt = \alpha\times\Delta hot+\beta\times\Delta act+\gamma\times\Delta len-\varphi\times\Delta bu-\eta\times M_c \tag{14}$$

$$\begin{aligned} \Delta hot &= \mathrm{hot}(\tau) - \mathrm{hot}\,(\tau^*) \\ \Delta act &= \mathrm{act}(\tau) - \mathrm{act}\,(\tau^*) \\ \Delta bu &= bu(\xi) - bu\,(\xi^*) \\ \Delta len &= \mathrm{len}(\xi) - \mathrm{len}\,(\xi^*) \end{aligned} \tag{15}$$

where (ξ) and (ξ^*) denote the link wavelength resource state before and after migration, respectively. $len(\cdot)$ returns the number of hops of the ring link, and $bu(\cdot)$ returns the number of wavelength resources. Allocating more wavelengths after migration reduces the transmission time for gradient updates between workers, as the increased bandwidth accelerates data transfer. Δ denotes before migration minus after migration. Thus, the reward function integrates the hotspot status, computing resources, bandwidth resources and the number of migrations.

D. Training Mechanism

Algorithm 1 describes the training mechanism of an agent that implements the actor-critic process of the DeepHM and DeepHM-DWA algorithms. The actor-critic process adopts two DNNs representing the policy and the value networks. The algorithm first obtains the state of the ICCI network and the tidal characteristics from the SDN controller (line 1). The tidal characteristics include the areas where the network nodes are located and the tidal variations of node and link resources. Secondly, the algorithm sets a maximum value for the number of training iterations and initializes the current number of episodes (line 2). For each DMT task, the state s_t of the DRL model is calculated from the task characteristics and the network state of the ICCI network (line 3). Lines 6-10 calculate the probability distribution π of the action based on s_t. Different actions will be generated depending on the specific values of π and δ, with δ initially set to 1. If the minimum of the probability distribution is greater than δ, sampling is performed based on the probability; otherwise, a random strategy is adopted. This approach balances the exploration of new actions and the exploitation of the current optimal strategy. In line 11, the DMT task attempts the action, which is migrated worker and time or not migrated. Then the algorithm calculates the immediate reward for this action (line 12). The current state, action, and immediate reward are stored in the experience buffer, which constitutes a sample (line 13). When the experience buffer capacity reaches Ψ the long-term cumulative reward is calculated for each sample in the buffer based on Eq. 1, and the gradient is generated to update the DNNs (lines 14-19). Finally, the experience buffer is emptied, and the probability function is updated in preparation for the next iteration (lines 20-22). DeepHM and DeepHM-DWA algorithms have a similar procedure but have different state, action and reward function acquired in different networks.

E. Motivation for using A3C

In this paper, DeepHM and DeepHM-DWA use the Asynchronous Advantage Actor-Critic (A3C) algorithm [28]. The reason is that A3C can effectively tackle the temporal and dynamic issues of ICCI. The computing resource in ICCI fluctuates due to factors such as user demand characteristics and traffic load. Therefore, it is necessary to adopt a framework that can be adapted to changing conditions. Due to the A3C algorithm allowing agents to continuously interact with the environment and update policies in real-time, it effectively captures the tidal characteristics of dynamic network changes. The network of actors can efficiently adapt to complex network states and select suitable actions, such as deciding when and where to migrate DMT tasks. At the same time, the network of critics can evaluate the results, which provides valuable feedback for improving the decision-making strategy. This approach is particularly suitable in ICCI where the relationships between DMT tasks, computing resources, and network conditions are complex and interrelated. In conclusion, the adaptability of A3C to temporal dynamics and its ability to capture and recognize differences make it the best choice for migrating DMT tasks in the ICCI network.

Algorithm 1 DeepHM and DeepHM-DWA procedure

```
1:  Initialize $G(V, E, N,$ type $)$ with tidal characteristics
2:  $Iter = 0$
3:  while Iter $< Iter_{\max}$ do
4:      for each $M_i\,(D_i, P_i, \theta_i, T_i, S_i, It_i)$ do
5:          calculate $S_t$ with $M_i$ and $G(V, E, N,$ type $)$
6:          calculate of $\pi$ based on $a_t$
7:          if $\underset{a}{argmin}\{\pi(a)\} > \delta$
8:              obtain action $a_t$ by sampling $\pi$
9:          else
10:             $a_t = \mathrm{argmin}\{\pi(a) \geq \mathrm{rand}()\}$
11:         DMT task migration according to $a_t$
12:         calculate the immediate reward $r_t$ for action $a_t$
13:         store $\{s_t, a_t, r_t, s_{t+1}\}$ into experience buffer
14:         if |experience buffer| $= \Psi$
15:             for each $\{s_{t'}, a_{t'}, r_{t'}, s_{t+1'}\}$ in buffer do
16:                 calculate $\Phi_t$, with Eq. (1)
17:             end
18:         calculate the policy and value gradients
19:         apply the gradients to update the DNNs
20:         empty experience buffer and update $\delta$
21:     end
22: $Iter = Iter$ +1
23: end
```

V. Simulation Evaluation

A. Simulation Setup

We simulated the ICCI network scenario described in this paper on the San Francisco metropolitan network topology with 22 nodes and 38 links, as illustrated in Fig. 4(a). We considered the nodes within the red dashed line as business area, which were assumed to be busy during daytime working hours, while the nodes in the green dashed line were considered within a residential area. We assumed bidirectional fiber links with 40 wavelengths and 100 Gbps of bit rate per wavelength. The tidal traffic in the network is randomly generated, with arrival times following a Poisson distribution (with different arrival rates for different areas), and the duration following an

TABLE II
Simulation parameters

Parameters	Value	Parameters	Value
Computing resource per GPU	20 TFLOPS	N_i^{worker}	[3,10]
Number of GPUs per node	100	λ	10^{12}
χ	100 Gbps	α	6
Wavelengths per link	40	β	2
Time	24 h	η	1
D_i	[700, 2000]	γ	2
θ_i	[0.1, 0.4]	φ	2
P_i	[70, 200] B	ω	0.95
T_i	[10, 24]	Ψ	50

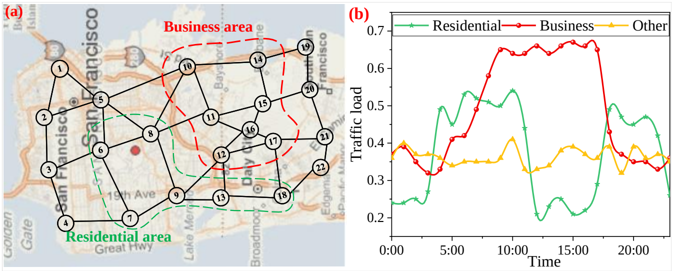


Fig. 4. (a) Mesh topology for metropolitan agglomeration connections in San Francisco, CA, USA, (b) tidal traffic variations over 24 hours in a day.

exponential distribution (10 min^{-1}). The tidal characteristics arise from the varying arrival rates across different areas and times. For example, during the peak hours from 9:00 to 16:00, the average arrival rate in the business area is 150 min^{-1}, while in the residential area, it is only 50 min^{-1}. Fig. 4(b) showed the traffic variations over 24 hours in a day, where the traffic load is normalized relative to the maximum load to ensure generality [29, 30]. Fig. 4(b) exhibits varying traffic loads due to different arrival rates in different areas at different times. Each node had one hundred A100 GPUs. We refer to the hotspot threshold defined in [11, 12], which indicates that when the workload of a DC reaches 80%, the job completion time increases significantly. Similarly, we considered a node as a hotspot when the ICC usage exceeds 80%. The data size of each DMT task ranged from 700GB (like GPT-3) to 2000GB (like PaLM), and the updated gradient size ranged from 70 to 200 billion [31]. The accuracy was randomly generated in [0.1, 0.4] and the number of iterations was represented by the $I(\theta_j) = v \times log(1/\theta_j)$, where v depended on the data size and the condition number [32, 33]. This number of iterations was an upper bound, exceeding it may not improve the performance of the model. In the simulation, both the pre-training and testing data are randomly generated, meaning that the tidal characteristics and DMT task parameters for pre-training and testing are different. As a result, the system does not have prior knowledge of the tidal and task characteristics during evaluation. The specific parameter values are provided in Table II.

We configured the DRL framework for asynchronous training using A3C [28] with 16 concurrent randomly initialized environments and agents. Each agent consisted of two DNNs with 5 hidden layers with 128 neurons per layer. The activation function used for the DNN was exponential linear unit (ELU). The Ψ and learning rate were set to 50 and 10^{-5}, respectively. In addition, we set the *K*, *M*, and *L* to 5, 2, and 3, respectively.

B. Comparison Heuristic Algorithms

Our proposed DRL algorithms are compared with three heuristics. The algorithms included node-proximity-aware migration [34], full node migration [35], and random node migration [36]. When a hotspot occurred in the training ring of a DMT task, the node-proximity-aware migration algorithm selectively migrated a single worker to the nearest node with the idlest CUs. In contrast, the full node migration strategy migrated all workers (excluding task generating node) to the top nodes with the idlest CUs. Finally, the random node migration approach migrated tasks with hotspot nodes randomly to other nodes to minimize the hotspots in the ICCI network.

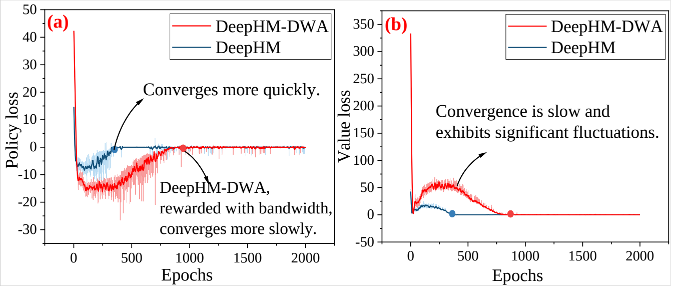


Fig. 5. Training performance in terms of (a) policy loss and (b) value loss of the DRL models.

C. Migration Performance

We considered various performance metrics including node remaining computing resources, the number of hotspot CU-Time blocks in different areas, bandwidth utilization, migration workers, downtime, and average task completion time. The CU-Time block is a metric that summarizes, for a given period, how many CUs were used. It is obtained by multiplying, for each unit of time, the number of CUs used during that unit of time. For instance, assuming hours as the time unit, and a situation where in the first hour 2 CUs were used, while in the second hour 5 CUs were used, the total CU-Time is 10 CU-Hours. A hotspot CU-Time block refers to a single ICC node at a specific time where the remaining CUs (number of GPUs) are less than 20% in the two-dimensional space. In this study, the migration cost is defined as the number of workers migrated per DMT task. The downtime is the transmission time for transmitting data and parameters to the target GPU.

1) **DRL convergence**

Fig. 5 showed the policy and value losses as the training of the agents progresses. The convergence of policy loss in Fig. 5(a) indicated that the policy of the agents had different behavior. DeepHM converged more quickly, at around 340 epochs. DeepHM-DWA, on the other hand, required 920 epochs to converge. This indicated that including the bit rate in the problem makes it harder to learn, but the agent can still learn the policy. The convergence of the value loss in Fig. 5(b) indicated that the agent learned to estimate the value of states with convergence similar to the policy network.

2) **Hotspots and CU performance**

We first compared the initial state (i.e., the system without any migration), DeepHM, DeepHM-DWA, and the three comparison algorithms after migration of the remaining CUs, which is the remaining number of idle GPUs in each node. Fig. 6 illustrated a heatmap of the remaining CUs per node, where we consider a hotspot when less than 20% of the remaining CUs are idle. In the initial state, hotspot nodes were very common reaching 25%, which were mainly located in residential and business areas. The number of hotspots was

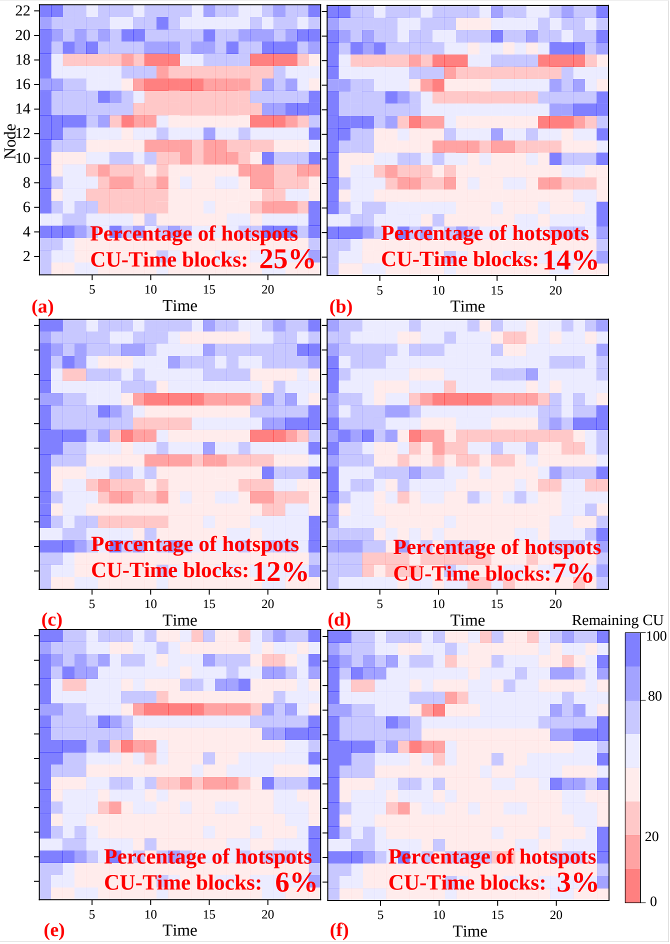


Fig. 6. Heatmap of remaining CUs at each node (a) initial; (b) proximity-aware; (c) random node; (d) DeepHM-DWA; and (e) DeepHM; (f) full node.

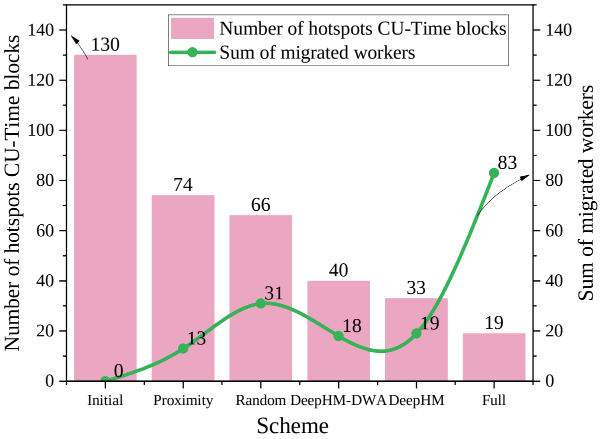


Fig. 7. Number of hotspot CU-Time blocks and total number of migrated workers.

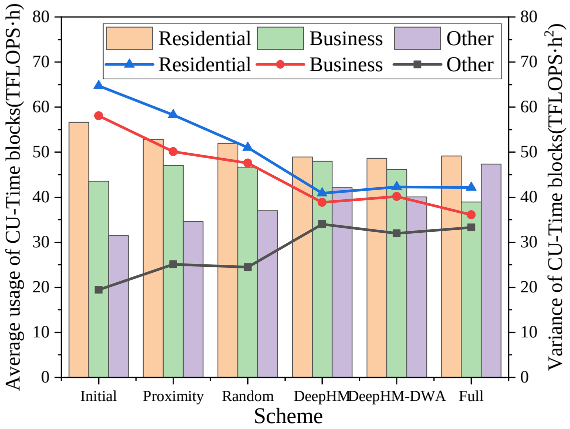


Fig. 8. Average usage and variance of CU-Time blocks in each area.

reduced after migration. Full node migration reduced the CU-Time blocks by 21%. However, this performance comes at the expense of migrated workers, as will be shown next. The proposed DeepHM and DeepHM-DWA algorithms reduced the hotspots by 19% and 18%, respectively. Node proximity-aware migration and random migration reduced similar CU-Time blocks, with 6% and 3%, respectively. The quantitative analysis of these observations was provided in Fig. 7.

The number of hotspots CU-Time blocks decreased substantially in the order of initial state, node-proximity-aware migration, random node migration, DeepHM-DWA, DeepHM, and full node migration. The full node migration reduced CU-Time blocks by 85% compared to the initial state. However, this performance in terms of CU-Time blocks comes at the cost of a high number of migrated workers. Notably, the DeepHM method demonstrated effectiveness, reducing hotspot CU-Time blocks by 74%, with the migration of only 19 workers. The DeepHM-DWA reduced hotspots CU-Time blocks by 69%, migrating only 18 workers. In both cases, the number of migrated workers was relatively low. Node-proximity-aware migration and random node migration exhibited similar reductions in hotspot CU-Time blocks, but the latter migrated 18 additional workers.

Next, we evaluated the effectiveness of different migration schemes in mitigating the tidal characteristics. The average usage and variance of CU-Time blocks in the residential, business, and other areas were shown in Fig. 8. In the initial state, the difference in the average CU usage between the residential, business, and other areas was more than 10 $TFLOPS \cdot h^2$, while the variance difference was about 90 $TFLOPS \cdot h^2$. After DeepHM migration, the average CU usage in the three areas exhibited a reduced disparity of 6 $TFLOPS \cdot h$, with a diminished variance difference of 78 $TFLOPS \cdot h^2$. Moreover, the difference in average CU usage between residential and business areas was only 1 $TFLOPS \cdot h$. In terms of CU equalization, the three areas of DeepHM-DWA have an average difference of 3 $TFLOPS \cdot h$ in CU usage and a variance difference of 22 $TFLOPS \cdot h^2$, which was a 68 $TFLOPS \cdot h^2$ reduction from the initial state. It is noteworthy that the average CU usage disparity of the other three migration algorithms exceeded DeepHM and DeepHM-DWA. The DeepHM exhibited the most superior performance in terms of different areas of equalization.

Furthermore, we compared the distribution of hotspot CU-Time blocks in different areas, as shown in Fig. 9. In the initial state, most of the hotspots were localized in residential and business areas, 46% of the hotspots were in residential areas, 40% of the hotspots were in business areas, while only 14% of the hotspots were in other areas. The percentage of hotspots

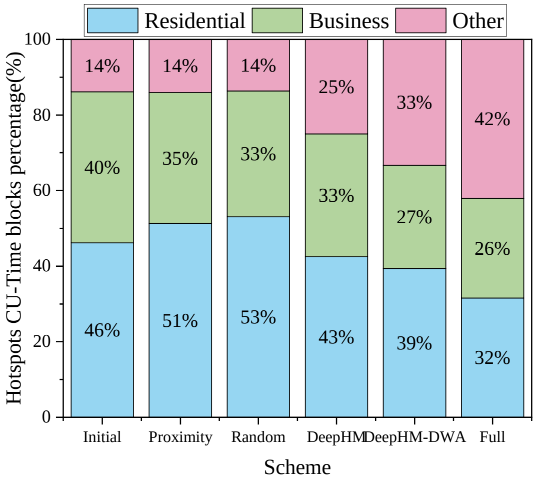


Fig. 9. Percentage of hotspot CU-Time blocks in each area.

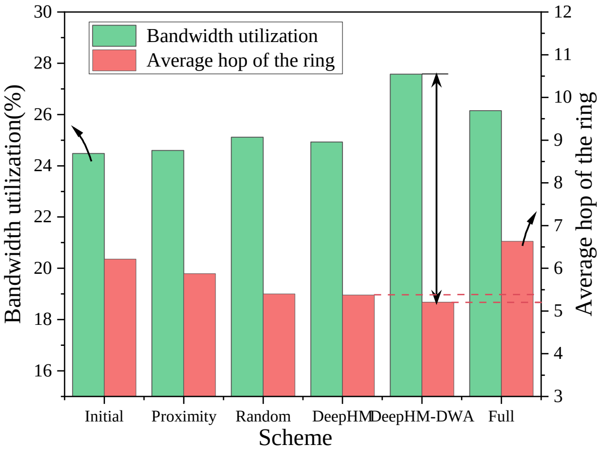


Fig. 10. Bandwidth utilization and average number of hops of the rings.

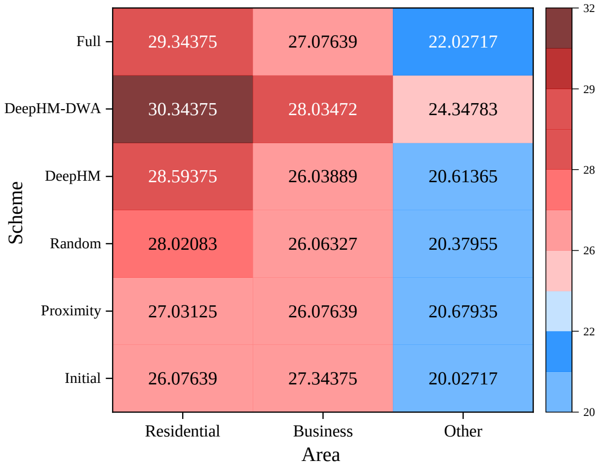


Fig. 11. Average bandwidth utilization in each area.

in other areas remained unchanged after node proximity-aware migration and random node migration, but the percentage of hotspots in residential areas increased to 51% and 53%, respectively. DeepHM equalized the distribution of hotspot CU-Time blocks in three areas with the percentage of hotspots in the three areas distributed as 43%, 33%, and 25%. The full node migration achieved the least number of hotspots, but the hotspots were centered in the other areas with 42%. DeepHM-DWA algorithm achieved the most balanced distribution of hotspots in six cases, where 39% of the hotspots were in residential areas, 33% in other areas, and 27% in business areas.

3) Network bandwidth performance

In terms of network bandwidth performance, we first compared the network bandwidth utilization and the average hop number of the training ring in the initial state and after migration, as shown in Fig. 10. In the initial state, the average hop number of the DMT training ring was more than 6, and the bandwidth utilization was 24.5%. Since proximity node migration migrated nodes to closer nodes, most of the rings were the same used before and after the migration, so the average hop number and bandwidth utilization of the rings changed very little. The random migration strategy introduced randomness in the migration nodes, resulting in a change in the training ring.

The average hop number decreased to 5.5. However, the bandwidth utilization increased to 25.1%, implying an increase in the number of wavelengths utilized. The bandwidth utilization of full node migration also increased significantly compared to the initial state, with a corresponding increase in the average number of hops of the rings used. In the DeepHM scheme, the bandwidth utilization was 25%, with an average hop number of 5.3 for the utilized rings. In contrast, DeepHM-DWA, which considered bandwidth factors, achieved the lowest average hop number for the utilized training rings at 5.1. However, the maximum bandwidth utilization reached 27.6%, indicating an increase in the number of wavelengths utilized for training after migration. This increase in the number of wavelengths used is attributed to the reward function, where a higher number of wavelengths used leads to greater rewards, thereby reducing communication time after migrating.

In Fig. 11, we compared the average bandwidth utilization of the three areas. Generally, the average bandwidth utilization of DeepHM-DWA was higher than the other cases in each area. Next was the full node migration scheme with higher bandwidth utilization than the other four cases, while DeepHM, proximity-aware migration, random node migration, and the initial state exhibited similar performance. Moreover, the bandwidth utilization in DeepHM-DWA was highest in residential areas, followed by business areas, and lowest in other areas, which was the same as the average usage trend of CU. In other areas, the disparity between DeepHM-DWA and the other scenarios was most pronounced. The DeepHM-DWA scheme surpassed full-node migration by 2%, and outperformed other schemes by 4%. The bandwidth utilization of other cases in the other areas exhibited similar performance, averaging around 20%. The residential area under the DeepHM-DWA algorithm achieved the highest bandwidth utilization, reaching 30.3%, while the lowest was observed in the other areas under the initial scheme, at only 20%. Overall, the trend of bandwidth utilization was generally consistent with the trend of CU utilization. Since DeepHM did not consider bandwidth factors, it did not demonstrate significant superiority when compared

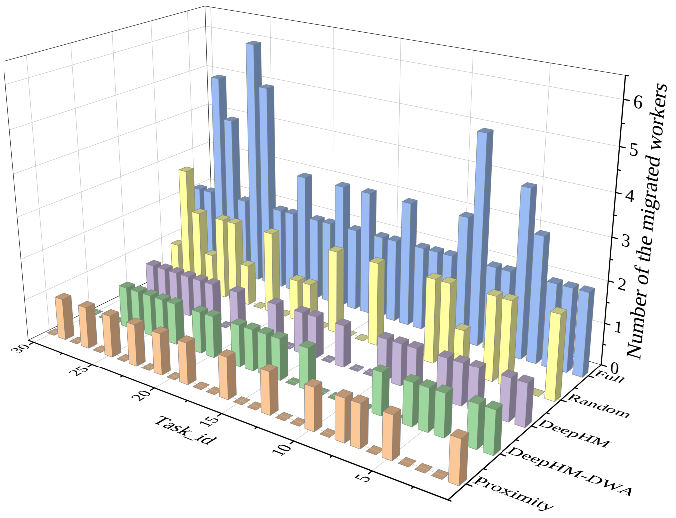


Fig. 12. Number of migrated workers in the five algorithms.

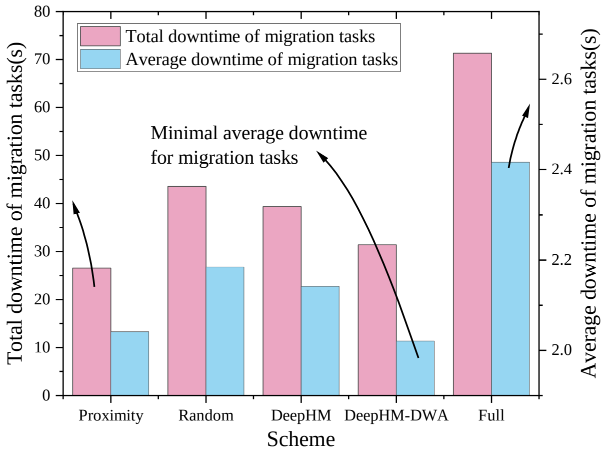


Fig. 13. Total and average downtime of migrated tasks.

to the other schemes.

4) **DMT tasks performance**

Fig. 12 provided a detailed presentation of the migration workers associated with each task for the five schemes. The full node migration strategy migrated all workers except the task generating nodes, which significantly increased the migration workers as well as the cost of GPU switching, while the node proximity-aware migration scheme had the least migration workers. In contrast, the DeepHM and DeepHM-DWA algorithms demonstrated notable efficiency compared to the full migration strategy, with only a few tasks experiencing an increase in the number of workers migrated.

In the network, we defined the downtime of DMT tasks as the time taken for transmitting the task data and parameters to the destination node. Here, we also consider the time for Optical-Electrical-Optical (OEO) conversion and GPU switching, which is expressed according to the following:

$$\begin{aligned} t_{\text{downtime}} &= \sum_{w_0=1}^{w} \frac{D_j^{w_0} + \Omega_j^{w0} \times 10^{11} \times 8}{100_{(Gbps)} \times 10^9 \times mb_j} \\ &+ \frac{d}{v_{\text{fiber}}} + (t_{OEO} \times 2 + t_{GPU}), \end{aligned} \tag{16}$$

where w represents the worker to be migrated. $D_j^{\omega 0}$ is the data size of task j at worker $\omega 0$. $\Omega_j^{\omega 0}$ denotes the parameters of task j currently iterated at $\omega 0$. mb_j signifies the number of wavelengths used for migration. d and v_{fiber} are the distance to the migrating node and the speed of propagation through the fiber, respectively [37]. t_{OEO} represents the time for OEO conversion, with two conversions required for each transmission, and t_{GPU} is the downtime for GPU [38, 39].

The total downtime and average downtime of migration tasks were shown in Fig. 13. DeepHM-DWA achieved the minimum average downtime for migration tasks because we maximized the number of wavelengths after migration, which reduced the communication time for task training. In DeepHM-DWA, the average downtime for migration tasks was 2 seconds, with a total migration time of 31 seconds. In proximity migration, the total downtime and average downtime were smaller because the migration distance was closer and fewer workers needed to be migrated. The average downtime for DeepHM was 2.1 seconds, with a total of 19 workers migrated, resulting in a total downtime of approximately 40 seconds. Full node migration had the largest average downtime and total downtime because it migrated more nodes per task and migrated more workers, with a total downtime of 70 seconds.

In terms of DMT task performance, we also considered the average ahead-of-time (AOT) ratio [26] of a DMT task improves after migration compared to its deadline. The accelerated completion time of migrated DMT tasks is attributed to the reduction in communication time, which results from changes in ring links and the number of wavelengths. The specific calculation is performed using the following:

$$\text{AOT \_ ratio} = \frac{1}{|M|} \sum_{\forall i} \frac{T_i - T_{\text{complete}}}{T_i} \times 100\%, \tag{17}$$

where $|M|$ represent the number of the DMT task. $T_{complete}$ is the completion time of the DMT task.

The average AOT ratio for the five migration schemes was depicted in Fig. 14 DeepHM-DWA exhibited a significantly faster task completion time compared to other migration schemes, with the AOT ratio for migrated tasks reduced by 4.6% compared to the deadline. Due to occupying shorter training rings and utilizing more wavelengths after migration, DeepHM-DWA reduced communication time for training. The ratio increased sequentially for random node migration, DeepHM, proximity-aware migration, and full-node migration. The ratio was related to the hops and wavelength occupied by the task in Fig. 10, with a similar trend and the number of ring hops after migration. However, despite random node migration involving more hops than DeepHM, tasks were completed faster. This was attributed to the occupation of more wavelengths, leading to higher wavelength utilization. Although full node migration had a higher wavelength utilization, the significant increase in the number of hops in the links increased the task completion time, resulting in only a 2% improvement. DeepHM can effectively minimize hotspots but

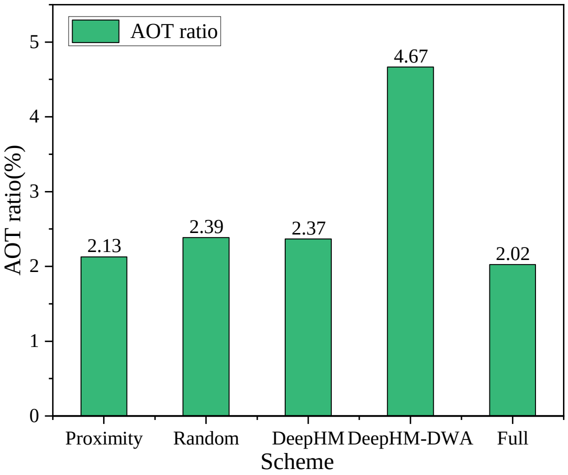


Fig. 14. Average ahead-of-time (AOT) ratio for the five schemes.

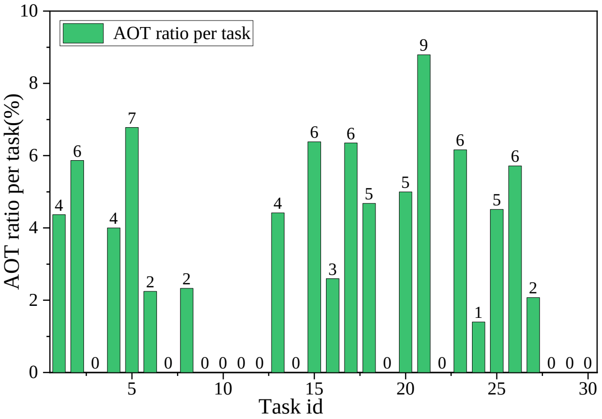


Fig. 15. AOT ratio per task when adopting DeepHM-DWA.

does not reduce task completion time. In contrast, DeepHM-DWA improves task completion time but results in higher wavelength utilization.

Next, we presented the average AOT ratio for each task in DeepHM-DWA in Fig. 15. The maximum reduction of task completion time is 9%, while the minimum reduction is 1%. When tasks were not migrated, there was no reduction in task completion, consistent with the number of workers migrated in DeepHM-DWA shown in Fig. 12.

VI. Conclusion

In this paper, we propose two DRL-based hotspot management algorithms called DeepHM and DeepHM-DWA. The algorithms work by migrating DMT tasks in ICCI networks considering their tidal characteristics. Additionally, we adopt three comparison algorithms to validate the effectiveness of the proposed algorithms. Simulation results show that DeepHM and DeepHM-DWA reduce hotspot CU-Time blocks by 19% and 18% with fewer number of worker migrations, respectively. Furthermore, both algorithms decrease variance across different network areas, effectively mitigating the tidal characteristics and balancing the network resource utilization. Moreover, DeepHM and DeepHM-DWA reduce the completion time of DMT tasks by 2% and 5%, respectively. Overall, the proposed algorithms perform hotspot management through task migration at a lower cost and accelerate the training of DMT tasks. Future works may extend this work by considering heterogeneous GPUs with varying amounts of processing and memory capacity.

References

[1] C. mobile, *China mobile intelligent computing center technology framwork white paper*. China mobile, 2023.

[2] J. G. Meyer, R. J. Urbanowicz, P. C. Martin *et al.*, “ChatGPT and large language models in academia: opportunities and challenges,” *BioData Mining*, vol. 16, no. 1, p. 20, 2023.

[3] B. Lei and G. Zhou, “Exploration and practice of computing power network (CPN) to realize convergence of computing and network,” in *Proc.OFC*, 2022, p. M4A.2.

[4] 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,” in *China communications*, vol. 18, no. 2. IEEE, 2021, pp. 175–185.

[5] R. Pang, H. Li, Y. Ji, G. Wang, and C. Cao, “Energy-saving mechanism based on tidal characteristic in computing power network,” in *Proc.NaNA*, 2021, pp. 50–54.

[6] H. Fang, W. Lu, Q. Li, J. Kong, L. Liang, B. Kong, and Z. Zhu, “Predictive analytics based knowledge-defined orchestration in a hybrid optical/electrical datacenter network testbed,” *Journal of Lightwave Technology*, vol. 37, no. 19, pp. 4921–4934, 2019.

[7] M. Yu, Y. Tian, B. Ji, C. Wu, H. Rajan, and J. Liu, “Gadget: Online resource optimization for scheduling ring-all-reduce learning jobs,” in *Proc. INFOCOM*, 2022, pp. 1569–1578.

[8] M. Yu, B. Ji, H. Rajan, and J. Liu, “On scheduling ring-all-reduce learning jobs in multi-tenant GPU clusters with communication contention,” in *Proc. MobiHoc*, 2022, pp. 21–30.

[9] C. Zhao, X. Li, J. Xin, Y. Liu, D. Li, and S. Huang, “Intelligent agent-based dynamic reliability evaluation for optical networks: A comprehensive framework and case study,” *IEEE Communications Magazine*, vol. 63, no. 3, pp. 151–157, 2025.

[10] W. Fan, F. Xiao, X. Chen, L. Cui, and S. Yu, “Efficient virtual network embedding of cloud-based data center networks into optical networks,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 32, no. 11, pp. 2793–2808, 2021.

[11] C. Li, J. Zhang, T. Ma, H. Tang, L. Zhang, and Y. Luo, “Data locality optimization based on data migration and hotspots prediction in geo-distributed cloud environment,” *Knowledge-Based Systems*, vol. 165, pp. 321–334, 2019.

[12] R. Chen, B. Liu, W. Lin, J. Lin, H. Cheng, and K. Li, “Power and thermal-aware virtual machine scheduling optimization in cloud data center,” *Future Generation Computer Systems*, vol. 145, pp. 578–589, 2023.

[13] X. Dong, T. El-Gorashi, and J. M. Elmirghani, "Energy-efficient IP over WDM networks with data centres," in *Proc. ICTON*, 2011, pp. 1–8.

[14] C. Guo, K. Xu, G. Shen, and M. Zukerman, "Temperature-aware virtual data center embedding to avoid hot spots in data centers," *IEEE Transactions on green communications and networking*, vol. 5, no. 1, pp. 497–511, 2020.

[15] S. Ilager, K. Ramamohanarao, and R. Buyya, "ETAS: Energy and thermal-aware dynamic virtual machine consolidation in cloud data center with proactive hotspot mitigation," *Concurrency and Computation: Practice and Experience*, vol. 31, no. 17, p. e5221, 2019.

[16] U. Paul, J. Liu, S. Troia, O. Falowo, and G. Maier, "Traffic-profile and machine learning based regional data center design and operation for 5G network," *Journal of Communications and Networks*, vol. 21, no. 6, pp. 569–583, 2019.

[17] A. Asrari, S. Lotfifard, and M. Ansari, "Reconfiguration of smart distribution systems with time varying loads using parallel computing," *IEEE Transactions on Smart Grid*, vol. 7, no. 6, pp. 2713–2723, 2016.

[18] H. Ma, J. Zhang, Z. Gu, H. Yu, T. Taleb, and Y. Ji, "DeepDefrag: spatio-temporal defragmentation of time-varying networks in computing power network based on reinforcement learning," in *Proc. ECOC*, 2022, pp. 1–4.

[19] M. Xu, D. Xu, C. Lou, L. Zhang, G. Huang, X. Jin, and X. Liu, "Efficient, scalable, and sustainable DNN training on SoC-clustered edge servers," *IEEE Transactions on Mobile Computing*, 2024.

[20] Y. Fan, and Y. Li, and B. Zhang, and L. Chen, and Y. Wang, and J. Guo, and W. Wang, and Y. Zhao, and J. Zhang, "GPU-efficient deployment of ring all-reduce-based distributed model training in tidal computing power network," in *Proc. ACP*, 2023, pp. 1–4.

[21] C. U. R. Institute, *White Paper on High-Throughput Data Network Architecture and Key Technologies*. China Unicom Research Institute, 2023.

[22] W. Xiao, R. Bhardwaj, R. Ramjee, M. Sivathanu, N. Kwatra, Z. Han, P. Patel, X. Peng, H. Zhao, Q. Zhang *et al.*, "Gandiva: Introspective cluster scheduling for deep learning," in *Proc.USENIX OSDI*, 2018, pp. 595–610.

[23] S. Hosseinalipour, S. Wang, N. Michelusi, V. Aggarwal, C. G. Brinton, D. J. Love, and M. Chiang, "Parallel successive learning for dynamic distributed model training over heterogeneous wireless networks," *IEEE/ACM Transactions on Networking*, 2023.

[24] C. Jo, Y. Cho, and B. Egger, "A machine learning approach to live migration modeling," in *Proc. SoCC*, 2017, pp. 351–364.

[25] T. Yang, H. Jiang, Y. Hou, and Y. Geng, "Carbon management of multi-datacenter based on spatio-temporal task migration," *IEEE Transactions on Cloud Computing*, vol. 11, no. 1, pp. 1078–1090, 2021.

[26] L. Chen, Y. Li, C. Natalino, Y. Li, B. Zhang, Y. Fan, W. Wang, Y. Zhao, and J. Zhang, "Reliable and efficient RAR-based distributed model training in computing power network," *Journal of Optical Communications and Networking*, vol. 16, no. 5, pp. 527–540, 2024.

[27] X. Chen, B. Li, R. Proietti, H. Lu, Z. Zhu, and S. B. Yoo, "DeepRMSA: A deep reinforcement learning framework for routing, modulation and spectrum assignment in elastic optical networks," *Journal of Lightwave Technology*, vol. 37, no. 16, pp. 4155–4163, 2019.

[28] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu, "Asynchronous methods for deep reinforcement learning," in *Proc. PMLR*, vol. 48, Jun 2016, pp. 1928–1937.

[29] Z. Zhong, N. Hua, M. Tornatore, Y. Li, H. Liu, C. Ma, Y. Li, X. Zheng, and B. Mukherjee, "Energy efficiency and blocking reduction for tidal traffic via stateful grooming in IP-over-optical networks," *Journal of Optical Communications and Networking*, vol. 8, no. 3, pp. 175–189, 2016.

[30] Y. Xiao, J. Zhang, and Y. Ji, "Energy-efficient DU-CU deployment and lightpath provisioning for service-oriented 5G metro access/aggregation networks," *Journal of Lightwave Technology*, vol. 39, no. 17, pp. 5347–5361, 2021.

[31] A. Chowdhery, S. Narang, J. Devlin, M. Bosma, G. Mishra, A. Roberts, P. Barham, H. W. Chung, C. Sutton, S. Gehrmann *et al.*, "Palm: Scaling language modeling with pathways," *Journal of Machine Learning Research*, vol. 24, no. 240, pp. 1–113, 2023.

[32] N. H. Tran, W. Bao, A. Zomaya, M. N. Nguyen, and C. S. Hong, "Federated learning over wireless networks: Optimization model design and analysis," in *Proc. INFOCOM*, 2019, pp. 1387–1395.

[33] J. Konečnỳ, Z. Qu, and P. Richtárik, "Semi-stochastic coordinate descent," *optimization Methods and Software*, vol. 32, no. 5, pp. 993–1005, 2017.

[34] W. Fan, F. Xiao, X. Chen, L. Cui, and S. Yu, "Efficient virtual network embedding of cloud-based data center networks into optical networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 32, no. 11, pp. 2793–2808, 2021.

[35] T. Taleb and A. Ksentini, "Follow me cloud: interworking federated clouds and distributed mobile networks," *IEEE Network*, vol. 27, no. 5, pp. 12–19, 2013.

[36] J. Liu, S. Wang, H. Xu, Y. Xu, Y. Liao, J. Huang, and H. Huang, "Federated learning with experience-driven model migration in heterogeneous edge networks," *IEEE/ACM Transactions on Networking*, 2024.

[37] N. I. Gorlov, I. V. Bogachkov, and E. T. Kitova, "Investigation of unauthorized connection to passive fiber optical access networks," in *Proc. APEIE*, 2018, pp. 154–156.

[38] T. Xie, J. Wang, Z. Wang, C. Ma, Y. Yu, J. Zhu, and J. Yu, "Long-range, high-precision, and high-speed absolute distance measurement based on alternately oscillating optoelectronic oscillators," *Optics Express*, vol. 27, no. 15, pp. 21 635–21 645, 2019.

[39] L. Liu, H. Xu, Z. Niu, J. Li, W. Zhang, P. Wang, J. Li, J. C. Xue, and C. Wang, "ScaleFlux: Efficient stateful scaling in NFV," *IEEE Transactions on Parallel and Distributed Systems*, vol. 33, no. 12, pp. 01–17, 2022.

VII. Biography Section

Yingbo Fan received his B.E. degree in electronic engineering from the Hebei Agricultural University, China, in 2022. He is currently working toward a Ph.D. degree at the School of Electronic Engineering, Beijing University of Posts and Telecommunications (BUPT). His primary research focuses on optical network and computing power network optimization.

Yajie Li received his Ph.D. degree in communication and information system from Beijing University of Posts and Telecommunications (BUPT) in 2018. He is currently with the State Key Laboratory of Information Photonics and Optical Communications, BUPT, Beijing, and School of Information Science and Technology, Tibet University, Lhasa. He was a visiting doctoral student at KTH Royal Institute of Technology from Oct. 2016 to Dec. 2017. His research interests include optical communication security, software defined optical networks, edge computing and 5G optical transport networks.

Carlos Natalino received the Ph.D. degree in electrical engineering from the Federal University of Pará, Brazil, in 2016. He is currently a Researcher with the Optical Networks Unit, Chalmers University of Technology. His research focuses on network automation and on the challenges and opportunities for application of machine learning in the network automation context. Over the past years, he has been researching how to leverage machine learning for optical network design and operation, in problems such as resource efficiency (e.g., spectrum) and physical layer security. He has been involved in several national and international projects funded by research bodies in EU, Sweden, and Brazil.

Yahui Wang received her B.E. degree in Communication Engineering from Henan Polytechnic University, China, in 2022. She received her M.S. degree in Electronic Science and Technology from the School of Electronic Engineering, Beijing University of Posts and Telecommunications (BUPT) in June 2025. Her primary research focuses on computing power network.

Jiaxing Guo received his B.E. degree in electronic engineering from the Hebei University of Science and Technology, China, in 2022. He received his M.S. degree in Electronic Science and Technology from the School of Electronic Engineering, Beijing University of Posts and Telecommunications (BUPT) in June 2025. His primary research focuses on multi-party benefit optimization in computing power networks.

Wanping Wu is currently pursuing an M.E. degree in Next-Generation Electronic Information Technology at the School of Electronic Engineering, Beijing University of Posts and Telecommunications (BUPT). Her research interests include computing power networks, load balancing, and network security.

Rongrong Ruan obtained a bachelor's degree in Applied Physics from Nanjing University of Posts and Telecommunications in 2023. He is currently pursuing a master's degree at the School of Electronic Engineering, Beijing University of Posts and Telecommunications. His main research direction is power balancing at the physical layer of optical fiber communication.

Wei Wang received the BE and PhD degrees in communications engineering from the Beijing University of Posts and Telecommunications (BUPT), in 2013 and 2018, respectively. He is currently with the State Key Laboratory of Information Photonics and Optical Communications, BUPT, Beijing, China. During 2016-2017, he was a visiting PhD student with University of California, Davis, CA, USA. During 2018–2020, he was also a member of technical staff with NSBU, VMware Inc. His research interests include network virtualization and optimization in datacenter or telecomnetworks.

Yongli Zhao received the B.S. degree in communication engineering and the Ph.D. degree in electromagnetic field and microwave technology from the Beijing University of Posts and Telecommunications (BUPT) in 2005 and 2010, respectively, where he is currently a Professor with the School of Electronic Engineering. He has published over 150 journal and conference articles. His research interests include software-defined optical networks, flexirid optical networks, and network virtualization.

Jie Zhang received the bachelor's degree in communication engineering and the Ph.D. degree in electromagnetic field and microwave technology from BUPT. He is currently a Professor and the Dean of School of Electronic Engineering, Beijing University of Posts and Telecommunications (BUPT), China. He has published over 300 technical papers. He has authored eight books, submitted 17 ITU-T recommendation contributions, and six IETF drafts. He holds 17 patents. He has served as a TPC member of a number of conferences such as ACP, OECC, PS, ONDM, COIN, and ChinaCom. His research interests include architecture, protocols, and standards of optical transport networks.