面向智能体互联网的AIGC任务分布式自适应智能卸载策略

袁晓铭[1]，张馨灵[1]，邓庆绪[1]，李长乐[2]，王嘉诚[3]，策力木格[4]

（1.东北大学秦皇岛分校河北省海洋感知网络与数据处理重点实验室，河北 秦皇岛 066004；2.西安电子科技大学空天地一体化综合业务网全国重点实验室，陕西 西安 710071；3.新加坡南洋理工大学计算机与数据科学学院，新加坡 308232；4.电气通信大学信息理工学研究科，日本东京都调布市 181-8585）

摘要：针对智能体互联网环境下人工智能生成内容任务的高算力需求与动态服务质量特性，本文提出一种基于PPO的智能计算卸载策略。首先，构建包含卸载位置决策与生成质量等级选择的二维联合动作空间，以适应AIGC任务的可伸缩特性；其次，设计融合用户偏好模式的用户服务体验质量奖励函数。进一步地，针对多维状态空间量纲差异导致的训练不稳定，引入状态与奖励双重在线归一化机制，并结合动态学习率调度策略，提升算法对复杂环境的特征提取效率。仿真实验结果表明，该方法在收敛速度与稳定性上显著优于基准算法。

关键词：智能体互联网；边缘侧大模型；移动边缘计算；深度强化学习；人工智能生成内容

中图分类号：TP393

文献标志码：A

doi：10.11959/j.issn.1000

Distributed Adaptive Intelligent Offloading Strategies for AIGC Tasks in the Internet of Agents

YUAN Xiaoming[1], ZHANG Xinling[1], DENG Qingxu[1], LI Changle[2], WANG Jiacheng[3], WU Celimuge[4]

1. Hebei Key Laboratory of Marine Perception Network and Data Processing, Northeastern University, Qinhuangdao 006004, China
2. State Key Laboratory of Integrated Services Networks, Xidian University, Xi'an 710071
3. College of Computing and Data Science, Nanyang Technological University, Singapore 308232
4. Graduate School of Informatics and Engineering, The University of Electro-Communications, Chofu-shi, Tokyo 182-8585, Japan

Abstract: The Internet of Agents environment imposes high computational demands and requires dynamic Quality of Service (QoS) for AIGC tasks. To address these challenges, this paper proposes an intelligent computational offloading strategy based on PPO. First, a two-dimensional joint action space encompassing offloading location decisions and generation quality level selection was constructed to accommodate the scalable characteristics of AIGC tasks. Second, a Quality of Experience reward function integrating user preference modes is designed. Furthermore, to address training instability caused by dimensional discrepancies in the multidimensional state space, a dual online normalization mechanism for states and rewards is introduced. This is combined with a dynamic learning rate scheduling strategy to enhance the algorithm's feature extraction efficiency in complex environments. Simulation results demonstrate that the proposed method significantly outperforms baseline algorithms in terms of convergence speed and stability.

Key words: internet of agents, edge-based large models, mobile edge computing, deep reinforcement learning, AIGC

收稿日期：XXXX–XX–XX；**修回日期：**XXXX–XX–XX

通信作者：袁晓铭，yuanxiaoming@neuq.edu.cn

基金项目：国家自然科学基金面上项目（No. 62371116）；河北省研究生创新资助项目（No. CXZZSS2026175）

Foundation Items： The National Natural Science Foundation of China (No. 62371116), Hebei Postgraduate Innovation Funding Project (No. CXZZSS2026175)

0 引言

人工智能（Artificial Intelligence, AI）[1]正加速向通用人工智能（General Artificial Intelligence, AGI）演进，智能体已成为实现此愿景的关键路径。这些智能体正从简单的自动化程序转变为能够自主感知[2]、决策并与环境交互的实体[3]。随着现实世界任务复杂性日益增加，多智能体间的分工协作将成为必然趋势。此外，通信网络也正经历着向智能体互联网（Internet of Agents, IoA）的变革，力求构建一个以智能体为中心的、旨在最大化交互效率的网络范式。然而，IoA尚处于发展初期，其体系架构、交互协议等关键技术仍不成熟[4]，这为人工智能生成内容（Artificial Intelligence Generated Content, AIGC）等新兴应用在IoA环境中的高效运行带来了挑战。

在当前阶段，IoA的合理技术路径应是将其视为运行在成熟物理基础设施之上的、以服务为中心的应用层范式。在此分层架构中，移动边缘计算（Mobile Edge Computing, MEC）作为基础设施层，依托信道状态信息获取和自适应物理层技术[5]，提供分布式计算、存储与网络能力。IoA则作为服务抽象层负责实体间交互，将优化目标从传统资源效率转向以内容质量为核心的用户体验质量（Quality of Experience, QoE）。AIGC任务是驱动IoA发展的典型应用，其生成过程需处理海量数据并依赖复杂模型推理，往往对实时性与低延迟有严苛要求[6]。这些任务正越来越多地由智能汽车、无人机和扩展现实设备等边缘终端发起。然而，在IoA的愿景下，这些边缘终端被赋予智能体的属性，意味着它们不仅是服务的请求者，更是具备自主决策、本地计算与内容生成能力的独立实体。正因如此，AIGC任务的执行不再是简单的上传与下发过程，而演变为智能体根据自身状态与环境反馈，在本地自主生成与边缘协同处理之间进行的智能权衡。考虑到大规模IoA环境中实体间的隐私壁垒与通信开销，完全中心化的联合优化机制难以落地。本文所提出的智能卸载模型，正是为IoA中的个体智能体构建了一套核心决策引擎。通过去中心化的强化学习赋能，每个智能体通过感知服务器负载等环境变量隐式地捕捉资源竞争态势，在复杂的IoA环境中自适应地决定生成任务的质量与部署位置，从而实现个体利益的最大化。

MEC能将任务卸载至邻近边缘服务器[7]（Edge Server, ES），显著增强处理能力、降低时延并优化能耗。文献[8]提出支持边缘-云端与边缘-边缘协同的6G-MEC架构，将协同卸载建模为马尔可夫决策过程（Markov Decision Process, MDP），设计集中式与分布式软演员-评论员（Soft Actor Critic, SAC）算法。文献[9]设计三级MEC架构，基于深度确定性策略梯度（Deep Deterministic Policy Gradient, DDPG）算法实现时延-能耗双目标优化，并结合区块链保障隐私。文献[10]针对多用户场景提出基于Q学习和DDPG的部分卸载方案，利用本地与边缘并行计算最小化系统时延。然而，这些传统MEC卸载方法在面对AIGC任务时面临新的挑战。与传统的边缘计算任务相比，面向IoA环境中AIGC任务的计算卸载呈现出几个本质性的差异与挑战。首先是服务质量的可伸缩性。AIGC任务的服务质量具有内在可变性，同一任务可在不同质量等级下完成，生成质量、资源消耗与时延间存在复杂的非线性权衡。这导致决策维度从传统的一维位置决策扩展至多维联合决策空间，问题复杂度显著增加。其次，优化目标呈现多维耦合特性。传统任务性能评估相对单一，主要关注时延和能耗权衡。而AIGC任务的QoE需综合考虑生成质量、时延和能耗等多个强耦合维度，如何在多目标冲突中找到最优平衡是一项新挑战。最后，动态适应性需求显著提升。AIGC任务的最优策略高度依赖上下文，用户偏好随场景变化，网络状态和资源可用性实时波动，不同任务类型资源需求模式差异显著。这要求卸载策略具备对AIGC特性的深度感知和自适应调整能力，而传统方法往往基于静态资源分配模式，难以满足需求。这些特性表明，面向AIGC任务的卸载问题需要重新审视问题建模、决策空间设计和优化目标定义。

为应对这些挑战，学术界探索了多种先进的建模与优化范式以提升MEC系统的决策智能性。数字孪生技术与边缘计算的融合[11-14]通过创建实时数据映射和虚拟仿真，辅助决策智能体进行更为精准的资源调度和任务卸载。博弈论方法被引入MEC系统以提升资源调度精度[15-17]，通过刻画多用户或多智能体之间的策略性交互，寻求达到某种均衡状态下的资源优化配置。文献[18]针对智能互联网辅助的边缘计算网络中AIGC任务卸载存在的

动态决策不稳定、资源过载及多目标优化失衡问题，提出去中心化AIGC任务卸载架构，设计改进近端策略优化算法（Proximal Policy Optimization, PPO），为AIGC服务的高效计算卸载提供了智能决策新框架。文献[19]提出一种DRL与扩散模型融合的ADSAC算法，将注意力机制嵌入扩散模型作为策略网络以捕捉任务与ES的关键关联特征，并通过用户效用函数实现了对不同类型的AIGC任务的服务提供商选择，为边缘网络中多类型AIGC服务的高效调度提供了新思路。文献[20]针对移动边缘网络中单智能体GenAI-LLM系统能力单一、扩展性不足及个性化欠缺的问题，设计协同、辩论、竞争三类交互范式与分层、集中式等四类交互结构，通过任务分解与专业化智能体分工实现复杂任务优化。尽管上述研究为AIGC卸载提供了先进的算法框架，但它们的决策核心仍主要聚焦于为给定的AIGC任务选择最优的计算节点，本质上是一个一维的位置决策问题。这些工作普遍忽略了AIGC服务一个更内在的本质，其服务质量本身是可伸缩的。因此，现有研究普遍未能将生成质量等级作为核心决策变量纳入模型，也缺乏一个能够融合用户动态偏好[21]的自适应用户体验质量优化机制，仍有很大的提升空间。

针对以上问题，本文的主要研究工作如下。

（1）设计了一种面向AIGC任务的可伸缩QoS感知计算卸载框架。本文首次将AIGC服务的“可伸缩QoS感知”计算卸载问题形式化为MDP，并为此设计了一个新颖的、包含“部署位置”和“生成质量等级”的二维组合动作空间，以应对AIGC任务的独有特性。更进一步，本文构建了一个创新的、多维度、上下文感知的QoE奖励函数，从任务时延、系统能耗和内容生成质量三方面综合评估系统性能。

（2）提出了SE-PPO（Service-optimized Experience-aware PPO）智能卸载算法。该算法在PPO原始框架基础上，针对IoA环境下AIGC任务卸载的独特挑战进行了系统性改进。本文引入了状态与奖励双重归一化机制以应对异构多维状态特征的量级差异，设计学习率预热与动态衰减策略以平衡初期探索稳定性与后期收敛精度，并构建自适应QoE奖励函数处理多目标动态冲突。这些改进使得SE-PPO能够在复杂动态环境中高效学习最优卸载策略。

（3）本文通过搭建多边缘服务器、多AIGC任务类型的复杂仿真环境，对所提方法进行了系统验证。实验结果表明，与传统启发式算法及主流DRL算法（如PPO、DQN、A2C等）相比，所提SE-PPO在动态环境下获得了更高的长期累积QoE奖励，同时有效降低了平均任务时延和系统能耗，延长了终端设备续航，验证了该方法在AIGC服务计算卸载场景中的有效性与优越性。

1　系统模型和问题建模

1.1　系统模型

本文提出的面向IoA的AIGC任务智能卸载系统架构如图1所示，它的应用范围十分广泛，涵盖智慧工厂、智能交通和智能家居等多个领域[22]。系统中的用户终端，也是AIGC任务的发起者和潜在处理节点。这些移动设备集合记为 $\mathcal{M}=\{1,2,\cdots,M\}$。每个设备 $m\in\mathcal{M}$ 拥有实时电量 $\text{bat}\in[0,100]$，该状态是智能体决策的一个输入数据，将间接影响优化目标QoE函数的权重的动态调整。移动设备还具备一定的本地处理能力 f_m^{loc}(G-cycles/s)，并维护一个本地任务队列 $Q_m^{\text{loc}}(t)$ 和一个传输任务队列 $Q_m^{\text{trans}}(t)$。用户请求AIGC服务即移动设备 $m\in\mathcal{M}$ 在时间t时生成一个AIGC任务 $\text{Task}_{m(t)}$。任务的基本属性为 $\varphi_{m(t)}=\left(u_{m(t)},D_{m(t)},C_{m(t)},u_{m(t)}^{\text{mode}}\right)$。分别表示任务编号、任务数据大小(MB)、任务的计算密度(G-cycles/bit)和用户对该任务选择的偏好模式。为确定用户的偏好模式 $u_{m(t)}^{\text{mode}}$，应用程序可提供“快速预览模式”、“均衡模式”和“最佳质量模式”的选项供用户选择所需类型；或者，也可根据应用程序的功能类别来预定义任务的默认实现模式。在实际运行中，系统队列往往同时存在多个不同偏好模式的异构任务，它们按先进先出策略排队等待处理。

针对新生成的AIGC任务 $\varphi_{m(t)}$，系统中的DRL智能体将全面感知当前环境状态，包括任务的自身属性、用户所设定的偏好模式 $u_{m(t)}^{\text{mode}}$、移动设备m的实时电池电量 $\text{bat}_{m(t)}$、以及无线网络带宽 B_t 和所有ES的动态负载 $\mathbf{L}_n(t)$ 等关键信息。基于这些观测，智能体将做出最大化用户体验的二维决策：一是确定任务的部署位置（即是在移动设备m本地处理，还是卸载到某个边缘服务器n）；二是选择

AIGC 内容的生成质量等级[23]。这一决策旨在动态平衡用户对时延、能耗和生成质量的多重需求，并有效管理移动设备的续航压力和 ES 的负载，从而最大化长期累积的用户体验。其中，ES 集合表示为 $\mathcal{N}=\{1,\cdots,N_S\}$。对于每个边缘服务器 $n\in\mathcal{N}$，它的计算资源 $\mathbf{F}^{\text{edge}}$(G-cycles/s)和负载 $\mathbf{L}_n(t)$(G-cycles)会随着任务的部署和完成而动态变化。此外，每个服务器 n 维护一个共享的 AIGC 任务队列 $Q_n^{ES}(t)$，新到达的任务会进入该队列等待处理。为了应对 AIGC 任务独有的特性，智能体的决策不仅决定任务的部署位置还要决定任务的生成质量等级。我们定义部署位置决策变量为 $\text{loc}_{m(t)}\in 0\cup\mathcal{N}$。其中，当 $\text{loc}_{m(t)}=0$ 时，表示任务 $\varphi_{m(t)}$ 将在移动设备 m 本地处理；当 $\text{loc}_{m(t)}=n$（其中 $n\in\mathcal{N}=\{1,\cdots,N_S\}$）时，表示任务将被卸载至第 n 个 ES 中进行处理。同时，我们定义生成质量等级决策变量为 $q_{m(t)}\in\mathcal{Q}=\{q_0,q_1,\cdots,q_{K-1}\}$，表示任务将以 $q_{m(t)}$ 质量等级进行生成。与传统 MEC 任务不同，AIGC 任务具有显著的计算可伸缩性，即生成内容的质量并非固定不变，而是可以在计算资源与生成质量之间进行灵活权衡。为在数学模型中精确刻画这一特性，我们将生成质量等级 $q_{m(t)}$ 显式纳入决策空间，并将其映射为模型推理中的迭代采样步数[24]。进一步地，我们将优化问题从传统 MEC 卸载的一维位置决策转变为卸载位置与生成质量的二维联合决策。AIGC 服务以用户体验而非单纯的系统资源效率为核心优化目标，因此我们的奖励函数在设计时也充分融合了用户偏好模式 $u_{m(t)}^{\text{mode}}$ 对质量、时延与能耗的动态权衡，而非简单地对时延和能耗进行线性加权求和。

1.2　时延模型

对于每个 AIGC 任务 $\varphi_{m(t)}$ 在时隙 t，根据 DRL 智能体选择的动作 $a_t=\left(\text{loc}_{m(t)},q_{m(t)}\right)$，计算其总时延 $T_{u_{m(t)}}^{tal}$[25]。智能体为任务 $\varphi_{m(t)}$ 选择质量 $q_{m(t)}$ 后，任务实际消耗的计算资源为：

$$C_{m(t)}^{act}=C_{m(t)}\times\omega_i^q \tag{1}$$

ω_i^q 为每个质量等级 q 对应的计算乘子。该公式揭示了 AIGC 任务质量与计算资源正相关的本质特

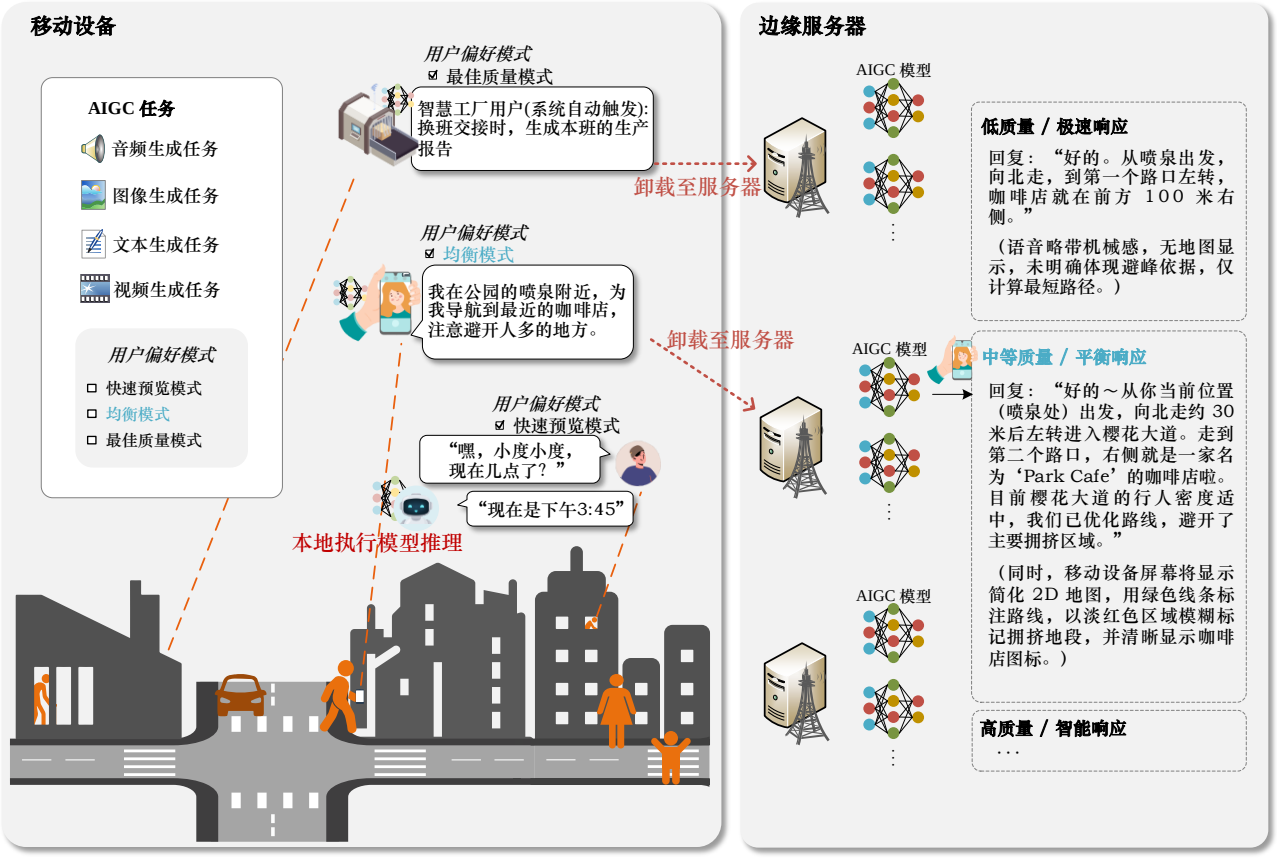


图 1　面向 IoA 的 AIGC 任务智能卸载系统架构

性。质量等级$q_{m(t)}$越高，计算乘子越大，从而直接导致计算资源消耗$C_{m(t)}^{act}$显著上升。

当任务在设备m本地执行时，设本地的排队时延为T_{loc}^{w}，则有：

$$T_{u,loc}^{w} = T_{u,loc}^{end}(u - 1) - t + 1 \tag{2}$$

队列中均采用先进先出的任务调度策略，这一时延主要取决于该节点的当前队列中现有任务的累积计算量和其最大处理能力。

我们将本地的计算时延表示为$T_{\text{loc}}^{\text{com}}$，则有：

$$T_{u,loc}^{\text{com}} = \frac{D_{m(t)}C_{m(t)}^{act}\delta_{m,t}}{f_{m}^{\text{loc}}} \tag{3}$$

因此总时延为：

$$T_{u,m}^{tal} = T_{u,loc}^{w} + T_{u,loc}^{com} \tag{4}$$

当任务卸载到边缘服务器n执行时，涉及到任务的传输时延，表示为T_{u}^{tran}。该时延包括任务输入数据的上传$T_{u,edge}^{up}$和模型生成结果的回传$T_{u,\text{edge}}^{\text{down}}$。除此之外，还有服务器排队时延$T_{u,edge}^{w}$和服务器的计算时延$T_{u,edge}^{\text{com}}$，它们分别表示为：

$$T_{u,edge}^{up} = \frac{D_{m(t)}\left(1 - \delta_{m,t}\right)}{r_{m,n}^{\text{trans}}} \tag{5}$$

$$T_{u,\text{edge}}^{\text{down}} = \frac{D_{m(t)}^{\text{out}}\left(1 - \delta_{m,t}\right)}{r_{m,n}^{\text{trans}}} \tag{6}$$

$$T_{u,\text{edge}}^{w} = T_{u,\text{edge}}^{\text{end}}(u - 1) - t + 1 \tag{7}$$

$$T_{u,edge}^{\text{com}} = \frac{D_{m(t)}C_{m(t)}^{act}\left(1 - \delta_{m,t}\right)}{f_{n}^{\text{edge}}} \tag{8}$$

传输速率$r_{m,n}^{\text{trans}}$的计算定义为：

$$r_{m,n}^{trans} = B_{t}lb\left(1 + \frac{P_{m}\left|h_{m,n}\right|^{2}}{\sigma^{2}}\right) \tag{9}$$

其中，B_{t}为带宽，P_{m}是移动设备的传输功率，$h_{m,n}$为信道增益，σ^{2}表示信道内的高斯噪声功率。

因此总时延为：

$$T_{u,n}^{\text{tal}} = T_{u,\text{edge}}^{up} + T_{u,\text{edge}}^{\text{down}} + T_{u,\text{edge}}^{w} + T_{u,\text{edge}}^{\text{com}} \tag{10}$$

1.3　能耗模型

由于不考虑任务在队列中等待的静态等待能耗，故系统总能耗由传输能耗和计算能耗两部分组成。此外，移动设备存在电池电量参数，电量会因移动设备的行为而有所消耗。当移动设备待机时，电量也会随时间略有减少。由于待机能耗数值较小对系统能耗影响不大，此处我们也不考虑[26]。

那么，对于每个移动设备而言，当任务在移动设备上处理时，设$E_{m,t}^{loc}$代表本地移动设备m的本地计算能耗，则有：

$$E_{m,t}^{loc} = \kappa T_{u,loc}^{com}\left(f_{m}^{loc}\right)^{3} = \kappa\left(f_{m}^{loc}\right)^{2}D_{m(t)}C_{m(t)}^{act} \tag{11}$$

当任务需要传输至ES时，设$E_{m,n}^{\text{tran}}$代表传输能耗，则有：

$$E_{m,n}^{\text{tran}} = P_{m,n}^{\text{tran}}\left(T_{u,\text{edge}}^{up} + T_{u,\text{edge}}^{\text{down}}\right) \tag{12}$$

因此，这部分的总能耗为：

$$E_{loc} = \sum_{t=1}^{T}\sum_{m=1}^{M}\left(E_{m,t}^{loc} + E_{m,n}^{tran}\right) \tag{13}$$

对于边缘服务器，当任务卸载至ES处理时，$E_{n,t}^{edge}$代表服务器的计算能耗，则有：

$$E_{n,t}^{edge} = \kappa T_{u,edge}^{com}\left(f_{n}^{edge}\right)^{3} = \kappa\left(f_{n}^{edge}\right)^{2}D_{m(t)}C_{m(t)}^{act} \tag{14}$$

因此，这部分的总能耗为：

$$E_{\text{edge}} = \sum_{t=1}^{T}\sum_{m=1}^{M}E_{n,t}^{\text{edge}} \tag{15}$$

1.4　优化目标

为了提供低时延的AIGC服务，我们的目标是为AIGC任务的计算卸载策略寻求一个最优解，以最大化用户在长期运行中的体验质量。QoE综合考虑了任务的生成质量、处理时延和系统能耗，并根据移动设备的电量和用户偏好进行权衡。

我们将优化问题定义为：

$$F_{m(t)} = \begin{cases} w_{q}\cdot\beta\left(q_{m(t)}\right) - w_{t}\cdot\dfrac{T_{u,m}^{\text{tal}}}{T^{\text{norm}}} - w_{e}\cdot\dfrac{E_{loc}}{E^{\text{norm}}} & \left(\text{loc}_{m(t)} = 0\right) \\ w_{q}\cdot\beta\left(q_{m(t)}\right) - w_{t}\cdot\dfrac{T_{u,n}^{\text{tal}}}{T^{\text{norm}}} - w_{e}\cdot\dfrac{E_{edge}}{E^{\text{norm}}} & \left(\text{loc}_{m(t)} \in \mathcal{N}\right) \end{cases} \tag{16}$$

其中，$\beta\left(q_{m(t)}\right)$是任务$\varphi_{m(t)}$在质量等级$q_{m(t)}$下的质量得分，用于量化不同等级下的用户主观感官体验，用户感知的质量增益随计算资源投入呈非线性饱和趋势[27]。T_{u}^{tal}和$E_{\text{loc/edge}}$分别是任务的总时延和总能耗，T^{norm}和E^{norm}归一化常数。w_{q},w_{t},w_{e}是根据系统状态和任务属性（特别是用户偏好模式$u_{m(t)}^{\text{mode}}$和电池电量$\text{bat}_{m(t)}$）动态调整的偏好权重。

这些权重用于在质量、时延和能耗之间进行动态权衡。

本文的优化目标是最大化长期累积的折扣 QoE 奖励期望，设 loc_t 和 q_t 分别代表在时隙 t 智能体为任务 $\varphi_{m(t)}$ 所做的部署位置和质量等级决策。则有：

$$\begin{aligned}&\max \mathbb{E}\left[\frac{1}{T}\frac{1}{M}\sum_{t=1}^{T}\sum_{m=1}^{M}F_{m(t)}\left(\delta_{m,t}\right)\right]\\&\text{s.t. } \mathrm{C}_1\text{: } \mathrm{loc}_{m(t)}\in\{0\}\cup\mathcal{N},\quad \forall m\in\mathcal{M},\forall t\in\mathcal{T}\\&\mathrm{C}_2\text{: } q_{m(t)}\in\mathcal{Q}=\left\{q_0,q_1,\cdots,q_{K-1}\right\},\quad \forall m\in\mathcal{M},\forall t\in\mathcal{T}\\&\mathrm{C}_3\text{: } \mathrm{bat}_m(t)\geqslant \mathrm{bat}_m^{\min},\quad \forall m\in\mathcal{M},\forall t\in\mathcal{T}\\&\mathrm{C}_4\text{: } L_n(t)\leqslant L_n^{\max},\quad \forall n\in\mathcal{N},\forall t\in\mathcal{T}\end{aligned}\tag{17}$$

其中，C_1 定义了任务部署位置的有效决策空间：当 $\mathrm{loc}_{m(t)}=0$ 时表示任务在移动设备本地处理，当 $\mathrm{loc}_{m(t)}=n$ 时表示任务被卸载至对应的 ES 处理。C_2 定义了 AIGC 任务生成质量等级的有效决策空间，要求选择的质量等级必须属于预定义集合 Q，其中 K 为质量等级总数。C_3 和 C_4 分别表示移动设备的最低电量限制和 ES 的最大负载限制；这两个约束构成了环境的终止条件，即当移动设备电量耗尽或任意服务器过载时，当前训练回合将终止。

2 基于深度强化学习的 AIGC 任务卸载算法

针对 IoA 环境中 AIGC 任务所特有的可伸缩 QoS 特性和用户个性化的偏好模式，以及多边缘服务器负载的动态性，传统的启发式算法难以在高度动态和状态空间复杂的环境中做出最优决策[28]。此外，我们所提出的 AIGC 任务卸载问题涉及“卸载位置”与“生成质量等级”的联合决策，导致动作空间呈现高维且离散的特征。为此，本文提出一种基于深度强化学习的改进型 PPO 算法（SE-PPO, Service-optimized Experience-aware PPO），用于解决 AIGC 任务的可伸缩 QoS 感知卸载问题。该算法在标准 PPO 框架的基础上，创新性地引入了状态与奖励的双重在线归一化机制，并结合动态学习率调度策略，以显著提升算法在异构特征环境下的训练稳定性与特征提取效率。本文提出的 SE-PPO 算法作为一个轻量级的决策网络，被部署在各个智能终端设备上。它通过全面感知当前环境状态，为终端发起的 AIGC 任务做卸载与服务质量决策。

2.1 马尔可夫过程建模

本文将 AIGC 任务卸载问题建模为马尔可夫决策过程，其基本要素包括状态空间、动作空间和奖励函数[29]。

在时隙 t 时，智能体需要感知 AIGC 任务、移动设备以及 MEC 环境的实时状态。状态空间应包含 AIGC 任务属性，表示为 $\varphi_{m(t)}=\left(u_{m(t)},D_{m(t)},C_{m(t)},u_{m(t)}^{\mathrm{mode}}\right)$，分别代表任务编号、任务数据大小(MB)、任务的计算密度(G-cycles/bit)和用户对该任务选择的偏好模式；还需包含设备当前的电池电量 $\mathrm{bat}_{m(t)}$，本地计算资源 f_m^{loc}、边缘服务器计算资源 $\mathbf{F}^{\mathrm{edge}}=\left[f_1^{\mathrm{edge}},f_2^{\mathrm{edge}},\cdots,f_{N_S}^{\mathrm{edge}}\right]$；此外还需包含当前的无线网络带宽 $B(t)$ 与所有边缘服务器的实时负载向量 $\mathbf{L}(t)=\left[L_1(t),\cdots,L_N(t)\right]$。因此，系统在时刻 t 的整体状态空间可表示为：

$$s_t=\left\{u_{m(t)},D_{m(t)},C_{m(t)},u_{m(t)}^{\mathrm{mode}},bat_{m(t)},f_m^{\mathrm{loc}},\mathbf{F}^{\mathrm{edge}},B(t),\mathbf{L}(t)\right\}\tag{18}$$

为了应对 AIGC 任务独有的特性，智能体需要决定任务的部署位置和生成质量等级。其中部署位置决策为 $\mathrm{loc}_{m(t)}\in\{0\}\cup\mathcal{N}$，其中 0 表示本地执行，$\mathcal{N}=\{1,\cdots,N_S\}$ 表示边缘服务器集合；生成质量等级决策为 $q_{m(t)}\in\mathcal{Q}=\{q_0,\cdots,q_{K-1}\}$。我们将这两个维度的决策映射为一个离散的组合动作 a_t。动作空间可表示为：

$$a_t=\left(\mathrm{loc}_{m(t)},q_{m(t)}\right)\tag{19}$$

其中，动作空间的大小为 $|\mathcal{A}|=\left(1+N_S\right)\times K$，智能体输出一个对应卸载决策位置和选择的生成质量组合的索引。

本文构建了一个多维度、上下文感知的 QoE 奖励函数 $R\left(s_t,a_t\right)$，旨在综合评估时延、能耗和任务的生成质量。特别地，该函数额外引入了基于状态的自适应权重机制。因此，本文构建的奖励函数定义为：

$$R\left(s_t,a_t\right)=\alpha\cdot\left(w_q\cdot\beta\left(q_{m(t)}\right)-w_t\cdot\frac{T_u^{tal}}{T^{\mathrm{norm}}}-w_e\cdot\frac{E_u^{tal}}{E^{\mathrm{norm}}}\right)\tag{20}$$

其中，α 为权重因子，用于平衡奖励规模；$\beta\left(q_{m(t)}\right)$

为质量得分；T^{norm} 和 E^{norm} 为归一化常数。权重系数 w_q,w_t,w_e 依据观察到的当前状态 s_t 中的 $u_{m(t)}^{\text{mode}}$、$\text{bat}_{m(t)}$ 和 $\mathbf{L}_n(t)$ 进行动态调整：在 $\text{bat}_{m(t)}$ 或 $\mathbf{L}_n(t)$ 较低时，系统自动增大 w_e 并减小 w_q，以优先保障设备续航；反之则依据 $u_{m(t)}^{\text{mode}}$ 侧重于模型生成质量或速度。

2.2　基于PPO的AIGC任务卸载算法

Schulman 等提出的近端策略优化（Proximal Policy Optimization, PPO）算法[30]因其在策略更新上的稳定性与数据利用的高效性，成为当前深度强化学习领域的主流算法之一。PPO 算法基于 Actor-Critic 架构，通过引入裁剪机制限制策略更新的步长，有效防止了因学习率过大导致的策略坍塌，保证了训练过程的单调提升。尽管 PPO 算法在标准基准测试中表现优异，但直接将其应用于 IoA 环境下的 AIGC 任务卸载中仍面临诸多挑战，例如缺乏对异构多维状态的感知敏感度，从而输出不符合长期生存需求的决策；难以在动态多目标冲突中维持策略稳定性；冷启动阶段缺乏资源保护意识，无约束的探索可能严重影响系统的鲁棒性与服务连续性等[31]。

为解决上述问题，本文提出 SE-PPO 算法。该算法在 PPO 原始框架基础上，结合 AIGC 卸载场景特性进行了以下关键性改进：

（1）构建二维联合动作空间。针对 AIGC 任务特性，联合建模卸载位置与生成质量等级，实现对计算资源与服务质量的协同控制。

（2）设计自适应 QoE 奖励函数。根据用户偏好模式与设备电池状态动态修改权重系数，引导策略在保障续航的前提下最大化长期提升用户的服务体验。

（3）引入状态与奖励双重归一化机制。针对 AIGC 场景下电池电量、信道增益与离散生成质量之间极端的量纲差异，采用在线计算方法进行动态归一化，防止小量级特征被淹没，确保算法能捕捉质量权衡逻辑。

（4）应用学习率预热与动态调整策略。针对 AIGC 任务二维联合动作空间带来的搜索压力，以及 QoE 奖励随用户偏好剧烈震荡的高方差特性，在初期采用 Warmup 机制防止策略坍塌，后期通过线性衰减提升收敛精度。

基于上述改进策略，本节将详细阐述我们所提出的算法的整体网络架构，并给出关键机制的数学原理与公式。

为了消除不同状态特征之间的量级差异，算法引入观测归一化模块。设原始状态向量为 s_t，我们维护一个全局运行均值 μ_s 和运行方差 σ_s^2。在每个时间步，利用 Welford 算法[32]在线更新统计量，并对状态进行标准化处理。

$$s_t' = \text{clip}\left(\frac{s_t - \mu_s}{\sqrt{\sigma_s^2 + \epsilon}}, -C, C\right) \tag{21}$$

其中，ϵ 为防止除零的微小常数，C 为截断阈值，clip $(\cdot)$ 函数用于过滤异常值。

同理，为了稳定 Critic 网络的价值评估，我们对奖励信号进行归一化。不同于状态归一化，奖励归一化基于折扣回报的标准差。设 R_t 为当前的折扣回报估计，我们维护其标准差 σ_R，则归一化后的奖励 r_t' 为：

$$r_t' = \text{clip}\left(\frac{r_t}{\sigma_R + \epsilon}, -C, C\right) \tag{22}$$

通过双重归一化，Actor 和 Critic 网络的输入和目标值均被映射到标准正态分布附近，显著提升了梯度下降的效率。

为了进一步提升训练稳定性，我们设计了动态学习率调度策略 $\alpha(k)$，其中 k 为当前训练步数。在训练初期 $\left(k < K_{\text{warm}}\right)$，采用 Warmup 策略，学习率将线性增加到基础学习率 α_{base}，以避免初始阶段梯度过大破坏训练特征。

$$\alpha(k) = \frac{k}{K_{\text{warm}}}\alpha_{\text{base}} \tag{23}$$

在预热结束后 $\left(k \geqslant K_{\text{warm}}\right)$，采用线性衰减策略，使学习率随训练逐渐降低，从而在训练后期实现更精细的收敛。

$$\alpha(k) = \alpha_{\text{base}}\left(1 - \frac{k - K_{\text{warm}}}{K_{\max} - K_{\text{warm}}}\right) \tag{24}$$

算法采用 Critic 网络 $V_\phi\left(s_t'\right)$ 来评估归一化状态的价值。为了平衡方差与偏差，算法采用广义优势估计 (Generalized Advantage Estimation, GAE) 来计算优势函数 $\widehat{A}_t$。首先计算时序差分残差（TD-Error）δ_t。

$$\delta_t = r_t' + \gamma V_\phi\left(s_{t+1}'\right) - V_\phi\left(s_t'\right) \tag{25}$$

其中，r'_t是由公式(22)计算得到的归一化后的即时奖励，γ是折扣因子。基于δ_t，GAE优势函数定义为：

$$\widehat{A}_t = \sum_{l=0}^{T-t-1} (\gamma\lambda)^l \delta_{t+l} \tag{26}$$

其中，$\lambda \in [0,1]$是GAE平滑因子，用于调节多步回报的权重。

Critic网络的更新目标是最小化预测价值与目标价值之间的均方误差。目标价值通常定义为$R_t = \widehat{A}_t + V_\phi(s'_t)$。Critic网络的损失函数定义为：

$$\mathcal{L}_{\text{critic}}(\phi_k) = \frac{1}{|\mathcal{B}|} \sum_{(s_t,R_t)\in\mathcal{B}} \left(V_{\phi_k}(s'_t) - R_t\right)^2 \tag{27}$$

其中$\mathcal{B}$表示经验回放中的一个小批量样本集合。

Actor网络旨在最大化累积奖励，即寻找最优策略π_θ。该算法的核心在于利用裁剪（Clipping）机制限制策略更新的步长，防止因更新幅度过大导致策略坍塌，从而保证训练的单调性和稳定性。定义新旧策略的比率$r_t(\theta)$为：

$$r_t(\theta) = \frac{\pi_\theta(a_t|s'_t)}{\pi_{\theta_{\text{old}}}(a_t|s'_t)} \tag{28}$$

算法的目标函数$L^{CLIP}(\theta)$包含两部分：原始的策略梯度目标和裁剪后的保守目标。具体的优化目标如下：

$$L^{CLIP}(\theta) = E_t\left[\min\left(r_t(\theta)\widehat{A}_t, \text{clip}(r_t(\theta),1-\epsilon,1+\epsilon)\widehat{A}_t\right)\right] \tag{29}$$

其中，ϵ是裁剪超参数，clip(·)函数将比率限制在$[1-\epsilon,1+\epsilon]$区间内。

为了鼓励智能体进行探索并避免过早收敛到局部最优，我们在最终的目标函数中加入策略熵$S\pi_\theta$作为正则项。综上所述，智能体的总优化目标是最小化以下总损失：

$$\mathcal{L}_{\text{total}} = -L^{CLIP}(\theta) + c_1\sum_{k=1}^{2} L_{\text{Critic}}(\phi_k) - c_2 S[\pi_\theta] \tag{30}$$

其中，c_1和c_2分别是价值损失系数和熵正则化系数。

本文提出的算法伪代码如算法1所示。

算法1　SE-PPO

输入 $u_{m(t)}$、$D_{m(t)}$、$C_{m(t)}$、$u^{\text{mode}}_{m(t)}$、$\text{bat}_{m(t)}$、f^{loc}_m、f^{edge}_n、$B(t)$、$\mathbf{L}(t)$、α_{base}和K_{warm}

输出 卸载决策及生成质量等级

1) 初始化Actor网络参数θ和Critic网络参数ϕ，观测归一化和奖励归一化器，经验缓冲区D，设置超参数学习率α，折扣因子γ，裁剪阈值ϵ，GAE因子λ，更新轮数K;

2) **for** 每一轮训练episode = 1 to M **do**

3) 初始化环境状态s_0;

4) **for** 每个时间步 step t = 0 to T － 1 **do**

5) 根据公式(21)归一化状态s'_t;

6) 利用Actor网络根据概率分布采样动作：$a_t \sim \pi_\theta(\cdot|s'_t)$，其中$a_t = (\text{loc}_{m(t)}, q_{m(t)})$;

7) 在环境中执行动作a_t，观测即时奖励r_t和新状态s_{t+1};

8) 根据公式(22)归一化奖励r'_t;

9) 将轨迹数据$(s'_t, a_t, r'_t, s_{t+1}, \pi_{\theta_{old}}(a_t|s'_t))$存入缓冲区D;

10) **end for**

11) 根据公式(23)计算当前学习率$\alpha(k)$并更新优化器;

12) 计算状态价值$V(s'_t)$，并根据公式(26)计算优势函数$\widehat{A}_t$;

13) **for** epoch = 1 to K **do**

14) 从D中随机抽取小批量样本;

15) 根据公式(27)更新Critic网络参数ϕ以最小化价值估计误差;

16) 根据公式(30)更新Actor网络参数θ以最大化裁剪后的期望回报;

17) **end for**

18) 清空缓冲区D;

19)**end for**

3　仿真分析

3.1　仿真场景与仿真参数设置

为验证本文提出的基于SE-PPO的AIGC任务卸载算法的性能，我们基于Python 3.9.23和PyTorch 2.5.1深度学习框架构建了仿真平台。实验模拟了一个动态MEC环境，其中网络带宽和服务器负载随时间动态变化[33-34]。

针对移动终端，我们模拟了典型的高性能工业手持设备或车载单元配置。其本地CPU时钟频率设定为2.0GHz，足以处理基础运算。为反映真实

的续航压力，设备初始电池能量设为10,000J。考虑到复杂的通信环境，我们将终端的最大发射功率设定为2.0W，以确保高吞吐量AIGC数据传输的稳定性。ES则被建模为一个高性能资源汇聚点，拥有16.0GHz的聚合计算能力。网络环境方面，我们着重模拟了动态波动的无线信道特性。信道带宽并非固定值，而是在[50, 150]Mbps的区间内进行随机游走，同时系统引入了10ms的基础网络时延，以贴近真实的动态网络场景。AIGC任务的生成遵循均匀分布规律，其输入数据量分布在[10, 25]MB之间，基础计算需求则位于[3, 8]G-cycles范围内。这些任务随机分布在各移动设备和时隙中，使得系统队列中通常同时存在多个不同偏好模式的任务等待处理，这要求算法能够根据任务的业务类型特性做出差异化的卸载决策。值得注意的是，任务的实际资源消耗将动态取决于智能体所选的质量等级，我们的等级数量设置为3，三种等级分别对应0.5、1.0和2.0倍的计算量乘子。

针对本文提出的SE-PPO算法，超参数的设置充分考虑了AIGC任务奖励的稀疏与滞后特点。我们在Actor和Critic网络中分别采用了5×10^{-4}和1.5×10^{-3}的学习率，以在策略更新的稳定性与价值评估的收敛速度之间取得平衡。折扣因子γ设定为0.95，GAE平滑因子λ设为0.9，这有助于智能体更准确地进行长期回报的信度分配。详细的仿真参数汇总如表1所示。

表1　　仿真参数

仿真参数	参数值
移动设备计算资源/GHz	2.0
边缘服务器计算资源/GHz	16.0
移动设备发射功率/W	2.0
移动设备接收功率/W	0.5
移动设备待机功率/W	0.5
移动设备初始电量	10000
网络带宽/Mbps	[50, 150]
任务数据大小/MB	[10.0, 25.0]
任务计算量/G-cycles	[3.0, 8.0]
Actor/Critic学习率	5e-4/1.5e-3
折扣因子	0.95
GAE因子	0.9
裁剪系数	0.2
训练总步数	1e6

3.2　仿真结果分析

超参数的选择直接决定了深度强化学习模型的收敛效率与最终性能。为了确定SE-PPO算法在AIGC任务卸载场景下的最佳配置，我们重点探究了学习率这一关键参数对模型收敛的影响

学习率作为控制模型参数更新步长的核心因子，对训练过程的稳定性起着决定性作用。如图2所示，当网络采用较大的学习率时，虽然智能体在训练初期能获得较快的奖励增长，但由于参数更新幅度过大，策略网络容易在最优解附近剧烈震荡，难以实现稳定收敛。反之，若学习率设置过小，尽管收敛过程较为平滑，但策略提升极其缓慢，显著增加了训练时间成本。此外，由实验结果可以看出，Actor与Critic设置不同的学习率会比它们设置不同的学习率效果更好。经过多组实验对比，我们将Actor网络的学习率设定为5×10^{-4}将Critic网络的学习率设定为1.5×10^{-3}，该数值在保障快速收敛的同时，有效维持了训练后期的稳定性。

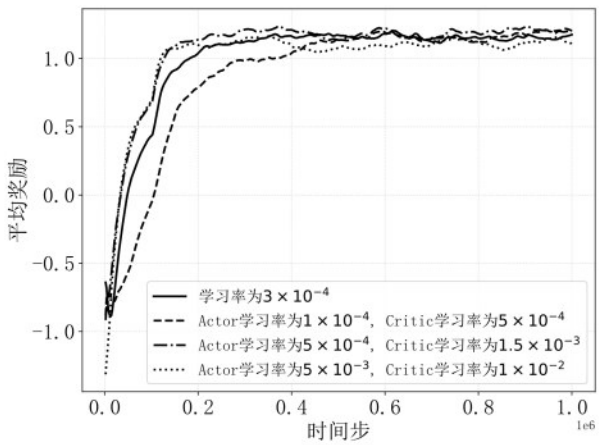


图2　不同学习率下的平均奖励收敛曲线

为了验证本文所提改进机制对算法性能的独立贡献，我们进行了消融实验。我们将SE-PPO与三种变体算法进行了对比：仅移除动态学习率机制的PPO-Norm、仅移除双重归一化机制的PPO-LR，以及移除所有改进机制的原始PPO。实验结果如图3所示，分别展示了各变体在平均奖励、任务时延及系统能耗上的收敛表现。

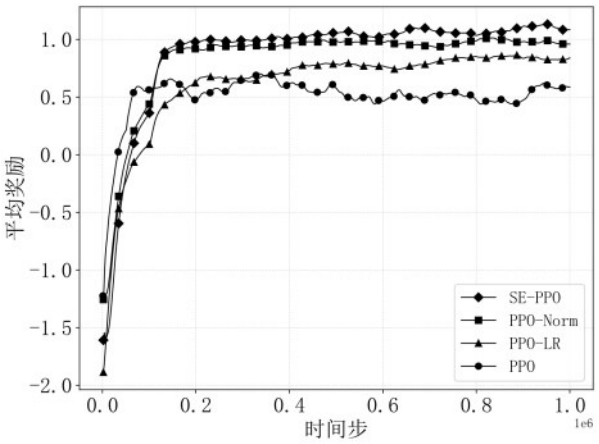
首先，分析图3(a)的平均奖励曲线可知，归一化机制在训练初期起到了决定性作用。引入了归一

化模块的SE-PPO与PPO-Norm算法，有效避免了梯度更新被大数值特征主导，保护了数值较小但对AIGC任务至关重要的生成质量特征不被淹没，从而极大加速了特征提取效率。相比之下，缺乏归一化的PPO与PPO-LR算法在初期收敛极为缓慢，尤其是原始PPO算法，始终难以摆脱低水平震荡。其次，动态学习率调度策略显著提升了训练后期的稳定性与收敛极值。任务卸载位置与质量的二维联合动作空间，导致其搜索规模较传统任务呈指数级增长。而SE-PPO通过在后期衰减学习率，实现了策略的精细化微调，使其能够锁定全局最优解。最后，综合观察图3(b)的时延曲线与图3(c)的能耗曲线，SE-PPO算法实现了综合性能的最优化。这充分证明，针对AIGC任务高算力消耗与动态QoS需求的特性，单独使用任一技术都无法达到最佳效果。

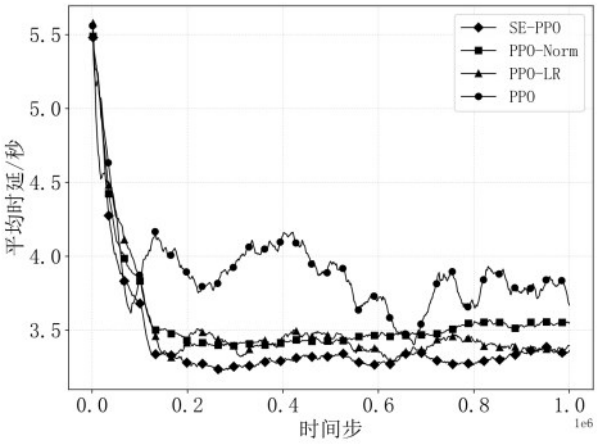
为了进一步验证AIGC特性建模相对于PPO训练技巧的独立贡献，我们设计了第二组消融实验。具体而言，我们构建了以下变体。一个是将质量等级固定为中等质量，移除质量决策维度，仅保留位置决策；另一个是将QoE奖励函数中的动态权重固定为等权配比，不随用户偏好模式和电量状态变化。实验结果如图4所示。

分析图4(a)的平均奖励曲线可知，SE-PPO显著优于所有变体算法。固定质量的SE-PPO和固定权重的SE-PPO虽然保留了训练技巧的改进，但奖励均大幅下降至SE-PPO的一半左右。这一结果有力地证明了AIGC特性建模对最终性能的决定性贡献。此外，对比同一建模条件下SE-PPO变体与PPO变体的差异，可以看出训练技巧的改进在各种建模条件下均能提供稳定的性能增益，但其贡献幅度远小于AIGC建模本身。

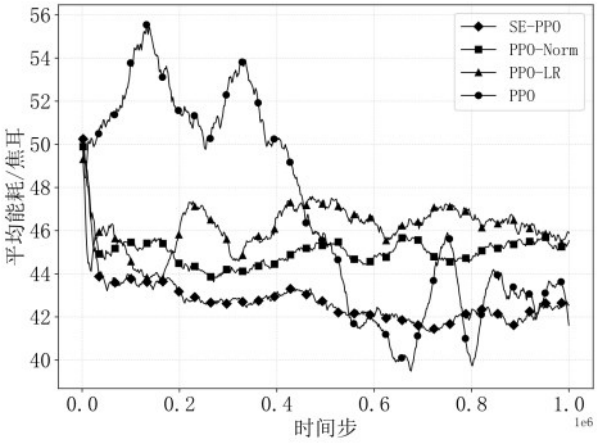
值得注意的是，观察图4(b)的时延曲线和图4(c)的能耗曲线，固定质量和固定权重的变体在时延和能耗上反而略低于完整版SE-PPO。这一现象并非SE-PPO的劣势，反而恰恰体现了AIGC质量决策的核心价值。固定质量的变体被限制在中等质量等级，天然回避了高质量生成带来的额外计算开销。而SE-PPO则能够在资源充裕时主动选择高质量等级，以适度增加的时延和能耗为代价，换取生成质量的大幅提升。综合奖励曲线可知，这种以资源换质量的策略使得SE-PPO在QoE综合评价上远优于所有变体，证明了二维联合决策空间使智能体具备了在多目标之间灵活权衡的能力。



(a)　平均奖励收敛曲线



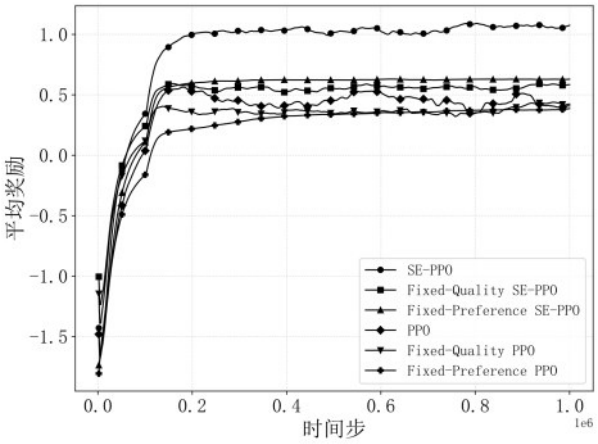
(b)　平均时延收敛曲线



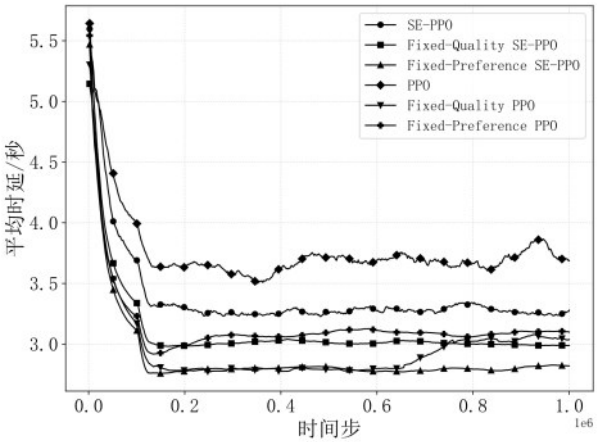
(c)　平均能耗收敛曲线

图3　算法消融实验结果对比

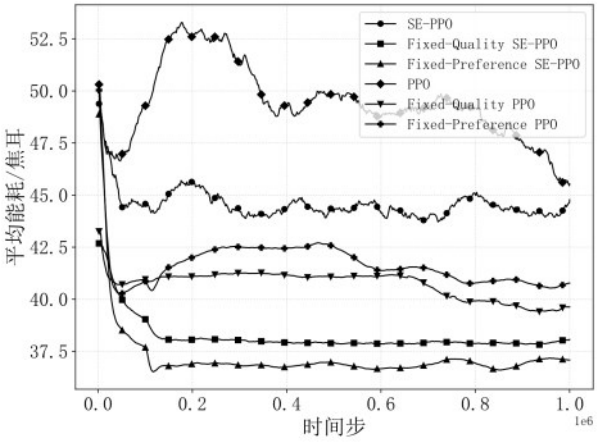
为了验证SE-PPO算法在处理动态AIGC任务



(a)　平均奖励收敛曲线



(b)　平均时延收敛曲线



(c)　平均能耗收敛曲线

图4　AIGC建模消融实验结果对比

卸载问题时的学习能力与收敛特性，我们记录了训练过程中智能体的平均累积奖励、平均时延和平均能耗的变化趋势，结果如图5至图7所示。

图5为不同算法的平均奖励曲线，首先，与A2C算法相比，SE-PPO在训练初期的收敛轨迹呈现出更为迅猛的上升趋势，且在大约150,000个时间步便率先达到了稳态收敛。在复杂的IoA边缘计算环境中，电池电量、计算负载与网络带宽等状态特征在数值量级上存在巨大差异。A2C等基线算法往往因难以处理这种不平衡的特征尺度，导致梯度更新方向被大数值特征主导，从而严重拖慢了学习效率。而SE-PPO通过在线归一化将所有输入特征映射至标准正态分布，极大地加速了神经网络对关键特征的提取与拟合效率。其次，相较于原始PPO和A2C算法，SE-PPO在最终收敛阶段展现出了更优越的稳定性与更高的奖励上限。基线算法在面对AIGC任务这种包含离散动作与高动态环境的复杂场景时，容易陷入局部最优解，具体表现为奖励曲线震荡剧烈且收敛值偏低。尽管原始PPO引入了裁剪机制以限制策略更新幅度，但在处理本文构建的部署位置与质量等级二维联合动作空间时，其表现仍不及SE-PPO。这一结果有力证明了我们设计的自适应QoE奖励函数的有效性。最后，关于DQN算法的表现，虽然其凭借贪心策略在训练初期能够快速锁定高回报动作，表现出较快的上升速度；但观察仿真后期可知，与SE-PPO相比，DQN的奖励曲线呈现出剧烈的震荡波动，且最终收敛均值显著偏低，缺乏长期稳定性。面对复杂的联合动作空间，DQN难以像PPO等策略梯度方法一样平滑地微调策略分布，导致其在后期难以精细化地逼近全局最优解。此外，图中显示的贪婪算法（Greedy）基线在整个周期内保持了较低且停滞的奖励水平。由于贪心策略仅关注当前时间步的即时收益最大化，会导致其频繁陷入次优陷阱。AIGC Heuristic是一种具备AIGC服务质量感知能力的启发式算法，它在整个训练周期内均保持了较低且停滞的奖励水平。尽管AIGC Heuristic相比Greedy引入了对质量等级的显式感知，但其奖励表现与Greedy相当，并未带来实质性提升。这一结果深刻揭示了，仅凭静态规则感知AIGC质量特性，缺乏对环境动态变化的端到端学习能力，无法有效优化长期累积QoE。

AIGC服务通常涉及高频的人机即时交互，对任务响应速度有着极为严苛的要求，因此平均任务时延是衡量系统QoE的核心指标之一。如图6所

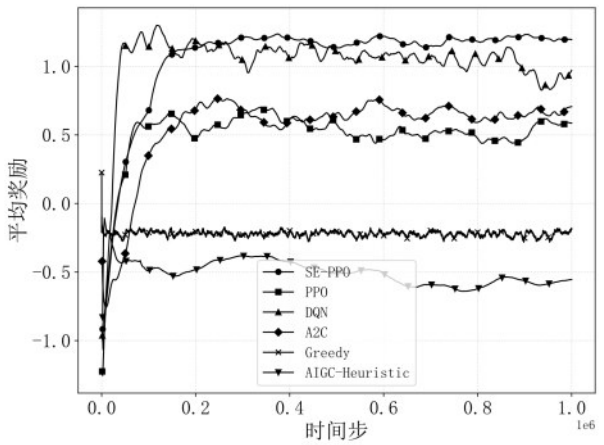


图5　不同算法的平均奖励曲线对比

示，SE-PPO算法展现出卓越的收敛特性，能够迅速将平均时延降低并稳定在极低的水平。相比之下，DQN与原始PPO算法的最终收敛值均显著高于SE-PPO，Greedy算法的平均时延也维持在较高水平，AIGC Heuristic算法的平均时延呈持续上升趋势且表现很差。值得注意的是，尽管A2C算法在训练后期的平均时延数值略低于SE-PPO，但其收敛曲线呈现出剧烈的震荡波动，缺乏保障服务质量所需的稳定性；而SE-PPO则在维持低时延的同时保持了极高的鲁棒性，从而实现了综合性能的最优化。

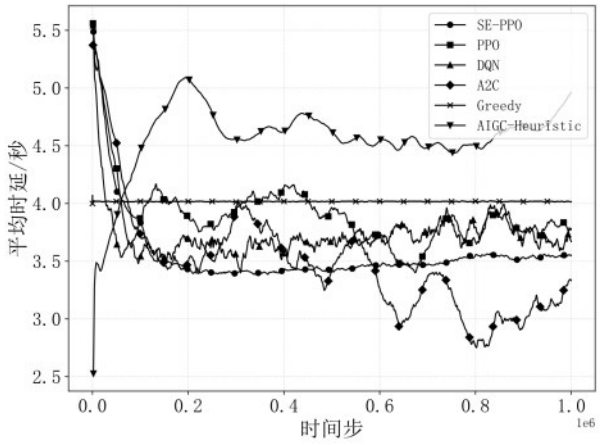


图6　不同算法的平均时延曲线对比

如图7所示，本文提出的SE-PPO算法在系统平均能耗上展现出卓越的优化能力。其能耗曲线在训练初期快速下降后，能够收敛并稳定在较低水平。相比之下，对比算法的能耗曲线均呈现出不同程度的震荡与较高的平均值。Greedy算法的能耗曲线始终处于高位，AIGC Heuristic算法的能耗曲线仍缺乏稳定的收敛行为。标准PPO算法缺乏对能耗的动态感知，其最终能耗值仍高于SE-PPO。尽管DQN算法在训练中也表现出降低能耗的趋势，但其基于价值的更新方式使其策略容易受到Q值高估的影响，导致其在能耗控制上缺乏稳定性，曲线波动较为明显。A2C算法的能耗曲线最不稳定，难以在动态环境中形成有效的节能策略。

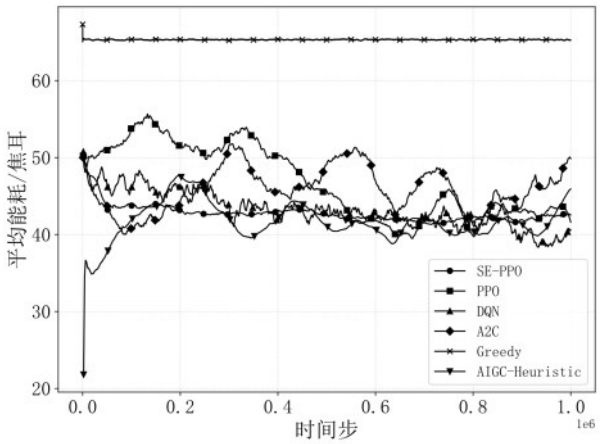


图7　不同算法的平均能耗曲线对比

图8展示了不同ES数量下，各算法收敛后平均奖励的变化趋势。这直观反映了不同算法在面对计算资源扩充与网络规模扩大时的可扩展性与资源利用效率。从图中可以观察到，随着服务器数量的增加，SE-PPO算法能够敏锐地捕捉到计算资源的增益，其奖励值呈现出稳步上升的态势，证明其能够灵活地将新增的物理资源转化为用户体验的实质提升。相比之下，基线算法在资源扩展场景下的表现存在明显局限。原始PPO和A2C算法在服务器增多时，性能并未出现预期的增长，甚至出现停滞或微幅震荡。DQN算法虽表现出一定的增长，但受限于离散动作空间的探索瓶颈，其对多服务器并发资源的调度能力不如SE-PPO精细。综合不同服务器规模下的实验数据，SE-PPO算法的平均收敛奖励分别比DQN、A2C和原始PPO提高了14.08%、78.02%和107.69%。因此，在本文研究的环境下，相较于其他基线方法，SE-PPO算法具备更强的可扩展性，能够更充分地挖掘边缘侧算力的潜能。

图9对比了不同算法在不同ES数量下的平均能耗。实验结果清晰地表明，本文提出的SE-PPO

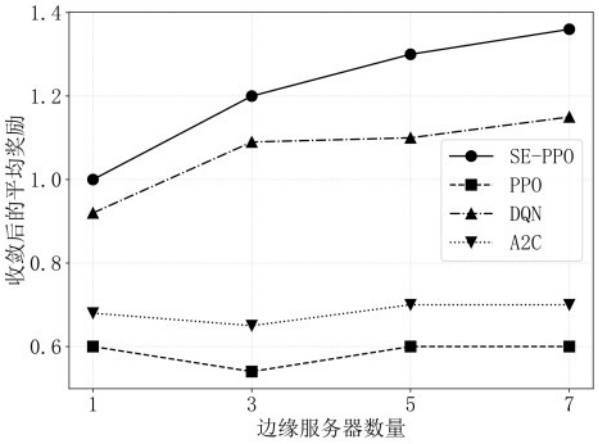


图8　不同边缘服务器数量下不同算法收敛后的平均奖励对比

算法在系统能耗优化上展现出卓越的性能。其能耗曲线不仅在所有场景下均保持最低，且呈现出稳定下降的趋势。相比之下，基线算法的表现则存在明显不足。标准PPO算法因缺乏对能耗的动态感知，其策略在能耗控制上表现随机，导致平均能耗在所有算法中最高；A2C算法的能耗曲线则最为不稳定，上下波动剧烈，反映出其策略在动态环境中难以收敛至有效的节能模式；DQN算法也展现出了一定的节能趋势，其数值紧随SE-PPO。然而，DQN作为基于价值的算法，其策略容易因Q值高估而变得短视，可能以过度牺牲服务质量为代价来换取即时能耗的降低。而SE-PPO通过策略梯度的平滑更新和多目标的动态权衡，能够在降低能耗的同时更好地维持整体QoE的平衡，实现更优的综合性能。

智能体互联网中的用户需求具有高度个性化与动态变化的特征。为了验证SE-PPO算法对不同用户服务体验偏好的自适应能力，我们设置了三种典型的用户偏好模式并统计了相应的性能指标，质量等级 $Q=\{0,1,2\}$，结果如图10所示。观察可知，SE-PPO算法能够敏锐地感知奖励函数中权重参数的变化，并自适应地调整卸载与资源分配策略。这一结果表明，本文提出的SE-PPO策略并非固定不变，而是具备良好的感知能力，能够根据特定的业务场景需求，在多维QoS指标之间实现灵活的按需配置。

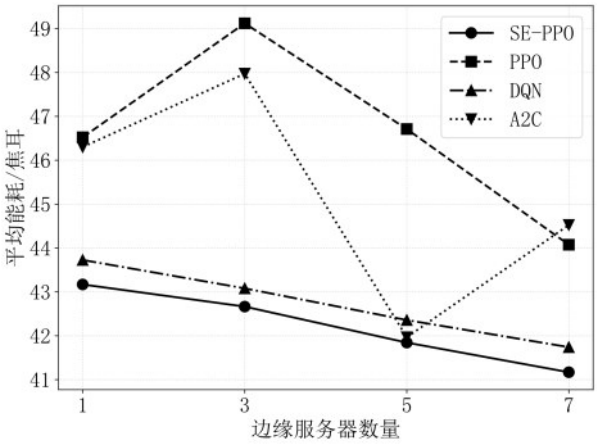


图9　不同边缘服务器数量下不同算法的平均能耗对比

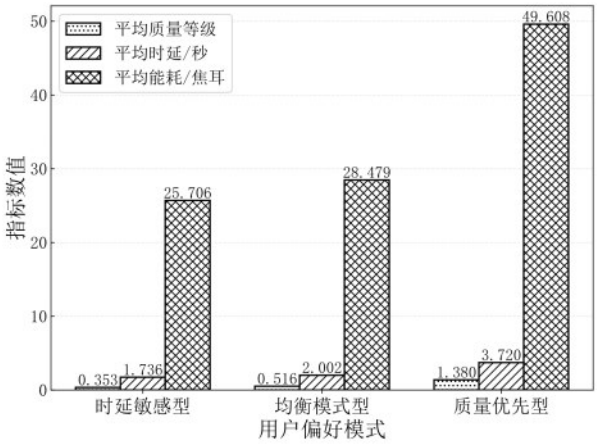


图10　不同用户偏好模式下的性能对比

移动边缘终端的续航能力是制约IoA长期运行的关键瓶颈。为了验证算法的电池保护机制，我们统计了智能体在低电量（10%-30%）、中电量（30%-50%）及高电量（50%-100%）三个区间的动作选择概率分布，结果如图10所示。实验结果揭示了策略随电池状态转移的演进规律，有力地证明了电池电量动态权重机制的有效性。智能体选择本地和高质量的决策组合的概率不到0.1%，这里我们就不绘制这一动作组合的比例了。

图中可以看到，高电量区间电池约束较弱，智能体采用较为激进的策略。虽然本地和低质量的动作占主导，但智能体仍保持了较高的卸载和高质量动作组合的选择比例，以获取高额奖励。随着电量下降至中电量区间，智能体表现出明显的保守倾向。当电量跌入危急区间时，策略出现了有趣的结构性调整。虽然直观上应进一步降低能耗，但数据显示本地和低质量比例反而下降，而卸载和高质量动作组合的比例激增至24.0%。这一现象深刻反映了本地计算本身即是巨大的能耗来源。在电量即将耗尽的极端情况下，智能体习得了计算转移策略。

通过承担单次传输能耗，将高强度的计算任务完全转移至ES执行。这不仅避免了本地CPU高负荷运行导致的快速关机，同时利用边缘侧的充裕算力换取了高质量回报，实现了绝境下的长期收益最大化。

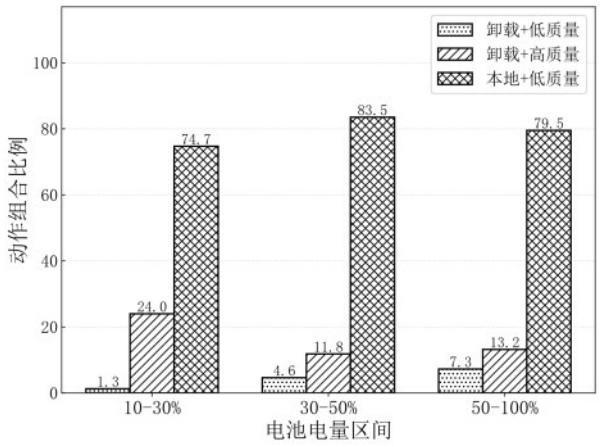


图11　不同电池电量区间下的动作选择比例

4　结束语

本文围绕IoA中AIGC任务的高算力需求与QoS可伸缩特性，构建了一个多维的边缘计算卸载模型，并提出了一种基于近端策略优化的智能卸载算法。首先，详细阐述了IoA环境下AIGC任务的通信、时延及能耗模型，重点刻画了涵盖卸载位置与生成质量等级的二维联合决策空间，并在此基础上设计了融合用户偏好模式与设备实时电量的自适应QoE奖励函数，将复杂的多目标优化问题转化为旨在最大化长期累积用户体验奖励的马尔可夫决策过程。随后，针对多维状态空间量纲差异大导致DRL算法难以收敛的问题，提出了SE-PPO算法，该算法通过引入状态与奖励的双重在线归一化机制，并结合Warmup预热与动态学习率调度策略，显著增强了智能体对异构环境特征的提取效率，有效解决了传统算法在训练初期策略震荡及后期陷入局部最优的难题。最后，通过仿真实验将提出的算法与原始PPO、A2C及DQN等多种基线算法进行了全面的性能对比，结果表明该算法在收敛速度、训练稳定性及平均任务时延等关键指标上均展现出显著优势，不仅能够根据网络波动和负载变化灵活调整卸载策略，更能在保障边缘终端续航的同时有效实现AIGC服务体验的最优化，证明了其在动态IoA环境中的鲁棒性与有效性。未来工作可将当前的单智能体决策框架扩展至多用户场景，在利用多智能体强化学习研究协同与竞争策略的同时，进一步引入更切合实际的信道干扰与并发任务处理模型。

参考文献：

[1] Xi Z H, Chen W X, Guo X, et al. The rise and potential of large language model based agents: A survey[J]. Science China Information Sciences, 2025, 68(2): 121101.

[2] 杨杰, 黄艺璇 , 杜涛, 等. 通信感知一体化原型验证的研究现状与发展趋势[J]. 通信学报, 2023, 44(11): 43-54.

[3] Wang Y T, Guo S L, Pan Y H, et al. Internet of Agents: Fundamentals, Applications, and Challenges[J]. IEEE Transactions on Cognitive Communications and Networking, 2025.

[4] Wang Y T, Pan Y N, Guo S L, et al. Security of Internet of Agents: Attacks and Countermeasures[J]. IEEE Open Journal of the Computer Society, 2025, 6: 1611-1624.

[5] Yuan X M, Lin Y B, Zhang R C, et al. PP-MoE: A Physics-Prioritized Mixture of Experts Scheme for Adaptive Channel Estimation[J]. IEEE Transactions on Wireless Communications, 2026, 25: 13654-13668.

[6] Du H Y, Niyato D, Kang J W, et al. The Age of Generative AI and AI-Generated Everything[J]. IEEE Network, 2024, 38(6): 501-512.

[7] Baghban H, Rezapour A, Hsu C H, et al. Edge-AI: IoT Request Service Provisioning in Federated Edge Computing Using Actor-Critic Reinforcement Learning[J]. IEEE Transactions on Engineering Management, 2022, 71: 12519-12528.

[8] Sun C, Wu X W, Fan Q L, et al. Cooperative Computation Offloading for Multi-Access Edge Computing in 6G Mobile Networks via Soft Actor Critic[J]. IEEE Transactions on Network Science and Engineering, 2021, 11(6): 5601-5614.

[9] He Q, Feng Z, Chen Z X, et al. Low-Cost Data Offloading Strategy With Deep Reinforcement Learning for Internet of Things[J]. IEEE Transactions on Services Computing, 2024, 18(3): 1543-1556.

[10] Deng X H, Yin J, Guan P Y, et al. Intelligent Delay-Aware Partial Computing Task Offloading for Multiuser Industrial Internet of Things Through Edge Computing[J]. IEEE Internet of Things Journal, 2021, 10(4): 2954-2966.

[11] Chen X C, Cao J N, Sahni Y, et al. Mobility-Aware Dependent Task Offloading in Edge Computing: A Digital Twin-Assisted Reinforcement Learning Approach[J]. IEEE Transactions on Mobile Computing, 2025, 24(4): 2979-2994.

[12] Yuan X M, Tian H S, Zhang X L, et al. Digital Twin-Driven MADRL Approaches for Communication-Computing-Control Co-Optimization [J]. IEEE Journal on Selected Areas in Communications, 2025, 43(10): 3596-3611.

[13] 袁晓铭, 田汉森, 黄锟达, 等. 数字孪生架构下基于GAN增强的多智能体深度强化学习边缘推理与异构资源协同优化[J]. 计算机学报, 2025, 48(8): 1763-1780.

[14] 陈赓, 宋政翰, 夏聪慧, 等. 数字孪生辅助下基于D3QN的车载网络协同卸载算法[J]. 通信学报, 2025, 46(8): 90-104.

[15] Wu H Y, Shi B W, He Q, et al. A Game-Theoretic Approach for Mi-

croservice Request Dispatching in Mobile Edge Computing Systems [J]. IEEE Transactions on Services Computing, 2025, 18(5): 2503-2516.

[16] Sun Z M, Sun G, Liu Y H, et al. BARGAIN-MATCH: A Game Theoretical Approach for Resource Allocation and Task Offloading in Vehicular Edge Computing Networks[J]. IEEE Transactions on Mobile Computing, 2024, 23(2): 1655-1673.

[17] 高玉芳, 姬智, 赵康健, 等. LEO 星座边缘计算网络中的动态计算卸载策略[J]. 通信学报, 2024, 45(7): 61-69.

[18] Zhang X X, Li S B, Tang J H, et al. DRL-Enabled Computation Offloading for AIGC Services in IIoT-Assisted Edge Computing Networks[J]. IEEE Internet of Things Journal, 2025, 12(9): 12829-12844.

[19] Liu Y J, Li S Y, Lin X, et al. QoS-Aware Multi-AIGC Service Orchestration at Edges: An Attention-Diffusion-Aided DRL Method[J]. IEEE Transactions on Cognitive Communications and Networking, 2025, 11 (2): 1078-1090.

[20] Zheng X Y, Sun G, Li J H, et al. Exploring Multi-Agent Dynamics for Generative AI and Large Language Models in Mobile Edge Networks [J]. IEEE Wireless Communications, 2025.

[21] Tian S J, Chang C, Long S Q, et al. User Preference-Based Hierarchical Offloading for Collaborative Cloud-Edge Computing[J]. IEEE Transactions on Services Computing, 2023, 16(1): 684-697.

[22] Feng W J, Zhang R J, Zhu Y C, et al. Exploring Collaborative Diffusion Model Inferring for AIGC-Enabled Edge Services[J]. IEEE Transactions on Cognitive Communications and Networking, 2025, 11(2): 946-960.

[23] Wu J Q, Zhuang X Y, Tang M, et al. QoE-Aware Offloading and Resource Allocation for MEC-Empowered AIGC Services[J]. IEEE Transactions on Mobile Computing, 2025, 24(10): 9664-9682.

[24] Rombach R, Blattmann A, Lorenz D, et al. High-Resolution Image Synthesis with Latent Diffusion Models[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). 2022: 10684-10695.

[25] Cao H F, Peng Y Z, Wang H P, et al. Multi-Satellite Cooperative Computing Task Offloading Strategy Based on Deep Reinforcement Learning[C]// Proceedings of the 2024 4th International Conference on Computer Communication and Artificial Intelligence (CCAI). Piscataway: IEEE Press, 2024: 464-471.

[26] Wang S, Li X Y, Gong Y. Energy-Efficient Task Offloading and Resource Allocation for Delay-Constrained Edge-Cloud Computing Networks [J]. IEEE Transactions on Green Communications and Networking, 2024, 8(1): 514-524.

[27] Xu M R, Du H Y, Niyato D, et al. Unleashing the Power of Edge-Cloud Generative AI in Mobile Networks: A Survey of AIGC Services [J]. IEEE Communications Surveys & Tutorials, 2024, 26(2): 1127-1170.

[28] Pan J L, Mcelhannon J. Future Edge Cloud and Edge Computing for Internet of Things Applications[J]. IEEE Internet of Things Journal, 2018, 5(1): 439-449.

[29] 许小龙, 方子介, 齐连永, 等. 车联网边缘计算环境下基于深度强化学习的分布式服务卸载方法[J]. 计算机学报, 2025.

[30] Schulman J, Wolski F, Dhariwal P, et al. Proximal Policy Optimization Algorithms[J]. arXiv Preprint, arXiv: 1707.06347, 2017.

[31] Li H, Xiong K, Lu Y P, et al. Collaborative Task Offloading and Resource Allocation in Small-Cell MEC: A Multi-Agent PPO-Based Scheme[J]. IEEE Transactions on Mobile Computing, 2025, 24(3): 2346-2359.

[32] Efanov A A, Ivliev S A, Shagraev A G. Welford's algorithm for weighted statistics[C]// Proceedings of the 2021 3rd International Youth Conference on Radio Electronics, Electrical and Power Engineering (REEPE). Piscataway: IEEE Press, 2021: 1-5.

[33] Gao H H, Huang W Q, Liu T, et al. PPO2: Location Privacy-Oriented Task Offloading to Edge Computing Using Reinforcement Learning for Intelligent Autonomous Transport Systems[J]. IEEE Transactions on Intelligent Transportation Systems, 2023, 24(7): 7599-7612.

[34] Rahmaty I, Shah-mansouri H, Movaghar A. QECO: A QoE-Oriented Computation Offloading Algorithm Based on Deep Reinforcement Learning for Mobile Edge Computing[J]. IEEE Transactions on Network Science and Engineering, 2025, 12(4): 3118-3130.

袁晓铭（1990- ），女，东北大学副教授、博士生导师，主要研究方向为边缘智能、资源管理、多智能体协同、生成式人工智能。。

张馨灵（2002- ），女，东北大学硕士生，主要研究方向为移动边缘计算、深度强化学习。

邓庆绪（1970- ），男， 东北大学教授、博士生导师， 主要研究方向为实时嵌入式系统、智能物联网。

李长乐（1976- ），男，现任西安电子科技大学教授、博士生导师，研究方向为 6G 与未来智能无线网络、车联网与自动驾驶技术。

王嘉诚（1992-），男，新加坡南洋理工大学计算机与数据科学学院研究员，主要研究方向为低空无线网络，通感一体化网络、生成式人工智能。

策力木格（1979-），男，博士，日本电气通信大学教授，日本工程院外籍院士，亚太人工智能学会会士，主要研究方向为无线网络、物联网系统、人工智能、边缘计算。