知识驱动的智能体互联网服务资源自主编排方法

钱琪杰[1,2]，宋阳子[2]，秦蓁[2]，钟旭东[2]，郭永安[1]，任保全[2]
（1.南京邮电大学通信与信息工程学院，江苏 南京 210003；2.军事科学院系统工程研究院，北京 100141）

摘要：虚拟网络嵌入作为核心资源分配手段，在动态网络环境中面临严峻挑战。针对现有深度强化学习方法因隐式知识编码导致的适应性差、可解释性弱等问题，提出一种知识驱动的资源编排框架。该框架将时序知识图谱作为智能体的显式世界模型，实现网络动态知识的结构化表征与推理，进而设计一种混合Actor-Critic智能体，通过双循环机制融合实时状态、历史经验与领域启发式知识，协同实现快速战术适应与长期战略优化。实验表明，所提框架在请求接受率与收益成本比上表现优异，在冻结策略参数下的跨拓扑迁移与在线知识适配场景下，模型由合成拓扑部署至真实拓扑时不需要额外重训练，仍能保持稳定优势，其中在指数业务场景下相较最强接纳率基线提升6.50%，相较部分深度强化学习基线的相对提升超过80%，为智能体互联网的高效可靠服务提供了关键技术支撑。

关键词：智能体互联网；知识驱动；虚拟网络嵌入；资源编排；时序知识图谱
中图分类号：TN915.0
文献标志码：A
doi：10.11959/j.issn.1000–436x.TXXB260271

Knowledge-driven autonomous orchestration method of service resources for the Internet of agents

Qian Qijie[1,2], Song Yangzi[2], Qin Zhen[2], Zhong Xudong[2], Guo Yong'an[1], Ren Baoquan[2]
1. School of Communications and Information Engineering, Nanjing University of Posts and Telecommunications, Nanjing 210003, China
2. System Engineering Institute, AMS PLA, Beijing 100141, China

Abstract: As a core resource allocation method, virtual network embedding (VNE) faces severe challenges in dynamic network environments. To address the poor adaptability and weak interpretability of existing deep reinforcement learning methods, which stem from their reliance on implicit knowledge encoding, a knowledge-driven resource orchestration framework was proposed. The framework pioneered the use of a temporal knowledge graph (TKG) as an explicit world model for the agent, enabling the structured representation and reasoning of network dynamics. Furthermore, a hybrid Actor-Critic agent was designed to integrate real-time states, historically curated experience, and domain heuristics through a dual-loop mechanism, achieving a synergy between rapid tactical adaptation and long-term strategic optimization. Experimental results demonstrate that the proposed framework achieves excellent performance in terms of the request acceptance ratio and revenue-to-cost ratio. In cross-topology transfer and online knowledge adaptation scenarios with frozen policy parameters, it maintains a consistent advantage; specifically, in exponential traffic scenarios, it improves upon the strongest baseline acceptance rate by 6.50% and achieves a relative improvement of over 80% compared to certain DRL baselines. This work provides key technical support for enabling efficient and reliable service provisioning in the Internet of agents.

Key words: IoA, knowledge-driven, VNE, resource orchestration, TKG

收稿日期：2026–05–13；**修回日期：**2026–07–26
通信作者：任保全，renbq88@126.com
基金项目：国防重点实验室基金重点项目（No.6142006240401）；国家自然科学基金资助项目（No.62571259）；江苏省研究生科研与实践创新计划基金资助项目（No.KYCX231089）
Foundation Items: The Key Laboratory Fund of National Defense (No.6142006240401), The National Natural Science Foundation of China (No.62571259), The Postgraduate Research & Practice Innovation Program of Jiangsu Province (No.KYCX231089)

0 引言

新一轮人工智能变革正推动大语言模型等基础模型向具备逻辑推理、工具操作、持续学习与自我组织能力的智能体演进[1-2]。随着多智能体系统[3]分工协作成为释放生产力的重要形态，通信网与互联网也正在从“人与人连接”向“智能体连接”升级，并逐步演化为以智能体为核心的新型网络范式——智能体互联网（Internet of agent，IoA）[4-5]。在IoA中，网络需要优先满足智能体间高效交互与协作需求，其体系架构、交互协议、安全与服务治理等技术仍在快速迭代，亟须形成面向自治运行与可持续演进的基础方法与工程能力[6]。

区别于传统互联网主要承载面向人类可理解的数据（文本、图像、音频等），IoA中的交换对象更趋于面向智能体的机器原生数据对象，如模型参数与梯度、加密令牌、潜在表征与知识摘要等[7]。同时，交互方式从图形用户界面演进为基于语义感知与目标驱动的自主协商通信[8-9]。上述变化使网络侧必须提供面向任务目标的服务治理与资源调度能力，包括服务发现与组合、算网融合资源编排、动态资源分配与优化，以及面向服务等级协议（service level agreement，SLA）的端到端服务质量（quality of service，QoS）保障等[10]。作为网络虚拟化与服务化架构的重要支撑技术，虚拟网络嵌入（virtual network embedding，VNE）能够将抽象的服务请求映射到共享的物理资源底座，实现计算与网络资源的协同编排，其性能直接影响IoA的规模化部署与服务质量。

智能体互联网服务任务图编排与5G网络切片编排并非同一问题的简单场景替换。5G网络切片通常依托软件定义网络（SDN）、网络功能虚拟化（NFV）和云原生基础设施，围绕切片实例及网络功能链的创建、配置、资源隔离、弹性调整和终止开展全生命周期管理[11]。IoA资源编排则位于上述资源环境之上，对于一次已经完成任务分解的多智能体协同任务，需要进一步确定推理、检索、规划、验证和安全审计等功能组件应部署到哪些计算节点，并为组件间的任务依赖建立满足资源约束的通信路径[12]。例如，一个多智能体协同研判任务由数据检索、模型推理、结果验证和安全审计4类组件构成。检索组件需要接近数据资源部署，推理组件具有较高计算需求，验证组件与推理组件之间存在高频中间结果交互，审计组件则可能部署于跨域网关附近。若仅依据节点剩余资源进行放置，策略容易将多个组件集中到高中心性节点，引发跨簇链路持续占用；而某一枢纽节点近期是否频繁造成映射失败，又无法由当前资源快照直接反映。

在IoA中，VNE的部署位置通常位于服务治理与编排层，并与NFV/云原生编排系统协同工作：当多智能体任务被分解为可部署的服务链或任务图时，编排器将其抽象为虚拟网络请求（virtual network request, VNR），其中虚拟节点对应可部署的智能体功能组件（如推理、检索、规划、验证、日志审计或安全网关等微服务），虚拟链路对应组件间的通信依赖与带宽/时延约束；底层物理网络（PN）由边缘节点、区域节点与骨干链路构成[13]。编排器需要在满足SLA的前提下，将VNR映射到PN的计算与链路资源上，从而实现智能体服务的落地部署。

然而，传统VNE方法在IoA的动态性与复杂性面前暴露出固有局限。早期研究多采用启发式规则（如贪心算法[10]）与元启发式搜索（如粒子群优化[13-14]、鲸鱼优化[15]）来构造节点映射与链路映射策略。这类方法通常依赖人工设计的局部指标或固定规则，难以刻画多维资源约束与长期收益之间的权衡，在请求高并发到达、网络拓扑与链路状态频繁波动的条件下，其决策往往呈现短视性与不稳定性，缺乏从历史交互中积累经验并预测资源演化趋势的内在机制。

为克服上述不足，深度强化学习（deep reinforcement learning，DRL）被引入VNE并显示出端到端学习长期优化策略的潜力[16-17]。DRL-VNE能够通过与环境持续交互学习资源分配策略，从而在一定程度上摆脱手工规则设计。然而，现有DRL-VNE仍普遍采用“隐式知识编码”范式，即将历史经验、环境规律与决策逻辑隐含地压缩到策略网络参数中[18]。该范式带来三方面瓶颈：其一，决策可解释性弱，难以支持可审计与可运维的服务治理；其二，面对非平稳环境与组合动作空间时，训练效率与稳定性受限；其三，当部署场景发生分布偏移（如由合成拓扑迁移至具有真实社团结构的拓扑或流量模式显著变化）时，策略性能易出现明显衰减，这与IoA对持续自适应与可靠服务的要求相矛盾。

针对 DRL-VNE 的可解释性与泛化能力不足，已有工作尝试从结构表征或决策结构上进行改进，例如，采用 GCN/GAT 提升网络状态感知能力，或引入分层强化学习分解决策空间[18-19]。然而，这些方法多数仍未摆脱隐式知识编码的核心约束，缺乏一个可查询、可演化、可进行时序推理的显式世界模型来统一刻画“网络状态–编排行为–反馈结果”的因果关联。面向 IoA 的服务治理与资源调度，亟须一种以显式知识建模为中心、能够持续学习并具备跨拓扑迁移能力的编排智能体框架，从而支撑意图驱动网络的自治编排、态势感知与优化决策。

现有 IoA 的服务治理通常建立在 SDN/NFV 与网络切片等服务化底座之上。多智能体任务往往被拆解为可部署的服务链或任务图，例如，“感知/检索–推理–规划–验证–安全审计”等功能组件的组合，其部署过程可统一抽象为 VNR：虚拟节点对应智能体功能组件或微服务实例的计算资源需求，虚拟链路对应组件间交互依赖及带宽、时延等 QoS 约束；编排器需要将 VNR 映射到边缘云、MEC 节点与骨干链路等共享物理资源底座，实现服务的在线落地与持续运行。与传统面向单次映射的静态 VNE 不同，IoA 场景呈现“持续到达–在线执行–生命周期释放”的全程运行形态，且链路拥塞、节点负载与可用性随时间波动，因而 VNE 更接近面向动态环境的序贯决策问题。从更一般的任务驱动通信角度看，任务目标会进一步引起跨层参数耦合、多时间尺度资源响应和多节点自主决策冲突，使资源配置由相对独立的网络层优化演变为任务约束下的联合智能决策[20]。IoA 服务资源编排需要同时描述任务依赖、当前资源状态和历史交互知识。因此，本文主要研究将动态形成的智能体服务任务图映射到共享算网资源底座，并通过显式历史知识提高持续准入能力、路径经济性和跨拓扑适应能力。由此引出本文的核心研究问题：如何构建一个与策略网络解耦、能够在线更新和进行时序推理的知识世界模型，以支持 IoA 服务任务图的长期资源编排。

面向 IoA 或云原生服务化网络的 VNE/编排研究大体可归纳为 3 类方法。第一类是启发式与元启发式方法[10-15]，通过局部指标或搜索规则构造节点放置与链路映射策略，优点是计算代价低、易工程实现，但难以在高并发到达、强非平稳与多约束耦合条件下保持稳定的长期收益。第二类是学习驱动的 DRL-VNE 方法[16-22]，通常将网络状态编码为图表示（如采用 GCN/GAT 提升结构感知能力或采用分层强化学习分解决策空间），以端到端方式学习长期优化策略，能够在一定程度上缓解手工规则设计的局限。第三类是面向网络自治闭环的可解释增强方法，典型思路是将规则、约束或领域先验注入策略网络以提升可解释性与收敛效率，但这类方法多数仍依赖隐式知识编码，缺少一个可查询、可演化、可时序推理的显式世界模型来统一组织，因此在 IoA 的跨拓扑迁移、长期在线运行与运维审计等要求下仍存在明显不足。近年来，也有研究开始尝试将知识图谱、图表示学习或结构化先验引入网络资源编排过程，用以增强网络状态建模、资源依赖刻画和决策可解释性[23]。这类方法在一定程度上弥补了传统 DRL 策略对拓扑结构和历史经验利用不足的问题，但多数工作仍将知识结构作为静态辅助特征或状态编码器使用[24]，尚未将“网络状态–编排行为–反馈结果”的连续交互过程组织为可持续更新、可查询和可时序推理的显式世界模型。换言之，已有知识增强型 VNE 方法更多关注结构信息注入，而较少关注在线经验治理、知识生命周期维护以及跨拓扑迁移中的历史风险纠偏。

因此，面向 IoA 的资源自主编排需要一种以显式知识建模为中心的框架，使智能体能够将在线交互经验以可追溯、可治理的形式持续沉淀，并在动态非平稳环境中实现快速战术修正与长期战略优化的协同，从而支撑可部署、可演进的智能体互联网服务治理。

基于此，本文面向智能体互联网的服务治理与资源自主编排需求，提出知识驱动的智能体自主编排（knowledge-driven agent autonomous orchestration, KD-AAO）框架，以时序知识图谱（temporal knowledge graph, TKG）作为编排智能体的显式世界模型，刻画“网络状态–编排行为–反馈结果”的演化关联，并通过知识与策略解耦的双循环协同机制实现知识更新与策略优化的闭环自治。

在本文中，IoA 服务任务图、VNR、NFV/云原生资源底座和 TKG 分别对应业务语义、资源抽象、部署执行和知识决策 4 个层次。首先，上层任务分解过程根据任务意图形成由推理、检索、规划、验证和安全审计等功能组件构成的服务任务图，本文

将已经形成的任务图作为编排输入。其次，服务任务图被转换为VNR，功能组件抽象为具有计算需求的虚拟节点，组件间的数据和控制依赖抽象为具有带宽及时延需求的虚拟链路，任务到达、运行和结束分别对应VNR的到达、生命周期保持和资源释放。最后，NFV/云原生平台提供组件实例化及资源执行环境，KD-AAO输出的节点映射和链路映射方案可分别转化为服务实例放置与通信路径配置。

本文的主要研究工作如下。

（1）提出面向IoA服务任务图在线映射KD-AAO框架。该框架采用“知识—策略解耦”的架构思想，设计时序知识图谱作为显式世界模型，用于统一表征物理资源状态、服务请求特征、编排动作与交互反馈，并构建双循环协同机制：高频“知识环”负责基于在线反馈更新知识状态，低频“策略环”负责基于积累经验进行策略优化，实现面向IoA的编排与可解释服务治理。

（2）设计全生命周期知识管理机制和三模态融合决策网络。在知识管理方面，提出“创建–冷却–恢复–蒸馏”的完整流程，特别是自适应冷却与多路径恢复机制，以保证知识的新鲜度与可靠性并抑制噪声扩散；在决策方面，构建融合3类信息的Actor-Critic：由图神经网络刻画的实时网络状态“感知”、由时序GraphSAGE建模的历史经验“记忆”和由启发式排序器提供的领域规则“先验”[25]。该融合机制在降低组合动作空间复杂度的同时兼顾短期敏捷性与长期规划能力。

（3）通过多场景实验系统验证所提框架的有效性与关键特性。实验结果表明，KD-AAO在长期请求接受率与资源效率等指标上优于多种代表性基线方法；泛化测试中，模型由合成训练场景迁移至具有真实结构特征的拓扑时不需要微调仍保持显著优势，体现了面向IoA动态异构环境的实用性与可迁移性。

1 系统模型

为刻画智能体互联网中服务任务图在共享算网资源底座上的在线部署问题，本文采用虚拟网络嵌入作为统一建模形式。对于一个已经完成任务分解的多智能体服务，其功能组件及组件间通信依赖可抽象为VNR：虚拟节点表示待部署的智能体功能组件或微服务实例，虚拟链路表示组件间的通信关系与资源约束；边缘节点、区域计算节点和骨干链路等基础设施则抽象为PN。VNE的目标是在满足基础资源、连通性和时延约束的前提下，将持续到达的VNR映射到物理网络上，提升系统长期服务承载能力与资源使用效率。

本文所考虑的智能体任务已经完成任务分解和功能组件选择，研究对象是任务图形成后的资源准入、节点放置和链路映射。NFV/云原生系统作为底层执行环境，所提架构不修改其资源管理接口，通过VNR向其提供可执行的节点放置与路径映射方案，涉及的功能组件对应的模型选择、容器镜像构建、实例内部执行逻辑以及运行中任务图重构均为通用设置。智能体互联网服务任务图在线映射架构如图1所示。

1.1 物理网络模型设置

为反映IoA资源底座的动态性，PN建模为随时间演化的状态序列$\{G_p(t)\}_{t=0}^{\infty}$。其中，$G_p(t)=(N_p,L_p)$表示时刻$t$的物理拓扑，$N_p$与$L_p$分别为物理节点与物理链路集合。物理节点$p_i \in N_p$的可用计算资源为$C_t(p_i)$，物理链路$l_p \in L_p$的可用带宽为$B_t(l_p)$，链路基础时延为$D_t(l_p)$，其中$C_t(p_i)$与$B_t(l_p)$随VNR的到达、映射和释放动态变化，$D_t(l_p)$用于刻画任务组件间通信路径的基础QoS约束，可由传播时延、排队时延或链路测量结果给出。为了支持知识驱动的细粒度推理，本文将物理网络实体属性划分为3类：固有属性，如节点度、介数中心性等结构角色特征；动态属性，如$C_t(p_i)$、$B_t(l_p)$与$D_t(l_p)$等随嵌入行为变化的资源与QoS状态；经验属性，如节点置信度$c_t(p_i)$与成功率$s_t(p_i)$等长期统计指标，用于表征节点的历史可靠性，并与瞬时资源状态解耦，由后文的时序知识图谱进行维护与更新。在IoA服务任务图在线部署过程中，状态序列的演化主要由VNR到达与离开、节点放置与链路路由决策引起的资源占用变化，以及链路可用带宽和基础时延变化等因素共同驱动。基于上述时序建模，本文在2.2节进一步给出TKG世界模型与时间编码，用于对经验属性与事件新鲜度进行显式维护，从而支撑后续的时序推理与决策。本文模型重点考虑计算资源、带宽资源和链路时延3类基础约束。时延抖动、链路丢包、节点能耗和任务优先级等因素可进一步作为物理实

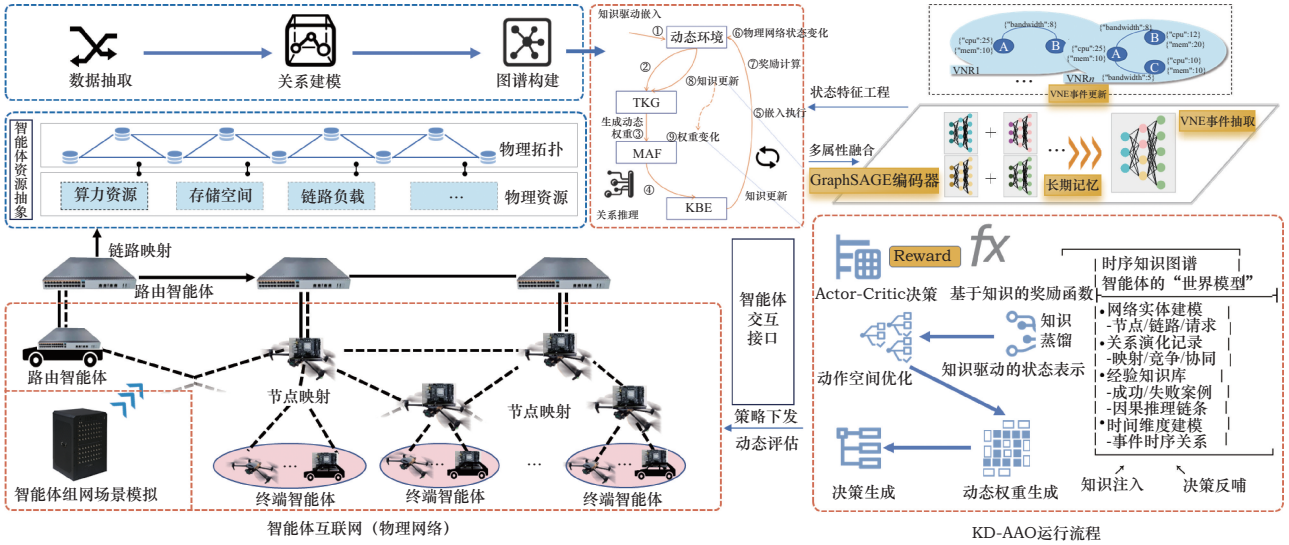


图 1　智能体互联网服务任务图在线映射架构

体或 VNR 的扩展属性写入 TKG，不考虑更细粒度的队列建模、能耗模型和差异化调度机制作为优化目标。

1.2　虚拟网络请求模型

VNR 用图 $G_v=(N_v,L_v)$ 表示，其中 N_v 与 L_v 分别为虚拟节点与虚拟链路集合。每个虚拟节点 $n_v\in N_v$ 具有计算需求 $\mathrm{req}(n_v)$，每条虚拟链路 $l_v\in L_v$ 具有带宽需求 $\mathrm{req}(l_v)$ 和最大允许时延 $D^{\max}(l_v)$。在 IoA 场景下，虚拟节点可对应推理、检索、规划、验证或安全审计等智能体功能组件，虚拟链路表示组件间的数据、上下文或控制消息交互依赖。VNR 具有有限生命周期，生命周期结束后其占用的物理资源被释放。若 G_v 被成功嵌入，则产生基础收益：

$$R_{\mathrm{base}}(G_v)=\sum_{n_v\in N_v}\mathrm{req}(n_v)+\alpha\sum_{l_v\in L_v}\mathrm{req}(l_v)\quad(1)$$

其中，α 为平衡节点资源与链路资源贡献的权重系数，时延作为可行性约束处理，不计入基础收益，避免 QoS 约束与资源收益混合建模。

如表 1 所示，智能体的引入并未改变 VNE 的基本资源约束，而是改变了 VNR 的生成语义、任务依赖结构、生命周期特征以及资源决策所需的历史状态。本文不重新定义 NFV 或 VNR，而是利用 TKG 将 IoA 任务运行产生的持续反馈转化为可用于后续映射的显式知识。

1.3　映射关系与可行性约束

记 $\mathrm{map}(\cdot)$ 为从虚拟实体到物理实体的映射函数。对任意虚拟节点 n_v，其映射结果为物理节点 $\mathrm{map}(n_v)\in N_p$；对任意虚拟链路 $l_v=(u_v,v_v)\in L_v$，其映射结果为连接 $\mathrm{map}(u_v)$ 与 $\mathrm{map}(v_v)$ 的物理路径 $\mathrm{path}(\mathrm{map}(l_v))$，由一组物理链路构成。可行嵌入需满足以下约束。

表 1　　**各要素与 TKG 的对应关系**

IoA 任务要素	VNR 抽象	NFV/云原生执行对象	TKG 知识表示
一次多智能体协同任务	VNR G_v	一次服务部署请求	VNR 实体及到达、接纳、拒绝、释放事件
推理、检索、规划等功能组件	虚拟节点 n_v 及计算需求	VNF、CNF 或微服务实例在物理节点上的部署	虚拟节点实体、承载关系、映射结果
组件间任务依赖	虚拟链路 l_v 及带宽、时延需求	服务实例间通信路径	虚拟链路、物理路径、占用及失败关系
边缘、MEC、区域云资源	物理节点 p_i	服务器或计算节点	固有属性、动态资源、历史成功率和风险值
骨干与跨簇通信资源	物理链路 l_p	SDN 配置的物理路径	带宽、时延、路径代价和历史拥塞风险
映射动作与执行反馈	$\mathrm{map}(n_v)$、$\mathrm{path}(\mathrm{map}(l_v))$	实例放置和路径配置结果	映射动作实体、成功/失败事件及时间戳

节点一对一映射约束，不同虚拟节点不得映射到同一物理节点，即：

$$\mathrm{map}(n_v) \neq \mathrm{map}(n_v') \tag{2}$$

节点计算资源约束，即：

$$C_t \mathrm{map}(n_v) \geqslant \mathrm{req}(n_v), \forall n_v \in N_v \tag{3}$$

链路时延与带宽约束，虚拟链路映射后的端到端路径时延不得超过其最大允许时延，且虚拟链路需求应由其对应物理路径上的每条物理链路满足：

$$\sum_{l_p \in \mathrm{path}(\mathrm{map}(l_v))} D_t(l_p) \leqslant D^{\max}(l_v), \quad \forall l_v \in L_v$$

$$B_t(l_p) \geqslant \mathrm{req}(l_v), \forall l_v \in L_v, \forall l_p \in \mathrm{path}(\mathrm{map}(l_v)) \tag{4}$$

其中，$D_t(l_p)$表示物理链路l_p在时刻t的基础时延，$D^{\max}(l_v)$表示虚拟链路l_v可接受的最大端到端时延。若任一虚拟链路不存在同时满足带宽和时延约束的物理路径，则当前VNR映射失败。

1.4 知识驱动的序贯决策建模

本文将动态VNE表述为知识驱动的马尔可夫决策过程（MDP）。对于包含$|N_v|$个虚拟节点的VNR，其嵌入过程被分解为$|N_v|$个连续决策步骤：智能体在每一步为当前虚拟节点选择一个承载的物理节点。状态s_t由三部分构成：当前物理网络拓扑与动态资源/QoS状态，包括节点可用CPU、链路可用带宽和链路基础时延；当前VNR的需求信息及部分映射进度，包括虚拟节点计算需求、虚拟链路带宽需求和最大允许时延；TKG的整体状态，用以编码网络实体的经验属性和历史反馈，智能体基于当前态势与历史规律联合推理。

动作a_t表示为当前虚拟节点选择物理节点$p_i \in N_p$。为降低组合爆炸，原始动作空间将通过可行性过滤得到候选集合C_{feasible}，仅保留满足节点计算资源、链路带宽、链路时延和连通性约束的物理节点作为可选动作。

上述CPU、带宽和时延条件首先作为硬可行性约束参与候选空间构造，在候选动作可行的基础上，奖励函数进一步引导策略在接纳收益、资源效率和负载均衡之间进行权衡。为同时鼓励收益、资源效率与负载均衡，本文采用复合奖励函数：

$$r(S) = R_{\mathrm{base}}(S) + w_{\mathrm{eff}} R_{\mathrm{eff}}(S) + w_{\mathrm{bal}} R_{\mathrm{bal}}(S) \tag{5}$$

其中，R_{base}由式(1)给出，R_{eff}与R_{bal}分别用于奖励资源效率与负载均衡，w_{eff}和w_{bal}为权重系数。资源效率项用于刻画一次嵌入在计算资源与链路资源上的单位收益能力，定义为：

$$R_{\mathrm{eff}}(t) = \frac{\mathrm{Rev}(G_v)}{\mathrm{Cost}_{\mathrm{cpu}}(G_v) + \mathrm{Cost}_{\mathrm{bw}}(G_v) + \epsilon} \tag{6}$$

其中，ϵ为防止分母为零的极小常数，$\mathrm{Cost}_{\mathrm{cpu}}(G_v)$表示虚拟节点映射所消耗的物理计算资源总量，$\mathrm{Cost}_{\mathrm{bw}}(G_v)$表示虚拟链路映射到多跳物理路径后产生的带宽占用开销。负载均衡项用于抑制资源过度集中消耗，定义为：

$$R_{\mathrm{bal}}(t) = -\left[\mathrm{Var}_{n \in V_s}\left(1 - \frac{C_n^{\mathrm{res}}(t)}{C_n^{\mathrm{cap}}}\right) + \mathrm{Var}_{e \in E_s}\left(1 - \frac{B_e^{\mathrm{res}}(t)}{B_e^{\mathrm{cap}}}\right)\right] \tag{7}$$

其中，$C_n^{\mathrm{res}}(t)$和$B_e^{\mathrm{res}}(t)$分别表示物理节点剩余CPU资源和物理链路剩余带宽，C_n^{cap}和B_e^{cap}表示其初始容量。该项越大，说明资源消耗越均衡。对于映射失败或高风险实体选择，进一步引入惩罚项：

$$R_{\mathrm{pen}}(t) = -\lambda_f \mathbb{I}_{\mathrm{fail}}(t) - \lambda_r \sum_{x \in \mathcal{X}_t} \mathrm{Risk}(x,t) \tag{8}$$

其中，$\mathbb{I}_{\mathrm{fail}}(t)$表示当前VNR是否嵌入失败，$\mathcal{X}_t$表示本次映射涉及的物理节点与物理链路集合，$\mathrm{Risk}(x,t)$为TKG维护的实体风险值。因此，完整奖励函数改写为：

$$R(t) = \alpha_1 R_{\mathrm{rev}}(t) + \alpha_2 R_{\mathrm{eff}}(t) + \alpha_3 R_{\mathrm{bal}}(t) + R_{\mathrm{pen}}(t) \tag{9}$$

在实验中，$\alpha_1, \alpha_2, \alpha_3$采用验证集网格搜索确定，并保持所有负载和拓扑迁移实验中的取值一致，以避免针对特定测试场景进行调参。默认设置下3类正向奖励归一化到相同量纲，失败惩罚和风险惩罚用于约束不可行映射和重复选择高风险实体。该设计使策略不仅追求短期收益，而且倾向于形成长期可持续的嵌入行为。同时，时延约束已在候选路径过滤阶段作为硬约束处理，奖励函数不再重复引入时延惩罚项，以避免增加额外权重参数并影响实验可复现性。

奖励函数权重参数设置如表2所示，完整训练参数、TKG容量参数及冷却/恢复阈值将在实验设置部分统一给出。

2 知识驱动资源编排方法设计

本节面向IoA服务任务图的在线节点放置与链路映射问题，给出KD-AAO的知识驱动资源编排

表 2　奖励函数权重参数设置

参数	含义	默认值	说明
α_1	收益权重	1.0	鼓励接纳高收益 VNR
α_2	资源效率权重	0.5	抑制高代价嵌入
α_3	负载均衡权重	0.5	避免资源集中消耗
λ_f	失败惩罚	1.0	惩罚嵌入失败
λ_r	风险惩罚	0.1	抑制高风险实体选择

方法。整体思路是将动态 VNE 过程中的网络状态、映射动作与反馈结果显式组织为可查询、可更新的时序知识图谱，并将其作为策略网络外部的世界模型。在此基础上，该方法通过三模态 Actor-Critic 网络融合当前物理网络状态、历史经验知识和启发式规则信号，生成满足资源与 QoS 约束的映射决策；同时，通过知识环和策略环的双循环机制分别完成在线经验写入与 PPO 策略优化[26]。需要说明的是，本节所述“编排”主要指服务任务图结构确定后的节点放置、链路映射、准入判断和知识更新，不涉及任务自动分解、运行中实例扩缩容和已部署服务迁移。具体服务资源自主编排运行流程如图 2 所示。

2.1　架构与双循环反馈机制

KD-AAO 通过构建 TKG 作为显式世界模型，将物理网络状态、VNR 特征、节点放置、链路映射以及成功/失败反馈以实体–关系–时间事件的形式进行结构化记录。在此基础上，框架引入双循环机制：知识环负责将每次映射事件及时写入 TKG 并更新实体经验属性，策略环负责基于一批 on-policy 交互轨迹执行 PPO 更新。二者在不同时间尺度上协同工作，使策略决策既能利用近期反馈，又能通过长期训练优化资源分配行为。

（1）知识环：事件触发的在线知识更新路径。知识环在每次 VNR 到达、离开、节点放置、链路映射成功或失败后触发，将相应事件写入 TKG，并同步更新相关物理节点和物理链路的经验属性，如置信度、成功率、失败率和风险值。其作用是为后续请求提供可追溯的历史反馈，使候选集合构造能够规避近期表现不稳定或风险较高的资源实体。

（2）策略环：批量式 PPO 策略优化路径。策略环在累计一批 VNR 交互轨迹后触发，将状态、动作、奖励、旧策略概率和价值估计组成 on-policy 轨迹缓存，并基于复合奖励函数计算回报与优势函数，随后采用 PPO 裁剪目标更新 Actor-Critic 网络参数。与知识环的事件级更新不同，策略环并不在每次映射后立即执行梯度更新，而是以批量轨迹为

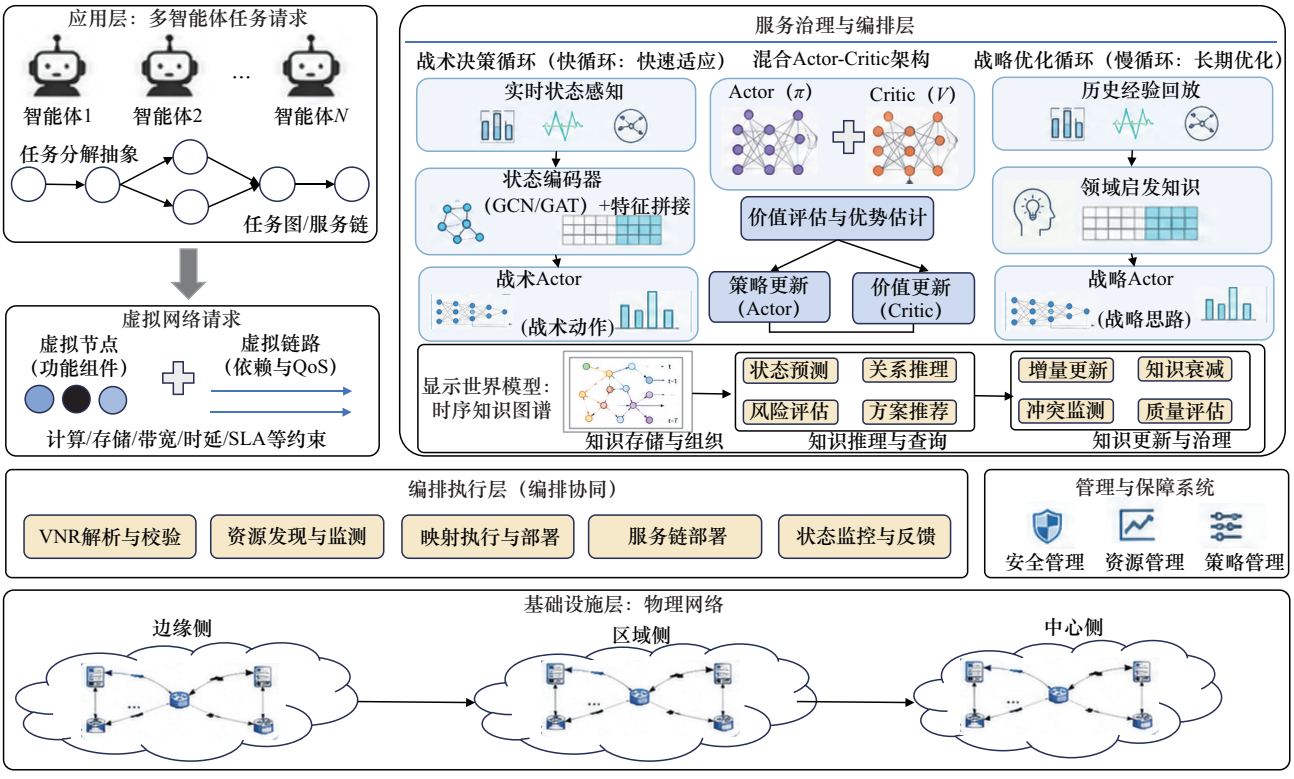


图 2　知识驱动的智能体网络服务资源自主编排运行流程

单位进行策略优化，从而提升训练稳定性并避免由单次反馈噪声导致的策略振荡。

2.2　时序知识图谱世界模型与时间编码

本节将服务任务图在线映射过程中产生的资源变化、节点放置、链路映射与成败反馈显式写入TKG，并通过时间编码构造可泛化的时序表征，为后续GraphSAGE记忆流提炼事件新鲜度与历史趋势信息提供基础。

（1）实体中心的知识建模。TKG采用实体–关系图结构描述IoA服务任务图映射环境，实体覆盖物理节点、物理链路、虚拟请求、虚拟节点、虚拟链路以及映射动作等关键对象，并显式刻画其间的关联关系（如“承载”“连接”“失败于”等）。每个实体维护3类属性：固有属性用于表征结构角色；动态属性用于表征可用资源等瞬时状态；经验属性用于表征历史交互规律与表现趋势，从而弥合“原始网络数据”与“可行动知识”之间的语义鸿沟。

（2）两阶段时间编码。标准GNN对时间不敏感，而在动态IoA环境中，事件的新鲜度与历史趋势会显著影响编排决策。直接使用绝对时间戳作为特征容易引入非平稳性，并在部署场景时间尺度变化时导致分布外泛化问题。为此，本文采用“局部归一化 + 高维映射”的两阶段时间编码，使模型获得稳定、可泛化的时间表征。

阶段1　在每个决策批次内以最早事件时间为参照，将事件时间戳转换为相对时间：

$$t_{\mathrm{norm},i} = t_i - t_{\min}^{\mathrm{batch}} + \varepsilon \tag{10}$$

其中，t_i为事件i的绝对时间戳，$t_{\min}^{\mathrm{batch}}$为该批次内最早事件时间，$\varepsilon$为保证正值的微小常数。该变换在局部上下文中保留事件的时序顺序与相对间隔，同时提供对绝对时间尺度更鲁棒的平稳信号。

阶段2　将$t_{\mathrm{norm},i}$通过正弦–余弦模块（借鉴Transformer位置编码机制[27]）映射为高维时间特征向量：

$$h_t^{(2k)} = \sin\left(\frac{t_{\mathrm{norm},i}}{B^{\frac{2k}{d_{\mathrm{embed}}}}}\right)$$

$$h_t^{(2k+1)} = \cos\left(\frac{t_{\mathrm{norm},i}}{B^{\frac{2k}{d_{\mathrm{embed}}}}}\right) \tag{11}$$

其中，k为维度索引，d_{embed}为嵌入维度，$B > 1$为时间尺度基数，用于控制不同维度上的频率跨度。本文沿用正弦–余弦位置编码中的常用设置，取$B = 10^4$。较低维度对应较高频率，有利于刻画近期事件变化；较高维度对应较低频率，有利于刻画长期趋势。该参数仅用于时间特征映射，并非风险冷却、恢复或策略更新中的经验阈值。将时间向量$\boldsymbol{h}_t$与实体特征进行拼接或加和后，TKG编码器能够区分短期波动与长期瓶颈，并学习“事件新鲜度–性能趋势”之间的关联，提升动态环境下的时序推理能力。

2.3　全生命周期知识管理机制

面向IoA服务任务图在线映射场景，环境反馈具有持续到达、强噪声和非平稳等特征。若将每次嵌入成败都等价写入知识库，容易被探索噪声或偶发拥塞误导，从而削弱知识的可信度与可用性。为此，本文在TKG世界模型之上引入全生命周期知识管理机制，以“创建–冷却–恢复–蒸馏剪枝”为主线，对知识的写入、筛选、纠偏与压缩进行统一治理，使世界模型在长期运行中保持可更新、可追溯和规模可控。

（1）创建：事件驱动的原子事实写入。知识环在每次嵌入尝试后即时触发，将“VNR到达、离开、节点映射、链路映射、成功/失败”等结果编码为带时间戳的原子事件，并写入对应实体与关系的时序轨迹中。该机制确保知识库与环境状态保持同步，从而将“策略执行结果”转化为可追溯的决策上下文。

（2）自适应冷却：统计过滤与候选池整肃。在线编排不可避免存在探索，因此单次失败并不等价于“坏知识”。本文引入自适应冷却作为统计过滤机制：系统为关键实体（尤其是物理节点）维护经验属性（如置信度、成功率、近期失败率等），当实体在滑动窗口内呈现可证明的持续低效（如近期失败率高且置信度低）时，将其加入冷却池并临时剔除出候选集合，从而避免策略在高风险资源上反复试错，提升后续候选集合构造的稳健性。

（3）多路径恢复：避免过度保守与信息过期。冷却机制虽然能提升稳健性，但若缺乏纠错，会导致系统过度保守或长期“误伤”潜在优质资源。为此，本文设计多路径恢复机制，使冷却实体可在满足任一路径条件时重新回到候选池。①性能恢复：若实体在近期交互中出现足够数量的成功嵌入，说

明其可靠性回升；②资源恢复：若实体可用资源显著补充（如某大VNR离开导致资源释放），先前失败诱因可能消失；③时间恢复：设置超时兜底，避免因历史数据过期导致永久排除。恢复后，系统对其经验属性进行“温启动”重置，以支持重新探索与再评估。

（4）经验蒸馏与剪枝：长期可扩展的知识压缩。在线运行中，TKG会持续累积瞬态实体（如已完成的VNR及其细粒度轨迹），若不加控制将导致存储与推理开销无界增长。本文采用“先蒸馏、后剪枝”的知识压缩机制：当知识库接近容量阈值时，将瞬态实体中的经验属性按关联关系（如“该VNR主要映射到的物理节点集合”）进行加权聚合，蒸馏为持久实体（物理节点/链路）的统计经验，再对瞬态实体进行剪枝，从而在控制知识规模的同时保留对未来决策有价值的长期规律。相关阈值和系数的具体取值将在实验设置部分的超参数表中统一给出，所有负载和跨拓扑实验保持一致。

量化更新规则如下。为提升知识治理机制的可复现性，本文进一步给出经验属性、冷却、恢复与蒸馏的量化规则。对于任一物理实体x，其成功率与失败率采用指数滑动平均更新：

$$\begin{aligned} S_x(t) &= \rho S_x(t-1) + (1-\rho)\mathbb{I}_{\text{succ}}(x,t) \\ F_x(t) &= \rho F_x(t-1) + (1-\rho)\mathbb{I}_{\text{fail}}(x,t) \end{aligned} \tag{12}$$

其中，$\rho \in (0,1)$为平滑系数。实体置信度定义为：

$$\text{Conf}_x(t) = \frac{N_x(t)}{N_x(t)+\kappa} \cdot S_x(t) \tag{13}$$

其中，$N_x(t)$表示实体累计参与映射次数，κ用于降低小样本实体置信度偏置。实体风险值定义为：

$$\text{Risk}_x(t) = \beta_1 F_x(t) + \beta_2 \text{Cong}_x(t) + \beta_3 \text{Cost}_x(t) \tag{14}$$

其中，$\text{Cong}_x(t)$表示实体拥塞程度，$\text{Cost}_x(t)$表示该实体导致的平均路径开销或资源占用代价。若实体在长度为W_c的滑动窗口中满足：

$$F_x^{W_c}(t) > \theta_f, \quad \text{Conf}_x(t) < \theta_c \tag{15}$$

则将其加入冷却池，并在后续候选集合构造中临时剔除。冷却实体满足以下任一条件即可恢复。

① 性能恢复。最近W_r次相关交互中成功次数不低于N_{succ}。

② 资源恢复。剩余资源增量满足：

$$\frac{R_x^{\text{res}}(t) - R_x^{\text{res}}(t_0)}{R_x^{\text{cap}}} > \theta_{\text{res}} \tag{16}$$

③ 时间恢复。冷却持续时间超过T_{cool}。

恢复后，实体经验属性采用温启动方式更新：

$$\begin{aligned} &\text{Risk}_x(t) \leftarrow \gamma \text{Risk}_x(t) \\ &\text{Conf}_x(t) \leftarrow (1-\gamma)\text{Conf}_x(t) + \gamma \text{Conf}_0 \end{aligned} \tag{17}$$

当TKG规模超过容量上限$C_{\max}$时，系统触发经验蒸馏与剪枝。对于瞬态VNR实体q，其经验按照关联物理实体集合$\mathcal{A}(q)$加权聚合：

$$\Delta \text{Exp}_x = \sum_{q: x \in \mathcal{A}(q)} \frac{w_{q,x}}{\sum_{x' \in \mathcal{A}(q)} w_{q,x'}} \text{Exp}_q \tag{18}$$

其中，$w_{q,x}$由资源占用比例、路径参与频次或失败归因权重确定。完成蒸馏后，系统优先剪枝生命周期结束且重要度低的瞬态实体与关联边，从而在控制图谱规模的同时保留对后续决策有价值的长期经验。

时序知识图谱世界模型的全生命周期知识管理与更新算法如算法1所示。

算法1　时序知识图谱世界模型的全生命周期知识管理与更新算法

定义　设知识图谱容量上限$N_{\max}$，蒸馏剪枝阈值α，指数滑动平均系数ρ，风险冷却系数η，恢复系数δ，时间编码维度d_{embed}，GraphSAGE层数L，采样规模$[s_1, s_2]$

输入　物理网络状态G^P（节点CPU、链路带宽、时延等），当前请求r_t（VNR），映射尝试事件流ε_t（放置/路由/成功或失败），当前TKG$\mathcal{K}_t$

输出　更新后的$\mathcal{K}_{t+1}$，实体嵌入$\boldsymbol{H}_{t+1}$，实体风险/可信度向量$\boldsymbol{q}_{t+1}$，用于决策的状态表示s_{t+1}

1)初始化：构建空时序知识图谱$\mathcal{K}_0$，初始化实体集合$\mathcal{V}$与关系集合$\mathcal{R}$，初始化实体嵌入$\boldsymbol{H}_0$与风险值$\boldsymbol{q}_0$。

2)初始化图编码器：设置GraphSAGE参数。

3)循环，for每次映射尝试（对应一次VNR的放置与路由过程）do：

4)获取当前事件流ε_t与批次时间戳集合$\{t_i\}$，得到批次最早时间$t_{\min}^{\text{batch}}$。

5)时间编码：对每个事件时间戳执行局部归一化，得到$t_{\text{norm},i}$（如式(10)所示）。

6)高维映射：将$t_{\text{norm},i}$通过正弦–余弦模块映射为时间特征$\boldsymbol{h}_t$（如式(11)所示）。

7)实体创建与对齐：解析事件ε_t中涉及的对象，若实体不存在，则创建实体（物理节

点、物理链路、VNR节点、VNR链路、映射动作等）。

8)关系更新：根据事件类型更新关系边（如“承载”“连接”“失败于”“占用”“释放”等），并写入时间属性。

9)属性更新：对每个实体，更新固有属性、动态属性与经验属性；

10)经验统计：对经验属性，采用指数滑动平均更新：EMA $\leftarrow \rho \cdot \text{EMA} + (1-\rho)\cdot x_t$。

11)图嵌入计算：将实体特征与时间特征$\boldsymbol{h}_t$拼接或加和，输入GraphSAGE得到实体嵌入$\boldsymbol{H}_t$。

12)根据$S_x(t),F_x(t),\text{Cong}_x(t),\text{Risk}_x(t)$更新实体经验属性。

13)若$F_x^{W_c}(t)>\theta_f$，且$\text{Conf}_x(t)<\theta_c$，则将实体加入冷却池。

14)若冷却实体满足性能恢复、资源恢复或时间恢复条件，则移出冷却池并温启动经验属性。

15)if 图谱容量$|\mathcal{G}_{\text{TKG}}|>C_{\max}$ then

知识蒸馏剪枝：对已完成VNR实体执行经验蒸馏，并剪枝低重要度瞬态实体与边；

16)end if

17)状态构建：以$\boldsymbol{H}_t$、$\boldsymbol{q}_t$以及当前请求r_t的编码为输入，生成决策状态s_t。

18)输出$\mathcal{K}_{t+1}\leftarrow\mathcal{K}_t$，$\boldsymbol{H}_{t+1}\leftarrow\boldsymbol{H}_t$，$\boldsymbol{q}_{t+1}\leftarrow\boldsymbol{q}_t$，$s_{t+1}\leftarrow s_t$。

19)end for

2.4 知识注入的候选动作构造与链路映射

VNE的动作空间随物理网络规模增长呈组合爆炸。为构建可训练且满足约束的候选集合，KD-AAO在策略网络输出动作前引入多阶段“知识漏斗”，逐步将全部物理节点集合收缩为可行候选集合C_{feasible}。该过程首先利用CPU、带宽、时延和节点唯一性等硬约束剔除不可行动作，然后利用TKG中的冷却状态和风险标记排除近期表现不稳定的候选，最后通过MAMC启发式排序提供规则特征。由此，策略网络只在满足基础约束的候选集合上进行选择，从而降低无效探索。

阶段1为物理约束过滤，首先剔除不满足CPU需求或已被当前VNR其他虚拟节点占用的物理节点；随后通过K最短路径搜索验证候选节点与已映射节点之间是否存在同时满足带宽和时延约束的可行路径。阶段2为经验知识过滤，利用TKG维护的冷却状态与风险标记，临时排除近期嵌入表现不佳的节点。该步骤将“历史可用性”显式注入动作空间，使探索更聚焦于统计上可靠的候选子集。阶段3为启发式知识排序，对于剩余候选节点进行结构与资源潜力评估。

在完成当前虚拟节点的物理节点选择后，系统需要对所有端点已确定的虚拟链路执行链路映射。设虚拟链路$e_v=(u_v,v_v)$的两个端点已分别映射到物理节点m_u与m_v，其带宽需求为$b(e_v)$。本文采用约束K最短路径搜索进行链路映射：首先在物理网络中生成从m_u到m_v的K条候选路径集合$\mathcal{P}_K(m_u,m_v)$，随后剔除任一物理链路剩余带宽小于$b(e_v)$或不满足时延约束的路径。由于时延已在路径可行性过滤阶段作为硬约束处理，路径代价函数主要用于在可行路径集合中进一步权衡跳数、带宽占用压力和历史风险。对于剩余可行路径p，定义路径代价为：

$$\mathcal{C}(p)=\omega_1|p|+\omega_2\sum_{e\in p}\frac{b(e_v)}{B_e^{\text{res}}(t)+\epsilon}+\omega_3\sum_{e\in p}\text{Risk}(e,t) \tag{19}$$

其中，$|p|$表示路径跳数，第二项刻画带宽占用压力，第三项为TKG维护的链路风险代价。若候选路径不满足式(4)中的带宽与时延约束，则不进入式(19)的代价计算。链路映射选择代价最小的可行路径：

$$p^*(e_v)=\arg\min_{p\in\mathcal{P}_K^{\text{feas}}(m_u,m_v)}\mathcal{C}(p) \tag{20}$$

若$\mathcal{P}_K^{\text{feas}}(m_u,m_v)=\varnothing$，则当前节点动作被判定为不可行，系统将失败原因、相关物理节点、候选路径瓶颈链路以及时间戳写入TKG，并触发风险更新与冷却治理；若链路映射成功，则沿路径扣减链路带宽资源，并记录路径时延满足情况，同时将“虚拟链路–物理路径–资源占用–QoS约束满足–映射结果”作为原子事件写入TKG。由此，链路映射不仅完成带宽约束下的路径选择，也与奖励函数中的资源效率、负载均衡项以及TKG风险演化形成闭环。约束K最短路链路映射算法如算法2所示。

算法2 约束K最短路链路映射算法

输 入 已完成节点映射的虚拟链路

$e_v=(u_v,v_v)$，物理端点m_u,m_v，带宽需求$b(e_v)$，最大允许时延$D^{\max}(e_v)$，TKG链路风险向量

输出　物理路径$p^*(e_v)$或映射失败标记

1)生成K条候选物理路径$\mathcal{P}_K(m_u,m_v)$；

2)删除任一链路剩余带宽小于$b(e_v)$或路径总时延大于$D^{\max}(e_v)$的路径，得到$\mathcal{P}_K^{\text{feas}}$；

3)若$\mathcal{P}_K^{\text{feas}}=\varnothing$，写入失败事件并返回失败；

4)对每条可行路径，计算$\mathcal{C}(p)$；

5)选择$p^*(e_v)=\arg\min \mathcal{C}(p)$；

6)更新路径带宽资源，记录路径时延满足情况，并将成功事件写入TKG；

7)返回$p^*(e_v)$。

本文采用改进的邻接加权多维容量启发式（MAMC）进行评分：

$$\text{MAMC}(p_i)=C(p_i)\times\sum_{p_j\in N(p_i)}B(p_i,p_j) \quad (21)$$

其中，$C(p_i)$为节点剩余CPU，$B(p_i,p_j)$为链路带宽，$N(p_i)$为物理节点邻居集合。该评分倾向于选择“自身资源充足且邻域带宽丰富”的节点，从而降低后续链路映射的路径开销风险。MAMC仅作为候选节点排序的启发式规则，时延约束已在候选路径过滤阶段处理，因此不再额外引入时延权重，以减少手工调参。

为将启发式信息以可学习方式注入策略网络，本文进一步将排序结果归一化为显式启发式特征：

$$r_i=1-\left|\frac{\text{rank}(p_i)}{C_{\text{feasible}}}\right|,\forall p_i\in C_{\text{feasible}} \quad (22)$$

其中，$\text{rank}(p_i)$表示p_i在候选集合中的排序名次；r_i可被视为“规则信号”，用于提升策略网络的收敛速度与可解释性。

2.5　三模态混合Actor-Critic决策网络

在候选集合C_{feasible}上，KD-AAO采用三模态混合Actor-Critic结构融合3类信息，由GCN编码的当前物理网络状态，由时间感知GraphSAGE从TKG提炼的历史经验与趋势，以及由MAMC排序产生的启发式规则特征。为使编码器从输入端即可利用历史上下文，本文对每个物理节点构建知识增强的核心特征向量：

$$x_{p_i}^{\text{core}}=[C(p_i),\text{BW}_{\text{aggr}}(p_i),c_i,s_i,d_i]^{\text{T}} \quad (23)$$

其中，$\text{BW}_{\text{aggr}}(p_i)$为节点邻接带宽聚合量，$c_i$与$s_i$分别为TKG维护的置信度与成功率，$d_i$为度中心性。链路时延主要通过候选路径过滤和链路映射过程参与决策，若在具体部署中需要进一步增强时延感知，也可将节点邻域平均时延或最短时延距离作为扩展动态特征并入$x_{p_i}^{\text{core}}$。

随后网络采用双流编码结构：一条GCN流以物理网络拓扑为图结构、以$x_{p_i}^{\text{core}}$为节点特征，提取当前时刻的结构与资源嵌入$\boldsymbol{h}_{p_i}^{\text{PN}}$，刻画“此时此地”的全局态势；另一条以TKG为图结构，采用时间感知GraphSAGE编码器，并引入2.2节中的时间编码向量，将事件新鲜度与长期趋势注入实体表征，得到历史经验嵌入$\boldsymbol{h}_{p_i}^{\text{TKG}}$。由于TKG实体与物理节点存在映射关系，本文将TKG流输出按实体对齐到物理节点维度，得到与$\boldsymbol{h}_{p_i}^{\text{PN}}$可融合的知识嵌入。

接着进行三模态融合与动作输出，对每个候选节点$p_i\in C_{\text{feasible}}$，将3类信息拼接形成融合表示：

$$\boldsymbol{h}_{p_i}^{\text{fuse}}=\boldsymbol{h}_{p_i}^{\text{PN}}\oplus\boldsymbol{h}_{p_i}^{\text{TKG}}\oplus r_i \quad (24)$$

并输入多层感知机（MLP）得到动作logits。为保证决策满足可行性约束，对不在C_{feasible}内的节点施加掩码，将其logit置为极小值，再经Softmax得到策略分布$\pi(a|s)$。该设计使策略同时具备数据驱动的深度、经验引导的稳健性与规则注入的效率，Critic用于评估当前状态的全局价值$V(s)$。本文对GCN节点嵌入进行全局平均池化得到物理网络图级向量$\boldsymbol{g}_{\text{PN}}$，并对TKG流得到的知识嵌入聚合形成图级向量$\boldsymbol{g}_{\text{TKG}}$，二者拼接后输入MLP输出标量价值：

$$V(s)=\text{MLP}_{\text{graph}}([\boldsymbol{g}_{\text{PN}}\oplus\boldsymbol{g}_{\text{TKG}}]) \quad (25)$$

该结构使价值评估同时感知“全局资源景观”与“历史表现摘要”，从而为策略更新提供更稳定的基线。

最后完成训练与闭环协同，Actor-Critic采用PPO进行优化[28]：策略环以一批VNR交互轨迹为单位，利用复合奖励与Critic价值估计计算优势函数并更新网络参数；知识环则在每次嵌入尝试后即时更新TKG经验属性与冷却状态。双循环将“高频战术修正”与“低频战略学习”解耦，使系统在动态环境中兼具快速适应与长期优化能力。KD-AAO的整体在线编排与PPO更新流程如算法3所示。

算法3 KD-AAO的整体在线编排与PPO更新流程

定义 Actor参数为θ，Critic参数为ϕ，PPO裁剪系数为ϵ_{clip}，折扣因子为γ，GAE参数为λ_{GAE}，轨迹缓存容量为T_{upd}，每轮PPO小批量更新次数为K_{ppo}

输入 物理网络G_p，VNR到达序列$\mathcal{R}$，初始TKGG_k，GraphSAGE-TKG编码器，GCN物理图编码器，MAMC排序器

输出 Actor策略π_θ，价值函数V_ϕ，VNR接纳/拒绝结果，对应节点映射和链路映射方案

1)初始化：构建TKG世界模型G_k，初始化Actor参数θ、Critic参数ϕ，令$\theta_{\text{old}} \leftarrow \theta$。

2)初始化on-policy轨迹缓存$\mathcal{D} \leftarrow \varnothing$。

3)循环，for每个到达的VNR$G_v \in \mathcal{R}$ do；

4)读取当前物理网络资源状态、当前VNR需求和TKG状态，构造初始状态s_t。

5)初始化当前VNR的部分映射集合$M_v \leftarrow \varnothing$，设置映射状态标记flag ← success。

6)循环，for G_v中每个待映射虚拟节点n_v do；

7)根据节点CPU约束、节点一对一约束、链路带宽约束和链路时延约束构造初始候选物理节点集合。

8)利用TKG中实体冷却状态、置信度和风险值过滤高风险候选节点。

9)采用MAMC对剩余候选节点进行启发式排序，生成候选集合C_{feasible}及其规则特征。

10)通过GCN编码当前物理网络状态，通过时间感知GraphSAGE编码TKG经验状态，并与MAMC规则特征进行三模态融合。

11)Actor根据融合表示输出动作分布$\pi_\theta(a_t|s_t)$，并在C_{feasible}上采样或选择动作a_t。

12)执行当前虚拟节点n_v到物理节点a_t的放置，并更新部分映射集合M_v。

13)对于端点均已完成映射的虚拟链路，调用算法2执行约束K最短路径链路映射，得到物理路径、资源占用变化和链路映射结果。

14)if 当前节点放置或链路映射失败 then

15)释放当前VNR已临时占用的物理资源；

16)将失败原因、相关物理节点/链路、瓶颈路径和时间戳写入TKG，并按算法1更新实体风险值与冷却状态；

17)设置flag ← fail，并跳出当前VNR的节点映射循环；

18)else

19)扣减对应CPU和带宽资源；

20)将节点放置、链路映射、资源占用和QoS约束满足情况作为成功事件写入TKG，并按算法1更新实体经验属性；

21)end if

22)根据式(9)计算当前步骤奖励r_t，获得下一状态s_{t+1}。

23)将 $(s_t,a_t,r_t,s_{t+1},\log\pi_{\theta_{\text{old}}}(a_t|s_t),V_\phi(s_t),\text{done}_t)$ 写入轨迹缓存$\mathcal{D}$。

24)令$s_t \leftarrow s_{t+1}$。

25)end for

26)if flag = success then

27)接纳当前VNR，并记录完整节点映射与链路映射方案；

28)else

29)拒绝当前VNR，保持物理资源状态一致性；

30)end if

31)当已接纳VNR生命周期结束时，释放其占用的物理资源，并将释放事件写入TKG。

32)if$|\mathcal{D}| \geqslant T_{\text{upd}}$ then

33)基于轨迹缓存中的奖励序列和Critic估计，计算回报$\mathcal{R}_t$与GAE优势函数$\hat{A}_t$：$\delta_t = r_t + \gamma V_\phi(s_{t+1}) - V_\phi(s_t)$；$\hat{A}_t = \sum_{l=0}^{\infty}(\gamma\lambda_{\text{GAE}})^l\delta_{t+l}$

34)计算新旧策略概率比：$r_t(\theta) = \dfrac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{\text{old}}}(a_t|s_t)}$

35)构造PPO裁剪策略目标：

$$L^{\text{clip}}(\theta) = \mathbb{E}_t[\min(r_t(\theta)\hat{A}_t,\text{clip}(r_t(\theta),1-\epsilon_{\text{clip}},1+\epsilon_{\text{clip}})\hat{A}_t)]$$

36)Critic价值损失：$L^V(\phi) = \mathbb{E}_t[(V_\phi(s_t) - R_t)^2]$

37)构造总优化目标：

$$L(\theta,\phi) = -L^{\text{clip}}(\theta) + c_v L^V(\phi) - c_e\mathcal{H}(\pi_\theta)$$

其中，$\mathcal{H}(\pi_\theta)$为策略熵正则项。

38)基于轨迹缓存$\mathcal{D}$按小批量大小B_{ppo}执行K_{ppo}轮梯度更新，并采用梯度裁剪阈值$g_{\max}$保持训练稳定。

39)$\theta_{\text{old}} \leftarrow \theta$，清空轨迹缓存$\mathcal{D}$。

40)end if

41)end for

42)输出训练后的策略 π_θ、价值函数 V_ϕ 以及在线映射结果。

本文引入MAMC作为动作脚手架，用于构造候选集合与规则特征，以缩减组合动作空间并提升收敛稳定性。需要强调的是，启发式仅用于提供结构化先验与可解释候选约束，最终动作仍由策略网络在交互数据驱动下学习得到。为降低候选截断带来的潜在偏差，本文在训练阶段采用自适应探索机制，以固定目标接受率 $ac^{\backslash *}$ 为参照，当当前接受率偏离 $ac^{\backslash *}$ 时，增强探索以扩大策略搜索；当当前接受率接近 $ac^{\backslash *}$ 时，降低探索以稳定收敛，从而避免对启发式规则的锁定依赖。

2.6　算法示例与复杂度分析

为便于理解算法1~算法3的协同关系，下面给出一个小规模在线映射示例。算法3作为总体流程，根据当前VNR和物理网络状态构造候选动作集合；算法1负责维护TKG中的实体经验属性、风险值和时间编码表示；算法2在虚拟链路端点确定后执行带宽与时延约束下的 K 最短路径链路映射。三者共同完成一次VNR的节点放置、链路映射、反馈写入和策略更新。设在时刻 t 的物理网络参数设置以及虚拟网络请求遵循初始定义，在线嵌入时，算法3首先依据当前VNR需求和物理网络状态对候选动作进行可行性过滤，并结合MAMC形成候选物理节点的启发式排序，从而构造候选集合 C_{feasible} 以缩减动作空间；随后，GCN对 $G_p(t)$ 的拓扑与资源状态进行结构编码。同时，算法1维护的TKG为候选节点/链路提供“经验属性”（如历史失败模式、拥塞风险与收益趋势），并通过时间编码刻画事件新鲜度。二者融合后，Actor在动作掩码约束下确定虚拟节点 n_v 的映射结果 $\text{map}(n_v) \in N_p$，并进一步为虚拟链路 l_v 选择物理路径 $\text{path}(\text{map}(l_v))$。随后，系统对候选映射执行可行性校验。一方面，检查节点一对一约束 $\text{map}(n_v) \neq \text{map}(n_v')$，并验证节点资源约束 $C_t\text{map}(n_v) \geqslant \text{req}(n_v), \forall n_v \in N_v$。另一方面，对路径上的每条物理链路 $l_p \in \text{path}(\text{map}(l_v))$，若校验失败（如带宽或时延约束不满足），系统将“失败原因−对应的 $\text{map}(\cdot)$ 或 $\text{path}(\cdot)$−时间戳”等信息写入TKG，并触发风险冷却以降低相同高风险动作在后续决策中的选择概率；若映射成功，则将本次嵌入的收益、资源占用、带宽及时延约束满足情况作为事件写入TKG，形成可追溯的在线经验。该过程对 G_v 中各虚拟节点与虚拟链路迭代执行，直至完成一次VNR的放置与路由映射；生命周期结束后，释放占用的物理资源，为后续请求到达提供新的可用状态。

在线开销主要由“候选过滤与掩码构造、图编码推理、候选排序输出、必要的约束校验与知识更新”构成。物理网络在时刻 t 的状态为 $G_p(t) = (N_p, L_p)$，其中 $|N_p|$ 与 $|L_p|$ 分别为物理节点与物理链路数量；虚拟网络请求为 $G_v = (N_v, L_v)$，其中 $|N_v|$ 与 $|L_v|$ 分别为虚拟节点与虚拟链路数量；候选集大小为 K，嵌入维度为 d。对单次VNR的在线编排过程如下。

（1）可行性过滤与动作掩码构造主要为线性扫描，复杂度约为 $O(|N_p| + |L_p|)$。

（2）物理图GCN前向编码的开销与链路规模近似线性相关，可写为 $O(L_g \cdot |L_p| \cdot d)$，其中 L_g 为GCN层数。

（3）TKG的GraphSAGE在采样策略下的前向开销与采样规模相关，可近似写为 $O(L_s \cdot S \cdot d)$，其中 L_s 为GraphSAGE层数，S 为每层采样邻居数。

（4）候选排序与策略输出通常不超过 $O(K \log K)$。

约束校验（如链路带宽/时延可行性验证）仅在生成映射方案后触发，其实际开销随 $|L_v|$ 近似线性增长；知识环更新（事件写入、冷却与触发式剪枝等）同样与 $|N_v| + |L_v|$ 近似线性相关。总体而言，Top-K候选机制将组合动作空间从 $|N_p|$ 压缩到 K，使在线推理主要计算集中在规模可控的候选集合上，从而在不改变整体框架主体的前提下保证在线编排效率。

除单次VNR在线推理复杂度外，本文进一步分析TKG存储复杂度与策略环训练复杂度。设TKG中持久实体数量为 $|V_K^p|$，瞬态VNR及映射事件实体数量为 $|V_K^t|$，关系边数量为 $|E_K|$，实体嵌入维度为 d，则TKG的存储复杂度为：

$$O\big((|V_K^p| + |V_K^t|)d + |E_K|\big) \tag{26}$$

由于本文设置容量上限$C_{\max}$，并在超过阈值后对瞬态实体执行经验蒸馏与剪枝，因此长期运行下有：

$$|V_K^p| + |V_K^t| \leqslant C_{\max} \tag{27}$$

从而TKG的存储开销被约束为：

$$O(C_{\max} d + |E_K|) \tag{28}$$

对策略环训练而言，设每次PPO更新使用批量大小为B，每条轨迹平均包含$\bar{n}_v$个虚拟节点决策，Actor-Critic网络参数规模为$|\Theta|$，PPO每轮更新次数为U，则单次策略环训练复杂度可近似表示为：

$$O\left(UB\bar{n}_v\left(L_g|E_s|d + L_k S^{L_k} d + |\Theta|\right)\right) \tag{29}$$

其中，L_g为GCN层数，$|E_s|$为物理网络链路数，L_k为GraphSAGE层数，S为每层采样规模。该复杂度对应PPO的on-policy批量更新过程，不涉及目标网络软更新或离线经验回放。式(29)表明，训练成本由物理图编码和策略网络反向传播决定；TKG由于采用采样式GraphSAGE和容量约束维护，其存储与推理开销不会随历史事件无界增长。

3　性能评价指标

本节面向IoA服务任务图在线映射问题定义性能评价指标。与完整智能体服务治理系统不同，本文评价对象限定为任务图结构确定后的资源映射层，即在VNR持续到达、执行和释放过程中，编排器能否在共享算网资源底座上完成有效的节点放置与链路映射。

本文给出可扩展的链路时延约束形式，当前实验重点评价CPU和带宽资源条件下的在线映射性能，时延参数的实网标定与实验验证将在后续工作中开展。

因此，本文从4个方面评价方法性能：①长期接纳率，用于刻画系统对持续到达任务图的承载能力；②平均收益成本比，用于刻画映射路径开销和资源经济性；③时延约束满足率，用于检查已接纳请求是否满足基础QoS约束；④资源利用波动，用于衡量在线运行过程中物理资源使用状态的稳定性。此外，本文在仿真实验中报告单次在线决策时延，用于评估方法作为控制面任务部署算法的执行开销。上述指标共同刻画“准入能力–资源经济性–基础QoS约束–运行稳定性”的综合性能，但不覆盖任务自动分解、运行中实例扩缩容和已部署服务迁移等完整IoA服务治理能力。

3.1　长期接纳率（long-term acceptance ratio, LAR）

长期接纳率用于刻画系统在VNR持续到达与资源动态释放条件下的总体承载能力。设仿真期间到达的VNR总数为N_{arrived}，其中成功完成节点放置、链路映射并被系统接纳的请求数为N_{accepted}，则长期接纳率定义为：

$$\text{LAR} = \frac{N_{\text{accepted}}}{N_{\text{arrived}}} \tag{30}$$

LAR度量的是资源映射层面的接纳能力，而非完整业务流程中的服务发现成功率或运行中的业务可用率。该指标越高，说明系统能够在长期运行过程中为更多智能体服务任务图提供可行资源映射，体现更强的任务部署准入能力。

3.2　平均收益成本比（revenue-to-cost ratio, R/C）

平均收益成本比用于刻画嵌入决策的资源经济性，重点反映虚拟链路映射为多跳物理路径时产生的额外链路占用开销。对任一被接纳的请求G_v，其基础收益$R(G_v)$按式(1)计算。为度量路径拉长带来的额外资源消耗，本文将嵌入成本定义为基础资源需求与路径诱导开销之和：

$$C(G_v) = R(G_v) + \sum_{l_v \in L_v} \text{req}(l_v) \cdot (|\text{path}(\text{map}(l_v))| - 1) \tag{31}$$

其中，$|\text{path}(\text{map}(l_v))|$表示虚拟链路$l_v$映射到的物理路径跳数。当虚拟链路映射为单跳物理链路时，该项不产生额外路径开销；当路径跳数增加时，额外带宽占用随之增加。在此基础上，平均收益成本比定义为被接纳请求集合上的累计收益与累计成本之比：

$$\text{R/C} = \sum_{G_v \in \text{Accepted}} \frac{R(G_v)}{\sum_{G_v \in \text{Accepted}} C(G_v)} \tag{32}$$

当大多数虚拟链路能够映射为短路径时，路径诱导开销较小，R/C更接近1；当映射普遍依赖长路径跨域连接时，额外占用增大，R/C下降。因此，R/C不仅反映“是否接纳”，更反映“以何种资源代价接纳”，可用于评估资源编排的可持续性与经济性。

3.3　资源利用波动（resource utilization fluctuation, RUF）

除长期接纳率和收益成本比外，IoA服务任务图在线映射还需要关注资源状态在长期运行中的稳定性。若资源使用在少数节点或链路上剧烈波动，

容易导致局部拥塞和后续请求失败。为此，本文定义资源利用波动衡量节点CPU和链路带宽利用率在时间维度上的变化程度。设物理节点p_i在时刻t的CPU利用率和链路l_j在时刻t的带宽利用率分别为：

$$U_i^C(t) = 1 - \frac{C_i^{\text{res}}(t)}{C_i^{\text{cap}}}$$
$$U_j^B(t) = 1 - \frac{B_j^{\text{res}}(t)}{B_j^{\text{cap}}} \tag{33}$$

RUF判别公式为：

$$\text{RUF} = \frac{1}{T}\sum_{t=1}^{T}\left[\text{Var}_{p_i \in N_p}\left(U_i^C(t)\right) + \text{Var}_{l_j \in L_p}\left(U_j^B(t)\right)\right],$$
$$\text{且 RUF} = -\frac{1}{T}\sum_{t=1}^{T} R_{\text{bal}}(t) \tag{34}$$

RUF可直接用实验过程中记录的负载均衡奖励或剩余资源状态计算。RUF越低，说明节点CPU和链路带宽利用状态越平稳，系统不易出现资源使用剧烈振荡；RUF越高，说明在线映射过程可能导致局部资源状态频繁起伏，增加后续请求映射失败的风险。

3.4　在线决策时延

为评估KD-AAO在在线部署阶段的执行开销，统计单次VNR映射决策的在线决策时延（online decision latency, ODL）。该指标由物理图编码、TKG-GraphSAGE推理、三模态融合与策略输出、知识更新等模块的前向耗时组成，可表示为：

$$\text{ODL} = T_{\text{GCN}} + T_{\text{TKG}} + T_{\text{fusion}} + T_{\text{update}} \tag{35}$$

其中，T_{GCN}、T_{TKG}、T_{fusion}和T_{update}分别表示物理网络图编码、时序知识图谱推理、融合策略输出和知识写入更新耗时。本文报告平均值和P90耗时，以反映不同请求规模和网络状态下的尾部时延。需要说明的是，ODL用于衡量服务任务图部署阶段的控制面编排开销，本文方法不面向毫秒级数据面转发或智能体交互消息调度。因此，若目标场景要求毫秒级闭环控制，还需进一步引入增量图编码、嵌入缓存复用和并行路径搜索等工程优化。

综上，本文以LAR和R/C为主要性能指标，分别评价服务任务图在线映射的长期准入能力和资源经济性；以RUF为辅助工程指标，分别评价基础时延QoS约束满足情况和资源使用稳定性；以ODL刻画在线部署阶段的控制面推理开销。后续仿真实验将围绕“准入能力–资源经济性–基础QoS–运行开销”的综合目标，验证KD-AAO在标准负载、高负载和跨拓扑迁移条件下的有效性。

4　仿真分析

本节围绕IoA服务任务图在线映射问题开展仿真实验，重点验证KD-AAO在长期准入能力、资源经济性、动态负载适应、跨拓扑迁移和在线决策开销方面的性能。本文实验评价对象为任务图结构确定后的节点放置与链路映射，不涉及任务自动分解、运行中实例扩缩容、已部署服务迁移和跨管理域知识共享。

4.1　仿真设置

实验在统一配置（OS：Ubuntu Linux 20.04，CPU：2×Intel Xeon Gold 5218R（2.1 GHz，20 cores），内存：256 GB DRAM，GPU：2 ×NVIDIA Tesla A100）下完成，使用Python 3.9与PyTorch 2.7.1实现深度模型；基线实现参考两个成熟的开源平台框架（层次强化学习驱动的VNE-HRL框架[29]和基于图注意力学习的GAL-VNE框架[30]），以保证对比公平性与复现可行性。为控制随机性影响，所有实验均进行多次独立运行并更换随机种子，最终结果报告均值。

为保证结果可复现并贴合IoA专题关注的“服务治理与资源调度、智能体驱动网络智能化、典型应用与验证”，本节从3条主线组织实验：组件治理贡献、高负载可扩展性和真实拓扑可迁移性。IoA的资源底座通常呈现“多域边缘节点–区域汇聚/网关节点–骨干互联”的层级结构：边缘侧节点在同一局部域内连接较为密集以支撑高频交互与低时延协作，而跨域通信往往需要经过少量网关/汇聚节点并沿骨干链路转发，从而形成具有“社团结构与枢纽瓶颈”的网络形态。本文实验采用的合成拓扑与真实拓扑均可在结构上刻画上述特征：一方面，合成拓扑用于模拟不同规模与连通密度下的边缘–骨干资源底座，以评估算法在多种IoA网络状态下的稳健性；另一方面，真实拓扑包含更复杂的社团结构与枢纽链路，有助于验证方法在跨域通信与枢纽拥塞风险下的泛化能力。因此，尽管实验拓扑来源不同，但其结构属性与IoA典型的“多簇协同、网关转发、跨域链路代价”具有一致性，可用于支撑IoA场景下的资源自主编排评估。

物理网络采用Waxman随机图模型生成，参数

$\alpha = 0.5, \beta = 0.2$，规模为100个节点、约500条骨干链路。节点CPU与链路带宽容量服从[50,100]的均匀分布。收益计算中资源权重取$\alpha = 1.0$，对节点与链路资源赋予同等重要性。采用时隙模型，每个时间单位对应100 s现实运行窗口。

IoA请求抽象设置，即VNR生成。VNR按泊松过程到达，标准负载下平均到达率为每100个时间单位4个请求；每个VNR节点数在2~10均匀采样，节点对以50%概率连边；节点与链路需求均在[0,50]内均匀采样；生命周期服从指数分布，平均为1 000个时间单位。

学习类模型统一采用Adam（学习率1×10^{-4}）、折扣因子$\gamma = 0.95$训练至收敛；所有DRL方法使用相同训练数据（5 000个VNR）并保存最优checkpoint。对于KD-AAO，PPO采用on-policy轨迹缓存收集状态、动作、奖励、旧策略概率和价值估计，轨迹缓存容量设置为10 000条转移记录，小批量大小为64。该缓存仅用于当前策略轨迹的批量更新，不进行离线经验回放。TKG最多维护10 000个实体，超过容量上限后触发容量约束下的经验蒸馏与剪枝，从而使历史经验能够持续沉淀，同时保证长期在线运行中的维护开销可控。TKG实体统计量采用指数滑动平均维护，图嵌入采用两层GraphSAGE（16维），每层邻居采样数依次设置为10和5，并引入本文提出的时间编码以表征事件新鲜度与趋势。主要超参数设置如表3所示。

表3中的参数在后续消融实验、高负载测试和跨拓扑迁移测试中保持一致，未针对Brain拓扑或特定业务分布进行单独调参。

对比方法选取8个代表性方法，覆盖传统启发式与DRL：D-ViNE、PPO-VNE、GRC-VNE、MCTS、GAL-VNE、DeepViNE、RL-VNE、DDPG-VNE。消融实验100次独立运行；高负载与泛化实验200次独立运行，每次更换随机种子，报告均值±标准差。这些基线覆盖3类代表性方法：传统启发式与搜索方法，包括D-ViNE、GRC-VNE和MCTS；深度强化学习方法，包括PPO-VNE、DeepViNE、RL-VNE和DDPG-VNE；图学习增强方法，包括GAL-VNE。该设置能够从规则搜索、端到端策略学习和图结构感知3个维度评估KD-AAO的相对优势。本文选取了覆盖启发式搜索、深度强化学习、图神经网络增强VNE和连续动作策略学习的代表性方法。其中，GAL-VNE代表图注意力学习驱动的VNE方法，PPO-VNE与DDPG-VNE代表基于策略优化的DRL-VNE方法，MCTS代表搜索式在线决策方法。通过这些基线，可以分别检验KD-AAO在结构感知、长期策略优化、动作空间搜索和跨拓扑泛化方面的性能差异。

表3 主要超参数设置

类别	参数	默认值
PPO训练	Adam优化器学习率	1×10^{-4}
PPO训练	长期奖励折扣因子γ	0.95
PPO训练	PPO裁剪系数ϵ_{clip}	0.20
PPO训练	广义优势估计参数λ_{GAE}	0.95
PPO训练	轨迹缓存容量	10 000
PPO训练	小批量训练样本数Batch-size	64
PPO训练	每次轨迹收集后的PPO更新轮数K_{ppo}	4
PPO训练	熵正则系数	0.01
PPO训练	价值损失系数	0.5
PPO训练	梯度裁剪阈值	0.5
TKG维护	TKG实体容量上限$N_{\max}$	10 000
TKG维护	成功率/失败率EMA平滑系数ρ	0.90
TKG维护	小样本置信度修正系数κ	10
TKG维护	时间编码尺度基数B	10^4
TKG维护	时间编码维度	16
TKG维护	GraphSAGE层数	2
TKG维护	GraphSAGE采样规模	10、5
风险计算	失败率风险权重β_1	0.50
风险计算	拥塞风险权重β_2	0.30
风险计算	路径/资源代价风险权重β_3	0.20
冷却机制	冷却判断滑动窗口W_c	20
冷却机制	冷却失败率阈值θ_f	0.60
冷却机制	冷却置信度阈值θ_c	0.50
恢复机制	恢复判断窗口W_r	10
恢复机制	性能恢复所需成功次数N_{succ}	3
恢复机制	资源恢复阈值θ_{res}	0.20
恢复机制	最大冷却持续时间T_{cool}	50
恢复机制	温启动系数γ_w	0.50
链路映射	K最短路径候选路径数	5
链路映射	路径跳数代价权重ω_1	0.40
链路映射	带宽占用压力权重ω_2	0.40
链路映射	链路风险代价权重ω_3	0.20

所有对比方法均冻结其训练得到的网络参数，并接收相同的目标拓扑、当前物理资源状态、VNR 需求和资源释放信息。各方法均不允许执行基于目标拓扑样本的梯度更新。KD-AAO-OnlineTKG 特有的事件写入和经验属性更新属于所提方法的运行时状态更新，而非策略再训练。为避免将该在线状态更新的收益与策略本身的跨拓扑能力混为一谈，本文通过 StaticTKG 与 OnlineTKG 对比单独量化在线知识适配的贡献。

为区分策略参数迁移与在线知识适配，跨拓扑实验采用统一的冻结策略协议。模型在 Waxman 拓扑上完成训练后，Actor、Critic、GCN 编码器、GraphSAGE 编码器以及融合 MLP 的参数在 Brain 拓扑测试阶段全部冻结，测试过程中不执行反向传播、优化器更新、参数微调或拓扑特定再训练。物理网络的剩余 CPU、带宽及 VNR 生命周期状态仍根据映射和释放过程正常变化，这是环境状态演化，而非模型训练。

对于 KD-AAO，Brain 测试开始时依据目标拓扑重新构建物理节点、物理链路及其关系实体，不直接复制 Waxman 拓扑中的实体级事件记录，模型仅迁移已经训练完成的编码器和策略参数。为进一步区分 TKG 在线更新所带来的作用，本文设置两种测试模式：KD-AAO-StaticTKG，冻结策略网络参数，同时固定 TKG 经验属性、事件历史及冷却状态，仅允许物理资源状态随环境变化；KD-AAO-OnlineTKG，冻结全部神经网络参数，但允许按照算法 1 写入测试阶段的映射事件，并更新成功率、失败率、置信度、风险值和冷却/恢复状态。两种模式采用相同初始经验属性、相同 VNR 序列和相同随机种子。

为保证知识治理机制的可复现性，本文在实验中固定冷却、恢复、蒸馏和 GraphSAGE 编码相关参数，并在所有实验场景中保持一致。TKG 在每次映射尝试后进行事件写入和经验统计更新，当实体规模超过容量上限 $N_{\max}$ 时，触发容量约束下的经验蒸馏与剪枝。该设置避免了针对特定负载或特定拓扑进行额外调参，同时保证长期在线运行中的图谱规模可控。

为进一步量化推理时延，本文在实验平台上对“单次在线决策前向推理”进行微基准测试（micro-benchmark）。具体地，在关闭反向传播的条件下，对每个 VNR 的在线决策过程统计前向耗时，并按模块拆分为 GCN 编码、TKG-GraphSAGE 推理、融合与策略输出，以及知识更新并写入与冷却。在固定网络规模与请求规模下重复多次采样，报告平均耗时与 P90 耗时。该量化结果可直接反映 KD-AAO 在时延敏感 IoA 场景中执行在线编排的推理开销。

表 4 给出了在线推理的微基准测试结果，并将单次在线决策的前向耗时拆分为 GCN 编码、时序知识图谱推理、融合与策略输出和知识更新 4 个模块。可以看到，各模块的 P90 耗时均高于平均耗时，这符合在线推理在不同网络状态与请求规模下存在尾部波动的特点。总体而言，在线推理耗时主要由物理图的 GCN 编码占据（平均 863.91 ms，P90 为 978.15 ms），而时序知识图谱推理、融合与策略输出以及知识更新的开销相对较小（平均分别为 118.91 ms、38.00 ms 与 27.79 ms）。该拆分结果表明，KD-AAO 的时序推理与知识维护并未成为主要时延瓶颈，系统开销主要集中在对物理图结构状态的编码阶段，为后续工程优化提供了明确方向。

表 4　在线推理的微基准测试结果

模块	平均耗时/ms	P90 耗时/ms	说明
GCN 编码	863.91	978.15	物理图前向
时序知识图谱推理	118.91	126.11	局部子图采样推理
融合与策略输出	38.00	53.80	MLP/Actor 前向
知识更新	27.79	34.35	写入+冷却
合计	1 048.61	1 192.41	单次在线决策

为保证跨拓扑可迁移实验的可验证性，本文明确区分训练拓扑与测试拓扑：训练阶段仅使用 Waxman 合成拓扑生成 VNR 交互轨迹，测试阶段将训练好的策略直接部署到 Brain 真实拓扑，测试过程中不进行梯度更新、参数微调或拓扑特定再训练。二者的主要差异在于：Waxman 拓扑更接近随机几何图结构，而 Brain 拓扑具有更明显的社团结构和枢纽节点，能够模拟 IoA 跨域部署中常见的簇内密集连接与跨簇瓶颈链路特征。因此，该实验用于检验策略是否能够从合成结构迁移到具有真实结构特征的异构网络环境。

4.2 自治机制贡献:消融实验

在智能体驱动网络自治框架中，关键问题不是“是否能学习一个策略”，而是“是否具备可积累经验、可纠错、可稳健执行的自治闭环”。为量化框架各自治组件的边际贡献，本文在标准负载（λ=0.04）下进行消融实验，设置4种配置：KD-AAO-Full（完整模型），KD-AAO-NoTKG（移除时序知识图谱，剥夺历史知识），KD-AAO-NoRank（移除启发式排序，削弱动作脚手架），KD-AAO-NoFeedback（移除反馈奖励，弱化风险治理）。每种配置在10次独立运行下报告均值与标准差。

表5给出了不同组件变体在标准负载下的消融结果。KD-AAO-Full的LAR为84.20%±0.80%，R/C为54.30%±0.40%，体现出较好的服务准入能力与资源经济性。KD-AAO-NoTKG带来最明显的性能退化：LAR降至78.90%±1.50%（下降5.30个百分点），R/C降至45.80%±0.35%（下降8.50个百分点）。这说明显式世界模型在自治编排中发挥核心作用。TKG将历史交互与失败模式沉淀为可行动知识，使智能体能够提前规避潜在的长路径与高开销映射，从而同时改善吞吐量与成本控制。与此同时，KD-AAO-NoTKG的波动也显著增大（标准差由0.80增至1.50），表明缺乏历史记忆会削弱策略的稳定性与可预测性，而稳定性正是网络自治可信运行的重要基础。

表5 不同组件变体在标准负载下的消融结果

变体	LAR	R/C	RUF
KD-AAO-Full	**84.20%±0.80%**	**54.30%±0.40%**	**0.10±0.01**
KD-AAO-NoTKG	78.90%±1.50%	45.80%±0.35%	0.17±0.02
KD-AAO-NoRank	80.70%±0.85%	48.60%±0.30%	0.14±0.02
KD-AAO-NoFeedback	82.90%±1.15%	54.10%±0.45%	0.12±0.01

KD-AAO-NoRank同样造成性能的显著下降：LAR为80.70%±0.85%（下降3.50个百分点），R/C为48.60%±0.30%（下降5.70个百分点）。这表明Ranker提供了有效的结构归纳偏置，使智能体更倾向于选择拓扑位置更优、后续链路映射更短的承载点，从而降低路由代价并提升长期接纳效率。该机制可被视为一种面向自治决策的“动作脚手架”，通过缩小搜索空间降低无效探索，提升学习与执行效率。

KD-AAO-NoFeedback使LAR小幅下降至82.90%±1.15%（下降1.30个百分点），而R/C基本保持不变（54.10%±0.45%，仅下降0.20个百分点）。这表明反馈机制的主要贡献不在于单次映射的经济性提升，而在于对高风险节点的在线治理与稳健执行：通过降低重复试错和短期波动，系统能在动态环境中维持更稳定的服务准入能力，即便对路径代价的影响相对有限。

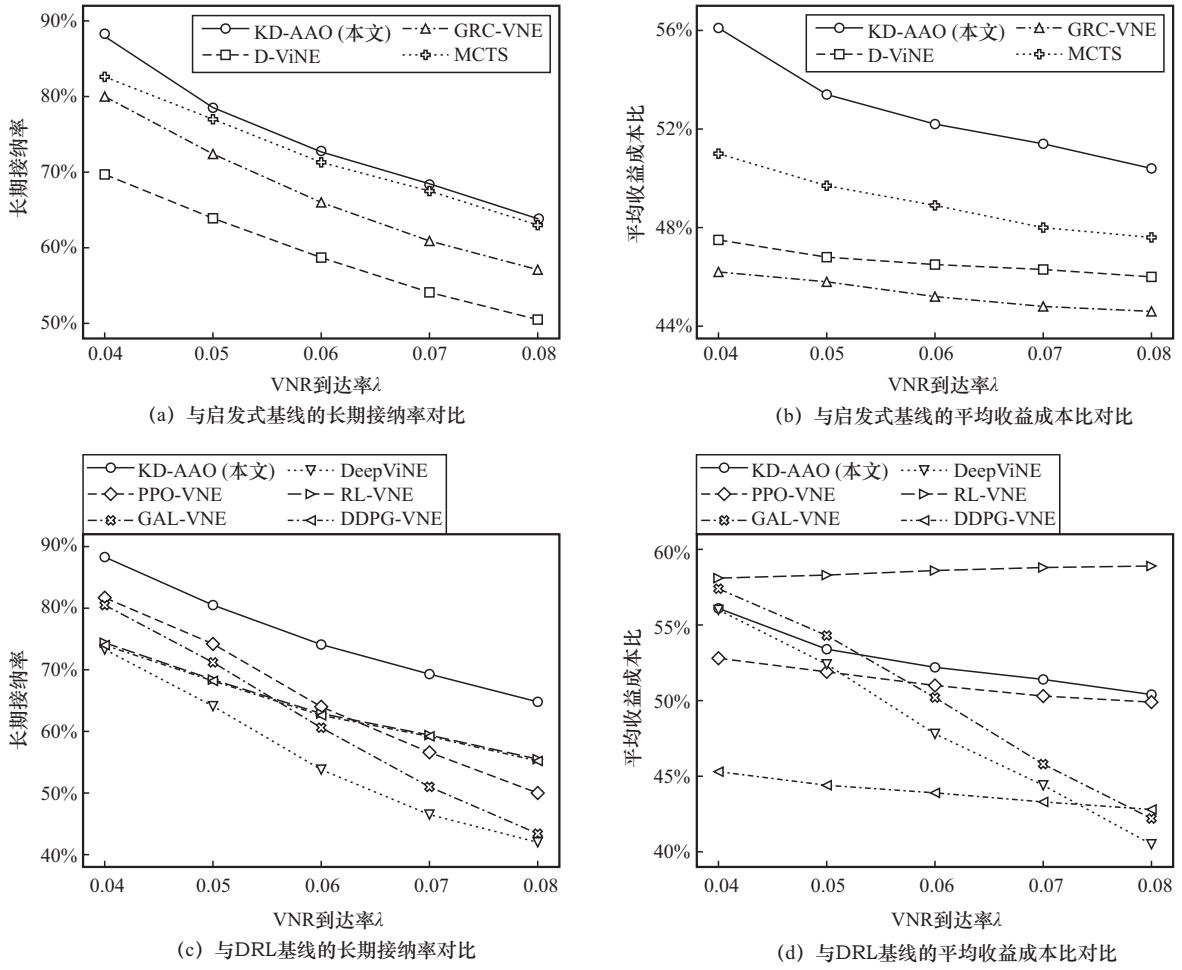
综上，消融实验从3个侧面验证了自治闭环的关键支柱：TKG提供长期记忆与趋势认知，Ranker提供可解释的结构先验与动作约束，反馈奖励机制提供风险治理与稳健执行能力。三者协同支撑IoA场景下的智能体驱动网络自治优化。

4.3 高负载压力测试与可扩展性验证

为评估在资源竞争加剧且请求密集到达条件下的自治鲁棒性，本文进行高负载压力测试，将VNR到达率λ从0.04逐步提升至0.08。为消除“在线再训练”带来的适配收益，所有学习型方法均采用单一预训练模型直接部署，测试过程不进行任何额外训练或参数更新，从而以零适配成本考察策略在更苛刻负载下的自适应表现。各设置下独立运行20次并取均值，结果如图3所示。

图3(a)和图3(c)给出的LAR曲线显示，KD-AAO在所有到达率下均保持最高的长期接纳率，并在负载上升过程中维持稳定领先。为评估请求密集到达条件下的负载适应能力，本文将VNR到达率由λ=0.04逐步提高至λ=0.08。KD-AAO的LAR分别为87.6%、79.8%、73.5%、68.7%、64.0%。在基准负载λ=0.04下，KD-AAO相对最强启发式基线MCTS（81.8%）与最强DRL基线PPO-VNE（80.9%）分别领先5.8与6.7个百分点。随着λ增大，各方法LAR均下降，但KD-AAO在λ=0.08时仍保持显著优势（64.0%），说明显式世界模型与知识驱动规划能够在资源趋紧时更有效地进行长期资源安排，从而缓解拥塞累积并维持更高的服务吞吐量。

图3(b)和图3(d)的R/C曲线进一步揭示了不同策略在“吞吐量”与“资源经济性”之间的取舍。RL-VNE在各负载下保持最高R/C（57.7%~58.5%），表明其更偏好短路径映射，倾向于以经



(a) 与启发式基线的长期接纳率对比　(b) 与启发式基线的平均收益成本比对比

(c) 与DRL基线的长期接纳率对比　(d) 与DRL基线的平均收益成本比对比

图 3　高负载条件下各模型长期接纳率与平均收益成本比的对比

济性优先的方式降低资源消耗，但其LAR相对受限。相比之下，KD-AAO在保持最高LAR的同时仍维持具有竞争力的R/C（由55.6%下降至50.0%），体现出更符合IoA自治目标的综合折中：系统不仅能接纳更多服务请求，也能在资源代价可控的前提下保持持续运行能力。

4.4　跨域自治与拓扑陷阱识别能力验证

IoA的真实部署往往面临显著的拓扑异质性，如存在社团簇结构与少数跨域枢纽节点。在此类网络中，短视策略容易反复占用枢纽导致跨域拥塞，形成典型的“拓扑陷阱”。为评估跨域自治能力，本文将仅在合成Waxman拓扑上训练的模型直接迁移到Brain真实拓扑（161个节点）进行测试，该拓扑具有自然簇结构与高度连接的枢纽节点。关键设置是：全程不进行拓扑特定再训练，从而检验方法在未知异质拓扑上的自治适应能力。

首先，本文在与训练阶段一致的Uniform VNR分布下评估跨拓扑泛化能力，结果如表6所示。由于策略在同质Waxman网络上训练，而测试拓扑具有更明显的簇结构与枢纽节点特征（可视作异构智能体组网环境的抽象），各类方法在迁移后均出现不同程度的性能下滑。尽管如此，KD-AAO依然保持了更强的自治适应性：其LAR达到31.60%±0.48%，相比接纳率最高的DRL基线PPO-VNE（25.10%±0.33%）提升6.50个百分点，同时维持较高的资源经济性（R/C为76.90%±0.55%）。从对比结果可以看出，多数学习型基线在该场景下的LAR主要分布在15%~17%（如GAL-VNE为17.30%±0.55%，DeepViNE为15.40%±0.42%，DDPG-VNE为15.60%±0.45%），表明其策略更容易固化为对训练拓扑的结构偏好，迁移到簇化且枢纽显著的网络后，难以及时修正节点质量评估与路径代价预期。KD-AAO的优势在于其TKG提供了“经验纠偏层”：系统会将映射失败、路径代价上升与枢纽拥

塞等现象持续写入世界模型，使枢纽节点在高频过载时被快速标记为高风险目标，抑制短视的枢纽堆叠选择，保留跨簇连通性与可用编排空间。该过程本质上体现为一种在线自治能力，即在未知拓扑中通过经验积累识别“拓扑陷阱”，并以此指导后续决策，使网络自治在异质环境下仍具备可持续运行的可靠性。

表6　智能体组网跨拓扑泛化结果（均匀业务）

方法	LAR	R/C
D-ViNE	15.90%± 0.32%	51.40%± 1.20%
PPO-VNE	25.10% ± 0.33%	55.60%± 0.70%
GRC-VNE	15.20%± 0.35%	70.90%± 1.55%
MCTS	22.90%± 0.40%	55.30%± 0.40%
GAL-VNE	17.30%± 0.55%	72.60%± 0.85%
DeepViNE	15.40%± 0.42%	51.90%± 2.80%
RL-VNE	16.10%± 0.90%	79.70%± 1.30%
DDPG-VNE	15.60%± 0.45%	69.80%± 1.40%
KD-AAO	**31.60%± 0.48%**	**76.90%± 0.55%**

为模拟更贴近实际部署的业务形态，本文进一步采用指数分布生成VNR（scale=2.5）。其中约70%的请求包含2~4个节点，其余请求为5~8个节点，评估结果如表7所示。与Uniform VNR相比，该设置下各方法整体性能均有所回升，主要原因在于小规模请求更容易在单一簇内完成映射，从而降低了对跨簇枢纽节点的持续占用，缓解了异质拓扑下的结构性瓶颈。在更现实的请求分布下，KD-AAO的优势进一步凸显。其LAR达到82.30%±0.25%，显著高于PPO-VNE的75.80%±0.28%，领先6.50个百分点，同时在资源经济性指标上也保持最优，R/C达到83.40%±0.68%。这一结果表明，当编排目标从“在复杂拓扑中勉强维持可用”转向“在高到达密度下进行高效打包与精细调度”时，KD-AAO的知识驱动机制更能发挥作用：一方面，TKG记录的历史代价与失败模式帮助策略持续识别潜在拥塞点并避免重复试错；另一方面，结构引导与动作脚手架将决策集中在更具拓扑优势的候选区域，使得在高密度小请求场景中能够更快找到低代价、可持续的映射组合，从而同时提升吞吐能力与资源利用效率。

表7　智能体组网跨拓扑泛化结果（指数业务）

方法	LAR	R/C
D-ViNE	45.10%± 0.70%	57.90%± 0.65%
PPO-VNE	75.80%± 0.28%	65.90%± 0.55%
GRC-VNE	45.00%± 0.68%	72.20%± 1.05%
MCTS	64.80%± 0.50%	62.10%± 0.45%
GAL-VNE	61.50%± 1.60%	77.90%± 0.60%
DeepViNE	46.20%± 0.80%	58.00%± 1.30%
RL-VNE	51.60%± 1.70%	79.80%± 0.95%
DDPG-VNE	44.70%± 0.30%	71.80%± 1.25%
KD-AAO	**82.30%± 0.25%**	**83.40%± 0.68%**

表8进一步区分了策略参数迁移和在线知识适配的作用。KD-AAO-StaticTKG在策略及TKG经验状态均冻结时取得29.00%±0.27%的LAR和69.90%± 0.82%的R/C，其相较PPO-VNE的差异主要反映三模态结构、知识增强状态表示和候选动作构造所带来的迁移能力。允许TKG在线更新后，KD-AAO-OnlineTKG的LAR进一步提升至31.60%和82.30%，说明目标拓扑上的成功/失败反馈能够通过经验属性和风险状态修正后续候选资源评价。

KD-AAO在Brain拓扑上的性能收益由两部分组成：一部分来自冻结参数条件下的结构化知识表示和策略迁移能力，另一部分来自不涉及梯度更新的在线知识适配。

综合消融、高负载与跨拓扑泛化实验可以得出，KD-AAO在智能体驱动网络自治方面具备明确优势。其一，TKG作为显式世界模型记忆层，提供长期经验与趋势认知，是提升吞吐量、经济性与

表8　冻结策略参数下TKG在线知识适配的贡献

方法	策略编码器参数	TKG经验属性	Uniform LAR	Uniform R/C	Exponential LAR	Exponential R/C
PPO-VNE	冻结	无TKG	25.10%± 0.33%	55.60%± 0.70%	75.80%± 0.28%	65.90%± 0.55%
KD-AAO-StaticTKG	冻结	冻结	29.00%±0.27%	69.90%± 0.82%	80.10%± 0.33%	78.10%± 0.47%
KD-AAO-OnlineTKG	冻结	在线更新	31.60%± 0.48%	76.90%± 0.55%	82.30%± 0.25%	83.40%± 0.68%

稳定性的核心；其二，启发式动作脚手架为自治策略注入可解释的结构先验，显著降低无效探索与短视放置；其三，自适应冷却机制提供轻量级风险治理能力，提高在线执行的稳健性。在高负载无再训练部署与真实拓扑无再训练迁移条件下，KD-AAO 仍保持领先表现，说明该框架能够在动态与异质环境中实现持续自治优化，这与智能体互联网专题中“智能体驱动的网络智能化与自治演进”研究方向高度契合。

5　结束语

本文面向智能体互联网中服务任务图在共享算网资源底座上的在线部署问题，提出了一种知识驱动的资源编排方法KD-AAO。不同于仅依赖策略网络参数隐式记忆历史经验的DRL-VNE方法，KD-AAO以时序知识图谱为显式世界模型，将物理资源状态、VNR特征、节点放置、链路映射以及成功/失败反馈组织为可查询、可更新的结构化知识，并通过知识环和策略环分别实现在线经验写入与PPO策略优化。在决策阶段，KD-AAO进一步融合GCN编码的实时物理网络状态、时间感知GraphSAGE提炼的历史经验知识，以及MAMC启发式排序提供的规则先验，从而在满足资源与基础QoS约束的候选空间内完成节点放置和链路映射。

未来，将进一步面向IoA的开放环境与安全挑战，探索更细粒度的自治治理策略与信任评估机制，支持多智能体间意图协商、服务组合与跨域算网融合调度，并在更大规模原型系统中开展持续运行验证，以推动智能体互联网从概念走向可部署、可演进的网络基础设施。

参考文献：

[1] 尹浩, 魏急波, 赵海涛, 等. 面向有人/无人协同的智能通信与组网关键技术: 现状与趋势[J]. 通信学报, 2024, 45(1): 1-17.
Yin H, Wei J B, Zhao H T, et al. Intelligent communication and networking key technologies for manned/unmanned cooperation: states-of-the-art and trends[J]. Journal on Communications, 2024, 45(1): 1-17.

[2] Park J S, O’Brien J, Cai C J, et al. Generative agents: interactive simulacra of human behavior[C]//Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology. New York: ACM, 2023: 1-22.

[3] Wooldridge M. An introduction to multi-agent systems[M]. 2nd ed. Hoboken: John Wiley & Sons, 2009.

[4] Wu S, Chen N, Xiao A L, et al. AI-empowered virtual network embedding: a comprehensive survey[J]. IEEE Communications Surveys & Tutorials, 2025, 27(2): 1395-1426.

[5] Satpathy A, Narayan Sahoo M, Swain C, et al. Virtual network embedding: literature assessment, recent advancements, opportunities, and challenges[J]. IEEE Communications Surveys & Tutorials, 2025, 27(6): 3861-3914.

[6] 段晓东, 孙滔, 陆璐, 等. 智能体互联网: 概念、架构及关键技术[J]. 电信科学, 2025, 41(10): 1-10.
Duan X D, Sun T, Lu L, et al. Internet of agents: conception, architecture and key technologies[J]. Telecommunications Science, 2025, 41 (10): 1-10.

[7] Fischer A, Botero J F, Beck M T, et al. Virtual network embedding: a survey[J]. IEEE Communications Surveys & Tutorials, 2013, 15(4): 1888-1906.

[8] Kreutz D, Ramos F M V, Veríssimo P E, et al. Software-defined networking: a comprehensive survey[J]. Proceedings of the IEEE, 2015, 103(1): 14-76.

[9] Mijumbi R, Serrat J, Gorricho J L, et al. Network function virtualization: state-of-the-art and research challenges[J]. IEEE Communications Surveys & Tutorials, 2016, 18(1): 236-262.

[10] Foukas X, Patounas G, Elmokashfi A, et al. Network slicing in 5G: survey and challenges[J]. IEEE Communications Magazine, 2017, 55(5): 94-100.

[11] 夏玮玮, 王博业, 夏雅星, 等. 基于多时间尺度协同的无蜂窝RAN切片资源分配算法[J]. 通信学报, 2025, 46(7): 60-77.
Xia W W, Wang B Y, Xia Y X, et al. Cellular-free RAN slicing resource allocation algorithm based on multi-timescale collaboration[J]. Journal on Communications, 2025, 46(7): 60-77.

[12] 段晓东, 黄正磊, 陆璐, 等. 智能体通信网络(ACN): 面向6G的网络发展新范式[J]. 通信学报, 2025, 46(11): 332-346.
Duan X D, Huang Z L, Lu L, et al. AI agent communication network (ACN): a new network paradigm for 6G[J]. Journal on Communications, 2025, 46(11): 332-346.

[13] 霍如, 沙宗轩, 吕科呈, 等. GAI使能网络智能化: 基于LLM的网络操作系统[J]. 通信学报, 2025, 46(6): 32-44.
Huo R, Sha Z X, Lyu K C, et al. General artificial intelligence enables network intelligence: a network operating system based on large language model[J]. Journal on Communications, 2025, 46(6): 32-44.

[14] Chowdhury M, Rahman M R, Boutaba R. ViNEYard: virtual network embedding algorithms with coordinated node and link mapping[J]. IEEE/ACM Transactions on Networking, 2012, 20(1): 206-219.

[15] Haeri S, Trajković L. Virtual network embedding via Monte Carlo tree search[J]. IEEE Transactions on Cybernetics, 2018, 48(2): 510-521.

[16] Yan Z X, Ge J G, Wu Y L, et al. Automatic virtual network embedding: a deep reinforcement learning approach with graph convolutional networks[J]. IEEE Journal on Selected Areas in Communications, 2020, 38(6): 1040-1057.

[17] Mirjalili S, Lewis A. The whale optimization algorithm[J]. Advances in Engineering Software, 2016, 95: 51-67.

[18] Su S, Li Z, Yao H, et al. Virtual network embedding with coordinated

node and link mapping based on graph optimization and path splitting[J]. IEEE/ACM Transactions on Networking, 2014, 22(1): 89-102.

[19] Song A, Chen W N, Gu T L, et al. Distributed virtual network embedding system with historical archives and set-based particle swarm optimization[J]. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 2021, 51(2): 927-942.

[20] 钟天尧, 姚欢, 冯智斌, 等. 任务场景驱动的无线通信跨层智能适变方法[J]. 通信学报, 2026, 47(2): 61-72.
Zhong T Y, Yao H, Feng Z B, et al. Task scenario-driven cross-layer intelligent adaptation approaches for wireless communication[J]. Journal on Communications, 2026, 47(2): 61-72.

[21] Henderson P, Islam R, Bachman P, et al. Deep reinforcement learning that matters[J]. Proceedings of the AAAI Conference on Artificial Intelligence, 2018, 32(1): 3207-3214.

[22] Kipf T N, Welling M. Semi-supervised classification with graph convolutional networks[C]//International Conference on Learning Representations (ICLR). Vancouver: ICLR, 2017: 1-14.

[23] Veličković P, Cucurull G, Casanova A, et al. Graph attention networks[C]//International Conference on Learning Representations (ICLR). Vancouver: ICLR, 2018: 1-12.

[24] Hamilton W L, Ying R, Leskovec J. Inductive representation learning on large graphs[C]//Proceedings of the 31st International Conference on Neural Information Processing Systems. New York: ACM, 2017: 1025-1035.

[25] Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need[C]//Advances in Neural Information Processing Systems (NeurIPS). Vancouver: ICLR, 2017: 5998-6008.

[26] Schulman J, Wolski F, Dhariwal P, et al. Proximal policy optimization algorithms[PP]. V2. (2017-08-28) [2026-05-13]. arXiv: arXiv.1707.06347.

[27] 孙春霞, 杨丽, 王小鹏, 等. 结合深度强化学习的边缘计算网络服务功能链时延优化部署方法[J]. 电子与信息学报, 2024, 46(4): 1363-1372.
Sun C X, Yang L, Wang X P, et al. Optimized deployment method of edge computing network service function chain delay combined with deep reinforcement learning[J]. Journal of Electronics & Information Technology, 2024, 46(4): 1363-1372.

[28] Abgaz Y, McCarren A, Elger P, et al. Decomposition of monolith applications into microservices architectures: a systematic review[J]. IEEE Transactions on Software Engineering, 2023, 49(8): 4213-4242.

[29] Cheng J, Wu Y L, Lin Y M, et al. VNE-HRL: a proactive virtual network embedding algorithm based on hierarchical reinforcement learning[J]. IEEE Transactions on Network and Service Management, 2021, 18(4): 4075-4087.

[30] Huang Z J, Sun Y Z, Wang W. Generalizing graph ODE for learning complex system dynamics across environments[C]//Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. New York: ACM, 2023: 798-809.

[作者简介]



钱琪杰（1997– ），男，南京邮电大学博士生，主要研究方向为人工智能、物联网和未来网络。



宋阳子（1997– ），女，西安电子科技大学博士生，主要研究方向为人工智能、计算机网络和未来网络。



秦蓁（1995– ），女，博士，军事科学院系统工程研究院在站博士后，主要研究方向为低空智能信息网络、边缘智能。



钟旭东（1991– ），男，博士，军事科学院系统工程研究院副研究员，主要研究方向为智能信息网络技术、卫星通信、无线资源管理与优化。



郭永安（1981– ），男，博士，南京邮电大学教授，主要研究方向为智能通信网络与智能无人系统等。



任保全（1974– ），男，博士，军事科学院系统工程研究院研究员，主要研究方向为物联网、无线通信、移动通信网络技术等。