中国科学 : 信息科学 2026 年 第 56 卷 第 11 期: 2860–2878
SCIENTIA SINICA Informationis
Agentic AI 赋能的 6G 融合通信专题 • 论文

AI-RAN 资源波动环境下异构 LLM 推理负载编排算法

黄开宇[1,2], 高帅[3], 侯雨茜[4], 徐凡[3*], 史清江[1,2*]

1. 同济大学计算机科学与技术学院, 上海 201804
2. 深圳市大数据研究院, 深圳 518172
3. 同济大学电子与信息工程学院, 上海 201804
4. 浙江大学信息与电子工程学院, 杭州 310027

* 通信作者. E-mail: xxiaof999@tongji.edu.cn, shiqj@tongji.edu.cn

收稿日期: 2026–03–30; 修回日期: 2026–08–11; 接受日期: 2026–10–16; 网络出版日期: 2026–11–09

国家重点研发计划项目 (批准号: 2022YFA1003900)、国家自然科学基金联合基金项目 (批准号: U25A20394)、国家自然科学基金重点项目 (批准号: 62231019)、国家自然科学基金青年科学基金项目 (C 类) (批准号: 62501424) 和上海市战略前沿专项项目 (批准号: 24DP1500704) 资助

摘要 随着 6G 网络向通算深度融合方向演进, 人工智能无线接入网 (artificial intelligence radio access network, AI-RAN) 为大语言模型 (large language model, LLM) 推理等生成式 AI 业务提供了新的边缘承载方式. 然而, 在 AI-RAN 环境中, RAN 通信业务对计算资源的动态挤占, 以及 LLM 自回归生成导致的输出长度随机性和显存需求波动, 使得实现高性能、高可靠的 LLM 推理服务面临严峻挑战. 针对上述问题, 本文提出一种资源波动环境下的异构 LLM 推理负载编排算法. 首先, 建立考虑通信业务动态资源竞争、连续批处理非线性资源消耗以及随机输出长度的系统模型, 并在独立正态近似下将显存机会约束转化为可处理的确定性二阶锥形式. 其次, 将长短期决策耦合的负载编排问题建模为马尔可夫决策过程 (Markov decision process, MDP), 设计涵盖硬件负载与执行风险的多维状态空间, 以权衡系统收益与违约风险. 最后, 提出一种基于行为克隆引导的异构近端策略优化算法 (behavior cloning-guided heterogeneous PPO, BCH-PPO), 利用双头策略网络与任务感知特征重排机制, 实现节点分配与优先级控制的并行决策, 提升算法在动态环境中的负载编排性能. 仿真结果表明, 该算法能够有效应对 AI-RAN 环境下的资源波动, 在保障 LLM 推理服务等级目标 (service level objective, SLO) 的基础上, 提高系统的推理吞吐量与调度性能.

关键词 人工智能无线接入网, 大语言模型, 负载编排, 深度强化学习, 近端策略优化

1 引言

随着无线接入网 (radio access network, RAN) 向服务化、云原生和智能内生方向持续演进, 通信与计算深度融合已成为未来 6G 网络的重要发展趋势 [1,2]. 在这一背景下, 人工智能无线接入网 (artificial intelligence radio access network, AI-RAN) 被学界和业界提出, 并被认为是支撑 AI-native 网络的重要架构. AI-RAN 架构通过在基站侧部署与基站共址的边缘服务器, 将无线通信处理能力与异构计算资源深度融合, 使 RAN 逐渐演化为兼具

引用格式: 黄开宇, 高帅, 侯雨茜, 等. AI-RAN 资源波动环境下异构 LLM 推理负载编排算法. 中国科学: 信息科学, 2026, 56: 2860–2878, doi: 10.1360/SSI-2026-0116
Huang K Y, Gao S, Hou Y Q, et al. Workload orchestration for heterogeneous LLM inference in resource-fluctuating AI-RAN environments. Sci Sin Inform, 2026, 56: 2860–2878, doi: 10.1360/SSI-2026-0116

© 2026《中国科学》杂志社

通信与计算能力的智能基础设施 [3, 4]. 由于 RAN 通常需按峰值业务需求进行资源配置以保障服务质量, 基站计算资源的平均利用率较低 (通常仅为 30% 左右), 存在大量空闲算力可供 AI 业务挖掘 [5]. 与此同时, 大语言模型 (large language models, LLMs) 推理由于其在智能助手和复杂任务执行等智能体化 AI (agentic AI) 任务中的核心作用, 正逐渐成为重要的 AI 工作负载 [6, 7]. 由于 LLM 推理极高的计算负载, 将其放置在基站端运行不仅能够解决 LLM 推理的计算需求, 也能充分利用基站的计算资源. 然而, 在 AI-RAN 场景中运行 LLM 推理任务仍面临显著挑战. 一方面, RAN 通信业务对计算资源具有绝对优先权, 使得可用于 AI 推理的资源随着通信负载变化呈现动态波动; 另一方面, 不同于计算模式相对固定的传统 AI 工作负载, LLM 推理的序列长度高度随机, 且执行效率主要受显存带宽限制, 导致其资源需求呈现显著的动态波动与不确定性 [8]. 因此, 在保障 RAN 通信业务优先级的前提下, 实现高效的 LLM 推理负载编排是 AI-RAN 系统中的关键问题.

为了解决这一问题, 越来越多研究开始探索在无线接入网中通信与 AI 计算的深度融合机制. 文献 [9] 引入了数字孪生与服务化架构, 在开放式无线接入网 (open RAN, O-RAN) 的基础上提出了端到端的 AI 融合框架, 为可编程与智能化的 6G 系统奠定了初步的技术支撑. 文献 [10] 探讨了 O-RAN 与 AI-RAN 的具体融合路径, 通过在服务管理与编排器中集成专用 AI 编排器, 初步实现了边缘站点通信业务与 AI 工作负载的统一调度. 针对通算结合场景下的资源利用率问题, 部分研究开始探索跨域调度策略. 文献 [11] 在 O-RAN 体系下设计了基于强化学习的跨域资源智能调度器, 在保证 RAN 服务满足率的同时兼容 AI 工作负载的编排, 极大提高了资源利用率. 文献 [12] 设计了基于 LLM 的 O-RAN 资源管理机制, 通过智能提示动态优化网络切片的资源分配, 显著提高了系统吞吐量和切片服务可靠性. 然而, 上述研究主要关注 AI-RAN 的宏观体系架构设计及传统 AI 工作负载调度, 尚未结合 LLM 特有的变长序列特性、自回归生成模式和细粒度服务约束进行针对性设计, 导致其在面对高并发、高性能要求的 LLM 推理请求调度时, 缺乏精细化的负载编排能力.

注意到上述研究普遍考虑在 AI-RAN 节点执行传统确定性 AI 负载的资源调度, 随着生成式 AI 逐渐成为 AI-RAN 节点的主要负载, 学术界也开始针对生成式 AI 负载开展调度优化研究. 文献 [13] 确立了基于注意力机制的 Transformer 架构, 其自回归生成模式构成了大模型推理的理论基础, 但也直接引发了 LLM 推理在执行动态性与资源异构性方面的核心挑战. 具体而言, 受自回归模式影响, 请求的输出长度在推理完成前无法预知, 这种执行动态性会导致传统批处理模式产生严重的计算气泡. 同时, 自回归特征要求持续复用历史键 – 值缓存 (key-value cache, KV Cache), 且请求间显著的变长序列特征极易引发严重的显存碎片化与空间利用率低下问题. 针对第一个问题, 文献 [14] 引入了连续批处理技术, 通过迭代级调度打破了序列间的同步等待约束, 有效提高了算力利用率. 针对第二个问题, 文献 [15] 提出了分页动态内存管理技术, 通过解耦物理内存与逻辑序列, 实现了 KV 缓存的高效管理. 在此基础上, 后续研究进一步深入探讨了面向服务约束和服务等级目标 (service level objective, SLO) 的复杂调度模型. 文献 [16] 从理论角度建立了带 KV 缓存约束的在线调度模型, 通过精细化批处理算法最小化推理时延; 文献 [17] 提出了一种 LLM 服务动态调度系统, 通过多实例间的任务重调度与 GPU 内存状态动态迁移, 实现了负载均衡、显存碎片整理与任务优先级隔离. 针对多任务多 SLO 场景, 文献 [18] 利用序列长度预测与模拟退火策略优化了优先级映射方案, 显著提升了系统 SLO 满足率. 然而, 这类研究通常植根于计算资源稳定的数据中心环境, 尚未考虑 AI-RAN 场景中通信业务波动导致的算力瞬态抢占. 由于缺乏对非稳态环境下通算资源博弈的感知, 现有 LLM 调度方案在面对动态变化的无线算力资源时, 难以保障推理服务稳定性, 面临严重的 SLO 违约风险.

从更广义的 6G 大模型融合视角看, 文献 [19] 系统讨论了集成学习与通信框架, 并联合考虑任务分配、模型规模、带宽和功率控制; 文献 [20] 则从隐私、安全与可信角度总结了分布式无线大模型的关键挑战. 这些工作分别侧重学习 – 通信协同优化和可信部署, 为无线大模型系统提供了重要的体系化基础. 与其不同, 本文关注通信业务瞬态挤占边缘计算资源时的 LLM 推理编排, 核心问题是如何同时处理输出长度不确定性、连续批处理资源耦合及细粒度 SLO 约束. 针对 AI-RAN 场景下通信负载瞬态波动与 LLM 自回归推理长度不确定性带来的异构资源编排难题, 本文提出了一种基于深度强化学习的异构 LLM 推理负载编排算法, 主要创新如下.

(1) 针对 AI-RAN 环境, 建立能够联合刻画 RAN 与 AI 负载动态竞争、LLM 连续批处理资源耦合及随机输出长度的系统模型. 在独立正态近似下, 该模型将随机峰值显存机会约束转化为带安全缓冲的确定性二阶锥约

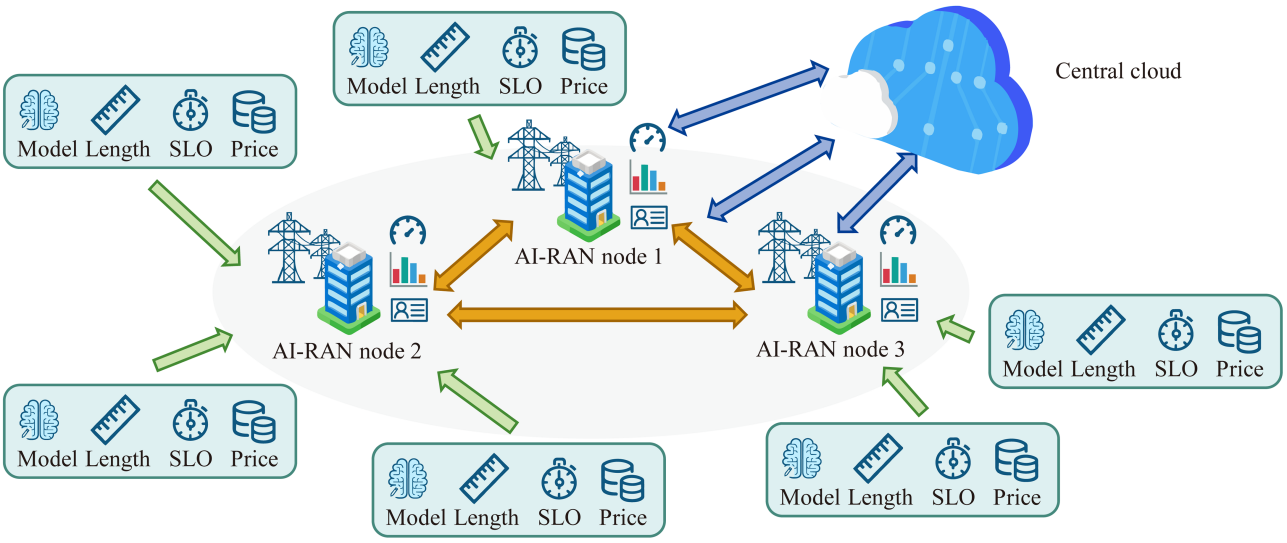

图 1　(网络版彩图) 系统模型.

Figure 1　(Color online) System model.

束, 从而以可调风险水平描述边缘侧资源边界与统计复用关系.

(2) 针对 LLM 自回归生成过程跨时隙演进带来的长短期决策耦合, 以及 LLM 推理服务中 SLO 的严苛保障需求, 将负载编排问题建模为最大化长期累积期望收益的马尔可夫决策过程 (Markov decision process, MDP). 通过设计涵盖物理硬件负载、逻辑执行瓶颈以及前沿执行风险的多维状态表征体系, 实现了对系统长期收益与服务违约风险之间权衡关系的统一描述.

(3) 针对动态并发环境下离散节点选择与连续准入顺序的异质决策, 提出基于行为克隆引导的异构近端策略优化算法 (behavior cloning-guided heterogeneous PPO, BCH-PPO). 本文的算法设计体现在面向上述编排问题构造双头策略、任务感知特征重排、可行性掩码以及专家引导的两阶段训练流程, 使节点分配与优先级决策能够在统一策略中协同优化.

2　系统模型

考虑一个由 N 个 AI-RAN 节点构成的边缘计算系统 (如图 1), 节点集合记为 $\mathcal{N}=\{1,2,\ldots,N\}$. 系统利用部署在基站侧的边缘服务器, 在满足通信处理任务的前提下为 LLM 推理任务提供低时延边缘服务. 系统时间被分割成 T 个等长时隙 $\mathcal{T}=\{1,2,\ldots,T\}$, 每个时隙 $t\in\mathcal{T}$ 持续时间为 Δt. 在时隙内, 系统接收到一组 AI 推理请求集合, 记为 $\mathcal{J}(t)$. 系统根据当前各节点的资源状态对到达的 AI 推理请求集合进行统一调度决策.

2.1　运行机制

在 AI-RAN 场景中, AI 推理请求在系统中的处理流程主要包括请求解析、负载编排与推理执行三个阶段, 具体过程如下. (1) 请求解析阶段: 请求到达边缘侧后, 入口节点通过深度包检测等技术对请求进行解析, 从中提取模型类型、输入序列长度以及 SLO 等请求特征. 同时, 系统基于历史统计或预测模型对请求的输出长度进行估计, 以获得后续资源调度所需的其他特征信息. (2) 负载编排阶段: 由于边缘节点的计算资源需要优先服务 RAN 通信负载, 其可用于 AI 推理的资源会随通信负载动态变化. 系统调度模块根据各边缘节点当前的资源状态, 决定每个新到达请求 $j\in\mathcal{J}(t)$ 的执行位置. 该决策过程由决策变量集合 $\mathbf{X}=\{x_{n,j}(t)\mid n\in\mathcal{N},j\in\mathcal{J}(t),t\in\mathcal{T}\}$ 完整描述: 具体地, 通过二元指派变量 $x_{n,j}(t)\in\{0,1\}$ 进行量化, 若系统在边缘节点集合 $\mathcal{N}$ 中匹配到满足条件的执行节点 n, 则 $x_{n,j}(t)=1$; 若所有边缘节点均无法满足任务需求, 请求将被转发至中心云执行, 即 $\sum_{n\in\mathcal{N}}x_{n,j}(t)=0$. (3) 推理执行阶段: 对于被分配至边缘节点的请求, 其推理过程通常由主流 LLM 推理引擎负责执行 [15]. 此类系

统普遍采用连续批处理机制, 将时隙 t 编排的新请求与尚未完成生成的存量请求动态组织为活跃执行批次 $\mathcal{B}_n(t)$. 在该机制下, GPU 计算核心与显存带宽由整个批次 $\mathcal{B}_n(t)$ 集体占用, 以执行统一的张量计算与访存任务. 由于 LLM 的自回归生成特性, 批次的实时规模与各任务的生成进度将共同决定系统当前的显存占用与单步解码时延特性, 呈现出显著的非线性耦合特征.

2.2　资源模型

2.2.1　节点静态资源模型

每个边缘节点 $n \in \mathcal{N}$ 均配置异构计算硬件以支撑 AI 推理任务. 定义节点的静态物理资源向量为 $\mathbf{C}_n = \{G_n, M_n, B_n\}$, 其中 G_n 表示 GPU 峰值计算吞吐量 (FLOPS), M_n 表示节点显存容量 (Bytes), B_n 表示显存带宽 (Bytes/s). 由于大模型部署的存储空间限制与高昂的加载时延, 边缘节点通常采用模型预加载策略. 记 $\mathcal{S}_n$ 为节点 n 已部署的模型集合. 因此, 节点 n 处理请求 j 的必要前提是该请求所需的模型 m_j 满足 $m_j \in \mathcal{S}_n$.

2.2.2　RAN 负载与动态资源模型

在 AI-RAN 架构中, RAN 负载与 AI 负载共享底层硬件, 且 RAN 负载具备绝对的资源优先级. 因此, AI 推理任务可用的计算资源具有显著的时变性与机会性. 现有实测研究表明, RAN 资源需求同时受到调制编码方案、信噪比和并发任务等多种因素影响, 各因素之间关系复杂, 难以建立易处理的精确模型 [11]. 尽管如此, 物理层基带处理的总体计算需求与活跃用户数及物理资源块规模仍呈整体正相关 [21]. 因此, 现有 AI-RAN 编排研究通常在系统级时间尺度上采用简化假设, 将用户数或归一化 RNTI 负载线性映射为 MIG 或共享 GPU 资源的需求比例 [11,22]. 遵循这一编排层建模思路, 本文采用线性映射近似刻画 RAN 负载对 AI 可用资源的平均挤占效应, 以保证在线编排问题的可处理性. 由此, AI 负载在时隙 t 可利用的动态剩余资源表示为

$$G_n^{\mathrm{res}}(t) = (1 - \alpha \cdot \rho_n(t)) \cdot G_n, \tag{1}$$

$$B_n^{\mathrm{res}}(t) = (1 - \beta \cdot \rho_n(t)) \cdot B_n, \tag{2}$$

其中, α 和 β 分别为 RAN 负载对计算和带宽的资源转换系数, 反映单位通信负载对硬件资源的平均挤占程度; $\rho_n(t) \in [0,1]$ 表示归一化网络负载. 调制编码方案、信噪比和码块结构会使相同 RAN 负载对应不同的基带处理工作量, MIG 划分粒度、内核调度和显存带宽竞争则会使基带工作量到实际 GPU 资源占用的转换呈现非线性. 因此, 式 (1) 和 (2) 的线性映射可能对剩余计算资源 $G_n^{\mathrm{res}}(t)$ 和剩余带宽 $B_n^{\mathrm{res}}(t)$ 产生估计偏差: 若低估 RAN 资源占用, 则会高估 AI 可用资源并低估推理时延, 从而增加 SLO 违约风险; 反之, 则可能造成过于保守的请求准入和资源利用率下降. 本文通过系数 α 和 β 对上述关系进行一阶近似, 实际部署时可依据具体硬件平台的测量结果进行校准. 与计算资源的动态波动不同, 为保障通信系统的极高可靠性并防止显存溢出, RAN 协议栈通常采用静态显存预留机制. 因此, AI 任务可用的显存资源被建模为不随时隙变化的固定值

$$M_n^{\mathrm{res}} = M_n - M_n^{\mathrm{RAN}}. \tag{3}$$

2.3　AI 负载模型

每个新到达的请求 $j \in \mathcal{J}(t)$ 由特征元组 $\Phi_j = \langle \mathcal{M}_j, l_j^{\mathrm{in}}, l_j^{\mathrm{out}}, \tau_j^{\mathrm{TPOT}}, \mathbf{p}_j \rangle$ 刻画. 其中, 模型需求 $\mathcal{M}_j$ 代表请求对特定 AI 模型的要求, 其既可以被限定为某一特定模型, 亦可设置为对多个模型的开放性选择, 从而允许系统根据灵活决策最优的执行模型 m_j. l_j^{in} 表示输入序列长度, 为已知的确定值; l_j^{out} 表示输出序列长度, 在请求完成前无法准确观测, 因而需要在请求执行前对其进行预测. 为此, 现有研究将 LLM 响应长度预测建模为不确定性感知回归问题, 从而能够同时获得逐请求的输出长度估计及其预测不确定性 [23]. 基于这一建模思路, 本文分别以 μ_j 和 σ_j^2 表示输出长度的预测均值和方差, 并采用如下逐请求正态分布对其进行近似:

$$l_j^{\mathrm{out}} \sim \mathcal{N}(\mu_j, \sigma_j^2). \tag{4}$$

τ_j^{TPOT} 用来刻画每个输出 token 时延的 SLO, 用户借此定义请求的最慢生成速度, 任意一个 token 生成超时都将被视为服务违约, 导致无法进行价值结算. 最后, 价格向量 $\mathbf{p}_j = \{p_j^{\mathrm{in}}, p_j^{\mathrm{out}}\}$ 刻画了 LLM 推理不同阶段的计费机制. 其中, p_j^{in} 对应预填充阶段处理输入上下文中每个 token 的单价; p_j^{out} 对应解码阶段生成每个 token 的单价.

为了提升推理效率, 边缘节点普遍采用连续批处理技术. 不同于传统边缘计算研究通常假设将节点算力、带宽等资源正交切分给独立任务 [11], LLM 推理中的硬件资源是直接作用于批次整体的. 在这种机制下, GPU 核心与显存带宽由当前活跃执行批次 $\mathcal{B}_n(t)$ 集体占用. 该批次的资源消耗强度由所有成员的状态共同决定. 这种整体性导致资源需求呈现出显著的非线性耦合特征.

2.3.1　显存需求模型

连续批处理按模型分别维护执行批次, 记 $m_n(t)$ 为节点 n 当前批次 $\mathcal{B}_n(t)$ 共享的模型. 该批次的瞬时显存占用 $M_{\mathrm{batch}}(\mathcal{B}_n(t))$ 取决于共享模型权重与各任务的 KV 缓存:

$$M_{\mathrm{batch}}(\mathcal{B}_n(t)) = \underbrace{M_{\mathrm{weight}}(m_n(t))}_{\text{批次共享权重}} + \sum_{j\in\mathcal{B}_n(t)} \underbrace{M_{\mathrm{KV}}(m_j)(l_j^{\mathrm{in}} + k_j(t))}_{\text{请求 KV 缓存}}, \tag{5}$$

其中 $M_{\mathrm{weight}}(m_n(t))$ 为批次共享模型的参数显存占用, $M_{\mathrm{KV}}(m_j)$ 为请求 j 所用模型每生成或处理一个 token 所需增加的 KV 缓存显存大小, $k_j(t)$ 为任务 j 截至当前时隙已生成的 token 数量.

由于自回归生成过程中 KV 缓存随时间单调递增, 系统编排决策必须前瞻性地约束推理结束时的峰值显存需求 $M_{\mathrm{batch}}^{\mathrm{peak}}(\mathcal{B}_n(t))$:

$$M_{\mathrm{batch}}^{\mathrm{peak}}(\mathcal{B}_n(t)) = M_{\mathrm{weight}}(m_n(t)) + \sum_{j\in\mathcal{B}_n(t)} M_{\mathrm{KV}}(m_j)(l_j^{\mathrm{in}} + l_j^{\mathrm{out}}). \tag{6}$$

考虑到所有请求同时达到理论最大长度的概率极低, 本文引入机会约束来描述显存安全性, 即保证在推理全周期内, 触发显存溢出的概率低于预设风险阈值 ϵ_{risk}:

$$\Pr\left(M_{\mathrm{batch}}^{\mathrm{peak}}(\mathcal{B}_n(t)) > M_n^{\mathrm{res}}\right) \leqslant \epsilon_{\mathrm{risk}}. \tag{7}$$

该假设契合了主流 LLM 推理引擎支持的显存卸载功能 [24]. 当瞬时显存达到饱和时, 系统可将部分 KV 缓存动态交换至 CPU 内存以维持服务连续性, 这意味着在受控的极低概率下允许显存溢出具有实际的物理可行性, 而不会导致系统崩溃. 基于此物理机制, 系统能够利用大数定律带来的统计复用增益, 显著缓解过度预留资源导致的闲置浪费, 从而在保障系统可靠性的前提下极大提升显存利用效率.

为将机会约束转化为可在线计算的确定性形式, 本文进一步假设同一批次内各请求的输出长度相互独立, 且分别满足式 (4). 在此条件下, 式 (7) 左侧的批次峰值显存占用仍服从正态分布, 其均值由模型权重占用和各请求的期望 KV 缓存占用共同构成, 方差则由各请求输出长度的不确定性累积得到. 因此, 将该随机量标准化后, 机会约束可等价表示为其 $(1-\epsilon_{\mathrm{risk}})$ 分位数不超过节点的可用显存, 从而得到如下确定性二阶锥约束 [25]:

$$\begin{aligned}\widehat{M}_{\mathrm{batch}}^{\mathrm{peak}}(\mathcal{B}_n(t)) = {} & \underbrace{M_{\mathrm{weight}}(m_n(t)) + \sum_{j\in\mathcal{B}_n(t)} M_{\mathrm{KV}}(m_j)(l_j^{\mathrm{in}} + \mu_j)}_{\text{期望峰值显存占用}} \\ & + \underbrace{\lambda_{\epsilon_{\mathrm{risk}}} \sqrt{\sum_{j\in\mathcal{B}_n(t)} \left(M_{\mathrm{KV}}(m_j)\sigma_j\right)^2}}_{\text{安全缓冲}} \leqslant M_n^{\mathrm{res}},\end{aligned} \tag{8}$$

其中, $\Phi(\cdot)$ 表示标准正态分布的累积分布函数, $\lambda_{\epsilon_{\mathrm{risk}}} = \Phi^{-1}(1-\epsilon_{\mathrm{risk}})$ 为与风险阈值对应的置信因子. 式 (8) 的第一项表示批次峰值显存占用的期望, 第二项则根据输出长度的预测不确定性预留安全缓冲. 当 ϵ_{risk} 减小时, $\lambda_{\epsilon_{\mathrm{risk}}}$ 相应增大, 因而系统将预留更多显存以降低溢出风险.

2.3.2　批处理时延模型

LLM 的推理过程由预填充和解码两个阶段组成. 预填充阶段主要负责处理请求的输入序列, 虽具有较高的计算强度, 但由于其仅在请求到达之初执行一次, 其时延开销在长序列生成任务中占比相对有限. 相比之下, 解码阶段涉及连续的自回归生成过程, 需要逐个产生输出 token, 这一阶段的持续性直接决定了用户感知的流畅度. 因此, 本文重点关注解码阶段的时延表现, 并以此作为 SLO 的核心约束.

在解码过程中, 边缘节点以批次为整体执行硬件操作. 每生成一个输出 token, 系统涉及两个核心物理动作: 一是计算核心执行张量运算, 二是将数据从显存搬运至寄存器. 在标准的全量注意力机制下, 单步生成所需的访存数据量等同于该批次当前的瞬时显存占用量 $M_{\text{batch}}(\mathcal{B}_n(t))$, 即需要完整读取一次模型权重参数及所有任务已积累的 KV 缓存. 不同于传统的计算密集型任务, 由于 LLM 庞大的模型参数规模与频繁的自回归访存需求, 这一过程呈现出显著的访存受限特征.

基于此执行特征, 单步解码时延 T_{batch} 可解构为计算分量、访存分量与系统调度开销 ϵ_{dec} 的叠加. 首先, 定义 $\omega(m_j)$ 为模型 m_j 生成单个 token 所需的计算量, 则批次 $\mathcal{B}_n(t)$ 在单步生成中的总计算负载 $W_{\text{batch}}(\mathcal{B}_n(t))$ 表示为

$$W_{\text{batch}}(\mathcal{B}_n(t)) = \sum_{j\in\mathcal{B}_n(t)} \omega(m_j). \tag{9}$$

结合显存模型 $M_{\text{batch}}(\mathcal{B}_n(t))$, 时延模型 T_{batch} 建模为

$$T_{\text{batch}}(\mathcal{B}_n(t)) = \underbrace{\frac{W_{\text{batch}}(\mathcal{B}_n(t))}{G_n^{\text{res}}(t)}}_{\text{计算时延}} + \underbrace{\frac{M_{\text{batch}}(\mathcal{B}_n(t))}{B_n^{\text{res}}(t)}}_{\text{访存时延}} + \underbrace{\epsilon_{\text{dec}}}_{\text{调度开销}}. \tag{10}$$

由式 (10) 可见, 由于访存分量在总时延中占据主导, 且 KV 缓存随序列长度单调增加, 单步解码时延表现出随生成进度演进的增长特性.

为了保障服务质量, 系统需关注任务在整个推理周期内的峰值时延. 为构造面向整个生成周期的保守时延估计, 本文作如下假设: 节点采用同步连续批处理和全量注意力, 时隙内 $G_n^{\text{res}}(t)$ 与 $B_n^{\text{res}}(t)$ 保持不变, 且 KV 缓存随生成长度单调增加. 定义任务 k 的概率有效长度为 $\tilde{l}_k = \mu_k + \lambda_{\epsilon_{\text{risk}}}\sigma_k$. 将各请求的概率有效长度代入式 (10), 得到批次单步解码时延的保守峰值估计

$$T_{\text{batch}}^{\text{peak}}(\mathcal{B}_n(t)) = \frac{\sum_{k\in\mathcal{B}_n(t)}\omega(m_k)}{G_n^{\text{res}}(t)} + \epsilon_{\text{dec}} + \frac{M_{\text{weight}}(m_n(t)) + \sum_{k\in\mathcal{B}_n(t)} M_{\text{KV}}(m_k)(l_k^{\text{in}} + \tilde{l}_k)}{B_n^{\text{res}}(t)}. \tag{11}$$

式 (11) 中, $\mathcal{B}_n(t)$ 表示时隙 t 内节点 n 的活跃请求批次; $\omega(m_k)$ 表示模型 m_k 生成单个 token 所需的计算量, 因而 $\sum_{k\in\mathcal{B}_n(t)}\omega(m_k)/G_n^{\text{res}}(t)$ 表示批次计算负载与节点剩余计算能力之比; ϵ_{dec} 表示一次解码步的固定调度开销. $M_{\text{weight}}(m_n(t))$ 表示节点当前加载模型的权重显存, $M_{\text{KV}}(m_k)$ 表示模型 m_k 每个 token 的 KV 缓存占用, l_k^{in} 与 $\tilde{l}_k$ 分别表示请求 k 的输入长度和概率有效输出长度, 因而分子中的显存项刻画模型权重与批次 KV 缓存的风险调整峰值占用; 该占用除以节点剩余显存带宽 $B_n^{\text{res}}(t)$ 后得到访存时延. 因此, 式 (11) 给出了式 (10) 在概率峰值状态下的风险感知上界. 对于批次中的任一请求 $j \in \mathcal{B}_n(t)$, 其 SLO 达标的物理前提是: 在请求的整个生命周期内, 其所在的批次时延始终不超过其容忍阈值, 即

$$T_{\text{batch}}^{\text{peak}}(\mathcal{B}_n(t)) \leqslant \tau_j^{\text{TPOT}}. \tag{12}$$

上述模型清晰地揭示了连续批处理中的负荷耦合效应: 节点是否准入新任务, 不仅取决于新任务自身的时延要求, 更取决于其加入后导致的批次峰值时延变化情况. 若 $T_{\text{batch}}^{\text{peak}}$ 超过了某任务 j 的阈值 τ_j^{TPOT}, 则该任务将面临 SLO 违约风险.

2.3.3　任务价值模型

系统的核心目标在于最大化边缘 AI 推理服务的长期累积经济效益. 在当前的商业实践与云服务标准中, 推理服务的收益结算通常高度依赖于实际处理的计算负载, 即以 token 吞吐量作为计价基准. 考虑到 LLM 推理的阶段性特征, 任务 j 的事后实际价值 V_j 由预填充阶段的输入处理价值与解码阶段的生成价值共同组成. 然而, 收益的最终获取必须以履行 SLO 为前提, 由于边缘节点采用连续批处理机制, 批次内所有请求共享相同的解码时延 T_{batch}, 因此一个请求能否成功结算, 取决于该批次在整个推理周期内是否始终满足该请求的容忍阈值. 基于上述逻辑, 任务 j 的事后实际价值 V_j 建模为

$$V_j(l_j^{\text{out}}) = (p_j^{\text{in}} \cdot l_j^{\text{in}} + p_j^{\text{out}} \cdot l_j^{\text{out}}) \cdot \mathbb{I}\left(\max_{t \in \text{lifetime}} T_{\text{batch}}(\mathcal{B}_n(t)) \leqslant \tau_j^{\text{TPOT}} \right), \tag{13}$$

其中 $\mathbb{I}(\cdot)$ 为指示函数.

值得注意的是, 边缘节点的 LLM 推理负载编排属于典型的事前决策过程. 由于 LLM 生成长度 l_j^{out} 的随机性, 系统在执行调度决策时无法精确预知每个任务的最终产出, 仅能依赖于任务特征中的统计信息. 为了弥合这种信息不对称性, 系统需基于输出长度分布的均值 μ_j 对潜在收益进行量化评估. 由此, 任务 j 的期望经济价值可先写为

$$\begin{aligned} \bar{V}_j &= \mathbb{E}\big[(p_j^{\text{in}} l_j^{\text{in}} + p_j^{\text{out}} l_j^{\text{out}}) \mathbb{I}(\text{SLO}_j)\big] \\ &\approx p_j^{\text{in}} l_j^{\text{in}} + p_j^{\text{out}} \mu_j, \end{aligned} \tag{14}$$

其中 SLO_j 表示请求生命周期内式 (12) 始终成立. 近似号依赖两个条件: 准入检查使 $\Pr(\text{SLO}_j)$ 接近 1, 且价格参数与随机输出长度独立. 若任务在执行中发生违约, 该近似会高估可结算价值, 因而后续奖励函数另行施加违约惩罚并采用过程价值反馈进行校正. 在实际编排中, 系统需通过准入控制合理配置活跃批次 $\mathcal{B}_n(t)$ 的规模, 以在提升系统吞吐量的同时, 避免因负载过高触发服务违约.

3　问题建模

本文的优化目标是在满足硬件物理资源限制与用户服务等级协议的前提下, 通过联合准入控制与节点分配, 最大化边缘 AI 推理服务的长期累积期望收益. 具体的优化问题如下:

$$\mathbf{P1}: \quad \max_{\mathbf{X}} \quad \sum_{t \in \mathcal{T}} \sum_{n \in \mathcal{N}} \sum_{j \in \mathcal{J}(t)} \bar{V}_j x_{n,j}(t) \mathbb{I}\Big(T_{\text{batch}}^{\text{peak}}(\mathcal{B}_n(t)) \leqslant \tau_j^{\text{TPOT}} \Big), \tag{15a}$$

$$\text{s.t.} \quad x_{n,j}(t) \in \{0,1\},\ \forall n \in \mathcal{N},\ j \in \mathcal{J}(t),\ t \in \mathcal{T}, \tag{15b}$$

$$x_{n,j}(t) \leqslant \mathbb{I}(m_j \in \mathcal{S}_n),\ \forall n \in \mathcal{N},\ j \in \mathcal{J}(t),\ t \in \mathcal{T}, \tag{15c}$$

$$\sum_{n \in \mathcal{N}} x_{n,j}(t) \leqslant 1,\ \forall j \in \mathcal{J}(t),\ t \in \mathcal{T}, \tag{15d}$$

$$\widehat{M}_{\text{batch}}^{\text{peak}}(\mathcal{B}_n(t)) \leqslant M_n^{\text{res}},\ \forall n \in \mathcal{N},\ t \in \mathcal{T}. \tag{15e}$$

在该优化问题中, 二进制决策变量 $x_{n,j}(t)$ 用于决定在时隙 t 是否将新到达请求 $j \in \mathcal{J}(t)$ 分配至边缘节点 n 执行. 目标函数旨在最大化所有准入任务的期望经济价值 $\bar{V}_j$, 并引入指示函数 $\mathbb{I}(\cdot)$ 确保收益结算的前提是活跃执行批次 $\mathcal{B}_n(t)$ 的预期峰值解码时延 $T_{\text{batch}}^{\text{peak}}$ 不超过请求个体的容忍阈值 τ_j^{TPOT} (即式 (12) 成立). 在约束体系中, 式 (15c) 通过选定模型 m_j 与节点预加载模型集合 $\mathcal{S}_n$ 的隶属关系确保模型匹配, 式 (15d) 限制任务分配的唯一性, 式 (15e) 则要求风险调整后的峰值显存估计不超过节点的固定剩余显存 M_n^{res}.

优化问题 **P1** 是一个具有长短期决策耦合特征的混合整数非线性规划问题, 具有高度复杂性, 主要体现在以下三个维度: 首先是请求异构性与资源的非线性耦合, 不同任务对模型及 SLO 的差异化需求在连续批处理机制下表现为高度非凸与不连续的物理特征; 其次是跨时隙的状态演进与时空耦合, 当前的决策会通过改变活跃批

次 $\mathcal{B}_n(t)$ 的构成, 持续影响未来存量任务的资源占用空间与 SLO 达标情况; 最后是 AI-RAN 资源剧烈波动与生成长度随机性带来的高度不确定性. 综上所述, 该问题难以通过静态规划手段解决, 亟需引入一种能够在线感知环境波动并进行长效增益权衡的智能编排算法.

4　算法设计

本节针对 AI-RAN 环境下计算资源随通信负载剧烈波动的特性, 提出一种异构 LLM 推理负载编排算法 BCH-PPO. 首先, 将联合优化问题 **P1** 转化为 MDP, 以刻画调度智能体与动态环境间的时空交互规律; 随后, 给出 BCH-PPO 算法的具体设计过程.

4.1　问题转化

在本文构建的 AI-RAN 架构中, 系统的全局编排器被视为执行决策的智能体. 由于 AI 推理任务的到达具有随机性, 且 AI-RAN 节点的剩余资源随通信负载动态波动, 本文将优化问题 **P1** 建模为一个离散时间的 MDP. 该 MDP 过程由五元组 $\langle \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma \rangle$ 表示. 其中, $\mathcal{S}$ 为环境状态空间; $\mathcal{A}$ 为智能体的动作空间; $\mathcal{P}$ 代表状态转移概率, 描述了系统在执行特定编排动作后, 其活跃批次 $\mathcal{B}_n(t)$ 和资源负载的演进规律; $\mathcal{R}$ 为奖励函数, 用于量化决策的优劣; γ 为折扣因子, 本文设定 $\gamma = 0.99$. 在每个时隙 $t \in \mathcal{T}$, 调度智能体感知当前的系统状态 $s(t) \in \mathcal{S}$, 依据策略选择动作 $a(t) \in \mathcal{A}$, 环境随后反馈即时奖励 r_t 并转移至下一状态 $s(t+1)$.

4.1.1　状态空间

在时隙 t, 系统状态 $s(t) \in \mathcal{S}$ 包含当前时刻到达的 LLM 推理请求特征和系统内 AI-RAN 节点的资源状态. 为解决任务数量 $|\mathcal{J}(t)|$ 随时间波动的问题, 本文设定系统最大并发请求数为 $J_{\max}$, 并采用补零策略对状态向量进行对齐, 状态表示为 $s(t) = \{\Psi_{\text{task}}(t), \mathbf{C}_{\text{node}}(t), \mathbf{q}(t)\}$. 其中, 补零位置的任务特征全部置 0, $q_j(t) \in \{0, 1\}$ 为有效位掩码: 真实请求取 1, 补零位置取 0.

任务特征集合 $\Psi_{\text{task}}(t)$ 针对对齐后的每个请求 j 提取多维特征向量. 其物理属性来源于系统模型中的特征元组 $\Phi_j = \langle \mathcal{M}_j, l_j^{\text{in}}, l_j^{\text{out}}, \tau_j^{\text{TPOT}}, \mathbf{p}_j \rangle$, 但真实输出长度 l_j^{out} 在决策时不可观测, 因而在状态中由统计量 (μ_j, σ_j) 替代. 在需求匹配方面, 模型需求特征 $\mathcal{M}_j$ 被编码为一个 L 维的 0/1 向量 $\mathbf{m}_j = [m_{j,1}, m_{j,2}, \ldots, m_{j,L}] \in \{0, 1\}^L$. 在该向量中, L 代表系统中已部署的模型种类总数, 若请求 j 需要调用第 l 种模型, 则对应分量 $m_{j,l} = 1$, 否则为 0, 从而实现任务需求与节点预加载模型的精确匹配. 在负载刻画层面, 针对自回归生成长度 l_j^{out} 的随机性, 状态向量提取其统计特征期望 μ_j 与标准差 σ_j 以预估长期的资源占用, 并整合归一化后的输入长度 l_j^{in} 与时延约束 τ_j^{TPOT} 来确立 SLO 的边界. 在价值转化层面, 系统利用价格向量 $\mathbf{p}_j$ 近似期望价值 $\bar{V}_j$, 显式地量化各任务对累积收益目标的潜在贡献.

节点资源状态 $\mathbf{C}_{\text{node}}(t)$ 遵循从物理边界到执行瓶颈, 再到前沿风险的递进逻辑构建. 首先, 基于实时 RAN 负载 $\rho_n(t)$ 与固定模型部署的独热向量 $\mathbf{s}_n \in \{0, 1\}^L$ 确立物理基准, 并依据式 (1) 与 (2) 动态缩放有效计算资源 $G_n^{\text{res}}(t)$ 与有效带宽资源 $B_n^{\text{res}}(t)$. 其次, 特征工程将抽象资源转化为执行度量: 在存储维度, 利用节点部署模型对应的 KV 缓存系数 $M_{\text{KV}}(m)$ 将剩余显存折算为 token 口径的 $T_{\text{KV}}^{\text{total}}$、$T_{\text{KV}}^{\text{used}}$ 与 $T_{\text{KV}}^{\text{left}}$; 在执行维度, 则依据式 (10) 量化由当前批次 $\mathcal{B}_n(t)$ 决定的计算时延 $T_{\text{comp}}(\mathcal{B}_n(t))$ 与访存时延 $T_{\text{mem}}(\mathcal{B}_n(t))$. 最后, 为增强稳健性, 状态向量集成反映系统瓶颈的紧张度指标. 其中, 时延紧张度定义为式 (11) 给出的 $T_{\text{batch}}^{\text{peak}}(\mathcal{B}_n(t))$ 与批次内最严 SLO $\min_{j \in \mathcal{B}_n(t)} \tau_j^{\text{TPOT}}$ 之比; 显存紧张度定义为式 (8) 左侧风险调整峰值显存与 M_n^{res} 之比. 因而, 高阶状态特征均可分别追溯到式 (1) 和 (2) 的动态物理资源、式 (10) 的执行时延以及式 (8) 和 (11) 的风险边界.

为了消除 AI-RAN 环境中不同物理量纲对神经网络梯度更新的影响, 本文在输入层引入了动态状态归一化机制. 由于状态特征的数值跨度巨大, 直接输入会导致神经元激活饱和. 为此, 智能体实时维护状态特征的滑动

均值 μ_{s} 与方差 σ_{s}^2, 并在决策前将原始状态映射为标准化特征:

$$\tilde{s}(t)=\mathrm{clip}\left(\frac{s(t)-\mu_{\mathrm{s}}}{\sqrt{\sigma_{\mathrm{s}}^2+\varsigma}},-c,c\right), \tag{16}$$

其中 ς 为防止除零的小常数, c 为裁剪阈值. 对任务槽位完成归一化后, 再将对应特征乘以 $q_j(t)$. 演员网络的动作采样、对数概率和损失统计均忽略 $q_j=0$ 的位置, 评论家网络也在聚合任务特征前应用同一掩码, 从而将补零位置与策略和价值计算隔离.

4.1.2　动作空间

动作空间 $\mathcal{A}$ 涵盖了调度智能体对当前时隙到达请求集合的负载编排决策. 针对 AI-RAN 环境下推理任务的复杂性, 本文设计了一种异质动作空间模型, 从空间维度的节点选择与时间维度的执行优先级两个层面实现联合优化. 定义联合动作向量为 $\mathbf{a}(t)=[\mathbf{a}_1(t),\mathbf{a}_2(t),\ldots,\mathbf{a}_{J_{\max}}(t)]$, 其中每个请求 j 的动作元组包含离散分量 $a_j^{\mathrm{node}}\in\{0,1,\ldots,N\}$ 以及连续分量 $w_j\in[0,1]$.

在空间维度, 离散分量 a_j^{node} 采用多离散编码方案以表征任务的归属决策. 具体而言, 当 $a_j^{\mathrm{node}}=0$ 时, 表示智能体决定不分配 AI-RAN 系统资源, 将该任务直接转发至云端处理; 当 $a_j^{\mathrm{node}}=n$ (其中 $n\in\{1,\ldots,N\}$) 时, 则表示将任务 j 分配给对应的边缘节点 n. 该设计将原始 $N\times J_{\max}$ 个相互关联的二进制变量改写为 $J_{\max}$ 个分类变量, 从参数化层面直接满足任务归属唯一性约束, 避免独立二进制采样产生多节点重复分配等无效编码. 两种表示的可行联合分配数均为 $(N+1)^{J_{\max}}$, 分类表示的主要作用是降低动作输出冗余和无效采样比例. 在时间维度, 引入连续动作分量 w_j 以表征任务在当前时隙资源竞争中的相对处理优先级. 在边缘节点资源不足以支撑所有分配请求时, w_j 表示同一时隙新请求的准入顺序. 系统先依据 a_j^{node} 对新请求分组; 对每个节点, 按 w_j 降序排列候选请求, 并依次将其暂加入当前批次. 每次暂加入后检查模型匹配、式 (8) 的显存约束以及式 (12) 对批次内所有请求的最严 SLO 约束; 仅当三者均满足时才正式准入, 否则将该请求转发至云端.

4.1.3　奖励函数

AI-RAN 环境下的负载编排具有显著的前瞻性, 但业务结算与 SLO 违约反馈存在滞后性. 由于 LLM 推理跨时隙且受环境演变影响, 仅靠即时奖励易导致智能体因短视而忽视资源波动与自回归特性, 易引发后期大规模违约; 而仅依赖后端结算则会面临信号稀疏与信度分配难题, 抑制收敛速度与精度. 为此, 本文构建了即时决策与过程价值协同的复合奖励机制, 驱动智能体在保证服务约束的前提下, 实现长期累积服务价值的最大化.

即时决策奖励 $r_{\mathrm{imm}}(t)$ 用于对调度智能体在负载编排瞬时的决策质量进行初步评价, 针对当前时隙对齐后的每一个任务 $j\in\{1,\ldots,J_{\max}\}$, 其即时回报 $r_j(t)$ 依据分配结果呈现为分段形式. 当任务被成功编排至目标节点且通过资源约束初步检查时, 系统赋予正向激励 $\omega_{\mathrm{suc}}\bar{V}_j$, 表征准入决策对系统收益的预期贡献. 若智能体选择将任务转发至云端 (即 $a_j(t)=0$), 系统则施加边际惩罚项 $-\omega_{\mathrm{rej}}\bar{V}_j$, 以抑制策略产生过度保守的调度倾向. 而对于因编排失当导致违反时延约束的 SLO 违约情况, 系统将施加严厉的违约惩罚 $-\omega_{\mathrm{fail}}\bar{V}_j$, 从而有效抑制盲目准入带来的资源风险与服务失效.

为了进一步刻画推理执行过程中的实际经济增益并解决延迟反馈问题, 本文引入了过程价值结算 $\Gamma_n(t)$. 该反馈机制通过实时监测各边缘节点在执行连续批处理任务时的物理产出, 量化编排方案在底层的实际价值实现程度. 节点 n 在时隙 t 产生的实际回报 $\Gamma_n(t)$ 定义为该节点内所有活跃请求所贡献的生成价值:

$$\Gamma_n(t)=\sum_{j\in\mathcal{B}_n(t)}p_j^{\mathrm{out}}\cdot l_j^{\mathrm{out}}(t), \tag{17}$$

其中 $l_j^{\mathrm{out}}(t)$ 表示任务 j 在当前时隙实际生成的 token 数量. 该公式使得奖励信号能够精准反映节点底层计算核心在当前时隙的有效产出效率. 综上所述, 时隙 t 的整体奖励函数 R_t 形式化表示为即时编排回报与过程价值结

算的加权总和:

$$R_t = \sum_{j=1}^{J_{\max}} r_j(t) + \lambda_{\text{tp}} \sum_{n\in\mathcal{N}} \Gamma_n(t), \tag{18}$$

其中 λ_{tp} 为平衡即时决策与长期执行效率的加权系数. 该复合奖励机制通过将物理约束硬化为即时惩罚信号, 并将基于价格向量的长期价值转化为动态引导, 迫使智能体在动态波动的 AI-RAN 环境中科学权衡负载编排的即时收益与系统长效处理的稳健性.

4.2　基于 BCH-PPO 的 LLM 推理负载编排算法

针对 AI-RAN 环境下异构 LLM 推理负载编排的复杂性, 本文提出一种基于行为克隆引导的异构近端策略优化算法 BCH-PPO, 伪代码如算法 1 所示, 算法架构如图 2 所示. 该算法通过异质策略表征实现对节点选择与执行优先级的联合建模, 并采用离线预训练与在线带约束微调的序列化训练流程. 其核心思想是构建一个能够同时处理离散与连续决策的复合策略架构, 并利用启发式规则的稳健性为智能体提供高质量的初始搜索点, 最终通过近端策略优化 (PPO) 算法 [26] 在动态环境中实现策略的自主调优与性能提升.

算法 1 Behavior cloning-guided heterogeneous PPO (BCH-PPO).

输入: 策略参数 θ, 价值函数参数 ϕ, 专家策略 π_{E}, 裁剪参数 ϵ_{clip}, 系数 $c_{\text{V}}, c_{\text{ent}}, \lambda_{\text{BC}}$, 学习率 $\alpha_\theta, \alpha_\phi$;
主迭代:
阶段 I: 基于专家示范的策略预训练
初始化专家数据集 $\mathcal{D}_{\text{E}} \leftarrow \emptyset$;
while 预训练未收敛 **do**
　构造重排状态 $\tilde{s}$、可行性掩码 g 与专家标签 $(a^{\text{E}}, w^{\text{E}})$;
　将 $(\tilde{s}, g, a^{\text{E}}, w^{\text{E}})$ 存入 $\mathcal{D}_{\text{E}}$, 并通过 $\mathcal{L}_{\text{BC}}$ 更新 θ;
end while
阶段 II: 融合行为克隆约束的 PPO 在线更新
初始化 $\theta_{\text{old}} \leftarrow \theta$;
for 迭代轮次 $= 1, 2, \ldots$ **do**
　在 AI-RAN 环境中运行 $\pi_{\theta_{\text{old}}}$ 共 T 步, 收集轨迹 τ、动作 $(a_{j,t}^{\text{node}}, w_{j,t})$ 及专家影子标签 $(a_{j,t}^{\text{E}}, w_{j,t}^{\text{E}})$;
　通过 GAE 计算 $\hat{A}_t$ 与 $\hat{R}_t$;
　for 轮次 $= 1, \ldots, K$ **do**
　　计算 $r_t(\theta)$ 与 $\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{policy}}^{\text{CLIP}} + c_{\text{V}}\mathcal{L}_{\text{V}} - c_{\text{ent}}\mathcal{H} + \lambda_{\text{BC}}\mathcal{L}_{\text{BC}}$;
　　利用 $\nabla\mathcal{L}_{\text{total}}$ 更新 θ 与 ϕ;
　end for
　更新 $\theta_{\text{old}} \leftarrow \theta$;
end for
输出: 优化后的策略参数 θ 与价值函数参数 ϕ.

4.2.1　异质策略表征与双头采样机制

为了支持异质动作空间 $\mathcal{A}$, 本文构建了双头策略网络架构. 对于状态观测 $\tilde{s}$, 策略网络并行输出离散与连续分布以实现协同采样. 在离散动作生成分支中, 针对每个请求 j 的节点选择决策, 网络输出经由结合可行掩码 g_j 的 masked-Softmax 得到分类分布 (categorical distribution) 的概率向量. 在训练阶段, 算法通过对该分布进行采样获得离散动作 a_j^{node}, 并记录其对数概率 $\log \pi_\theta^{\text{node}}(a_j^{\text{node}} \mid \tilde{s}, g_j)$ 为后续策略梯度更新提供依据.

在连续动作生成分支中, 针对执行优先级 w_j 的表征, 模型通过网络预测每个请求对应的状态相关均值 $\mu_{\theta,j}(\tilde{s})$, 并引入一个全局共享且可学习的标准差参数 σ 来构建正态分布 $\mathcal{N}(\mu_{\theta,j}(\tilde{s}), \sigma^2)$. 与均值 $\mu_{\theta,j}$ 不同, 标准差 σ 不随输入状态动态变化, 而是作为策略参数的一部分在训练过程中跨样本同步优化, 从而稳定探索尺度并提升收敛性. 在动作采样阶段, 优先级由正态分布直接采样并进行裁剪以满足物理有效性约束:

$$w_j = \text{clip}(\mu_{\theta,j}(\tilde{s}) + \sigma \cdot \epsilon,\ -c_p,\ c_p), \quad \epsilon \sim \mathcal{N}(0,1), \tag{19}$$

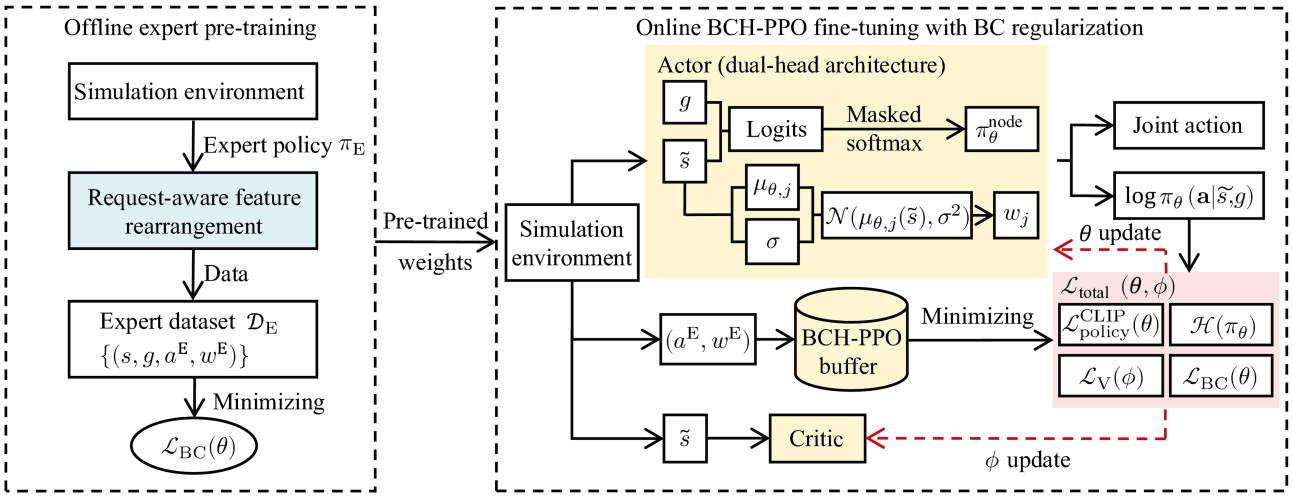

图 2　(网络版彩图) 算法架构.

Figure 2　(Color online) Algorithm architecture.

其中 c_p 为预设的优先级裁剪阈值.

在 BCH-PPO 的概率比率计算中, 离散节点选择与连续优先级两部分共同构成长度归一化的对数策略代理量. 为简化记号, 以下省略时隙下标. 令 $q_j \in \{0,1\}$ 表示请求位置掩码, 其中 $q_j = 1$ 表示第 j 个位置对应真实请求, $q_j = 0$ 表示补零位置; 令 $g_{j,a} \in \{0,1\}$ 表示离散动作 $a \in \mathcal{A}_j$ 对请求 j 是否可行, 并记 $\mathbf{g}_j = (g_{j,a})_{a\in\mathcal{A}_j}$. 进一步定义有效位置掩码 $v_j = q_j \mathbb{I}\left[\sum_{a\in\mathcal{A}_j} g_{j,a} > 0\right]$. 因此, $v_j = 1$ 当且仅当位置 j 为真实请求且至少存在一个可行动作. 对于 $\sum_{j=1}^{J_{\max}} v_j > 0$ 的样本, 长度归一化的异质动作对数策略代理量定义为

$$\ell_\theta^{\text{norm}}(\mathbf{a} \mid \tilde{s}, \mathbf{g}) = \frac{\sum_{j=1}^{J_{\max}} v_j \log \pi_\theta^{\text{node}}\left(a_j^{\text{node}} \mid \tilde{s}, \mathbf{g}_j\right)}{\sum_{j=1}^{J_{\max}} v_j} + \frac{\sum_{j=1}^{J_{\max}} v_j \log \mathcal{N}\left(w_j; \mu_{\theta,j}(\tilde{s}), \sigma^2\right)}{\sum_{j=1}^{J_{\max}} v_j}. \tag{20}$$

式 (20) 中除以 $\sum_j v_j$ 并非对每个请求的节点选择概率重新归一化. 节点选择概率已由 masked-Softmax 在动作维度完成归一化, 此处仅在请求槽位维度对异质动作的对数似然取平均. 若对有效位置的对数概率求和, 当有效请求数由 5 增至 25 时, 对数策略差、PPO 比率及策略梯度的尺度将随请求数增大, 使相同的学习率和裁剪阈值对应不同的实际更新强度, 并使 PPO 裁剪目标更易进入饱和区间. 采用每个有效请求的平均对数似然后, 不同并发请求数下的策略统计量具有可比尺度, 从而减小请求数量变化引起的额外梯度幅值波动.

其中, v_j 用于排除补零或无可行动作的位置, 而 $\sum_j v_j$ 仅用于稳定不同有效长度下的聚合尺度, 并不参与单个连续动作分布的参数化. 因此, 全局共享标准差仍为 σ, 无需按照 $\sum_j v_j$ 对其进行缩放, 以避免对连续动作的探索方差重复尺度化. 当 $\sum_j v_j = 0$ 时, 当前时隙不存在有效策略动作, 因而不将该样本计入演员网络更新.

4.2.2　基于专家示范的策略离线预训练

为解决冷启动问题, 本文利用启发式专家策略 π_E 交互生成示范数据集 $\mathcal{D}_E$. 预训练阶段采用逐请求目标化方式构造监督元组 (s, g, a^E, w^E), 仅对观测向量首位的目标请求提供有效标签, 其余位置补零作为上下文. 核心逻辑在于请求感知特征重排: 通过位置置换将目标请求属性固定置于观测首位, 其余请求及环境特征置于后侧. 这种以任务为中心的决策视图, 使得共享参数的策略网络能通过动态调整输入顺序聚焦首位请求的目标特征, 从而针对不同属性的请求输出差异化的调度指令.

在标签生成阶段, 专家策略采用最小延迟增量优先 (minimal latency increment, MLI) 规则. 对目标请求 j, 首先筛选满足部署条件的候选节点. 将请求 j 暂时加入节点 n 的批次后, 定义其峰值解码时延增量为

$$\Delta T_{n,j} = T_{\text{batch}}^{\text{peak}}\left(\mathcal{B}_n(t) \cup \{j\}\right) - T_{\text{batch}}^{\text{peak}}\left(\mathcal{B}_n(t)\right). \tag{21}$$

仅当加入请求后仍满足式 (8) 的概率显存约束和式 (12) 的最严 SLO 约束时, 节点 n 才进入请求 j 的可行集合 $\mathcal{N}_j^{\mathrm{fea}}$. 若该集合非空, MLI 专家选择峰值时延增量最小的节点, 即

$$a_j^{\mathrm{E}} = \arg\min_{n\in\mathcal{N}_j^{\mathrm{fea}}} \Delta T_{n,j}. \tag{22}$$

当多个节点具有相同的时延增量时, 选择节点编号较小者以保证标签生成的确定性. 若 $\mathcal{N}_j^{\mathrm{fea}} = \emptyset$, 则令 $a_j^{\mathrm{E}} = 0$, 将该请求转发至云端. 为了使连续优先级标签与 MLI 原则保持一致, 定义请求 j 的最小可行时延增量为

$$\delta_j = \min_{n\in\mathcal{N}_j^{\mathrm{fea}}} \Delta T_{n,j}. \tag{23}$$

记当前时隙存在可行节点的请求集合为 $\mathcal{J}^{\mathrm{fea}}(t)$, 并定义

$$\delta_{\min} = \min_{k\in\mathcal{J}^{\mathrm{fea}}(t)} \delta_k, \qquad \delta_{\max} = \max_{k\in\mathcal{J}^{\mathrm{fea}}(t)} \delta_k.$$

则连续优先级标签定义为

$$w_j^{\mathrm{E}} = 1 - \frac{\delta_j - \delta_{\min}}{\delta_{\max} - \delta_{\min} + \varsigma}, \qquad j \in \mathcal{J}^{\mathrm{fea}}(t), \tag{24}$$

其中, $\varsigma > 0$ 为防止分母为 0 的微小常数. 因此, 引起较小峰值时延增量的请求具有更高的准入优先级; 对于不存在可行节点的请求, 令 $w_j^{\mathrm{E}} = 0$. 当优先级相同时, 依次按照请求到达时间和请求编号确定顺序. 通过最小化联合行为克隆损失函数 $\mathcal{L}_{\mathrm{BC}}(\theta)$, 使智能体初步习得基础的调度逻辑, 其中 $\mathcal{L}_{\mathrm{BC}}(\theta)$ 定义如下:

$$\mathcal{L}_{\mathrm{BC}}(\theta) = \mathbb{E}_{(s,m,a^{\mathrm{E}},w^{\mathrm{E}})\sim\mathcal{D}_{\mathrm{E}}}\left[-\frac{\sum_{j=1}^{J_{\max}} v_j e_j \log \pi_\theta^{\mathrm{node}}(a_j^{\mathrm{E}} \mid \tilde{s}, m)}{\sum_{j=1}^{J_{\max}} v_j e_j + \varepsilon} + \frac{\sum_{j=1}^{J_{\max}} v_j \left\| \mu_{\theta,j}(\tilde{s}) - w_j^{\mathrm{E}} \right\|^2}{\sum_{j=1}^{J_{\max}} v_j + \varepsilon} \right], \tag{25}$$

其中第一项为针对离散选择的交叉熵损失, 第二项为针对连续优先级的均方误差损失. 这里 q_j 表示真实请求, $v_j = q_j \mathbb{I}[\sum_a g_{j,a} > 0]$ 表示该真实请求存在可行动作, $e_j = \mathbb{I}[a_j^{\mathrm{E}} \geqslant 0]$ 表示该位置存在专家标签. 在逐请求预训练样本中, 通常仅首位目标请求满足 $q_j = v_j = e_j = 1$, 其余位置仅作为带掩码的上下文. 这种预训练机制不仅赋予了智能体稳健的初始性能, 也为后续的在线演化提供了高质量的初始策略梯度.

4.2.3　融合行为克隆约束的 PPO 在线更新

在在线交互阶段, 智能体在仿真环境中采样得到 on-policy 轨迹, 并利用 PPO 的裁剪目标函数对策略进行微调. 为在探索新策略的同时不丧失专家先验知识, 本文将行为克隆项作为正则化约束引入在线更新的总损失函数中. 总优化目标 $\mathcal{L}_{\mathrm{total}}(\theta,\phi)$ 定义为策略裁剪项、价值函数误差、熵正则以及行为克隆约束的加权和:

$$\mathcal{L}_{\mathrm{total}}(\theta,\phi) = \mathcal{L}_{\mathrm{policy}}^{\mathrm{CLIP}}(\theta) + c_{\mathrm{V}}\mathcal{L}_{\mathrm{V}}(\phi) - c_{\mathrm{ent}}\mathcal{H}(\pi_\theta) + \lambda_{\mathrm{BC}}\mathcal{L}_{\mathrm{BC}}(\theta). \tag{26}$$

策略裁剪项采用标准 PPO-Clip 形式:

$$\mathcal{L}_{\mathrm{policy}}^{\mathrm{CLIP}}(\theta) = -\mathbb{E}\Big[\min\big(r(\theta)\hat{A},\ \mathrm{clip}(r(\theta), 1-\epsilon_{\mathrm{clip}}, 1+\epsilon_{\mathrm{clip}})\hat{A} \big) \Big], \tag{27}$$

其中比率项 $r(\theta)$ 基于异质联合动作的对数概率计算. 令联合对数概率 $\log \pi_\theta(\mathbf{a}_t \mid \tilde{s}_t, m_t)$ 按式 (20) 由离散节点选择与连续优先级两部分聚合可得

$$r(\theta) = \exp\Big(\log \pi_\theta(\mathbf{a} \mid \tilde{s}, m) - \log \pi_{\theta_{\mathrm{old}}}(\mathbf{a} \mid \tilde{s}, m) \Big). \tag{28}$$

式 (27) 中, 优势信号 $\hat{A}$ 由标准广义优势估计计算得到并在小批量内进行标准化, 回报目标 $\hat{R}$ 由优势与价值基线构造. 价值损失项用于回归回报目标, 采用均方误差形式. 熵正则项 $\mathcal{H}(\pi_\theta)$ 用于维持探索并防止策略过早收

敛. 针对离散节点选择与连续优先级的异质特性, 实现中对离散分类熵与连续正态熵进行聚合, 并仅在有效请求位置 v_j 上进行统计:

$$\mathcal{H}(\pi_\theta)=\frac{\sum_{j=1}^{J_{\max}} v_j\left[-\sum_{a\in\mathcal{A}_j}\pi_\theta^{\text{node}}(a\mid\tilde{s},m_j)\log\pi_\theta^{\text{node}}(a\mid\tilde{s},m_j)+\frac{1}{2}\left(\log(2\pi\mathrm{e}\sigma^2)\right)\right]}{\sum_{j=1}^{J_{\max}} v_j},\tag{29}$$

其中, 第一项为分类分布的香农熵, 通过最大化该项鼓励智能体尝试不同的节点分配方案; 第二项为正态分布的微分熵, 由可学习的标准差 σ 决定. 通过这种异质熵的协同优化, 算法能够在离散路径选择与连续权重调度两个维度上保持稳健的搜索能力.

值得注意的是, 即使在预训练结束后, 仿真环境仍会持续为在线决策生成专家影子标签 $(a^{\mathrm{E}},w^{\mathrm{E}})$ 并随 on-policy 数据一并存入 BCH-PPO 缓冲区. 在线行为克隆正则项沿用预训练阶段的监督形式, 由离散节点的交叉熵与连续优先级的均方误差共同构成, 并在掩码约束下仅对有效位置计入损失, 从而在策略优化过程中形成稳定的策略锚点, 减少高维探索带来的性能震荡.

4.3　复杂度分析

深度强化学习算法的计算复杂度主要与神经网络的层数和每层神经元的个数有关. 在本文构建的 AI-RAN 负载编排场景中, 智能体采用 BCH-PPO 算法, 其复杂度主要由演员网络和评论家网络的前向传播与反向传播开销构成. 演员网络负责将包含任务特征 $\Psi_{\text{task}}(t)$ 与网络资源状态 $\mathbf{C}_{\text{node}}(t)$ 的高维状态空间映射为异质联合动作的参数化输出, 其输入维度受最大并发请求数 $J_{\max}$ 影响. 若该网络由 L_π 层全连接神经网络构成, 则计算复杂度可表示为 $O(\sum_{l=1}^{L_\pi}C_{l-1}^{\pi}C_l^{\pi})$, 其中 C_l^{π} 表示网络第 l 层神经元的个数. 与此同时, 评论家网络用于估算当前状态的长期累积收益以指导策略优化, 其复杂度同样取决于其隐藏层架构, 表示为 $O(\sum_{l=1}^{L_v}C_{l-1}^{v}C_l^{v})$, 其中 L_v 和 C_l^{v} 分别为该网络的层数和每层神经元数量.

记 BCH-PPO 的总训练回合数为 M, 每个回合包含 T 个时间步, 每个更新周期进行 K 轮迭代优化, 则训练阶段的总计算复杂度为 $O\left(MTK\left(\sum_{l=1}^{L_\pi}C_{l-1}^{\pi}C_l^{\pi}+\sum_{l=1}^{L_v}C_{l-1}^{v}C_l^{v}\right)\right)$. 由上式可知, 训练复杂度由训练规模因子 M、T、K 和演员 – 评论家网络的规模共同决定. 在网络结构固定时, 计算量随 M、T 和 K 线性增长; 若各隐藏层的宽度近似为 C, 即 $C_l^{\pi}\approx C, C_l^{v}\approx C$, 则网络计算量随层数近似线性增长、随网络宽度近似平方增长, 因此隐藏层宽度是影响计算开销的主要网络结构参数.

在测试阶段, 调度智能体仅执行演员网络的前向传播, 其计算复杂度为 $O\left(\sum_{l=1}^{L_\pi}C_{l-1}^{\pi}C_l^{\pi}\right)$. 因此, 在线决策开销主要取决于演员网络的层数和各层宽度; 第 5.3 小节的微基准实验将进一步验证实际在线响应时延.

5　仿真与分析

本节通过大量仿真实验验证所提基于 BCH-PPO 的 LLM 推理负载编排算法在 AI-RAN 场景下的性能. 首先, 详细说明了仿真参数的设置. 然后, 将 BCH-PPO 的性能与对比基线在收敛性、成功率、拒绝率、违约率和系统总价值上进行比较.

5.1　仿真参数设置

本文构建了一个离散时隙的 AI-RAN 推理编排仿真环境, 时隙粒度设为 $\Delta t=1\,\mathrm{s}$, 旨在对齐 O-RAN 架构中近实时无线接入网智能控制器 (near-RT RIC) 的控制时间尺度. 系统共部署 $N=4$ 个基站节点, 涵盖宏基站与微基站两类, 其具体的资源配置参数详见表 1. RAN 负载利用服从 Weibull 分布的到达间隔生成时间序列 [27]. 宏基站的形状参数分别为 0.6 和 0.7, 微基站分别为 1.0 和 1.2; 尺度参数通过迭代校准, 使原始负载的均值/峰值比约为 0.3, 随后以各节点峰值归一化得到 $\rho_n(t)\in[0,1]$. RAN 负载对计算和带宽的资源转换系数设为 $\alpha=\beta=1$. AI 推理负载支持 Qwen3-14B、Qwen3-32B 以及自适应模型选择三类模式, 请求到达过程建模为速率为 10 s^{-1} 的泊松过程, 等价地, 相邻请求的到达间隔服从参数为 10 s^{-1} 的指数分布. 请求的输入长度 l_j^{in} 与输出长度均值

表 1　仿真中的节点配置.

Table 1　Node configurations in simulation.

Node type	Simulated GPU	Memory M_n (GB)	Compute G_n (TFLOPS)	Bandwidth B_n (GB/s)	RAN reserved M_n^{RAN} (GB)	Model set $\mathcal{S}_n$
Macrocell	H100	80	1979	3352	4	{Qwen3-32B}
Microcell	A100-40G	40	312	1555	2	{Qwen3-14B}

表 2　AI 推理工作负载定价.

Table 2　Pricing of AI inference workload.

Model $\mathcal{M}_j$	Input price p_j^{in}	Output price under SLO	
		$\tau_j^{\mathrm{TPOT}} = 0.03$ s	$\tau_j^{\mathrm{TPOT}} = 0.10$ s
Qwen3-14B	1.0	12.0	4.0
Qwen3-32B	2.0	24.0	8.0
Auto	1.5	18.0	6.0

表 3　关键仿真设置与 BCH-PPO 超参数.

Table 3　Simulation settings and BCH-PPO hyperparameters.

Parameter	Value	Parameter	Value
ϵ_{dec}	0.001	ϵ_{risk}	0.01
λ_{tp}	1	λ	0.95
ϵ_{clip}	0.1	c_{V}	0.5
c_{ent}	0.001	λ_{BC}	0.1
Learning rate	3×10^{-4}	Mini-batch size	2048
Update epochs	4	Gradient clipping	0.5

μ_j 均由 BurstGPT 数据集 [28] 逐请求抽样获得. 在实际部署中, μ_j 和 σ_j^2 可由不确定性感知的请求级输出长度预测器获得 [23]. 仿真中设置 $\sigma_j = 0.2\mu_j$, 并从截断于正整数范围的正态分布采样实际输出长度 l_j^{out}. 此外, SLO 设定为两档约束, 即 $\tau_j^{\mathrm{TPOT}} \in \{0.03, 0.1\}$ s, 对应的价格向量 $\mathbf{p}_j$ 随模型规格与 SLO 档位动态变化, 如表 2 所示.

本文基于 PyTorch 平台实现 BCH-PPO 智能体及对比基线, 并统一采用 Adam 优化器进行参数更新. BCH-PPO 智能体中的演员网络与评论家网络均由两个包含 256 个神经元的全连接隐藏层组成, 并使用 Tanh 作为激活函数. 在训练过程中, 每个回合设置为 100 个时间步. 为减小不同奖励分量的尺度差异, 本文依据训练轨迹对即时决策回报与过程价值进行归一化, 并仅利用训练集和独立验证集选择关键权重, 从候选配置中选择系统总价值较高者. 据此设置 $\lambda_{\mathrm{tp}} = 1$, $\lambda_{\mathrm{BC}} = 0.1$. 其余关键超参数如表 3 所示. 本文引入强化学习与启发式两类基准算法进行对比: 强化学习基线包括引入熵正则化以提升探索鲁棒性的 DSAC [29] 以及结合了确定性策略梯度与策略网络平滑技术的 DDPG [30], 启发式基线则涵盖按请求期望价值降序分配的最大价值优先策略 (highest value first, HVF)、旨在最小化节点峰值时延增量的最小延迟增量优先策略 (minimal latency increment, MLI), 以及在候选节点集合内按模型类型进行循环指派的轮询策略 (round-robin, RR).

5.2　性能分析

本文从收敛性、成功率、拒绝率、违约率及系统总价值 5 个方面评估 BCH-PPO 的性能. 三类比例均以到达请求总数为分母: 成功率表示边缘完成且全生命周期满足 SLO 的请求比例, 拒绝率表示未被边缘准入并转发云端的比例, 违约率表示已准入但至少一次未满足 SLO 的请求比例, 因而三者之和为 1. 在比较不同准入倾向的策略时, 还需结合条件违约率 Pr(违约 | 已准入) 理解风险差异. 图 3 给出了 AI-RAN 资源剧烈波动环境下, BCH-PPO 与 PPO、DDPG、DSAC 的平均回合奖励收敛曲线. 结果表明, BCH-PPO 在收敛速度和稳态奖励上

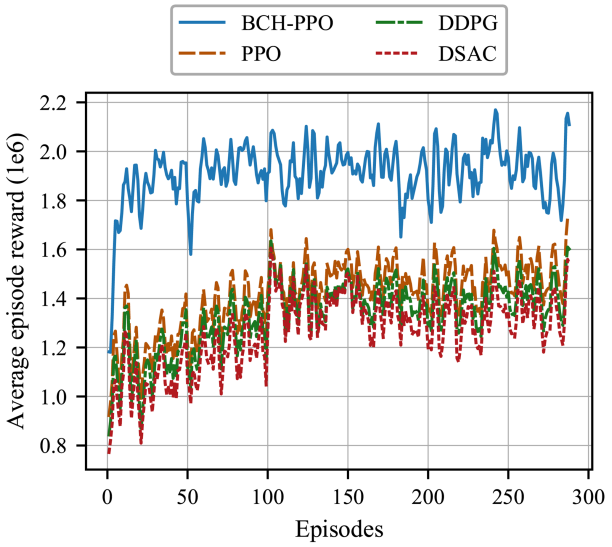

图 3　(网络版彩图) 不同算法在 AI-RAN 资源波动环境下的收敛性能.

Figure 3　(Color online) Convergence performance of different algorithms in the resource-fluctuating AI-RAN environment.

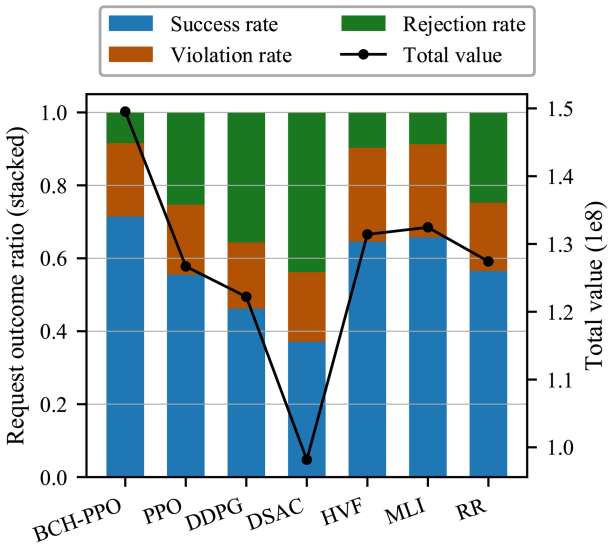

图 4　(网络版彩图) 不同算法在 AI-RAN 资源波动环境下的性能对比 (成功率、拒绝率、违约率和系统总价值).

Figure 4　(Color online) Performance comparison (success rate, rejection rate, unconditional violation rate, and total value) of different algorithms in the resource-fluctuating AI-RAN environment.

均占优, 在最后 50 个回合的平均奖励较 PPO、DDPG 和 DSAC 提升 27.6%~43.5%. 从机制上看, 专家预训练为策略提供满足基本可行约束的初始动作分布, 在线行为克隆正则可抑制资源突变导致的大幅策略漂移、双头策略和可行性掩码则减少无效动作.

图 4 展示了 AI-RAN 资源剧烈波动环境下, BCH-PPO 与 PPO、DDPG、DSAC 及 HVF、MLI、RR 在测试集上的成功率、拒绝率、违约率和系统总价值. 结果表明, BCH-PPO 的系统总价值为 1.495×10^{8}, 较各基线提高 12.9%~52.3%. 相比 PPO、DDPG 和 DSAC, 其成功率提高 15.9%~34.3%, 拒绝率降低 16.9%~35.4%. 相比 HVF 和 MLI, BCH-PPO 的成功率分别提高 6.9% 和 5.7%, 拒绝率分别降低 1.3% 和 0.3%, 且总价值分别提高 13.8% 和 12.9%. 同时, BCH-PPO 的违约率相对二者分别降低 21.7% 和 21.2%, 说明其在准入率相近的情况下实现了更高的成功率和系统价值. 该优势来自对显存机会约束、节点分配和执行优先级的联合优化, 使其能在资源受限时优先接纳高价值任务. 虽然 BCH-PPO 的违约率较 PPO、DDPG 等纯强化学习方法略高 4.9%~10.8%, 但较 HVF 和 MLI 仍降低 21% 以上, 说明其在接纳更多请求、降低拒绝率和控制风险之间取得了更优折中. 相比之下, HVF 和 MLI 缺乏对动态负载与长期风险的刻画, 虽然接纳率较高, 但单位准入带来的总价值较低; DDPG 和 DSAC 则因缺乏专家引导和耦合建模而更保守, 拒绝率均超过 35%. 因此, BCH-PPO 在资源波动环境下表现出更强的在线调度稳健性与系统收益优势.

本文进一步在理想的 AI-RAN 资源固定环境下评估算法性能, 并与资源波动场景进行对照. 由于资源供给更稳定, 各算法曲线波动减弱, 调度重点由高不确定风险控制转向稳定资源下的价值分配, 但策略差异仍然清晰. 图 5 展示了资源固定环境下 BCH-PPO 与 PPO、DDPG、DSAC 的平均回合奖励收敛曲线. 在最后 50 回合, BCH-PPO 的平均回合奖励较 PPO、DDPG、DSAC 提升 5.6%~18.8%, 说明专家引导与双头异构动作建模在平稳场景中仍能持续带来收益优势. 与资源波动环境相比, 其对 PPO 的提升有所收敛, 这是因为环境随机性下降后, 基线算法的无效探索减少; 但 BCH-PPO 的优势依然保留. 图 6 展示了资源固定环境下 BCH-PPO 与 PPO、DDPG、DSAC 及 HVF、MLI、RR 在测试集上的性能对比. BCH-PPO 的成功率为 95.0%, 高于其他基线方法 5.6%~12.4%; 对应拒绝率仅 5.0%, 远低于其他基线方法. 在系统总价值方面, BCH-PPO 较 PPO、DDPG、DSAC 提升 6.6%~30.6%, 同时高于 HVF、MLI、RR. 此外, 固定资源设定下各方法违约率均为 0,

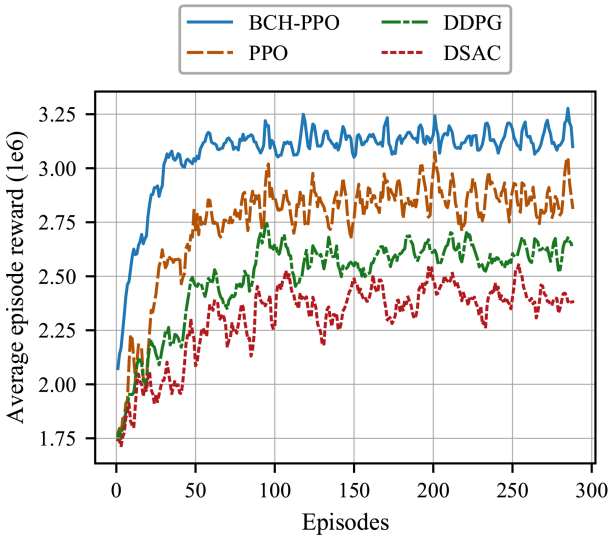

图 5　(网络版彩图) 不同算法在 AI-RAN 资源固定环境下的收敛性能.

Figure 5　(Color online) Convergence performance of different algorithms in the fixed-resource AI-RAN environment.

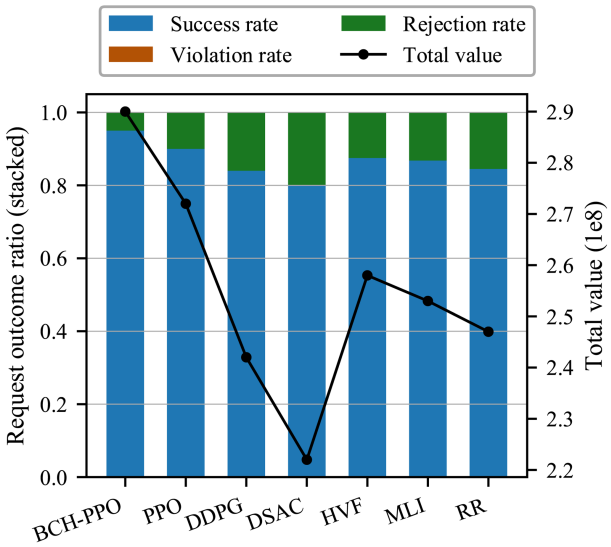

图 6　(网络版彩图) 不同算法在 AI-RAN 资源固定环境下的性能对比 (成功率、拒绝率、违约率和系统总价值).

Figure 6　(Color online) Performance comparison (success rate, rejection rate, violation rate, and total value) of different algorithms in the fixed-resource AI-RAN environment.

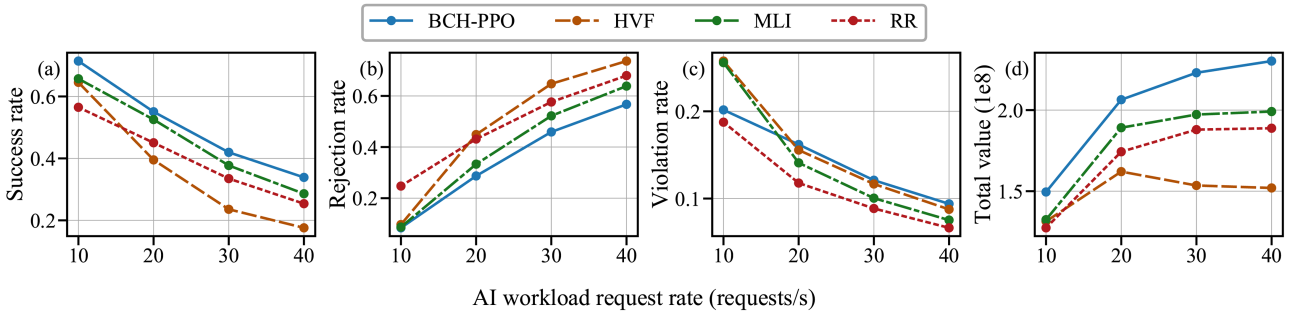

图 7　(网络版彩图) AI 负载请求率对不同算法的 (a) 成功率, (b) 拒绝率, (c) 无条件违约率, (d) 总价值的影响.

Figure 7　(Color online) The impact of the AI workload request rate on the (a) success rate, (b) rejection rate, (c) unconditional violation rate, and (d) total value of different algorithms.

说明系统瓶颈主要体现在准入决策; 在此条件下 BCH-PPO 仍保持最高成功率与总价值, 验证了其跨资源模式的泛化稳健性.

图 7 给出了资源波动环境下 AI 负载请求率由每秒 10 请求增至每秒 40 请求时, BCH-PPO 与 HVF、MLI、RR 的性能变化. 随着负载增加, 各方法的成功率下降、拒绝率上升, 表明系统逐步进入容量受限区间; 但 BCH-PPO 在各请求率下始终保持最优综合收益. 具体来看, BCH-PPO 的成功率由 72.9% 降至 33.9%, 拒绝率由 8.1% 升至 56.7%, 反映出高负载下准入竞争显著加剧. 与此同时, 其违约率由 20.2% 降至 9.4%, 说明策略会在高负载阶段主动收紧接纳边界, 优先保留更可能满足约束的请求. 在系统总价值方面, BCH-PPO 随请求率提升而持续增加, 且相较同请求率下最优启发式基线始终领先 9.1%~15.6%. 因此, 在资源波动与业务激增并存的 AI-RAN 场景中, BCH-PPO 具备更强的跨负载鲁棒性与系统级收益保持能力.

表 4　AI-RAN 资源波动环境下 BCH-PPO 专家引导机制的消融实验.

Table 4　Ablation of the BCH-PPO expert-guidance mechanism in the resource-fluctuating AI-RAN environment.

Method	Success rate (%)	Rejection rate (%)	Violation rate (%)	Total value (10^8)
BCH-PPO without expert guidance	62.18	20.96	16.86	1.3559
BCH-PPO	71.43	8.41	20.16	1.4953

5.3　算法验证

为量化专家引导对最终调度性能的贡献, 本文比较无专家引导 BCH-PPO 与完整 BCH-PPO 在 AI-RAN 资源波动环境下的成功率、拒绝率、违约率和系统价值. 前者保留相同的状态空间、双头网络、特征重排、可行性掩码和 PPO 训练设置, 但不使用 MLI 示范进行离线预训练, 且在线阶段令 $\lambda_{\mathrm{BC}} = 0$; 后者同时采用 MLI 示范预训练和在线行为克隆正则. 两种方法使用相同训练预算和测试轨迹, 以排除网络结构与环境样本差异的影响. MLI 与完整 BCH-PPO 的直接比较已由图 4 给出, 因此不在消融表中重复列示. 由表 4 可见, 引入 MLI 专家示范后, 完整 BCH-PPO 的成功率提高 9.25%, 拒绝率降低 12.55%, 系统总价值提高 10.28%, 违约率提高 3.30%. 这说明专家引导促使策略接纳更多具有较高收益但接近 SLO 边界的请求, 其主要作用是改善准入质量和系统收益, 而非单独降低违约率. 此外, 图 4 中完整 BCH-PPO 的总价值较示范来源 MLI 高 12.89%, 表明最终策略能够利用环境回报进一步超越专家, 未被行为克隆限制在 MLI 的性能水平.

本文调度器承担控制面的负载编排功能, 实际部署时通常运行于由通用 CPU 承载的 near-RT RIC 等控制平台, 无需占用执行 LLM 推理的 GPU 资源. 为验证 BCH-PPO 的在线响应能力, 在 Intel Xeon Silver 4314 CPU (2.40 GHz) 上使用 PyTorch 2.9.0 和单线程执行完整决策链路, 测试规模为 $N = 4$、$J_{\max} = 25$. 在上述硬件、线程配置和请求规模下, BCH-PPO 完整决策链路的平均时延为 4.305 ms, P99 时延为 5.529 ms, 表明其算法侧计算能够在所测设置下提供毫秒级在线决策.

6　结论

本文研究了 6G 通算融合演进背景下 AI-RAN 环境中的异构 LLM 推理负载编排问题, 旨在应对资源动态波动与自回归显存需求不确定性带来的严峻挑战. 首先, 针对 AI-RAN 通算资源竞争及 LLM 连续批处理的非线性特征, 构建系统级资源模型, 并在独立正态近似下利用二阶锥约束刻画随机输出长度带来的显存风险. 其次, 本文提出了一种行为克隆引导的异构近端策略优化算法 (BCH-PPO), 通过设计双头策略网络与任务感知特征重排机制, 实现了节点分配与任务优先级的并行决策, 显著提升了算法在复杂动态环境下的收敛速度与决策稳健性. 最后, 仿真结果从收敛性、成功率、拒绝率、违约率、系统总价值等方面验证了所提算法的有效性. 随着 6G 通算融合的持续演进和生成式 AI 应用的蓬勃发展, 在 AI-RAN 环境下为 LLM 推理提供可靠支持已成为 6G 边缘网络的研究热点. 未来的工作可以进一步探索多模型切片与流水线并行技术在 AI-RAN 中的应用, 以实现更大规模集群下的高能效智能算力调度.

参考文献

1　Saad W, Bennis M, Chen M. A vision of 6G wireless systems: applications, trends, technologies, and open research problems. IEEE Netw, 2019, 34: 134–142

2　Dang S, Amin O, Shihada B, et al. What should 6G be? Nat Electron, 2020, 3: 20–29

3　Wang Q T, Li Z X, Wang Y. AI RAN: technological evolution and key challenges for AI-native radio access network. Radio Commun Technol, 2025, 51: 548–554 [王晴天, 李泽旭, 王越. AI RAN: 面向智能化无线接入网的技术演进与关键挑战. 无线电通信技术, 2025, 51: 548–554]

4　Kundu L, Lin X, Gadiyar R, et al. AI-RAN: transforming RAN with AI-driven computing infrastructure. IEEE Commun Mag, 2025, 64: 168–174

5　SoftBank Corp. AI-RAN: telecom infrastructure for the age of AI. 2024. https://www.softbank.jp/en/corp/set/data/technology/research/topics/126/img/pdf/SoftBank_AI_RAN_Whitepaper_December2024.pdf

6 Wang L, Ma C, Feng X, et al. A survey on large language model based autonomous agents. Front Comput Sci, 2024, 18: 186345
7 Acharya D B, Kuppan K, Divya B. Agentic AI: autonomous intelligence for complex goals—a comprehensive survey. IEEE Access, 2025, 13: 18912–18936
8 Miao X, Oliaro G, Zhang Z, et al. Towards efficient generative large language model serving: a survey from algorithms to systems. ACM Comput Surv, 2025, 58: 1–37
9 Li Z, Wang Q, Wang Y, et al. The architecture of AI and communication integration towards 6G: an O-RAN evolution. In: Proceedings of the 30th Annual International Conference on Mobile Computing and Networking, 2024. 2329–2334
10 Polese M, Mohamadi N, D'Oro S, et al. Beyond connectivity: an open architecture for AI-RAN convergence in 6G. 2025. ArXiv:2507.06911
11 Shah S D A, Hafeez M, Salama A, et al. Proactive AI-and-RAN workload orchestration in O-RAN architectures for 6G networks. IEEE Open J Commun Soc, 2025, 6: 7939–7954
12 Wu X, Farooq J, Wang Y, et al. LLM-xApp: a large language model empowered radio resource management xApp for 5G O-RAN. In: Proceedings of the Workshop on Security and Privacy of Next-Generation Networks (FutureG), San Diego, 2025. 1–5
13 Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need. In: Proceedings of Advances in Neural Information Processing Systems 30, 2017. 5998–6008
14 Yu G I, Jeong J S, Kim G W, et al. Orca: a distributed serving system for Transformer-based generative models. In: Proceedings of the 16th USENIX Symposium on Operating Systems Design and Implementation, 2022. 521–538
15 Kwon W, Li Z, Zhuang S, et al. Efficient memory management for large language model serving with PagedAttention. In: Proceedings of the 29th Symposium on Operating Systems Principles, 2023. 611–626
16 Jaillet P, Jiang J, Mellou K, et al. Online scheduling for LLM inference with KV cache constraints. 2025. ArXiv:2502.07115
17 Sun B, Huang Z, Zhao H, et al. Llumnix: dynamic scheduling for large language model serving. In: Proceedings of the 18th USENIX Symposium on Operating Systems Design and Implementation, 2024. 173–191
18 Huang J, Xiong Y, Yu X, et al. SLO-aware scheduling for large language model inferences. 2025. ArXiv:2504.14966
19 Xu W, Yang Z, Ng D W K, et al. A new path to integrated learning and communication (ILAC): large AI models leveraging hyperdimensional computing. IEEE Trans Commun, 2026, 74: 4948–4973
20 Yang Z, Xu W, Liang L, et al. On privacy, security, and trustworthiness in distributed wireless large AI models. Sci China Inf Sci, 2025, 68: 170301
21 Auer G, Giannini V, Desset C, et al. How much energy is needed to run a wireless network? IEEE Wireless Commun, 2011, 18: 40–49
22 Shah S D A, Nezami Z, Hafeez M, et al. The interplay of AI-and-RAN: dynamic resource allocation for converged 6G platform. In: Proceedings of IEEE INFOCOM Workshops, 2025. 1–6
23 Shi B, Dong B, Zheng Q. Uncertainty-aware large language model response length perception. Sci China Inf Sci, 2025, 68: 210101
24 Sheng Y, Zheng L, Yuan B, et al. FlexGen: high-throughput generative inference of large language models with a single GPU. In: Proceedings of the International Conference on Machine Learning, 2023. 31094–31116
25 Nemirovski A, Shapiro A. Convex approximations of chance constrained programs. SIAM J Optim, 2007, 17: 969–996
26 Schulman J, Wolski F, Dhariwal P, et al. Proximal policy optimization algorithms. 2017. ArXiv:1707.06347
27 Arfeen A, Pawlikowski K, McNickle D, et al. The role of the Weibull distribution in modelling traffic in Internet access and backbone core networks. J Netw Comput Appl, 2019, 141: 1–22
28 Wang Y, Chen Y, Li Z, et al. BurstGPT: a real-world workload dataset to optimize LLM serving systems. In: Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining, 2025. 5831–5841
29 Christodoulou P. Soft actor-critic for discrete action settings. 2019. ArXiv:1910.07207
30 Lillicrap T P, Hunt J J, Pritzel A, et al. Continuous control with deep reinforcement learning. 2015. ArXiv:1509.02971

Workload orchestration for heterogeneous LLM inference in resource-fluctuating AI-RAN environments

Kaiyu HUANG[1,2], Shuai GAO[3], Yuqian HOU[4], Fan XU[3*] & Qingjiang SHI[1,2*]

1. *School of Computer Science and Technology, Tongji University, Shanghai 201804, China*
2. *Shenzhen Research Institute of Big Data, Shenzhen 518172, China*
3. *College of Electronic and Information Engineering, Tongji University, Shanghai 201804, China*
4. *College of Information Science and Electronic Engineering, Zhejiang University, Hangzhou 310027, China*

* Corresponding author. E-mail: xxiaof999@tongji.edu.cn, shiqj@tongji.edu.cn

Abstract As 6G networks evolve toward the deep integration of communication and computing, the artificial intelligence radio access network (AI-RAN) has emerged as a novel edge hosting paradigm for generative AI services, such as large language model (LLM) inference. However, in AI-RAN environments, achieving high-performance and reliable LLM inference faces severe challenges due to the dynamic preemption of computing resources by RAN communication services, as well as the stochastic output lengths and fluctuating resource requirements inherent in the autoregressive generation process of LLMs. To address these issues, this paper proposes a workload orchestration algorithm for heterogeneous LLM inference in resource-fluctuating environments. First, a system model is established that accounts for dynamic resource competition from communication services, non-linear resource consumption of continuous batching, and stochastic output lengths. Furthermore, under an independent Gaussian approximation, memory chance constraints are transformed into a tractable deterministic second-order cone form. Second, the workload orchestration problem, which involves coupled long-term and short-term decisions, is modeled as a Markov decision process (MDP). A multi-dimensional state space covering hardware load and execution risk is designed to balance system utility and violation risk. Finally, a behavior cloning-guided heterogeneous proximal policy optimization (BCH-PPO) algorithm is proposed. By employing a dual-head policy network and a task-aware feature reshuffling mechanism, the algorithm achieves parallel decision-making for node allocation and priority control, thereby enhancing orchestration performance in dynamic environments. Simulation results demonstrate that the proposed algorithm effectively navigates resource fluctuations in AI-RAN environments. While ensuring service level objectives (SLOs) for LLM inference, it significantly enhances system throughput and scheduling performance.

Keywords artificial intelligence radio access network, large language model, workload orchestration, deep reinforcement learning, proximal policy optimization