# 卫星边缘计算中考虑隐私保护的数字孪生系统任务卸载策略

林兵[1,2,3]，檀啸[1]，何凯[1]，吴华明[4]
（1.福建师范大学物理与能源学院，福建 福州 350117；2.福建省网络计算与智能信息处理重点实验室，福建 福州 350116；
3.大数据智能教育部工程研究中心，福建 福州 350116；4.天津大学应用数学中心，天津 300072）

**摘要：**针对卫星边缘计算中数字孪生任务卸载存在隐私与系统开销难权衡、依赖约束严格及依赖拓扑感知困难问题，提出了一种基于优先级图深度强化学习算法的计算卸载策略。首先，针对时延、能耗与隐私保护协同优化，构建面向隐私保护的星地数字孪生任务卸载架构。其次，设计优先级排序机制以满足依赖约束、简化求解空间。最后，提出优先级图深度强化学习算法，为排序任务求解最优卸载策略。实验结果表明，相比传统经典算法，所提算法的系统时延降低4.52%~46.16%，卫星能耗减少3.42%~32.31%，任务被攻破数量降低2.69%~57.71%。

**关键词：**卫星边缘计算；数字孪生系统；优先级调度；图注意力网络；深度强化学习
**中图分类号：**TP309
**文献标志码：**A
**doi：**10.11959/j.issn.1000–436x.TXXB260162

# Privacy-preserving task offloading strategy for digital twin system in satellite edge computing

Lin Bing[1,2,3], Tan Xiao[1], He Kai[1], Wu Huaming[4]
1. College of Physics and Energy, Fujian Normal University, Fuzhou 350117, China
2. Key Laboratory of Network Computing and Intelligent Information Processing in Fujian Province, Fuzhou 350116, China
3. Engineering Research Center for Big Data Intelligence of the Ministry of Education, Fuzhou 350116, China
4. Center for Applied Mathematics, Tianjin University, Tianjin 300072, China

**Abstract:** Satellite edge computing faces a trade-off between privacy protection and system overhead, strict task-dependency constraints, and the limited ability of conventional algorithms to capture complex dependency topologies in digital twin task offloading. To address these issues, a computation offloading strategy based on a priority graph deep reinforcement learning algorithm was proposed. Firstly, a privacy-preserving satellite-terrestrial digital twin task offloading architecture was constructed to collaboratively optimize delay, energy consumption, and privacy protection. Secondly, a priority sorting mechanism was designed to satisfy dependency constraints and simplify the solution space. Finally, a priority-based graph deep reinforcement learning algorithm was developed that extracted topological features through a graph attention network to determine the optimal offloading strategy for the sorted tasks. Experimental results demonstrate that compared with conventional algorithms, the proposed algorithm reduces system delay by 4.52%~46.16%, satellite energy consumption by 3.42%~32.31%, and the number of compromised tasks by 2.69%~57.71%.

**Key words:** satellite edge computing, digital twin system, priority scheduling, graph attention network, deep reinforcement learning

**收稿日期：**2026–03–30；**修回日期：**2026–06–17
**通信作者：**吴华明，whming@tju.edu.cn
**基金项目：**国家自然科学基金资助项目（No.62072108）；福建省高校产学合作基金资助项目（No.2022H6024, No.2021H6026）；福建省促进海洋与渔业产业高质量发展专项资金资助项目（No.FJHYF-L-2025-07-004）；福建省技术创新重点攻关及产业化基金资助项目（No.2024XQ004）；福建省高等教育研究院高等教育改革与研究项目（No.FGJG202510）
**Foundation Items:** The National Natural Science Foundation of China (No.62072108), University-Industry Cooperation of Fujian Province (No.2022H6024, No.2021H6026), Fujian Provincial Special Fund for Promoting the High-Quality Development of Marine and Fishery Industries (No.FJHYF-L-2025-07-004), Fujian Province Key Technology Innovation and Industrialization Projects (No.2024XQ004), The Higher Education Reform and Research Project of the Fujian Higher Education Research Institute (No.FGJG202510)

## 0　引言

作为虚实交互的核心技术，数字孪生已被广泛应用于工业制造领域[1]。数字孪生系统是强调“人–机–环境”深度互融的复杂系统，旨在通过感知层、孪生层与控制层的协同，实现物理实体与虚拟空间的全生命周期双向闭环控制[2]。为支撑系统的实时演化，其内部产生了大量具有严格依赖拓扑关系的计算密集型任务，这些任务必须在极短的时延约束内完成处理[3]。然而，由于算力约束，传统的本地计算模式难以支撑系统的高效处理。

云边协同计算将数字孪生系统中的任务卸载至边缘或云端，有效缓解了终端算力受限的问题[4]。然而，该模式高度依赖地面通信基础设施[5]。受限于极端的地理环境，偏远地区的设备常面临地面通信网络覆盖受限乃至完全缺失的困境。受制于高昂的部署与维护成本，传统地面云边协同难以在上述极端场景中落地，无法提供可靠的算力支撑[6]。

卫星网络是地面通信基础设施的重要延伸。卫星边缘计算（satellite edge computing, SEC）通过在卫星，特别是低地球轨道（low earth orbit, LEO）卫星上部署边缘服务器，能够为偏远地区用户提供算力服务，被视为一项极具前景的技术[7]。SEC 凭借其广域覆盖特性，能够规避复杂地理环境的物理约束，通过星地链路为偏远地区的设备提供持续的服务。此外，融合了地球静止轨道（geostationary earth orbit, GEO）卫星与地面云的多层级协同架构，能够更高效地处理任务。

近年来，已有诸多研究致力于优化卫星边缘计算环境下的卸载策略，以提升服务质量[8-11]。尽管其应用前景广阔，但在应对数字孪生系统任务的卸载时，仍面临多重严峻挑战。

（1）数字孪生系统任务隐私保护与系统开销难以兼顾。SEC 中的时延和能耗构成系统的开销。数字孪生系统中涉及大量重要的虚实交互数据，其任务的隐私保护至关重要[12]。卫星通信链路完全暴露在开放空间，使数据极易受到恶意卫星的非法窃听与篡改[13]，因此必须对隐私任务实施加密保护。高强度的加密算法会增加系统时延，因此如何权衡隐私保护与系统开销成为一大挑战。

（2）动态网络下数字孪生系统依赖任务的调度挑战。卫星网络因节点高速运动呈现高度动态的时变特性，使调度环境变得异常复杂。相较于传统的独立计算任务，数字孪生系统中的任务通常具有严格的依赖约束，其结构通常被建模为有向无环图（directed acyclic graph, DAG）[14]。如何在动态变化的卫星网络拓扑中，既满足数字孪生系统任务的依赖约束，又有效降低决策的计算复杂度，是一项具有挑战性的问题。

（3）传统算法在处理复杂依赖特征时的局限性。由依赖任务构成的 DAG 拓扑蕴含深层的结构关联特征。然而，传统算法（如深度强化学习（deep reinforcement learning, DRL））通常将环境状态映射为扁平化的一维向量，缺乏对任务间依赖拓扑结构的感知能力，导致智能体难以在复杂结构中精准识别影响系统性能的关键节点[15]。

针对上述 SEC 中数字孪生系统任务卸载背景下隐私保护与系统开销难以兼顾、动态网络中依赖约束严格，以及复杂拓扑结构难以被有效感知的挑战，本文提出了一种基于优先级图深度强化学习（priority-aware graph DRL, PGDRL）算法的计算卸载策略。本文的主要贡献如下。

（1）针对数字孪生系统任务隐私保护与系统开销难以兼顾的问题，提出了一种面向隐私保护的星地数字孪生卸载（privacy-preserving satellite-digital twin task offloading, P-SDTO）架构，利用 LEO、GEO 卫星和地面云协同计算，并融合基于高级加密标准（advanced encryption standard, AES）的多级加密机制，实现平衡隐私保护与系统开销的自适应决策。

（2）针对动态网络中数字孪生系统任务调度复杂度高、依赖约束严苛的难题，设计了一种优先级排序机制。该机制通过对任务序列进行优先级评估与重排，在严格保障任务间拓扑依赖的前提下，有效降低了模型求解的计算复杂度。

（3）针对传统算法难以感知依赖拓扑结构的问题，提出了 PGDRL 算法。通过基于图注意力网络（graph attention network, GAT）的状态感知算法，将任务属性与 DAG 拓扑重构为高维状态嵌入向量，打破了扁平化状态输入的感知瓶颈，使智能体可以依照优先级排序后的任务序列求解最优策略。

## 1　相关工作

本节回顾了云边协同计算中数字孪生系统的任务卸载、动态卫星边缘计算中依赖任务卸载以及 SEC 中的隐私保护研究。

在传统地面网络环境中，针对云边协同计算下的数字孪生任务卸载，现有研究主要通过协同架构设计与资源分配来提升系统效能。Li等[16]针对工业物联网场景中复杂且动态的计算需求，提出了一种由数字孪生驱动的云–边–端协同智能计算卸载方法。该方法通过构建边缘联盟，优化数字孪生映射和感知下的任务卸载决策。Hu等[17]设计了一种非正交多址接入辅助的云–边–端协同计算卸载策略。通过非正交多址接入技术改善无线传输效率，并对任务卸载路径与多维资源分配进行联合建模优化，有效降低了系统的处理时延与整体能耗。上述场景考虑的都是独立任务，而数字孪生系统产生的任务并非孤立存在[18]。近年来，学术界已针对依赖任务开展相关研究。Zhang等[19]将死锁避免机制融入多目标优化框架，设计了一种基于学习的无死锁依赖任务协同调度机制。随后，为了进一步降低死锁检测的计算开销，Zhang等[20]从理论层面进行创新，提出了一种具备多项式复杂度的死锁避免卸载策略。上述研究大多以完善地面通信基础设施为前提，但在通信网络匮乏或覆盖受限的偏远地区难以实现。

SEC凭借广域覆盖优势实现任务就近卸载，有效解决了上述问题。针对SEC高动态的特性，DRL算法因其快速决策能力，被广泛应用于SEC中。Zou等[21]提出了一种基于时空感知的DRL协同卸载方法，通过引入向上排序机制对依赖子任务的执行优先级进行划分，结合长短期记忆网络，实现了依赖任务的有效调度。Zhang等[22]提出了一种由终端–卫星–云组成的3层边缘计算架构，通过预留机制，联合优化任务卸载决策、通信和计算资源分配变量。为了应对更大规模的网络与并发任务，Lai等[23]构建了分层多智能体强化学习模型，有效缓解了单节点的计算压力。数字孪生系统任务间有着复杂的图拓扑结构，使传统DRL算法对拓扑感知存在局限性，因此部分研究开始尝试引入图神经网络以有效提取任务间的图拓扑特征。Cao等[24]将图神经网络与DRL深度结合，用于提取边缘计算中依赖任务的拓扑特征，显著提升了智能体对关联任务执行顺序的感知能力。尽管现有工作有效解决了动态网络下拓扑的感知决策问题，但卫星开放的通信链路容易使数据遭受恶意卫星的监听与篡改，这使数字孪生系统面临严峻的隐私泄露风险。

针对数字孪生系统任务的隐私保护问题，现有研究主要通过密码学技术或优化安全调度策略来应对威胁。Gong等[25]在空天地一体化网络中引入区块链技术，利用其去中心化和不可篡改的特性保障数字孪生卸载的安全性。为了在信息安全与性能之间寻求更好的平衡，Lan等[26]将安全风险纳入优化目标，提出了一种安全敏感的卫星任务卸载框架，通过将任务调度至低风险节点来规避威胁。此外，针对数据隐私保护需求，Xiao等[27]进一步设计了基于本地差分隐私的卸载方案，在保护隐私的同时优化系统效用。尽管现有方案在一定程度上缓解了数据泄露风险，但多采用静态加密，未将防护等级纳入动态决策，导致系统难以在安全需求与时延、能耗等资源约束之间实现有效平衡。

表1从7个维度对相关工作进行了比较，其中√表示该研究涉及或考虑了对应维度，×表示该研究未涉及或未考虑对应维度。这7个维度包括问题设定、任务约束、隐私建模、状态表示、动作空间定义、优化目标和应用场景。其中，问题设定包含地面静态环境与空天动态环境；任务约束区分了包含拓扑结构的依赖任务与独立任务；隐私建模分为考虑隐私保护与未考虑隐私保护；状态表示包括扁平一维向量与高维图嵌入特征；动作空间定义包含节点选择以及节点与安全等级联合选择；优化目标对比了独立目标优化与多目标联合优化；最后，应用场景涵盖了常规数据处理与孪生数据处理。目前，如何在动态网络中满足任务依赖约束、有效感知任务拓扑结构，并在隐私保护与系统开销之间实现权衡是当下研究亟须解决的难题。受上述观察的启发，本文提出了一种基于PGDRL算法的任务卸载策略，旨在动态多层级卫星边缘环境中实现面向隐私保护的数字孪生系统任务协同调度。

## 2 系统模型

本节构建了P-SDTO架构。如图1所示，在离散的时隙集合$\mathcal{T}=\{1,\cdots,T\}$上，P-SDTO架构由地面孪生终端层、LEO卫星层、GEO卫星层和地面云层构成。定义LEO卫星层包含$M$颗部署了边缘服务器的LEO卫星，表示为$\mathcal{M}=\{1,\cdots,M\}$。GEO卫星层与地面云层分别由GEO静止卫星节点$n_{\text{geo}}$与地面数据中心节点$n_{\text{gnd}}$组成。系统的全网计算节点集合定义为$\mathcal{N}=\mathcal{M}\cup\{n_{\text{geo}}\}\cup\{n_{\text{gnd}}\}$。

表 1　相关工作的比较

| 文献 | 问题设定 | | 任务约束 | | 隐私建模 | | 状态表示 | | 动作空间定义 | | 优化目标 | | 应用场景 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 地面静态 | 空天动态 | 依赖任务 | 独立任务 | 考虑隐私保护 | 未考虑隐私保护 | 扁平一维向量 | 高维图嵌入特征 | 节点选择 | 节点与安全等级联合选择 | 独立目标优化 | 多目标联合优化 | 常规数据 | 孪生数据 |
| 文献[16] | √ | × | × | √ | × | √ | √ | × | √ | × | √ | × | × | √ |
| 文献[17] | √ | × | × | √ | × | √ | √ | × | √ | × | × | √ | × | √ |
| 文献[19] | √ | × | √ | × | × | √ | √ | × | √ | × | × | √ | × | √ |
| 文献[20] | √ | × | √ | × | × | √ | √ | × | √ | × | √ | × | √ | × |
| 文献[21] | × | √ | √ | × | × | √ | × | √ | √ | × | √ | × | √ | × |
| 文献[22] | × | √ | √ | × | × | √ | × | √ | √ | × | × | √ | √ | × |
| 文献[23] | × | √ | √ | × | × | √ | × | √ | √ | × | √ | × | √ | × |
| 文献[24] | × | √ | √ | × | × | √ | × | √ | √ | × | √ | × | √ | × |
| 文献[25] | × | √ | × | √ | √ | × | √ | × | √ | × | √ | × | × | √ |
| 文献[26] | √ | × | × | √ | √ | × | √ | × | √ | × | √ | × | √ | × |
| 文献[27] | √ | × | × | √ | √ | × | √ | × | √ | × | √ | × | √ | × |
| 本文工作 | × | √ | √ | × | √ | × | × | √ | × | √ | × | √ | × | √ |

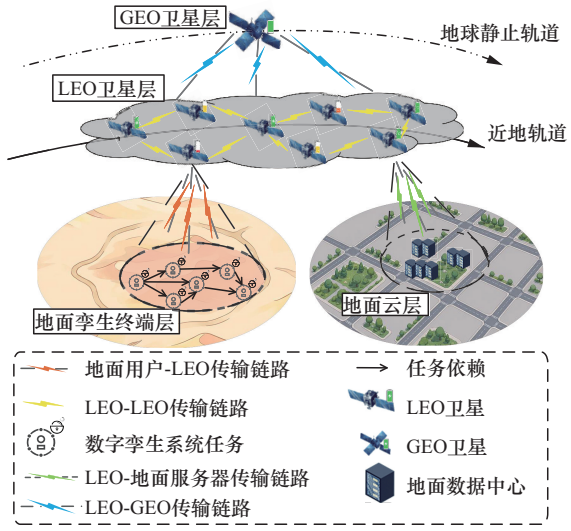



图 1　P-SDTO 架构

## 2.1　通信模型

本节首先基于离散时间快照技术对 LEO 卫星网络进行建模，随后阐述各链路传输速率的计算。

### 2.1.1　动态拓扑与离散跳数

考虑到 LEO 卫星的高速运动导致网络拓扑具有高度时变性，本文采用离散时间快照技术[28]对网络进行建模。图 2 展示了某一特定时隙 $t$ 下的 LEO 卫星网络拓扑快照，其中实线表示连接同一轨道面内卫星的轨道内链路，该类链路拓扑结构稳定且持续连通；虚线表示连接相邻轨道面卫星的轨道间链路，受卫星高速相对运动影响，该类链路呈现出动态时变或间歇性连通的特征。对于任意时刻 $t$ 下的静止快照，本文采用 Dijkstra 算法搜索任意两颗 LEO 卫星节点 $m$ 和 $m'$ 之间的最短通信路径。通过遍历全网节点，生成 $M$ 颗卫星在时隙 $t$ 的全网跳数矩阵 $\boldsymbol{H}^t=[h^t_{m,m'}]_{M\times M}$，其中 $h^t_{m,m'}$ 表示时隙 $t$ 中卫星节点 $m$ 到 $m'$ 的最小中继跳数，用于后续计算星间多跳传输的链路开销。

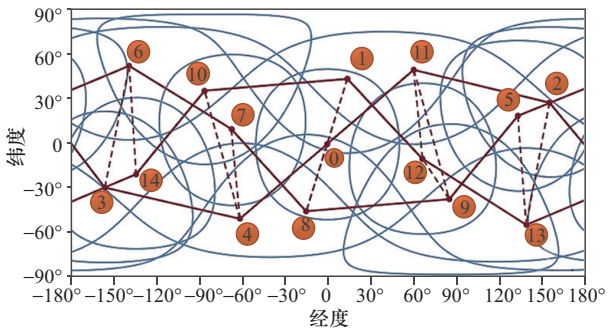



图 2　某一时隙下的 LEO 卫星网络拓扑快照

### 2.1.2　通信链路传输

本节具体介绍 P-SDTO 架构中的通信链路。星地接入链路用于将任务从地面孪生终端上传到 LEO

接入卫星。本文定义$Mm_{\text{acc}} \in \mathcal{M}$为当前时隙内地面孪生终端接入的LEO卫星节点，其星地上行链路传输速率$R_{\text{up}}$为：

$$R_{\text{up}} = B_{\text{up}} \text{lb}\left(1 + \frac{P_{\text{tx}}\beta_0}{\sigma^2 d_{\text{acc}}^2}\right) \tag{1}$$

其中，$B_{\text{up}}$为上行链路带宽，$P_{\text{tx}}$为地面孪生终端发射功率，$\beta_0$为信道增益参考值，$\sigma^2$为噪声功率，$d_{\text{acc}}$为地面与接入卫星$m_{\text{acc}}$间的距离。

星间链路（inter-satellite link, ISL）用于LEO卫星间的协同通信，假设数据在任意两颗建立连接的LEO卫星之间进行传输，则LEO卫星之间的传输速率$R_{\text{ISL}}$为：

$$R_{\text{ISL}} = B_{\text{ISL}} \text{lb}\left(1 + \frac{P_{\text{sat}}\beta_0}{\sigma^2 d_{\text{L2L}}^2}\right) \tag{2}$$

其中，$B_{\text{ISL}}$为星间链路的信道带宽，$P_{\text{sat}}$为LEO卫星的星间发射功率，$d_{\text{L2L}}$为LEO卫星之间的距离。

LEO-GEO中继链路用于实现LEO卫星向GEO卫星传输数据。本文定义$m_{\text{geo}}$为系统中的GEO卫星节点，则LEO与GEO卫星之间的传输速率$R_{\text{L2G}}$为：

$$R_{\text{L2G}} = B_{\text{L2G}} \text{lb}\left(1 + \frac{P_{\text{sat}}\beta_0}{\sigma^2 d_{\text{L2G}}^2}\right) \tag{3}$$

其中，$B_{\text{L2G}}$为LEO-GEO中继链路的信道带宽，$d_{\text{L2G}}$为LEO与GEO卫星之间的距离。

星地馈电链路用于连接信关卫星与地面云数据中心。本文定义$m_{\text{gw}} \in \mathcal{M}$为当前时隙内负责连接地面的信关节点，则下行馈电传输速率$R_{\text{down}}$为：

$$R_{\text{doum}} = B_{\text{doum}} \text{lb}\left(1 + \frac{P_{\text{gw}}\beta_0}{\sigma^2 d_{\text{gw}}^2}\right) \tag{4}$$

其中，$B_{\text{down}}$为星地馈电链路的信道带宽，$P_{\text{gw}}$为信关卫星的下行发射功率，其大小为LEO卫星的发射功率，$d_{\text{gw}}$为信关卫星与地面云数据中心的距离，其大小等于地面到LEO接入卫星的距离。

### 2.2 数字孪生系统任务模型

本节主要对数字孪生系统中的任务模型及协同卸载决策机制进行形式化定义。

#### 2.2.1 基于DAG的依赖任务建模

地面终端共生成$I$个任务，表示为$\mathcal{I}=\{1,\cdots,I\}$，每个任务包含两个属性，$D_i$和$\rho_i^C$分别表示数据大小和计算强度。不同于传统的独立任务，本文将地面孪生终端生成的复杂应用建模为具有严格拓扑约束的DAG，表示为$\mathcal{G}=(\mathcal{V},\mathcal{E})$，其中顶点集$\mathcal{V}=\mathcal{I}$表示任务集合，依赖关系$\mathcal{E}\subseteq\mathcal{I}\times\mathcal{I}$表示任务间的执行顺序约束。若$(i,j)\in\mathcal{E}$，则任务$i$是任务$j$的直接前驱。本文定义前驱任务集合$\textbf{pred}(j)=\{i\in\mathcal{I}|(i,j)\in\mathcal{E}\}$。任务$j$只有在$\textbf{pred}(j)$中所有前驱任务完成并将其输出数据传输到位后，才进入可调度状态。

#### 2.2.2 协同卸载决策机制

系统需为DAG中的每个子任务$i$选择一个最优的计算节点$n$进行处理，其中$n\in\mathcal{N}$。本文引入二元卸载决策变量$x_{i,n}\in\{0,1\}$，当$x_{i,n}=1$时，表示任务$i$被调度至最优计算节点$n$执行，否则为0。

### 2.3 信息安全模型

P-SDTO架构中的星间及星地通信依赖开放的无线信道，卸载任务面临被通信范围内恶意卫星窃听或篡改的风险。为保护数字孪生系统卸载任务的隐私，本文建立了相应的信息安全模型。

本文选用AES算法作为加密机制，任务$i$的数据将被划分为固定大小的数据块，并采用统一的密钥对这些数据块进行加密。

定义安全等级集合$\mathcal{L}=\{l_1,l_2,l_3\}$，分别对应低、中、高3种安全等级，其中$l_1<l_2<l_3$。对于任务$i$，其决策包含一个安全等级选择$k_i\in\mathcal{K}$。AES算法使用的密钥长度可表示为$L_i^{\text{key}}=l_{k_i}$。随着安全等级$k_i$的提升，密钥长度$l_{k_i}$增长，加密所需时延也会增加。

攻击能力上限参数$L_{\max}^{\text{att}}$表示当前环境下攻击者可能破解的最大密钥长度边界。对于选定的密钥长度$L_i^{\text{key}}$，其被成功破解的相对风险系数$P_i^{\text{risk}}$可表示为：

$$P_i^{\text{risk}} = \frac{L_{\max}^{\text{att}} - L_i^{\text{key}}}{L_{\max}^{\text{sys}} - L_{\min}^{\text{sys}}} \tag{5}$$

其中，$L_{\min}^{\text{sys}}$为系统支持的最小密钥长度，$L_{\max}^{\text{sys}}$为系统支持的最大密钥长度。

为体现不同的真实安全态势，本文通过不同数量的恶意卫星来模拟并评估任务被攻破的情况。假设系统通信范围内潜在恶意卫星的平均数量为$\xi$，其实际数量$X$服从参数为$\xi$的泊松分布，即$X\sim\text{Poisson}(\xi)$。任务的数据幸存概率$P_i^{\text{surv}}$定义为在所有$X$个恶意卫星的联合攻击下仍未泄露的概率，

即 $P_i^{\text{surv}}=(1-P_i^{\text{risk}})^X$。本文采用蒙特卡罗方法，在仿真中判定任务的最终状态。二值变量 $A_i$ 表示任务 $i$ 被攻破的结果，判定逻辑为：

$$A_i=\begin{cases}1, & \varpi > P_i^{\text{surv}}\\ 0, & \varpi \leqslant P_i^{\text{surv}}\end{cases} \tag{6}$$

其中，$\varpi$ 为服从[0,1]均匀分布的随机数。

本文将安全风险映射为与密钥长度相关的线性归一化概率模型[29]，采用泊松点过程对恶意卫星进行建模能够有效规避确定性轨道建模带来的复杂性[30]。最后，本文的研究重点聚焦于协同架构下的计算资源分配与计算卸载，参考边缘计算跨层优化的经典范式[27]，将底层安全协议的细节简化为系统可调度的计算开销与传输时延。

### 2.4　时延模型

基于不同的任务卸载决策，任务可能在接入 LEO 卫星、其他 LEO 卫星、GEO 卫星或地面云服务器上进行处理，具体如下。

#### 2.4.1　加密时延

为防止恶意卫星窃听或篡改，需要对卸载任务进行加密，加密时延取决于选择的安全等级。

根据 2.3 节的信息安全模型，任务 $i$ 的安全等级 $k_i$ 对应着不同的密钥长度 $L_i^{\text{key}}$。本文定义加密计算密度 $\rho_i^{\text{en}}$ 为完成 1 bit 数据加密所需的平均 CPU 时钟周期数，$\rho_i^{\text{en}}$ 的值与任务 $i$ 选择的安全等级 $k_i$ 呈正相关。

对于数据量为 $D_i$ 的任务 $i$，若其在计算能力为 $f_{\text{en}}$ 的地面孪生终端上执行加密，则产生的加密时延 $T_i^{\text{enc}}$ 可表示为：

$$T_i^{\text{enc}}=\frac{D_i\rho_i^{\text{en}}}{f_{\text{en}}} \tag{7}$$

其中，$f_{\text{en}}$ 为地面孪生终端加密计算频率。

#### 2.4.2　传输时延

为了计算任务 $i$ 的总传输时延 $T_i^{\text{trans}}$，本文将传输过程分解为上行传输时延和中继传输时延两部分。由于计算任务的输出数据量通常远小于输入数据量，本文忽略计算结果的回传时延[5]。

数据首先需要从地面孪生终端上传至当前的接入卫星 $m_{\text{acc}}$，根据前文定义的星地链路模型，此时的上行传输时延可表示为：

$$t_i^{\text{up}}=\frac{D_i}{R_{\text{up}}} \tag{8}$$

当数据到达接入卫星 $m_{\text{acc}}$ 后，根据任务卸载决策 $n$，后续的中继传输时延分为以下 3 种情况讨论。

（1）当任务 $i$ 的目标节点 $n$ 位于 LEO 卫星层时，若任务在接入卫星处理，则无中继传输时延；若任务卸载至邻近 LEO 卫星，数据需经星间链路传输，则：

$$T_i^{\text{relay}}=\frac{h^t_{m_{\text{acc}},n}D_i}{R_{\text{ISL}}} \tag{9}$$

其中，$h^t_{m_{\text{acc}},n}$ 为接入卫星到目标卫星的最短路径跳数，$R_{\text{ISL}}$ 为星间链路传输速率。

（2）当任务 $i$ 被卸载至 GEO 卫星时，则：

$$T_i^{\text{relay}}=\frac{D_i}{R_{\text{L2G}}} \tag{10}$$

其中，$R_{\text{L2G}}$ 为 LEO-GEO 激光链路的传输速率。

（3）当任务 $i$ 上传至地面云中心时，数据需先经多跳 ISL 到达信关卫星 $m_{\text{gw}}$，再通过馈电链路下传，则：

$$T_i^{\text{relay}}=\frac{D_i h^t_{m_{\text{acc}},m_{\text{gw}}}}{R_{\text{ISL}}}+\frac{D_i}{R_{\text{feed}}} \tag{11}$$

其中，$R_{\text{feed}}$ 为下行馈电链路的传输速率。

综上所述，任务 $i$ 的总传输时延 $T_i^{\text{trans}}$ 可表示为：

$$T_i^{\text{trans}}=T_i^{\text{up}}+T_i^{\text{relay}} \tag{12}$$

#### 2.4.3　传播时延

在 P-SDTO 系统中，传播时延由电磁波在物理链路中的飞行时间决定，设光速为 $c$。与传输时延类似，本文首先分析任务上传阶段的传播时延，随后讨论任务间依赖数据流转的传播时延。

对于任务上传阶段的传播时延，所有任务需由地面孪生终端经无线链路上传至接入卫星 $m_{\text{acc}}$。无论任务卸载决策如何，该阶段的传播时延取决于地面孪生终端与接入卫星的实时距离 $d_{\text{acc}}$，任务上传阶段的传播时延为：

$$T_i^{\text{prop_acc}}=\frac{d_{\text{acc}}}{c} \tag{13}$$

当数据到达接入卫星后，根据目标节点 $n$ 的位置，后续的中继传播时延 $T^{\text{prop_relay}}$ 分为以下 3 种情况讨论。

（1）当目标节点 $n$ 位于 LEO 卫星层时，若在本地执行，传播时延为 0；若卸载至邻近卫星，需经过多跳星间链路传播，则：

$$T_i^{\text{prop_relay}} = \frac{d_{\text{ISL}} h^t_{m_{\text{acc}},n}}{c} \tag{14}$$

其中，$d_{\text{ISL}}$为相邻LEO卫星间的链路距离。

（2）当任务被卸载至GEO卫星时，数据经LEO-GEO激光链路直接传输，则：

$$T_i^{\text{prop_relay}} = \frac{d_{\text{L2G}}}{c} \tag{15}$$

其中，$d_{\text{L2G}}$为LEO与GEO卫星间的中继链路距离。

（3）当任务卸载至地面云中心时，数据需先传输至信关卫星$m_{\text{gw}}$再下传至地面，则：

$$T_i^{\text{prop_relay}} = \frac{d_{\text{ISL}} h^t_{m_{\text{acc}},m_{\text{gw}}}}{c} + \frac{d_{\text{feed}}}{c} \tag{16}$$

其中，$d_{\text{feed}}$为信关卫星至地面站的馈电链路距离。

综上所述，任务$i$上传阶段的总传播时延$T_i^{\text{prop_up}}$为：

$$T_i^{\text{prop_up}} = T_i^{\text{prop_acc}} + T_i^{\text{prop_relay}} \tag{17}$$

对于依赖数据传播时延$T_{i,j}^{\text{del}}$，当任务$j$依赖前置任务$i$的输出结果，数据传播时延$T_{i,j}^{\text{del}}$可表示为：

$$T_{i,j}^{\text{del}} = \begin{cases} \dfrac{h_{m,n} d_{\text{ISL}}}{c}, \{m,n\} \subseteq \mathcal{M} \\ \dfrac{d_{\text{L2G}}}{c}, \{m,n\} \cap \mathcal{M} \neq \varnothing \wedge n_{\text{geo}} \in \{m,n\} \\ \dfrac{d_{\text{L2L}} h_{s,m_{\text{gw}}} + d_{\text{L2S}}}{c}, \{m,n\} \cap \mathcal{M} \neq \varnothing \wedge n_{\text{gnd}} \in \{m,n\} \\ \dfrac{d_{\text{L2G}} + d_{\text{ISL}} h^t_{m_{\text{geo}},m_{\text{gw}}} + d_{\text{feed}}}{c}, \{m,n\} = \{n_{\text{geo}}, n_{\text{gnd}}\} \end{cases} \tag{18}$$

其中，$s$表示链路中的LEO侧锚点节点，即$\{s\} = \{m,n\} \cap \boldsymbol{\mathcal{M}}$。符号∧表示逻辑“且”。上述4项条件分别对应节点对均为LEO集合子集、节点对包含LEO卫星节点且包含GEO卫星节点、节点对包含LEO卫星节点且包含地面节点和节点对由GEO卫星与地面节点构成4种情况。

#### 2.4.4 计算时延

当任务$i$被调度至计算节点$n$执行时，其总计算时延$T_i^{\text{comp}}$取决于任务的计算复杂度与节点的处理能力。任务$i$的计算工作量为$W_i = D_i \rho_i^C$，总计算时延$T_i^{\text{comp}}$分以下3种情况讨论。

（1）当任务被调度至LEO卫星网络中的任意节点$n \in \mathcal{M}$执行时，则有：

$$T_i^{\text{comp}} = \frac{W_i}{f_{\text{LEO}}} \tag{19}$$

其中，$f_{\text{LEO}}$为LEO卫星的计算频率。

（2）当任务被卸载至GEO卫星$n \in \{n_{\text{geo}}\}$执行时，则有：

$$T_i^{\text{comp}} = \frac{W_i}{f_{\text{geo}}} \tag{20}$$

其中，$f_{\text{geo}}$为GEO卫星的计算频率。

（3）当任务被回传至地面云服务器$n \in \{n_{\text{gnd}}\}$执行时，则有：

$$T_i^{\text{comp}} = \frac{W_i}{f_{\text{gnd}}} \tag{21}$$

其中，$f_{\text{gnd}}$为地面云服务器的计算频率。

#### 2.4.5 任务就绪时刻

在系统模型中，每个任务$i$的执行需要满足双重数据约束：一方面地面终端需加密并上传最新的任务数据，所需时间为$T_i^{\text{pre}} = T_i^{\text{enc}} + T_i^{\text{trans_up}} + T_i^{\text{prop_up}}$；另一方面需等待所有前置任务$\mathbf{pred}(i)$的计算结果到达目标节点。因此，任务$i$的就绪时刻$\tau_i^{\text{ready}}$取决于上传路径与前置依赖路径中较晚到达的时刻，任务就绪时刻$\tau_i^{\text{ready}}$可表示为：

$$\tau_i^{\text{ready}} = \tau_j^{\text{finish}} + \max\left\{T_i^{\text{pre}}, \max_{j \in \mathbf{pred}(i)} \{T_j^{\text{del}}\}\right\} \tag{22}$$

任务$i$的完成时刻$\tau_i^{\text{finish}}$由任务就绪时刻与节点的计算处理时延共同决定，因此$\tau_i^{\text{finish}} = \tau_i^{\text{ready}} + T_i^{\text{comp}}$。

### 2.5 能耗模型

在SEC系统中，星上能量受限于太阳能板供电与电池容量，是系统调度的关键约束资源。任务$i$的总能耗$E_i$定义为网络传输能耗与计算处理能耗之和，总能耗$E_i$分以下3种情况讨论。

（1）当卸载至LEO卫星时，总能耗包括从接入卫星$m_{\text{acc}}$到目标卫星$n$的星间链路传输能耗，以及目标卫星的计算能耗。

$$E_i = h_{m_{\text{acc}},n} P_{\text{sat}} \frac{D_i}{R_{\text{ISL}}} + \kappa \left(f_{\text{LEO}}\right)^2 W_i \tag{23}$$

其中，$P_{\text{sat}}$为星间链路的发射功率，若任务在接入卫星执行，则跳数$h = 0$，传输能耗为0。

（2）当卸载至GEO卫星时，总能耗由LEO-GEO链路的传输代价与GEO卫星上的计算代价构

成，则：

$$E_i = P_{\mathrm{L2G}} \frac{D_i}{R_{\mathrm{L2G}}} + \kappa \left( f_{\mathrm{LEO}} \right)^2 W_i \tag{24}$$

其中，$P_{\mathrm{L2G}}$为LEO卫星向GEO卫星发送数据时的发射功率。

（3）当卸载至地面云服务器时，本文只考虑维持卫星正常运行的能耗，因此只需考虑LEO卫星数据互传与下行传输的能耗，则：

$$E_i = h_{m_{\mathrm{acc}}, m_{\mathrm{gw}}} P_{\mathrm{sat}} \frac{D_i}{R_{\mathrm{L2L}}} + P_{\mathrm{gw}} \frac{D_i}{R_{\mathrm{down}}} \tag{25}$$

其中，$P_{\mathrm{gw}}$为下行馈电链路的发射功率。

### 2.6　优化目标

本节将P-SDTO系统中的任务卸载决策与安全等级配置问题构建为一个多目标联合优化问题。基于前文定义的离散时隙模型$T = \{1, \cdots, T\}$，本文的目标是寻找最优的任务卸载决策集合$\{x_{i,n}\}$与安全等级配置集合$\{k_i\}$，以在满足严格的时序与同步约束下，协同优化系统的时效性、能效性与安全性。

首先，针对系统总时延，由于任务间存在依赖关系，系统的性能取决于所有任务中最晚完成的时刻。将系统总时延记为$T_{\mathrm{sys}}$。基于前文定义的任务完成时刻$\tau_i^{\mathrm{finish}}$，系统总时延$T_{\mathrm{sys}}$可定义为：

$$T_{\mathrm{sys}} = \max_{i \in \boldsymbol{\mathcal{V}}} \left( \tau_i^{\mathrm{finish}} \right) \tag{26}$$

其次，对于系统总能耗$E_{\mathrm{sys}}$，本文仅考虑卫星维持运行与工作的能耗。总能耗$E_{\mathrm{sys}}$由各计算节点的计算处理能耗与节点间链路的传输能耗组成。总能耗$E_{\mathrm{sys}}$可定义为：

$$E_{\mathrm{sys}} = \sum_{i \in \boldsymbol{\mathcal{V}}} \sum_{n \in \boldsymbol{\mathcal{N}}} x_{i,n} \left( E_i^{\mathrm{comp}} + \sum_{i \in \mathbf{pred}(j)} E_i^{\mathrm{trans}} \right) \tag{27}$$

最后，为了量化系统的安全风险，本文定义累计被攻破次数为$A_{\mathrm{sys}}$。基于前文定义的攻击判定二值变量$A_i \in \{0,1\}$，若$A_i = 1$则表示任务$i$被攻破，被攻破的任务总数$A_{\mathrm{sys}}$定义为：

$$A_{\mathrm{sys}} = \sum_{i \in \boldsymbol{\mathcal{V}}} A_i \tag{28}$$

综上所述，本文将该多目标联合优化问题P1形式化为最小化总加权成本$\mathcal{J}$，即：

$$\begin{aligned}
&\mathrm{P1} \quad \min_{\{x_{i,n}\}, \{k_i\}} \quad = \omega_{\mathrm{L}} T_{\mathrm{sys}} + \omega_{\mathrm{E}} E_{\mathrm{sys}} + \omega_{\mathrm{A}} A_{\mathrm{sys}} \\
&\text{s.t.} \sum_{n \in \boldsymbol{\mathcal{N}}} x_{i,n} = 1, \quad \forall i \in \boldsymbol{\mathcal{V}} \\
&k_i \in \{1,2,3\}, \quad \forall i \in \boldsymbol{\mathcal{V}} \\
&\tau_i^{\mathrm{ready}} \geqslant \tau_j^{\mathrm{finish}} + T_i^{\mathrm{pre}}, \quad \forall i \in \boldsymbol{\mathcal{V}} \\
&\tau_i^{\mathrm{ready}} \geqslant \tau_j^{\mathrm{finish}} + T_i^{\mathrm{del}}, \quad \forall i \in \boldsymbol{\mathcal{V}}, \forall j \in \mathbf{pred}(i) \\
&x_{i,n} \in \{0,1\}, \quad A_i \in \{0,1\}
\end{aligned} \tag{29}$$

其中，$\omega_{\mathrm{T}}$为时延权重，$\omega_{\mathrm{E}}$为能耗权重，$\omega_{\mathrm{A}}$为隐私保护权重。为避免引入冗余的计算开销，仿照文献[26]，本文在构建总加权成本$\mathcal{J}$与后续强化学习奖励函数$r_t$时未采用显式的归一化函数，而是将量纲转换机制与多目标联合优化偏好隐式融于权重系数$\omega_{\mathrm{T}}$, $\omega_{\mathrm{E}}$,$\omega_{\mathrm{A}}$中。

上述构建的多目标联合优化问题本质上是一个混合整数非线性规划问题，已被证明是NP难的。由于隐私保护与系统开销难以兼顾，且DAG间的时序依赖约束极大地限制了可行解空间，传统方法难以在卫星网络的高动态时变环境下实现实时求解。第3节提出了PGDRL算法，旨在动态环境下寻找满足依赖约束且快速响应的任务卸载策略。

## 3　PGDRL算法

针对多维资源约束与复杂任务依赖导致传统算法求解效率低下的难题，本节考虑数字孪生系统任务的物理属性与任务间依赖的拓扑特征，提出了PGDRL算法，旨在动态环境下联合优化系统时延、卫星能耗及任务被攻破数量。图3展示了PGDRL算法框架，该框架首先通过优先级预处理模块与GAT模块对DAG任务进行排序与拓扑特征提取，构建高维状态输入，然后利用Actor-Critic网络输出兼顾时延、能耗与安全性的最优卸载策略。

### 3.1　优先级排序机制

针对任务调度复杂度高、依赖约束难以满足的问题，设计优先级排序机制以满足任务依赖约束并降低调度复杂度。

本文定义单任务预期成本为$\mathrm{Cost}_i$，考虑到在优先级排序阶段尚未确定任务的具体卸载决策，本文采用网络中所有可用计算节点的平均处理能力进行估算。任务$i$的预期成本$\mathrm{Cost}_i$表示为：

$$\mathrm{Cost}_i = \omega_{\mathrm{T}} \bar{T}_i + \omega_{\mathrm{E}} \bar{E}_i + \omega_{\mathrm{A}} \bar{A}_i \tag{30}$$

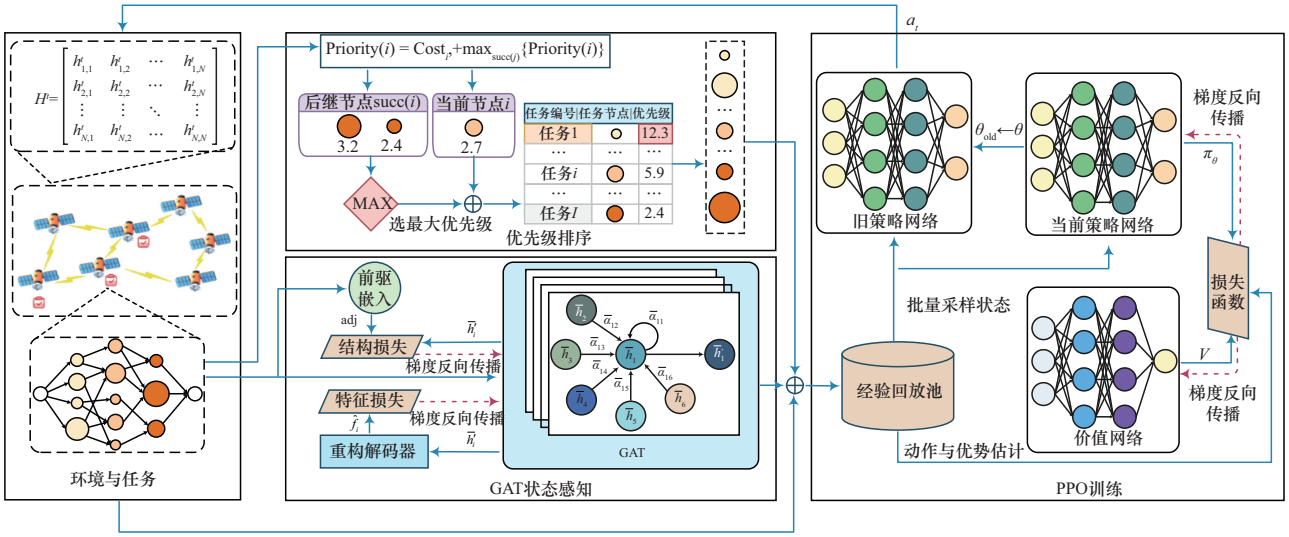



图3　PGDRL 算法框架

其中，$\omega_{\mathrm{T}}$、$\omega_{\mathrm{E}}$和$\omega_{\mathrm{A}}$是与系统总目标函数一致的全局权重系数，$\bar{T}_i$、$\bar{E}_i$与$\bar{A}_i$分别表示任务$i$在全网范围内的平均时延、平均能耗与基于平均恶意节点密度的被攻击概率。依照$\mathrm{Cost}_i$对任务序列$S$进行排列。

基于单任务预期成本，任务$i$的优先级$\mathrm{Priority}_i$反映了任务距离结束所需的累积最小开销。对于DAG中的出口任务，其优先级仅取决于自身成本，即$\mathrm{Priority}_i = \mathrm{Cost}_i$。对于DAG中的非出口任务，优先级$\mathrm{Priority}_i$表示为：

$$\mathrm{Priority}_i = \mathrm{Cost}_i + \max_{j \in \mathbf{succ}(i)}\left\{\mathrm{Priority}_j\right\} \quad (31)$$

在完成全图节点的递归计算后，系统依据各任务的优先级$\mathrm{Priority}_i$对任务集$\boldsymbol{\mathcal{V}}$进行降序排列，生成一个一维线性任务序列$\boldsymbol{\mathcal{Q}}_{\mathrm{task}}$，智能体严格按照线性任务序列$\boldsymbol{\mathcal{Q}}_{\mathrm{task}}$依次为任务分配计算节点与安全等级。参考现有动态网络中依赖任务调度的文献[21]，在实际决策时，智能体会根据每一步实时的网络状态做出动作，从而动态补偿并纠正了平均值近似带来的初始偏差。具体算法如算法1所示。

**算法1**　任务序列生成

**输入**　$\boldsymbol{\mathcal{G}} = (\boldsymbol{\mathcal{V}}, \boldsymbol{\mathcal{E}}), \omega_{\mathrm{L}}, \omega_{\mathrm{E}}, \omega_{\mathrm{A}}$

**输出**　$\boldsymbol{\mathcal{Q}}_{\mathrm{task}}$

1) 遍历任务集$\boldsymbol{\mathcal{V}}$，依式(30)计算预期成本$\mathrm{Cost}_i$

2) 对图$\boldsymbol{\mathcal{G}}$进行拓扑排序

3)　for 任务$i \in \mathrm{Reverse}\{\boldsymbol{\mathcal{S}}\}$ do

4)　　依式(31)计算优先级$\mathrm{Priority}(i)$

5)　end for

6) $\boldsymbol{\mathcal{Q}}_{\mathrm{task}}$←按优先级$\mathrm{Priority}(i)$降序排列$V$

7) return $\boldsymbol{\mathcal{Q}}_{\mathrm{task}}$

### 3.2　基于GAT的状态感知算法

为了使神经网络能够感知DAG的异构属性，本文首先定义了任务节点的原始特征。对于DAG中的任意任务$i$，构建其初始特征向量$\boldsymbol{h}_i^{(0)} = [D_i, \rho_i, \mathrm{Deg}_i^{\mathrm{in}}, \mathrm{Deg}_i^{\mathrm{out}}, I_i^{\mathrm{sche}}]$，其中$D_i$表示数据大小，$\rho_i$表示完成任务所需的CPU周期数，$\mathrm{Deg}_i^{\mathrm{in}}$表示入度，$\mathrm{Deg}_i^{\mathrm{out}}$表示出度，引入度数特征的物理意义在于，入度为0的节点为入口任务，出度为0的节点为出口任务。$I_i^{\mathrm{sche}} \in \{0,1\}$为二值标识位，用于指示该任务当前是否已被调度。

首先，通过可学习的共享权重矩阵$\boldsymbol{W}$对节点特征进行线性变换，并计算注意力系数$e_{i,j}^{(l)}$。

$$e_{i,j}^{(l)} = \mathrm{LeakyReLU}\left(\boldsymbol{a}^{\mathrm{T}}[\boldsymbol{W}^{(l)}\boldsymbol{h}_i^{(l)} \| \boldsymbol{W}^{(l)}\boldsymbol{h}_j^{(l)}]\right) \quad (32)$$

其中，$\boldsymbol{a}$为单层前馈神经网络的权重向量，$[\cdot \| \cdot]$为向量拼接操作，LeakyReLU为非线性激活函数。为了使系数具有概率含义，本文利用Softmax函数对其进行归一化，得到最终的注意力权重$\alpha_{i,j}^{(l)}$。

$$\alpha_{i,j}^{(l)} = \frac{\exp\left(e_{i,j}^{(l)}\right)}{\sum\limits_{k \in \mathbf{pred}(i)} \exp\left(e_{i,k}^{(l)}\right)} \quad (33)$$

归一化后的注意力权重$\alpha_{i,j}^{(l)}$动态衡量了前驱任务$j$对当前任务$i$的重要程度。最终，任务$i$的高维嵌入向量$\boldsymbol{h}_i^{(l+1)}$通过加权聚合其前驱特征进行更

新，具体为：

$$\boldsymbol{h}_i^{(l+1)}=\sigma\left(\sum_{j\in\mathbf{pred}(i)}\alpha_{i,j}^{(l)}\boldsymbol{W}^{(l)}\boldsymbol{h}_j^{(l)}\right) \tag{34}$$

最终嵌入向量 $\boldsymbol{h}_i^{(L)}$ 不仅包含任务的物理属性，还融合了其依赖链上的上下文信息。

本文构建了由特征与结构损失共同构成的联合损失函数。特征损失 $\mathcal{L}_{\text{feat}}$ 表示为：

$$\mathcal{L}_{\text{feat}}=\frac{1}{|\mathcal{V}|}\sum_{i\in\mathcal{V}}\|\boldsymbol{h}_i^{(L)}-\boldsymbol{h}_i^{(L)'}\|^2 \tag{35}$$

结构损失 $L_{\text{struct}}$ 表示为：

$$\mathcal{L}_{\text{struct}}=-\sum_{i\in\mathcal{V}}\sum_{j\in\mathbf{pred}(i)}\ln\left(\frac{1}{1+\exp(-\boldsymbol{h}_i^{(L)}\cdot\boldsymbol{h}_j^{(L)})}\right) \tag{36}$$

综上所述，GAT 模块的总训练损失定义为 $\mathcal{L}_{\text{GAT}}=\alpha_1\mathcal{L}_{\text{feat}}+\alpha_2\mathcal{L}_{\text{struc}}$。具体算法如算法2所示。

**算法2**　基于GAT的状态感知

**输入**　邻接矩阵 **adj**，原始特征 $\boldsymbol{h}$，预训练轮数 $E_{\text{pre}}$，GAT Encoder网络层数 $L_{\text{enc}}$，GAT网络层数损失权重 $\alpha_1,\alpha_2$

**输出**　GAT Encoder网络参数 $\Theta_{\text{enc}}$

1) 初始化Encoder和Decoder网络参数
2) for epoch = 1 to $E_{\text{pre}}$ do
3)　$\boldsymbol{h}^{(0)}\leftarrow\boldsymbol{h}$
4)　for $l_{\text{enc}}$= 1 to $L_{\text{enc}}$ do
5)　　依式(33)计算注意力权重 $\alpha_{i,j}$
6)　　$\boldsymbol{h}^{(l)}\leftarrow$ GAT_Layer $(\boldsymbol{h}^{(l-1)},\mathbf{adj})$
7)　end for
8)　$\boldsymbol{h}^{(0)}=\boldsymbol{h}^{(L_{\text{enc}})}$
9)　for $l_{\text{dec}}$= 1 to $L_{\text{enc}}$ do
10)　$\tilde{\boldsymbol{h}}^{(k)}\leftarrow$ Layer $(\tilde{\boldsymbol{h}}^{(k-1)})$
11)　end for
12)　$\boldsymbol{h}'\leftarrow\tilde{\boldsymbol{h}}^{(L_{\text{dec}})}$
13)　依式(34)和式(35)计算 $\mathcal{L}_{\text{GAT}}$，并更新 $\Theta_{\text{enc}}$
14) end for
15) return $\Theta_{\text{enc}}$

### 3.3　马尔可夫决策过程

本节将优化问题建模为一个离散时间的马尔可夫决策过程。具体的状态空间、动作空间及奖励函数定义如下。

（1）状态空间：在第 $t$ 步时，针对优先级队列中的待调度任务 $i$，本文将系统状态 $s_t$ 定义为 $s_t=[\boldsymbol{h}_i^{(L)},D_i,\rho_i,\boldsymbol{H}_t,X]$。具体各分量定义如下：任务拓扑结构嵌入 $\boldsymbol{h}_i^{(L)}$ 为GAT模块的最终输出，$D_i$ 和 $\rho_i$ 分别为任务 $i$ 的大小与计算复杂度，$\boldsymbol{H}_t$ 为全网跳数矩阵，真安全态势作为先验知识输入，用当前通信覆盖范围内侦测到的恶意卫星数量 $X$ 体现[27]。针对节点资源状态，本文假设各节点的可用资源在调度时隙初刷新，以此实现调度与资源管理的解耦[10]。

（2）动作空间：鉴于任务执行顺序已由优先级预处理模块确定，智能体在第 $t$ 步的核心任务是为任务 $i$ 匹配最优的资源与防护，则动作空间为 $\boldsymbol{a}_t=\{n,k\}$，分别确定节点位置和加密等级。

（3）奖励函数：奖励函数 $r_t$ 定义为系统总时延、网络总能耗与任务被攻破数量的加权和的负值。其数学表达式为：

$$r_t=-\mathcal{J} \tag{37}$$

### 3.4　深度强化学习训练框架

为了在复杂的动作空间中实现稳定的策略更新，本文采用近端策略优化（proximal policy optimization, PPO）算法，结合任务优先级与GAT模块优化目标。

利用广义优势估计来计算第 $t$ 步决策的优势函数 $\hat{A}_t$。具体计算式为：

$$\delta_t=r_t+\gamma V_{\Phi}(S_{t+1})-V_{\Phi}(S_t) \tag{38}$$

$$\hat{A}_t=\sum_{k=0}^{T-t-1}(\gamma\lambda)^k\delta_{t+k} \tag{39}$$

其中，$\gamma$ 为折扣因子，$\lambda$ 为GAE平滑系数，$V_{\Phi}$ 为Critic网络的价值估计。

GAE通过权衡偏差与方差，量化了当前动作 $a_t$ 相对于平均状态价值的优势程度。其次，为了衡量新策略 $\pi_\theta$ 与旧策略 $\pi_{\theta_{\text{old}}}$ 的差异，定义概率比率 $r_t(\theta)$ 为：

$$r_t(\theta)=\frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{\text{old}}}(a_t|s_t)} \tag{40}$$

其中，$\pi_\theta(a_t|s_t)=\pi_\theta^{\text{node}}(n|s_t)\cdot\pi_\theta^{\text{sec}}(k|s_t)$ 为双头输出动作的联合概率分布。

PPO算法的核心在于其策略截断损失，定义为：

$$L_t^{\text{CLIP}}(\theta)=\min\left(r_t(\theta)\hat{A}_t,\text{clip}(r_t(\theta),1-\varepsilon,1+\varepsilon)\hat{A}_t\right) \tag{41}$$

式(41)通过超参数 $\varepsilon$ 将概率比率限制在 $[1-\varepsilon,1+$

$\varepsilon]$内。当优势函数$\hat{A}_t>0$时，该机制增加优质动作的概率但防止过激更新；当$\hat{A}_t<0$时，则减少劣质动作的概率。

为了实现端到端的联合优化，本文将总损失函数$L(\theta)$构建为策略损失、价值误差与熵正则项的加权组合。

$$L_{\text{total}}(\theta)=\hat{\mathbb{E}}_t\left[-L_t^{\text{CLIP}}(\theta)+c_1L_t^{\text{VF}}(\theta)-c_2S[\pi_\theta](s_t)\right] \tag{42}$$

其中，$c_1$,$c_2$为权重系数。具体算法如算法3所示。

**算法3**　PGDRL

**输入**　预训练参数$\Theta_{\text{Enc}}$，原始特征$h$，邻接矩阵**adj**，任务队列$\mathcal{Q}_{\text{task}}$，最大回合数$E_{\max}$，更新轮数$E_{\text{pre}}$

**输出**　Actor网络参数$\theta$，Critic网络参数$\Phi$

1) 初始化Actor网络参数$\theta$和Critic网络参数$\Phi$
2) for episode = 1 to $E_{\max}$ do
3)　获取初始状态$S_0$
4)　for 任务$i\in\mathcal{Q}_{\text{task}}$ do
5)　　利用$\Theta_{\text{Enc}}$融合$\boldsymbol{h}$,**adj**并构建$\boldsymbol{h}_i^{(L)}$
6)　　更新状态$S_t$
7)　　依$\pi_\theta(S_t)$采样动作$\boldsymbol{a}_t$
8)　　执行动作$\boldsymbol{a}_t$，获得奖励$r_t$及下一状态$S_{t+1}$
9)　　将元组$(S_t,a_t,r_t,S_{t+1})$存入经验回放池$\mathcal{D}$
10) end for
11) 依式(39)计算优势估计$\hat{A}_t$
12) for epoch = 1 to $E_{\text{pre}}$ do
13)　依式(40)和式(41)计算$r_t(\theta)$和$L_t^{\text{CLIP}}$
14)　依式(42)计算总损失$L_{\text{total}}$
15)　更新Actor网络参数$\theta$和Critic网络参数$\Phi$
16) end for
17) end for
18) return $\theta$和$\Phi$

## 4　实验分析

本节将通过大量仿真实验来全面评估PGDRL算法在SEC数字孪生系统中的性能。具体而言，后续的实验分析将重点围绕以下4个研究问题（research question, RQ）展开探讨。

RQ1：时延、能耗及隐私保护的权重参数变化对系统总代价有何影响，不同任务规模下能否确定一组最优的权重配置以实现系统总加权成本的最小化？

RQ2：PGDRL算法相比其他对比算法，能否在多次实验下高效、稳定地实现收敛？

RQ3：相较于主流基准算法与消融变体，PGDRL算法能否在不同规模场景下实现系统总时延、总能耗及任务被攻破数量的最优协同优化？

RQ4：系统卸载决策的选择分布有何特征，展现出了怎样的自适应能力？

### 4.1　实验设置

仿真实验部署于配置了Intel Core i7-14700HX处理器、16 GB内存以及NVIDIA GeForce RTX 4060 GPU的计算平台上。考虑到强化学习算法对初始化的敏感性以及环境的动态性，所有性能评估均采用5个不同的全局随机种子进行多次独立测试，从而确保实验结果的统计显著性与可靠性。在后续的结果展示中，各指标均取自多次运行的统计均值，并在图表中以误差棒标明标准差，算法收敛曲线绘制置信区间阴影。在任务拓扑构建方面，实验选用了源自不同科学领域的真实工作流数据集[31]来模拟DAG结构，包括Montage、CyberShake、Epigenomics和Inspiral，依赖边分布、入口/出口比例等均按照工作流数据集确定。本文以经典的铱星星座架构为参考[32]，用Skyfield库计算LEO卫星的实时轨道位置，模拟了包含66颗LEO卫星的网络。同时，利用NetworkX库构建时变的网络拓扑图，生成系统模型所需的邻接矩阵与全网跳数矩阵[33]。在每一个离散时间隙中，可见LEO卫星集合大小动态维持在10~15颗之间。此外，为还原动态网络安全威胁态势，根据文献[30]，对空间分布服从泊松点过程的恶意卫星，在根据恶意卫星密度计算其数量后，在当前的可见LEO卫星通信范围内进行均匀离散采样。

在参数设置方面，为了验证算法在不同任务规模下的性能表现，本文分别设置了10~30任务的小规模场景和300~500任务的大规模场景。假设每个任务$i$的大小服从[2,3]MB区间内的均匀分布，任务的计算密度$\rho_i^{\text{C}}$ = 200 cycles/bit。在安全配置方面，加密计算密度为$c_{\text{enc}}$ = 20 cycles/bit，AES算法的密钥长度分别设置为$l_1$ = 128 bit、$l_2$ = 192 bit和$l_3$ = 256 bit[34]。根据文献[35]，所有仿真环境的参数设置如表2所示。

**表 2　　仿真环境参数设置**

| 参数 | 值 |
|---|---|
| LEO 卫星的计算频率 $f_{LEO}$/(Gcycle·s$^{-1}$) | 5 |
| GEO 卫星的计算频率 $f_{geo}$/(Gcycle·s$^{-1}$) | 50 |
| 地面云服务器的计算频率 $f_{gnd}$/(Gcycle·s$^{-1}$) | 100 |
| 地面孪生终端的加密频率 $f_{en}$/(Gcycle·s$^{-1}$) | 3 |
| 星地上行链路的传输速率 $R_{up}$/(Mbit·s$^{-1}$) | 250 |
| LEO 卫星之间的传输速率 $R_{ISL}$/(Gbit·s$^{-1}$) | 1.6 |
| LEO 卫星与 GEO 卫星之间的传输速率 $R_{L2G}$/(Gbit·s$^{-1}$) | 2 |
| 下行馈电链路的传输速率 $R_{feed}$/(Gbit·s$^{-1}$) | 1.8 |
| 地面与接入卫星之间的距离 $d_{acc}$/km | 784 |
| LEO 卫星之间的距离 $d_{L2L}$/km | 4 481 |
| LEO 卫星与 GEO 卫星之间的距离 $d_{L2G}$/km | 31 305 |
| 卫星的能量系数 $\kappa$/(J·(Hz$^3$·s)$^{-1}$) | $5\times10^{-27}$ |
| 星间发射功率 $P_{sat}$/W | 7 |
| LEO-GEO 的发射功率 $P_{L2G}$/W | 5 |
| 馈电下行链路的发射功率 $P_{gw}$/W | 8 |
| 光速 $c$/(m·s$^{-1}$) | $3\times10^8$ |

对于神经网络参数的设置，在网络结构与训练设置上，重构解码器采用 4 层多层感知机，隐层激活函数为 ReLU。GAT 的预训练轮次 $E_{pre}$ 为 500 轮，学习率 $N_G$ 为 $1\times10^{-3}$，联合损失的权重配比设定为 $\alpha_1 = 0.5, \alpha_2 = 0.5$。根据文献[36]，其他超参数设置如表 3 所示。

**表 3　　算法超参数设置**

| 参数 | 值 |
|---|---|
| 学习率 lr | $3\times10^{-4}$ |
| 网络层 $L_D$ | 4 |
| 神经元数量 $N_D$ | 512 |
| 折扣因子 $\gamma$ | 0.99 |
| GAE 参数 $\lambda$ | 0.95 |
| 裁剪系数 $\delta$ | 0.2 |
| 批量大小 $b$ | 256 |
| 最大回合数 $E_{max}$ | 5 000 |

## 4.2　对比算法

为全面评估 PGDRL 算法的性能，本文将系统时延、卫星能耗及任务被攻破数量作为评价指标，并与以下 4 种算法进行对比。

（1）PSO[37]：一种基于群智能的启发式搜索算法，通过模拟鸟群的觅食行为解决优化问题。其中，适应度函数定义为总成本，粒子群规模与最大迭代次数分别设为 50 和 1 000。

（2）TD3[38]：一种 DRL 算法，通过引入双 Critic 网络和时延更新机制，有效解决了 $Q$ 值高估问题，相关的参数设置同 PGDRL 算法。

（3）G-PPO：仅保留 GAT 状态感知模块的消融变体。去除了任务的优先级排序预处理机制，直接利用 GAT 提取拓扑特征并结合 PPO 算法进行调度决策。

（4）P-PPO：仅保留优先级排序机制的消融变体。去除了 GAT 模块，智能体仅依靠扁平化的状态输入并按照优先级序列进行调度决策。

## 4.3　实验结果与分析

### 4.3.1　RQ1 参数敏感性分析

为了寻找最合适的时延权重 $\omega_T$、能耗权重 $\omega_E$ 和隐私权重 $\omega_A$ 组合，本节通过实验分别在小规模和大规模场景开展了参数敏感性分析实验。图 4 展示了小规模场景下不同权重参数组合的总代价，图 5 展示了大规模场景下不同权重参数组合的总代价。

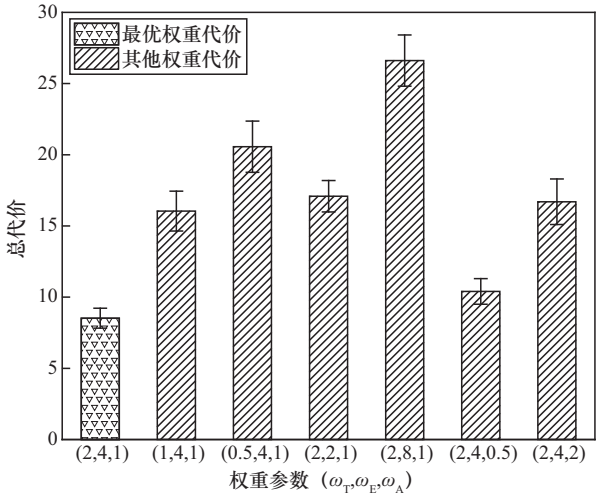



图 4　小规模场景下不同权重组合的系统总代价

从图 4 和图 5 中可以看出，当权重参数为(2, 4, 1)时，系统可以得到最小的总代价，且方差较小。在小规模场景下，该权重组合相比次优的权重组合降低了 16.78%，而大规模场景下相比次优的权重组合降低了 23.65%。

图4和图5的实验结果充分验证了(2, 4, 1)并非过拟合于单一参数设定的局部解，而是一组具备强鲁棒性与广泛适用性的最优配置，该配置将作为全部后续实验的权重配置。

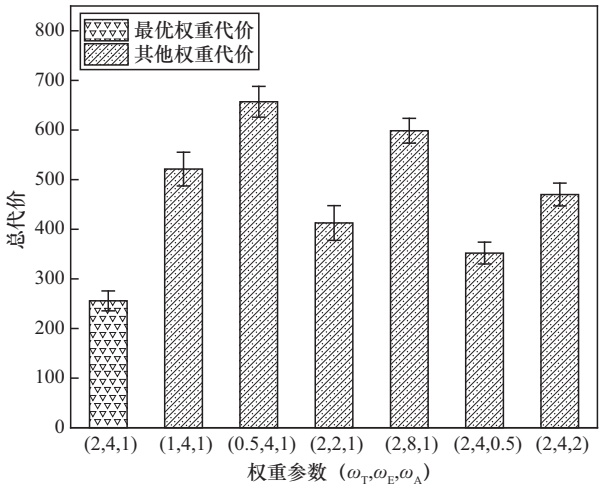



图5　大规模场景下不同权重组合的系统总代价

### 4.3.2　RQ2 算法收敛性分析

为了评估PGDRL算法能否高效、稳定地收敛至全局最优策略，本节将分析并讨论PGDRL算法与其他算法的奖励收敛性能对比差异。图6展示了PGDRL算法与其他算法相比平均奖励随训练回合数的收敛演变趋势以及奖励的置信区间。

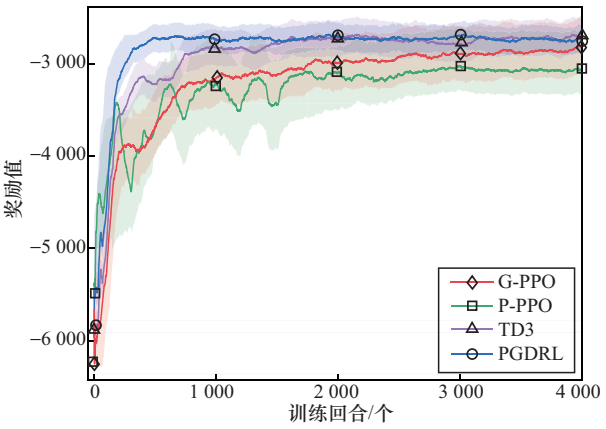



图6　不同算法训练过程中的平均奖励与置信区间

随着训练回合数增加，PGDRL算法在约750个回合的训练后率先达到稳定状态，并在后续迭代中维持着最高的平均奖励水平。相比之下，表现次优的TD3算法则需要约1 470个回合才能趋于收敛，这意味着PGDRL算法在收敛速度上提升48.24%。在与消融变体的对比中，G-PPO算法直到约1 800个回合才逐渐趋于稳定，PGDRL相较于G-PPO算法将收敛速度提升58.33%，而PGDRL相较于P-PPO算法在最终平均奖励上提升了32.1%。

图6表明，PGDRL算法可以高效、稳定地收敛至全局最优策略，这确保了卸载策略可以适应快速变化的环境。

### 4.3.3　RQ3 不同规模下的性能对比与消融实验分析

为了验证PGDRL算法在不同任务负载下的优势，并评估关键模块在不同任务规模下能否持续发挥效能，本节将详细分析不同任务规模下各算法的表现差异，并探讨核心机制的消融实验结果。图7对比了任务数从10~30个的小规模场景下各算法优化表现，图8评估了任务数从300~500个的大规模场景下各算法的表现。

图7(a)对比了小规模场景下各算法在系统时延优化上的表现。在10个任务规模下，PGDRL算法的系统时延相比其他算法保持最低，实验结果表明，相较于传统的启发式算法PSO，PGDRL算法的系统时延降低了11.67%。即使与主流的DRL算法TD3相比，PGDRL算法依然实现了6.54%的性能提升。随着任务规模扩展至30个，时延呈现上升趋势。PGDRL算法的平均时延相较于次优的TD3算法降低了4.52%。在消融实验中，PGDRL算法的平均时延较P-PPO与G-PPO分别降低了14.68%与10.07%。图7(b)对比了小规模场景下各算法在卫星能耗优化上的表现，卫星能耗随着任务数量增多而增大。在30个任务规模下，相比PSO算法，PGDRL在卫星能耗上降低了13.42%。与TD3算法相比，PGDRL将卫星能耗降低了3.42%。对比消融变体，PGDRL相较于P-PPO与G-PPO算法卫星能耗分别降低了11.12%和12.79%。图7(c)对比了各算法在任务被攻破数量优化上的表现。即便在最小的10个任务规模下，PGDRL相较于PSO算法任务被攻破数量也降低了37.44%，这是因为传统的启发式算法依赖静态的种群迭代寻优，无法实时响应服从泊松分布的恶意卫星数量变化，导致其卸载策略难以适应高风险环境。在与TD3算法的对比中，PGDRL算法将任务被攻破数量降低了2.69%。这是由于TD3作为确定性策略算法处理复杂的联合调度时易忽略安全风险。在消融实验中，PGDRL相较于P-PPO与G-PPO算法，任务被攻破数量分别减少了9.03%和8.32%。

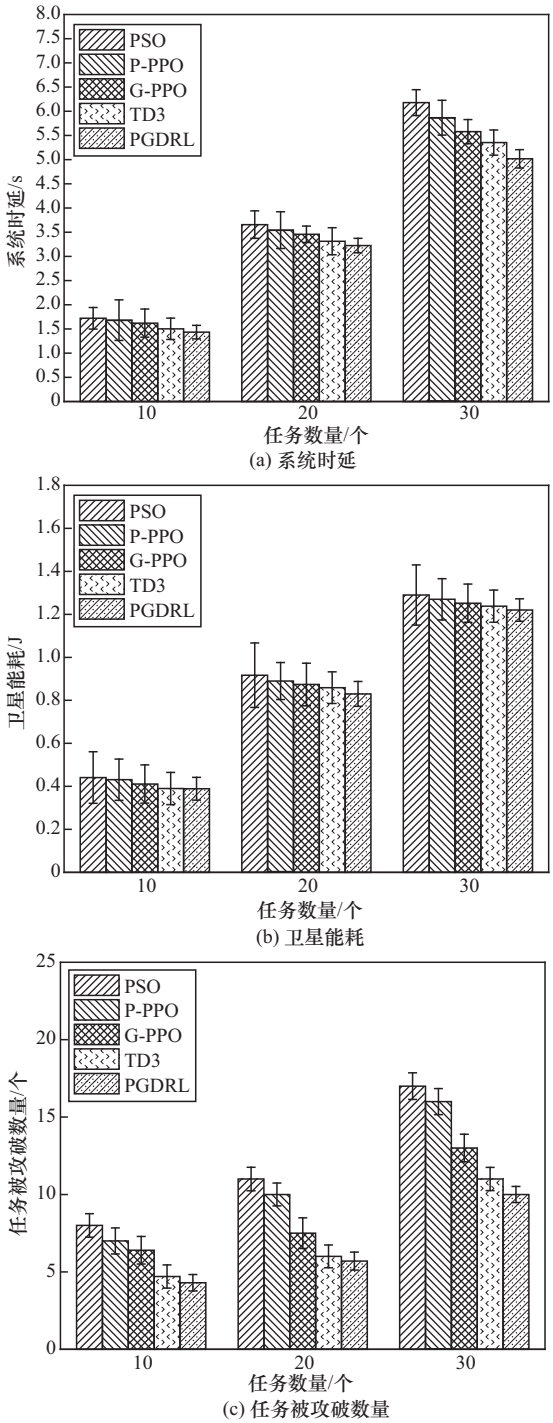



图 7　小规模场景下的算法性能对比

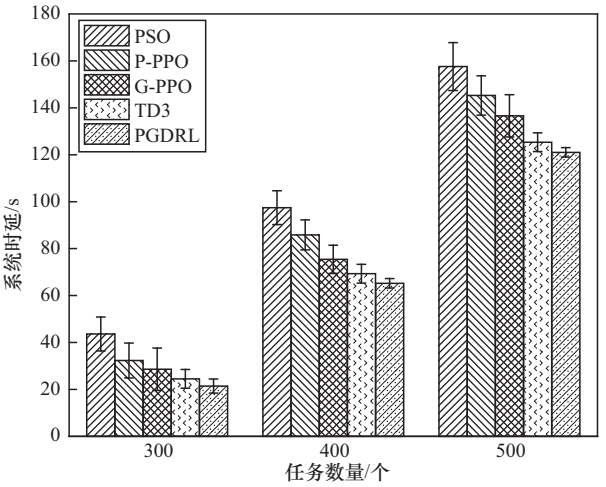
图 8(a)对比了大规模场景下各算法在优化系统时延上的表现。在大规模场景下，PGDRL 相比 PSO 算法性能提升 41.31%。相较于一般的深度强化学习算法，PGDRL 算法优化的时延保持最低，相比 TD3 算法降低了 11.19%。在与消融变体算法的对比中，PGDRL 相较于 P-PPO 与 G-PPO 算法平均时延分别降低了 25.73% 和 19.37%，这得益于 PGDRL 算法有效融合了结构表征与时序约束。图 8(b)对比了大规模场景下各算法在优化卫星能耗方面的表现。随着任务数量从 300 个增加到 500 个，卫星能耗显著增加。PGDRL 相比 PSO 算法在卫星能耗上降低了 28.95%。此外，PGDRL 因 PPO 算法训练稳定性以及兼顾计算效率与采样效率的优势，对比次优的 TD3 算法在卫星能耗上降低了 6.95%。在消融实验中，PGDRL 相比 P-PPO 与 G-PPO 算法性能分别提升 9.38% 和 8.24%。图 8(c)对比了大规模场景下各算法在优化任务被攻破数量上的表现。在 300 个任务的场景下，面对动态变化的场景，静态场景下的传统启发式算法难以适应，且大规模任务使算法难以探寻最优解，PGDRL 算法优化下的任务被攻破数量相比 PSO 算法减少了 52.89%。随着任务数量增加到 500 个，PGDRL 算法依旧可以保持最优，相比 TD3 算法性能提升了 17.58%。相比于消融变体算法，PGDRL 算法的任务被攻破数量相比 P-PPO 和 G-PPO 分别降低了 39.32% 和 28.98%。

图 7 与图 8 表明，在不同任务规模的场景下，PGDRL 算法都可以更好地满足复杂拓扑约束下的调度需求。这确保了系统在严格的资源限制下，能够实现时延、卫星能耗与安全的多目标协同优化。

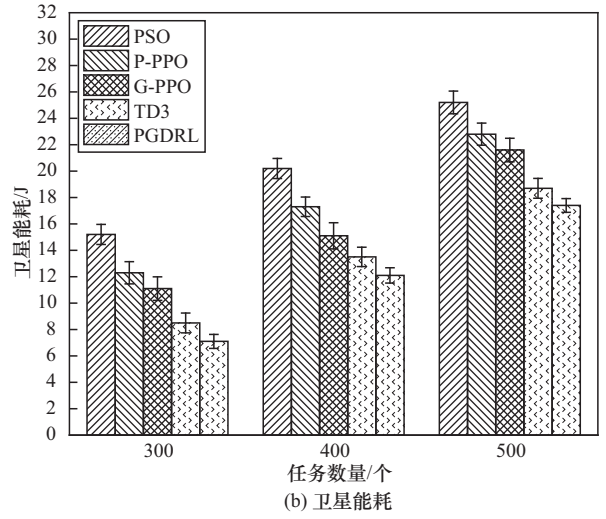
#### 4.3.4　RQ4 自适应加密策略有效性分析

为了验证不同威胁环境下 PGDRL 算法对多级加密策略的自适应能力，并评估所提方案能否在复杂动态环境中有效应对外部威胁的演化，本节分析 PGDRL 算法面对恶意节点增减时的策略动态演变过程。定义恶意卫星数量在区间[2, 5]内为低威胁环境，在区间[5, 9]内为中威胁环境，在区间[9, 13]内为高威胁环境。

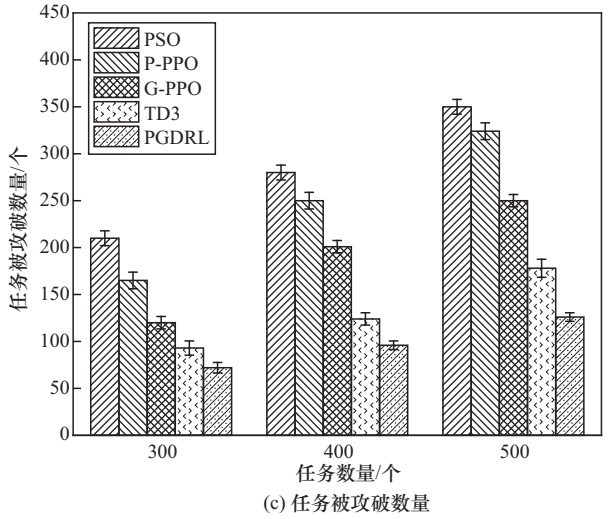
图 9 展示了不同任务数量下，系统卸载至各层计算节点的比例分布。在不同任务数量下，卸载至 LEO 卫星的任务比例始终占据主导，维持在 58.33%~80.25%。随着任务数量由 10 个逐渐增加至 60 个，卸载至 GEO 卫星和地面云服务器的任务比例逐渐上升。当任务数量达到 60 个时，GEO 卫星与地面云服务器中的任务比例分别增至 22.33% 和 20.33%。这主要是因为，随着任务数量的增加，LEO 卫星受限于自身的资源瓶颈，难以在保障所有任务服务质量的前提下实现多目标的联合优化。因此，系统会将部分计算任务转移至 GEO 卫星与地面云服务器。




(a) 系统时延




(b) 卫星能耗




(c) 任务被攻破数量

图8 大规模场景下的算法性能对比

图9表明，本文算法可以在复杂约束下寻找卸载与加密安全策略，并展现出自适应能力。系统不仅能够依据任务数量选择合适的计算节点卸载，还在多变的环境中实现了效能与安全的权衡。

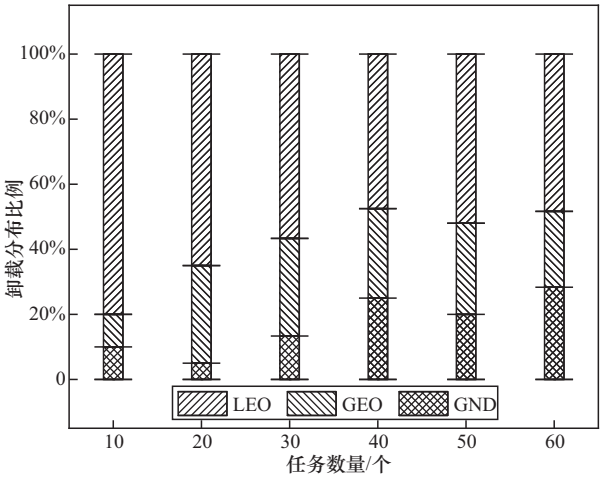



图9 不同任务数量下的卸载分布情况

## 5 结束语

本文提出了一种基于PGDRL算法的计算卸载策略，在SEC中求解偏远地区数字孪生系统任务的最优卸载策略。P-SDTO架构利用LEO、GEO卫星和地面云协同计算，并融合AES算法的多级加密机制，兼顾隐私保护与系统开销。此外，本文设计了优先级排序机制以降低计算复杂度，并提出了PGDRL算法，通过利用GAT感知的状态信息，使智能体可以依照优先级排序后的任务顺序求解最优策略。仿真结果表明，该策略具备在高动态环境下的适应能力。在不同规模任务场景下，PGDRL算法在降低系统的时延与能耗开销，并有效减少任务受攻击次数上，均优于对比算法PSO、TD3及消融算法P-PPO、G-PPO。此外，面对不同的威胁环境时，该系统能够有效实现隐私保护与执行效率的动态平衡。

未来工作中，在模型构建方面，将进一步考虑可变信道特性的影响，引入更复杂的链路衰落与干扰模型；在优化算法方面，将考虑结合多智能体强化学习技术，引入分布式协作机制，进一步提高算法对大规模异构卫星网络的探索能力与决策效率。

## 参考文献：

[1] Tao F, Zhang H, Liu A, et al. Digital twin in industry: state-of-the-art[J]. IEEE Transactions on Industrial Informatics, 2019, 15(4): 2405-2415.

[2] Lou S H, Hu Z X, Zhang Y R, et al. Human-cyber-physical system for industry 5.0: a review from a human-centric perspective[J]. IEEE Transactions on Automation Science and Engineering, 2025, 22: 494-511.

[3] Lu Y L, Huang X H, Zhang K, et al. Communication-efficient federated

learning and permissioned blockchain for digital twin edge networks[J]. IEEE Internet of Things Journal, 2021, 8(4): 2276-2288.

[4] 梁冰, 纪雯. 基于次模优化的边云协同多用户计算任务迁移方法[J]. 通信学报, 2020, 41(10): 25-36.
Liang B, Ji W. Multiuser computation offloading for edge-cloud collaboration using submodular optimization[J]. Journal on Communications, 2020, 41(10): 25-36.

[5] Mao Y Y, You C S, Zhang J, et al. A survey on mobile edge computing: the communication perspective[J]. IEEE Communications Surveys & Tutorials, 2017, 19(4): 2322-2358.

[6] Liu J J, Shi Y P, Fadlullah Z M, et al. Space-air-ground integrated network: a survey[J]. IEEE Communications Surveys & Tutorials, 2018, 20(4): 2714-2741.

[7] 李斌, 刘文帅, 费泽松. 面向空天地异构网络的边缘计算部分任务卸载策略[J]. 电子与信息学报, 2022, 44(9): 3091-3098.
Li B, Liu W S, Fei Z S. Partial computation offloading for mobile edge computing in space-air-ground integrated network[J]. Journal of Electronics & Information Technology, 2022, 44(9): 3091-3098.

[8] Zhang X Y, Liu J, Zhang R, et al. Energy-efficient computation peer offloading in satellite edge computing networks[J]. IEEE Transactions on Mobile Computing, 2024, 23(4): 3077-3091.

[9] Jia M, Zhang L, Wu J, et al. Deep multiagent reinforcement learning for task offloading and resource allocation in satellite edge computing[J]. IEEE Internet of Things Journal, 2025, 12(4): 3832-3845.

[10] Wen W C, Cui H X, He T. Multi-layer reinforcement learning assisted task offloading in satellite edge computing[J]. IEEE Transactions on Vehicular Technology, 2025, 74(4): 6561-6572.

[11] 夏玮玮, 胡静, 宋铁成. 低地球轨道卫星边缘计算场景中任务卸载与资源分配联合优化算法[J]. 通信学报, 2024, 45(7): 48-60.
Xia W W, Hu J, Song T C. Joint optimization algorithm for task offloading and resource allocation in low earth orbit satellites edge computing scenario[J]. Journal on Communications, 2024, 45(7): 48-60.

[12] 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.
Yuan X M, Tian H S, Huang K D, et al. A GAN-enhanced multi-agent deep reinforcement learning for digital twin-driven edge inference and heterogeneous resource co-optimization[J]. Chinese Journal of Computers, 2025, 48(8): 1763-1780.

[13] 张小亮, 涂勇策, 马恒太, 等. 一种适用于卫星通信网络的端到端认证协议[J]. 计算机研究与发展, 2013, 50(3): 540-547.
Zhang X L, Tu Y C, Ma H T, et al. An end-to-end authentication protocol for satellite communication network[J]. Journal of Computer Research and Development, 2013, 50(3): 540-547.

[14] 俞宏杰, 林兵, 卢宇, 等. 多云环境下面向代价和可靠性优化的工作流应用调度策略[J]. 福建师范大学学报(自然科学版), 2025, 41(1): 77-86, 123.
Yu H J, Lin B, Lu Y, et al. Optimization of workflow application scheduling in multi-cloud environments oriented towards cost and reliability[J]. Journal of Fujian Normal University (Natural Science Edition), 2025, 41(1): 77-86, 123.

[15] Yu S, Gong X W, Shi Q, et al. EC-SAGINs: edge-computing-enhanced space-air-ground-integrated networks for Internet of vehicles[J]. IEEE Internet of Things Journal, 2022, 9(8): 5742-5754.

[16] Li X H, Chen B T, Fan J C, et al. Cloud-edge-end collaborative intelligent service computation offloading: a digital twin driven edge coalition approach for industrial IoT[J]. IEEE Transactions on Network and Service Management, 2024, 21(6): 6318-6330.

[17] Hu X X, Shan L, Hua J L, et al. Joint resource allocation and task offloading for heterogeneous cloud-edge-end networks assisted by NOMA[J]. IEEE Transactions on Green Communications and Networking, 2026, 10: 1765-1778.

[18] 杨林瑶, 陈思远, 王晓, 等. 数字孪生与平行系统: 发展现状、对比及展望[J]. 自动化学报, 2019, 45(11): 2001-2031.
Yang L Y, Chen S Y, Wang X, et al. Digital twins and parallel systems: state of the art, comparisons and prospect[J]. Acta Automatica Sinica, 2019, 45(11): 2001-2031.

[19] Zhang R P, Feng Y X, Yang Y K, et al. Learning-based deadlock-free multi-objective task offloading in satellite edge computing with data-dependent constraints and limited buffers[J]. IEEE Transactions on Network Science and Engineering, 2025, 12(1): 356-368.

[20] Zhang R P, Feng Y X, Yang Y K, et al. Polynomial-complexity deadlock avoidance policy for tasks offloading in satellite edge computing with data-dependent constraints and limited buffers[J]. IEEE Transactions on Aerospace and Electronic Systems, 2024, 60(5): 6822-6838.

[21] Zou X, Xie R C, Xiong Z H, et al. Optimizing task offloading in dynamic satellite-terrestrial integrated computing power networks: a time-space-aware DRL approach[J]. IEEE Transactions on Cognitive Communications and Networking, 2026, 12: 3631-3649.

[22] Zhang H Y, Liu R K, Kaushik A, et al. Satellite edge computing with collaborative computation offloading: an intelligent deep deterministic policy gradient approach[J]. IEEE Internet of Things Journal, 2023, 10(10): 9092-9107.

[23] Lai J Y, Liu H S, Xu G Y, et al. Joint computation offloading and resource allocation for LEO satellite networks using hierarchical multi-agent reinforcement learning[J]. IEEE Transactions on Cognitive Communications and Networking, 2025, 11(4): 2554-2567.

[24] Cao J Q, Zhang S L, Chen Q X, et al. Computing-aware routing for LEO satellite networks: a transmission and computation integration approach[J]. IEEE Transactions on Vehicular Technology, 2023, 72(12): 16607-16623.

[25] Gong Y K, Yao H P, Xiong Z H, et al. Blockchain-aided digital twin offloading mechanism in space-air-ground networks[J]. IEEE Transactions on Mobile Computing, 2025, 24(1): 183-197.

[26] Lan W J, Chen K Y, Cao J N, et al. Security-sensitive task offloading

in integrated satellite-terrestrial networks[J]. IEEE Transactions on Mobile Computing, 2025, 24(3): 2220-2233.

[27] Xiao H Z, Zhao J, Feng J, et al. Joint optimization of security strength and resource allocation for computation offloading in vehicular edge computing[J]. IEEE Transactions on Wireless Communications, 2023, 22(12): 8751-8765.

[28] Duan R J, Zhang H L, Li X. Adaptive snapshot-based elastic route reconstruction for distributed LEO satellite networks[J]. IEEE Transactions on Aerospace and Electronic Systems, 2025, 61(2): 3516-3530.

[29] Sthapit S, Lakshminarayana S, He L G, et al. Reinforcement learning for security-aware computation offloading in satellite networks[J]. IEEE Internet of Things Journal, 2022, 9(14): 12351-12363.

[30] Okati N, Riihonen T. Nonhomogeneous stochastic geometry analysis of massive LEO communication constellations[J]. IEEE Transactions on Communications, 2022, 70(3): 1848-1860.

[31] Bharathi S, Chervenak A, Deelman E, et al. Characterization of scientific workflows[C]//Proceedings of the 2008 Third Workshop on Workflows in Support of Large-Scale Science. Piscataway: IEEE Press, 2008: 1-10.

[32] Fossa C E, Raines R A, Gunsch G H, et al. An overview of the IRIDIUM (R) low earth orbit (LEO) satellite system[C]//Proceedings of the IEEE National Aerospace and Electronics Conference. Piscataway: IEEE Press, 1998: 152-159.

[33] Zhang N T, Na Z Y, Tao J H, et al. Time-varying graph and binary tree search based routing algorithm for LEO satellite networks[J]. IEEE Transactions on Vehicular Technology, 2023, 72(10): 13683-13688.

[34] 王永娟, 樊昊鹏, 代政一, 等. 侧信道攻击与防御技术研究进展[J]. 计算机学报, 2023, 46(1): 202-228.
Wang Y J, Fan H P, Dai Z Y, et al. Advances in side channel attacks and countermeasures[J]. Chinese Journal of Computers, 2023, 46(1): 202-228.

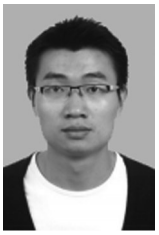
[35] Fan W H, Meng Q C, Wang G, et al. Satellite edge intelligence: DRL-based resource management for task inference in LEO-based satellite-ground collaborative networks[J]. IEEE Transactions on Mobile Computing, 2025, 24(10): 10710-10728.

[36] Cao Z Q, Deng X H, Yue S, et al. Dependent task offloading in edge computing using GNN and deep reinforcement learning[J]. IEEE Internet of Things Journal, 2024, 11(12): 21632-21646.

[37] Kennedy J, Eberhart R. Particle swarm optimization[C]//Proceedings of International Conference on Neural Networks. Piscataway: IEEE Press, 1995: 1942-1948.

[38] Fujimoto S, Hoof H, Meger D. Addressing function approximation error in actor-critic methods[C]//Proceedings of the 35th International Conference on Machine Learning (ICML). New York: ACM Press, 2018: 1587-1596.

[作者简介]



**林兵**（1986– ），男，博士，福建师范大学教授，主要研究方向为云计算技术与智能计算。



**檀啸**（2001– ），男，福建师范大学硕士生，主要研究方向为卫星边缘计算与数据布局。



**何凯**（2000– ），男，福建师范大学硕士生，主要研究方向为智能计算与能源管理。



**吴华明**（1986– ），男，博士，天津大学教授、博士生导师，主要研究方向为边缘计算、物联网与DNA存储。