Proceedings of CONF-SEML 2026 Symposium: Computational Analysis and Modeling in Complex Intelligent Systems
DOI: 10.54254/2755-2721/2026.GU33407

Research and Analysis on Task Unloading Strategy of Edge Computing Based on Deep Reinforcement Learning

Ruijie Li

School of Information Engineering, Henan University of Animal Husbandry and Economy, Zhengzhou, China
remyuan@outlook.com

Abstract. With the development of 5g/6G and the Internet of things, time delay sensitive applications pose a serious challenge to the terminal computing capacity. Mobile edge computing can effectively reduce delay and energy consumption by offloading tasks to the near end node. However, the dynamic change of wireless channel and the uneven load of nodes lead to the highly complex unloading environment, which is difficult to adapt to the traditional methods. With the advantages of no prior model and good at sequential decision-making, deep reinforcement learning has become a research hotspot in this field. This paper systematically combs the task unloading strategy based on deep reinforcement learning, classifies and compares the existing research from four dimensions of learning paradigm, agent architecture, optimization objectives and environmental characteristics, deeply analyzes the typical application scenarios, and summarizes the current research hotspots, challenges and future development direction, in order to provide a systematic reference for the follow-up research in this field.

Keywords: Deep reinforcement learning, Mobile edge computing, Task offloading, Offloading strategy

1. Introduction

As an important stage of the evolution of the distributed computing paradigm, edge computing effectively solves the problem of high transmission delay in the core network under the cloud computing mode by sinking computing resources to the edge of the network, enabling the terminal to unload tasks to the near end node [1]. However, the time-varying wireless channel state, the dynamic fluctuation of edge node load and user mobility make the task unloading environment highly complex [2].

In the traditional unloading methods, although precise algorithms such as integer linear programming can obtain the optimal solution, the computational complexity is too high, and heuristic methods such as genetic algorithm reduce the computational overhead, but the generalization ability is insufficient [3]. With the advantages of not requiring precise environmental models and being good at handling sequential decisions, deep reinforcement learning has rapidly developed into a research hotspot in this field since its introduction in 2017 [1].

© 2026 The Authors. This is an open access article distributed under the terms of the Creative Commons Attribution License 4.0 (https://creativecommons.org/licenses/by/4.0/).

DOI: 10.54254/2755-2721/2026.GU33407

In recent years, researchers have explored the application of deep reinforcement learning in task unloading from different angles. Wang et al. Combined federated learning with model reinforcement learning [4]. Song et al. Used evolutionary multi-objective reinforcement learning method to solve the multi-objective optimization problem in UAV aided edge calculation [5]. Sun et al. Used the soft actor critic algorithm to solve the collaborative unloading problem in 6G network [6]. Ren et al. Designed a hierarchical reinforcement learning framework for large-scale UAV assisted dynamic environment [3]. Yao et al. Proposed a hybrid multi-agent PPO algorithm for some observable environments in crowd edge computing [7]. These studies have promoted the development of this field from different dimensions, such as algorithm paradigm, agent architecture, and optimization objectives.

The current research has formed a variety of algorithm paradigms based on value function method, strategy gradient method, actor critical method, as well as centralized, distributed and other diversified agent architectures. Based on the literature retrieval of IEEE Xplore and other databases, this paper selects 10 representative literatures, classifies and compares the existing deep reinforcement learning unloading strategies from four dimensions of learning paradigm, agent architecture, optimization objectives and environmental characteristics, analyzes typical application scenarios, and summarizes research hotspots, challenges and future directions, so as to clarify the applicable conditions of different DRL algorithms in various edge computing scenarios for subsequent research, and provide the basis for algorithm selection and improvement direction.

2. Introduction to core technology

2.1. Deep reinforcement learning

Deep reinforcement learning is an interdisciplinary field that integrates the perception ability of deep learning and the decision-making ability of reinforcement learning. Its essence is to learn the optimal strategy to maximize the cumulative rewards through the continuous interaction between agents and the environment [8]. Traditional reinforcement learning uses Markov decision-making process. Through the interaction between agents and the environment, it can learn the optimal strategy and maximize the cumulative reward [9]. However, when the state space is large, it will face "dimensional disaster". This problem can be effectively overcome by using the deep reinforcement learning deep neural network to approximate the value function or strategy [8]. At present, the mainstream deep reinforcement learning algorithms can be divided into three categories, including the methods based on value function (such as dqn and ddqn) suitable for discrete action space, the methods based on strategy gradient (such as PPO and SAC) suitable for continuous action space, and the actor critic methods (such as A3C and ddpg) combining the advantages of both, which have both stability and efficiency. Multi agent reinforcement learning (such as maddpg) further solves the problem of cooperation and competition in the environment of multi-agent coexistence.

2.2. Problem of unloading edge computing tasks

Due to the limited computing and storage resources, terminal devices need to unload computing tasks to edge nodes or cloud processing to reduce delay and energy consumption [1]. From the perspective of tasks, there are two categories of unloading decisions: binary unloading and partial unloading [8]. The offload decision needs to consider a variety of dynamic factors such as wireless channel state, edge node load, task characteristics, user mobility, etc. it is not only a typical multi-

DOI: 10.54254/2755-2721/2026.GU33407

objective optimization problem, but also has been proved to be a NP hard problem [2,3,8]. Traditional methods such as integer linear programming can obtain the optimal solution, but the computational complexity is too high. Heuristic algorithms (such as genetic algorithm and particle swarm optimization) reduce the computational overhead, but the generalization ability is insufficient [1,3]. Due to its advantages of not requiring a priori model and being good at dealing with dynamic environment, deep reinforcement learning provides a new solution for task unloading [8].

3. Classification and comparison of task unloading strategies based on deep reinforcement learning

3.1. Classification by learning paradigm

According to the processing methods of decision problems, the existing research mainly adopts the methods based on value function, strategy gradient and actor critical. In addition, model reinforcement learning also appears in some works as an exploration direction.

Among the methods based on value function, deep Q network (dqn) and its variants are most widely used because they can effectively deal with high-dimensional state space. Its core is to approximate the optimal action value function through neural network, so as to select the action with the maximum expected cumulative return in each state. Aiming at the problem of distributed offloading of mobile devices under unknown edge node load dynamics, Tang et al. Combined the long-term and short-term memory network, competitive architecture and dual dqn, so that the device can make decisions only based on local observations (such as task size and historical edge load), effectively reducing the task rejection rate and average delay [9]. Ke et al. Used ddqn to solve the problem of partial unloading and resource allocation in security aware MEC, and avoided the dimensional disaster caused by the increase in the number of users with the help of decentralized learning architecture [2]. According to the statistics of zabihi et al., dqn and its extensions account for about 22% of the research related to task unloading [8].

Different from the value function method, the strategy gradient based method directly optimizes the strategy itself, which is more suitable for continuous action space. Sun et al. Used the soft actor critic (SAC) algorithm to deal with the collaborative offload problem in 6G networks [6]. Based on the maximum entropy framework, sac encourages extensive exploration by simultaneously optimizing the cumulative reward and strategy entropy, so it has stronger robustness and generalization ability. The centralized and distributed sac offload algorithms proposed by them use the cooperation of edge cloud and edge edge to significantly reduce the service delay.

Actor critical method combines the advantages of value function and strategy gradient. Critics evaluate the value of the current strategy, and actors update the strategy according to the feedback of critics. Ren et al. Proposed a hierarchical reinforcement learning algorithm ht3o for the dynamic environment of large-scale UAV assisted MEC [3]. In this method, the trajectory optimization and task unloading are processed in layers. The upper layer uses the depth deterministic strategy gradient (ddpg) algorithm to optimize the continuous trajectory of UAV, and the lower layer uses dqn to make discrete unloading decisions, which effectively solves the complex decision-making problem in large-scale scenes. Wang et al. Explored a new paradigm of combining model reinforcement learning and federated learning. The femrl algorithm they proposed first trains an integrated environment dynamic model through federated learning, and then uses the TRPO algorithm training strategy within the "world model", which greatly improves the sample efficiency, and theoretically proves the monotonic improvement characteristics of the algorithm [4].

DOI: 10.54254/2755-2721/2026.GU33407

In general, different learning paradigms have different emphasis in task unloading. The value function method is good at discrete decision-making and high-dimensional state processing, the strategy gradient method is suitable for continuous control tasks, and actor critical and its extensions (such as hierarchical reinforcement learning and model reinforcement learning) better meet the unloading challenges in large-scale, dynamic, multi-objective and other complex edge computing scenarios through structural innovation.

3.2. Classified by agent architecture

The distributed nature of edge computing environment determines the importance of agent architecture design. According to the degree of centralization and decentralization of decision-making, it can be divided into centralized, distributed and hybrid architectures.

A central controller collects global information and makes all unloading decisions for the centralized intelligent body framework. Although the global optimal solution can be obtained in theory, it faces huge communication overhead, single point of failure risk and scalability problems. The csaco algorithm is an example. Although its central agent can learn better strategies, it needs to collect the status of all edge servers, and the communication overhead is large [6].

In contrast, the distributed agent architecture allows each terminal device or edge node to make decisions independently, which has good scalability and robustness. The docrrl algorithm falls into this category. In docrrl, each wireless device has an independent agent to make unloading and bandwidth allocation decisions based on local observations (such as channel status and task queue). MEC server is only responsible for assisting in training the neural network, greatly reducing the communication overhead and dependence on the central node [2,9]. The emorl-tcto algorithm proposed by song et al., although its evolutionary learning process can be regarded as a centralized optimization, the final output is a Pareto optimal strategy set containing a variety of trade-off strategies. Each strategy can be independently implemented by agents to adapt to different preferences [5].

Centralized training distributed execution (ctde) is a hybrid architecture that uses global information during training and only relies on local information during execution. It has both the training efficiency of centralized architecture and the deployment flexibility of distributed architecture. Yao et al. Proposed a hybrid multi-agent near end strategy optimization algorithm (mixed mappo) for crowd edge computing scenarios. The algorithm is centrally trained in the cloud and uses the average output of multiple critical networks to evaluate the joint strategy more stably, while the trained actor network is deployed on each user device and can make distributed unloading decisions only based on local observations, effectively solving the problem of multi-agent cooperation in some observable environments [7].

It can be seen that the centralized architecture is suitable for scenarios with high global information requirements and small network scale, but its scalability is limited; The distributed architecture sacrifices part of the global optimality for good scalability and low communication overhead, which is suitable for large-scale and dynamic environments; The hybrid architecture achieves a good balance between training efficiency and deployment flexibility through centralized training and distributed execution, especially suitable for some observable multi-agent collaboration scenarios.

DOI: 10.54254/2755-2721/2026.GU33407

3.3. Classification by optimization objective

Task unloading usually needs to weigh multiple conflicting goals, so the setting of optimization goals is the key to strategy design. According to the number and nature of objectives, it can be divided into single objective optimization and multi-objective optimization.

Single objective optimization aims to minimize or maximize a core indicator, such as time delay, energy consumption or system utility [3]. Ma et al. Decomposed the problem into two sub problems, path recommendation and task unloading, and solved them with PPO and dqn algorithms respectively. The core optimization goal is the comprehensive performance of users (such as delay energy consumption) [10]. The core of femrl algorithm is to ensure the monotonous improvement of strategy through theory, and the ultimate goal is to minimize the gap between cumulative return and real environmental return [4].

Multi objective optimization directly faces the conflict between objectives. The research of song et al. Is a typical representative. They explicitly modeled the trajectory control and task unloading problem in UAV assisted MEC as a three objective optimization problem, aiming to simultaneously minimize task delay, UAV energy consumption and maximize the number of tasks collected. The emorl algorithm they adopted is a multi strategy and multi-objective RL algorithm, which can output a group of non dominated strategies approaching the real Pareto frontier in one operation cycle, providing great flexibility for decision makers to choose appropriate strategies according to different preferences (such as paying more attention to delay or energy consumption) [5]. As Mao et al. Pointed out, in MEC system, the goals of delay, energy consumption and so on conflict with each other, and it is necessary to find the best balance through joint wireless and computing resource management [1].

Single objective optimization is applicable to scenarios where the core indicators are clear and other constraints can be transformed into a single measure, and its implementation is relatively simple; Multi objective optimization is closer to the real needs of resource competition and performance trade-off in edge computing, and can provide a variety of Pareto optimal strategies for decision makers to choose on demand, but the algorithm design is more complex. The two paradigms have their own application scenarios, which can be flexibly selected in practice according to the clarity of application requirements and the preferences of decision makers.

3.4. Classification by environmental characteristics

The unloading strategy of tasks is closely related to the dynamics and complexity of the environment. According to the characteristics of the task, it can be divided into separable and indivisible tasks. The former allows the task to be divided into several parts for parallel processing. For example, Ma et al. And Ke et al. Used the partial unloading model to further improve the system performance by optimizing the unloading ratio [2,10]. In the latter, researchers Ren et al. And Tang et al. Locally process or unload tasks as a whole [3,9].

According to the number of users, it can be divided into single user and multi-user scenarios. In the multi-user scenario, there is competition between users for communication and computing resources, which makes the problem more complex. Mao et al. Systematically sorted out the resource management technology from single user to multi-user, and then to multi server collaboration [1]. The work of song et al. And Yao et al. Is a useful exploration in multi-user complex scenarios [5,7].

According to the dynamic nature of the environment, it can be divided into static environment and dynamic environment. Ren et al. Clearly pointed out in their research that most of the existing

DOI: 10.54254/2755-2721/2026.GU33407

studies assume a static environment, and the ht3o hierarchical reinforcement learning method they proposed is just to deal with the dynamic challenges brought about by the movement of user equipment and the real-time change of environment status [3]. The dynamic nature of the environment (such as wireless channel state, user mobility, random arrival of tasks) is one of the main motivations for reinforcement learning method to be applied to task unloading, because reinforcement learning can adaptively learn the optimal strategy through interaction with the dynamic environment without prior knowledge [8].

The granularity of unloading decision is determined by the divisibility of tasks, and partial unloading can make better use of edge resources; The number of users determines the degree of resource competition, and multi-user scenarios need more complex coordination mechanisms; The dynamic environment directly affects the adaptability of the algorithm, and the unloading problem in dynamic environment is more dependent on the online learning characteristics of reinforcement learning. Therefore, in practical applications, it is necessary to select or design a matching unloading strategy and DRL algorithm according to the task characteristics, user size and the degree of environmental change.

After completing the classification and comparison of the above four dimensions, this paper further summarizes the core features of various deep reinforcement learning algorithms in edge computing task unloading. As shown in Table 1, the value function based method, strategy gradient based method, actor critical method, hierarchical reinforcement learning, Multi-Agent Reinforcement Learning and model reinforcement learning are systematically compared from five aspects: algorithm categories, typical algorithms, action space types, core advantages and typical application scenarios. This table helps researchers quickly grasp the applicable conditions and performance characteristics of different DRL algorithms in task unloading, and provides an intuitive reference for algorithm selection and improvement.

Table 1. Comparison of different DRL algorithms in task unloading

DRL algorithm category	algorithm	Action space type	Core advantages	Typical application in MEC unloading
Method based on value function	DQN, DDQN, Dueling DQN	dispersed	Suitable for discrete decision problems; Through experience playback and target network stability training, ddqn can alleviate over estimation.	Binary offload decision (whether to offload), edge node selection, bandwidth proportional allocation (after discretization).
Strategy gradient based approach	PPO, TRPO	continuous or discrete	Can directly optimize the strategy, suitable for continuous action space; PPO guarantees the update stability by tailoring the objective function.	Continuous control problems (such as UAV trajectory, unloading ratio) multi-objective optimization and path planning.
Actor critical method	DDPG, SAC	continuity	Combining value function and strategy gradient, the sample efficiency is high; Sac encourages exploration by maximizing entropy and has strong robustness.	Continuous motion trajectory control Power control and collaborative unloading proportion optimization.

DOI: 10.54254/2755-2721/2026.GU33407

Table 1. (continued)

Hierarchical reinforcement learning	HRL (DDPG + DQN)	Mixing (upper layer continuous, lower layer discrete)	The complex problem is decomposed into sub problems with different time scales, Reduce learning difficulty and improve scalability.	Large scale multi UAV dynamic environment Joint scheduling of trajectory optimization and task unloading.
Multi agent reinforcement learning	MAPPO	Discrete or continuous	Adapt to distributed multi-user scenarios, Some observable problems are handled by centralized training distributed execution (ctde).	Crowd edge computing, multi-user competition and collaboration Distributed uninstallation in some observable environments.
Model reinforcement learning	Federation integration modelRL + TRPO	continuity	Strategy training through learning environment model, Greatly improve sample efficiency and reduce interaction with real environment.	Data privacy protection scenarios in edge computing, Using federated learning and training to integrate dynamic models, Optimize the strategy within the model.

4. Application area

The task unloading strategy driven by deep reinforcement learning has shown great potential in many frontier application fields. In the Internet of vehicles, the mobility of vehicles and the diversity of tasks put forward high requirements for the unloading strategy. Ren et al's Research scenario is a large-scale vehicle scenario, with multiple UAVs serving as edge computing nodes for mobile vehicles. The ht3o algorithm proposed by them effectively manages the dynamics brought by vehicle movement and minimizes the average task delay through hierarchical reinforcement learning [3].

In UAV aided mobile edge computing, UAV can not only act as a relay, but also as a mobile edge server carrying computing resources. Song et al's research is aimed at uav-mec system, in which UAV is responsible for collecting the task of ground intelligent equipment, and can select local processing or unloading to the base station. Their multi-objective algorithm aims to simultaneously optimize the delay, energy consumption and the number of task collections [5].

In the IOT scenario, a large number of resource constrained sensors and devices need efficient task processing solutions. Ke et al. Considered the task unloading problem of IOT devices under security threats. The docrrl algorithm proposed by Ke et al. Can minimize the weighted sum of system delay and energy consumption while protecting data privacy [2]. Yao's crowd edge computing extends the Internet of things to the level of human-computer collaboration. By using a large number of idle mobile device resources, it builds a huge distributed computing platform. Its mixed mappo algorithm provides a feasible scheme for distributed task scheduling in this highly dynamic and partially observable environment [7].

DOI: 10.54254/2755-2721/2026.GU33407

5. Challenges and future prospects

5.1. Challenges

Despite significant progress, many challenges remain in this area. First, the sample efficiency problem is still the main bottleneck restricting the practicability of the algorithm. Wang et al. Introduced the model-based method to solve this problem [3]. Secondly, the nonstationarity in the multi-agent environment leads to the difficulty of convergence analysis, and the problem of credit allocation among agents has not been completely solved [7]. Third, the state action space explosion of large-scale systems poses a challenge to the scalability of the algorithm. Ke et al. Used decentralized learning and Ren et al. Introduced hierarchical reinforcement learning to reduce complexity [2,3]. Finally, the convergence of most DRL algorithms in edge computing scenarios is still lack of strict theoretical proof [4].

5.2. Future prospects

Future research will develop in a more efficient, intelligent and secure direction. The deep integration of model reinforcement learning and federated learning is expected to further improve sample efficiency and privacy protection [4]. The combination of meta learning and reinforcement learning will enable the algorithm to quickly adapt to the new environment and reduce the retraining cost [8]. Multi agent reinforcement learning still needs continuous improvement in communication protocol design and credit allocation mechanism [7]. The combination of graph neural network and DRL can better model the topology and task dependency of edge network [8]. In terms of security and privacy, researchers can combine differential privacy, homomorphic encryption and other technologies to enhance data protection [2]. In addition, algorithm lightweight and hardware deployment will become the key to promote theoretical research to practical application, and more research on model compression and reasoning acceleration for resource constrained devices is needed [7,10].

6. Conclusion

This paper presents a systematic review of edge computing task offloading strategies based on deep reinforcement learning. Through the classification and comparison of the existing research from the four dimensions of learning paradigm, agent architecture, optimization objectives and environmental characteristics, it can be found that the methods based on value function (such as dqn and ddqn) are widely used in discrete unloading decision, the methods based on policy gradient (such as PPO and sac) are suitable for continuous control problems, and the actor critical method (such as ddpg) combines the advantages of the two. In terms of agent architecture, centralized, distributed and centralized training distributed execution (ctde) have their own advantages and disadvantages, which are suitable for different scale network scenarios. Multi objective optimization and dynamic environmental adaptability have become the focus of current research.

Deep reinforcement learning shows great potential in task unloading. In typical application scenarios such as the Internet of vehicles, UAV assisted communication, and the Internet of things, it can adapt to the dynamic environment and achieve an effective trade-off between multiple objectives such as time delay, energy consumption, and task completion rate. However, the existing research still faces challenges such as insufficient sample efficiency, difficulty in multi-agent

DOI: 10.54254/2755-2721/2026.GU33407

convergence analysis, limited system scalability and insufficient guarantee of theoretical convergence.

Future research should focus on the integration of model reinforcement learning and federated learning, the combination of meta learning and DRL, the optimization of multi-agent cooperation mechanism, the application of graph neural network and the lightweight deployment of algorithm. Through these explorations, deep reinforcement learning is expected to play a more important role in edge computing system.

References

[1] Mao, Y., You, C., Zhang, J., et al. (2017) A survey on mobile edge computing: The communication perspective. IEEE communications surveys & tutorials, 19(4): 2322-2358.
[2] Ke, H.C., Wang, H., Zhao, H.W., et al. (2021) Deep reinforcement learning-based computation offloading and resource allocation in security-aware mobile edge computing. Wireless Networks, 27(5): 3357-3373.
[3] Ren, T., Niu, J., Dai, B., et al. (2021) Enabling efficient scheduling in large-scale UAV-assisted mobile-edge computing via hierarchical reinforcement learning. IEEE Internet of Things Journal, 9(10): 7095-7109.
[4] Wang, J., Hu, J., Mills, J., et al. (2023) Federated ensemble model-based reinforcement learning in edge computing. IEEE Transactions on Parallel and Distributed Systems, 34(6): 1848-1859.
[5] Song, F., Xing, H., Wang, X., et al. (2022) Evolutionary multi-objective reinforcement learning based trajectory control and task offloading in UAV-assisted mobile edge computing. IEEE Transactions on Mobile Computing, 22(12): 7387-7405.
[6] Sun, C., Wu, X., Li, X., et al. (2021) Cooperative computation offloading for multi-access edge computing in 6G mobile networks via soft actor critic. IEEE Transactions on Network Science and Engineering, 11(6): 5601-5614.
[7] Yao, S., Wang, M., Ren, J., et al. (2025) Multi-agent reinforcement learning for task offloading in crowd-edge computing. IEEE Transactions on Mobile Computing, 24(10): 9289-9302.
[8] Zabihi, Z., Eftekhari Moghadam, A.M., (2023) Rezvani M H. Reinforcement learning methods for computation offloading: a systematic review. ACM Computing Surveys, 56(1): 1-41.
[9] Tang, M., Wong V.W.S., (2020) Deep reinforcement learning for task offloading in mobile edge computing systems. IEEE transactions on mobile computing, 21(6): 1985-1997.
[10] Ma, B., Xu, Y., Pan, Y., et al. (2024) A multi-user mobile edge computing task offloading and trajectory management based on proximal policy optimization. Peer-to-Peer Networking and Applications, 17(6): 4210-4229.