分布式Agents网络优化与调度技术

# **引言**

## **项目研究背景(张永康)**

随着人工智能的快速发展，机器学习、深度学习等技术在众多领域取得了显著成果，但其模型复杂度和参数量不断增加，对计算资源的需求呈指数级增长。例如，训练大型神经网络需要处理海量数据和进行复杂的计算，单机计算已经难以满足其时效性和资源需求，分布式 Agents 网络成为应对大规模计算任务的有效解决方案之一。同时，在强化学习等领域，也面临着训练效率、多智能体协作等问题，需要通过网络优化和调度来提升性能，以更好地实现智能决策和复杂任务求解。

近年来，以机器学习特别是深度学习为代表的人工智能技术在多个应用领域取得突破性进展。如图像识别[1]通过卷积神经网络实现超人类精度，语音识别[2]基于端到端建模突破传统算法瓶颈，机器翻译[3]依托Transformer架构实现跨语言语义理解，自动驾驶[4]则通过多模态感知系统重塑交通范式。这些成就的取得主要源于三大驱动力：第一，互联网与物联网技术催生了数据爆炸，全球数据总量从2010年的2ZB激增至2020年的59ZB，为模型训练提供了丰富的知识矿藏；第二，算法创新持续突破理论边界，从反向传播到注意力机制，从生成对抗网络到元学习，不断拓展机器认知的疆域；第三，计算范式发生根本性变革，GPU集群算力从2012年AlexNet所需的1.3TFLOPS跃升至GPT-3训练时的3.14EFLOPS，增幅超百万倍。

当前人工智能发展面临显著的算力鸿沟。OpenAI研究显示，2012至2020年间AI训练算力需求年均增长达10倍，而同期Nvidia旗舰GPU（从K20X到A100）的FP32峰值算力仅提升约23倍。这种剪刀差在Transformer类模型中尤为显著：GPT-3的参数量达1750亿，训练能耗相当于126个丹麦家庭的年用电量。在摩尔定律趋缓（晶体管密度年增长降至3%）与登纳德缩放定律失效的双重制约下，单机性能提升已难以满足大模型需求。这促使分布式机器学习向多维度演进：横向扩展层面，参数服务器架构支持千卡级并行训练；纵向深化层面，模型并行技术破解了单卡显存限制；拓扑创新层面，AllReduce通信优化使分布式训练效率提升40%以上。斯坦福大学最新研究[5]表明，通过混合并行策略，千亿参数模型训练时间可从数月压缩至数周，标志着分布式计算正式进入工程化应用阶段。

在大数据时代，分布式Agents系统在处理海量数据和复杂任务时展现出显著优势。机器学习训练任务通常需要对大量数据进行密集计算，以提升模型的准确度。然而，单机系统在计算速度和数据存储方面往往难以满足需求。例如，使用单块Nvidia Tesla V100 GPU训练自然语言处理模型BERT-large需要耗时78天，这显然是不切实际的。分布式Agents系统的目标是将训练任务分布式地部署到多个计算节点上，通过Agents之间的自主决策和协同工作，提高模型训练的速度，减少任务耗时。凭借其高效的任务分配和资源管理能力，分布式Agents系统在高并发场景中扮演着愈发重要的角色，如大规模实时数据处理、分布式任务调度等。

本报告系统梳理了分布式Agents网络优化与调度的最新技术进展，旨在为构建高效、可靠、可扩展的分布式智能系统提供理论参考与实践路径。

## **项目研究意义（张永康）**

分布式Agents网络作为一种新兴的计算范式和协作架构，为解决复杂系统中的众多难题提供了极具潜力的解决方案，在众多领域展现出广阔的应用前景和深远的研究价值。深入研究分布式Agents网络的优化与调度问题，对于推动相关技术发展、提升系统性能以及满足日益增长的应用需求具有极为关键的意义。

在分布式环境下，多个智能体（Agents）通过协同工作能够有效实现共同目标。以云计算中心为例，众多计算节点作为智能体，通过优化计算资源分配策略，可以精准地将任务分配给不同节点，充分发挥各节点的计算优势，从而有效避免资源闲置与过载并存的情况，显著提高整体性能。在卫星任务规划领域，多个卫星作为智能体，能够依据任务优先级、时间窗口以及轨道位置等多方面要求，合理安排观测任务，优化调度方案，进而提高任务成功率和资源利用率，确保卫星系统能够高效地完成复杂的太空观测任务。

分布式Agents网络的性能直接关系到分布式Agents网络的运行效果。优化网络结构和通信机制能够加快信息传递速度、减少通信延迟，从而有效提高系统的响应能力。例如在无人机编队飞行中，优化通信网络可确保无人机之间实时共享位置和状态信息，实现精确编队控制，增强编队的稳定性和灵活性，使其能够更好地应对复杂多变的飞行环境和任务要求。此外，优化后的分布式Agents网络还能显著增强系统对故障的容错能力，当网络中某个智能体出现故障时，其他智能体能够基于预设的容错机制和协同策略，快速调整任务分配和通信路径，维持系统的正常运行，有效降低单点故障对整个系统的影响，保障系统的可靠性与稳定性。

分布式Agents网络所处的环境通常是动态变化的，优化后的网络能够及时感知环境变化，并自适应地调整参数和策略，保持良好性能。在移动通信网络中，用户的移动、信号干扰以及网络负载变化等因素会导致网络拓扑结构和通信质量的实时变化，优化后的分布式Agents网络能够依据这些变化，动态调整资源分配、路由选择和功率控制等参数，以应对复杂多变的通信环境，确保通信质量和网络性能。在智能交通系统中，面对交通流量的实时变化、道路施工、交通事故以及特殊事件等情况，分布式Agents网络能够不断优化路径规划和信号控制策略，实时调整交通流的疏导方式，从而有效应对复杂多变的交通状况，提供持续、高效、可靠的交通管理服务，缓解城市交通拥堵，提升交通系统运行效率。

分布式Agents网络的优化与调度研究不仅能够推动多智能体系统在复杂任务执行中的高效协作，提升系统性能与稳定性，还能够使其更好地适应动态变化的环境，满足不同领域日益增长的应用需求，为解决众多复杂系统问题提供有力支持，对于促进相关技术领域的发展和创新具有不可估量的价值。

# **分布式Agents网络优化技术研究现状**

## **研究现状（张永康）**

随着分布式系统在大规模数据处理、云计算、边缘计算等领域的广泛应用，分布式Agents网络的优化和调度成为提升系统性能、可靠性和一致性的关键。近年来，在优化和调度方法上取得了显著进展，提出了多种基于最优控制、多层网络、对偶近端梯度法、分布式强化学习、数学方法和人工智能的解决方案。这些方法不仅能够有效应对高并发场景下的资源需求波动和任务复杂性，还能在动态环境中实现高效的资源分配和任务调度。以下将详细阐述这些优化和调度方法的研究现状。

### 优化方法

#### 基于最优控制的分布式优化方法

动态资源分配算法：利用最优控制理论，根据各个节点的实时资源状态（如CPU使用率、内存占用率等），动态调整Agents的部署和资源分配[6]。例如，通过计算局部梯度和海森矩阵，结合状态误差，实现资源的高效分配。这种方法可以确保在资源有限的情况下，各个Agents能够根据其任务类型和优先级获得合理的资源配额，从而提高系统的整体性能。

资源监控与弹性扩展：结合最优控制策略，建立资源监控体系，实时监测系统资源使用情况。在资源不足时自动触发弹性扩展机制，增加节点或Agents实例；在负载降低时自动缩减资源[7]。这种机制可以有效应对高并发场景下的资源需求波动，确保系统在不同负载下的稳定运行。

#### 基于多层网络的分布式优化方法

多层网络优化算法：利用多层网络（如能量-气体互依网络）的框架，提出分布式鞍点算法及其变体——分布式梯度下降算法[8]。通过分析多层网络的扩散动态，获得多层超拉普拉斯矩阵，从而优化网络性能。这种方法特别适用于复杂的多层网络环境，能够有效协调不同层之间的资源分配和任务调度。

协调调度策略：针对互依基础设施网络，提出协调调度策略，将优化问题扩展到多层网络环境中，分析不同层之间的扩散对共识时间的影响[9]。这种策略可以确保在多层网络中，各个Agents能够高效协作，实现全局优化目标。

#### 基于对偶近端梯度法的分布式优化方法

集群网络优化算法：在多簇网络中，提出基于对偶近端梯度法的分布式优化算法。该算法通过优化全局成本函数，使每个簇中的Agents在最优策略上达成一致，并满足全局耦合约束[10]。这种方法特别适用于大规模分布式系统，能够有效降低通信成本，同时保证算法的收敛性。

通信优化：该算法允许Agents仅与其直接邻居通信，降低了通信成本，同时保证了算法的收敛性。这种优化策略可以显著提高系统的可扩展性和效率，特别适用于资源受限的分布式环境[11]。

### 调度方法

#### 基于分布式强化学习的调度方法

分布式强化学习技术：[12]采用如A3C（Asynchronous Advantage Actor-Critic）等分布式强化学习方法，解决云边缘计算网络优化问题。例如，通过A3C方法，Agents可以独立学习最优的计算卸载策略，以减少系统延迟。这种方法可以有效应对动态环境中的资源调度问题，提高系统的响应速度和效率。

Transformer增强的调度技术：[13]提出Transformer增强的分布式强化学习技术（TF-DDRL），用于在边缘和云计算环境中调度物联网应用。该技术基于IMPALA架构，提高了在高度分布式环境中的可扩展性。通过引入Transformer模型，可以更好地处理大规模数据和复杂任务，进一步提升调度性能。

#### 基于数学方法的调度方法

纳什均衡调度：结合博弈论和强化学习，提出基于纳什均衡的智能负载分配方法（NILD），用于地理分布式数据中心的负载管理。该方法通过非合作博弈论方法，在全球范围内实现最优负载平衡。这种方法可以有效解决多Agents环境中的资源竞争问题，确保系统的公平性和效率。

数学+启发式方法：采用数学优化与启发式方法相结合的方式，如利用遗传算法与模拟退火算法结合的时空任务调度方法（STTS），实现地理分布式数据中心的最优任务调度。这种方法可以有效处理复杂的调度问题，提高系统的整体性能和资源利用率。

#### 基于人工智能的调度方法

多智能体拍卖机制：[14]提出基于多智能体拍卖机制的地理感知任务分配框架（GMTA），用于优化地理分布式容器化云中的科学工作流执行。该框架通过智能工作流划分和基于Agents的协商，简化依赖关系，增强并行执行能力。这种方法可以有效提高任务调度的灵活性和效率，特别适用于复杂的科学计算任务。

深度强化学习调度：采用深度强化学习方法，如改进的深度Q网络（IDQN），使Agents能够从奖励函数中学习，并持续选择最优的绿色数据中心和服务器，以最大化奖励[15]。这种方法可以有效应对动态环境中的资源调度问题，提高系统的自适应能力和效率。

通过上述优化和调度方法，分布式Agents网络可以在高并发场景下实现高效的资源分配和任务调度，从而显著提升系统的性能、可靠性和一致性。

### 业界调研（黄子昂 刘帅）

#### Cisco DAWN

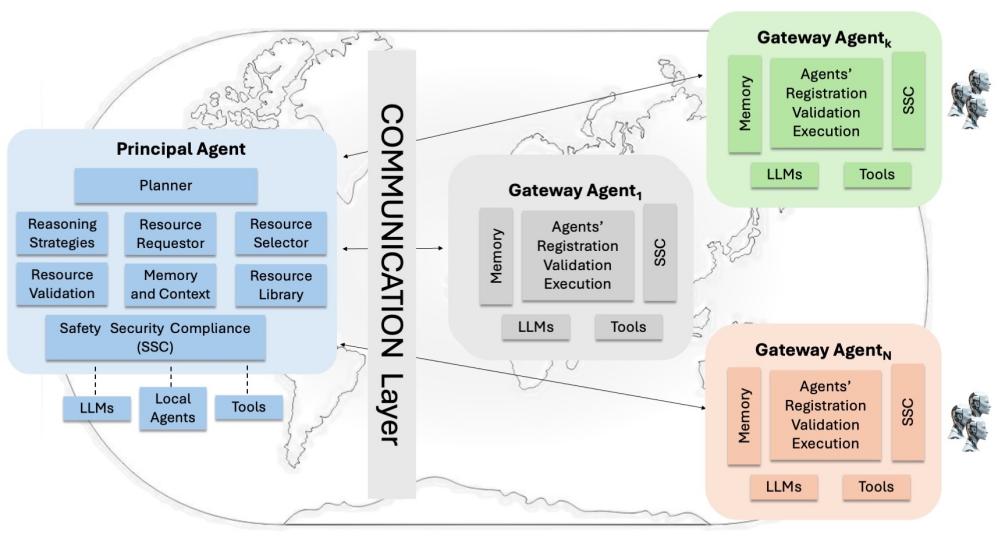


图2.1 代理之间的协作和通信

DAWN 平台旨在实现代理之间的协作和通信，其特点是具有模块化架构，包括主代理（Principal Agents）和网关代理（Gateway Agents）等关键组件。它将基于大语言模型（LLM）的代理集成到各种应用程序中，为代理合作提供了一个全面的策略。解决现有框架在管理大量代理和整合来自不同生态系统的异构代理方面存在的局限性。

No-LLM Mode模式最适合对可预测性和可靠性要求较高的任务，通过利用传统算法和工具来确保结果的一致性和可重复性。在这种模式下，人类操作员手动设计工作流程，不使用主管代理（Principal Agents），并且仅使用从网关代理（Gateway Agents）获取的非LLM工具，以最大化确定性。

Copilot Mode模式，代理充当智能助手的角色，通过创造性输入来增强确定性工作流程。人类操作员仍然手动设计工作流程，不使用主管代理。从网关代理获取的工具、代理以及代理应用程序将被用于中间步骤。

LLM Agents Mode模式，代理自主运行，基于复杂的推理实时做出决策。主代理自主定义工作流程，并从网关代理自主获取所需资源。这种模式特别适用于业务逻辑模糊且能够容忍随机结果的复杂任务。

DAWN实现了全球分布式代理之间的协作，利用它们的个体专业知识和集体智能来执行复杂任务。该框架支持各种任务，从确定性工作流到创造性和自主过程，使其适用于多种应用。

DAWN的设计确保了与不同工具和系统的无缝集成，促进了需要代理过程和传统软件系统混合的广泛应用。专门的安全、安全和合规层确保了代理操作遵守严格的安全协议，防止未授权访问，并确保符合特定行业的合规要求。

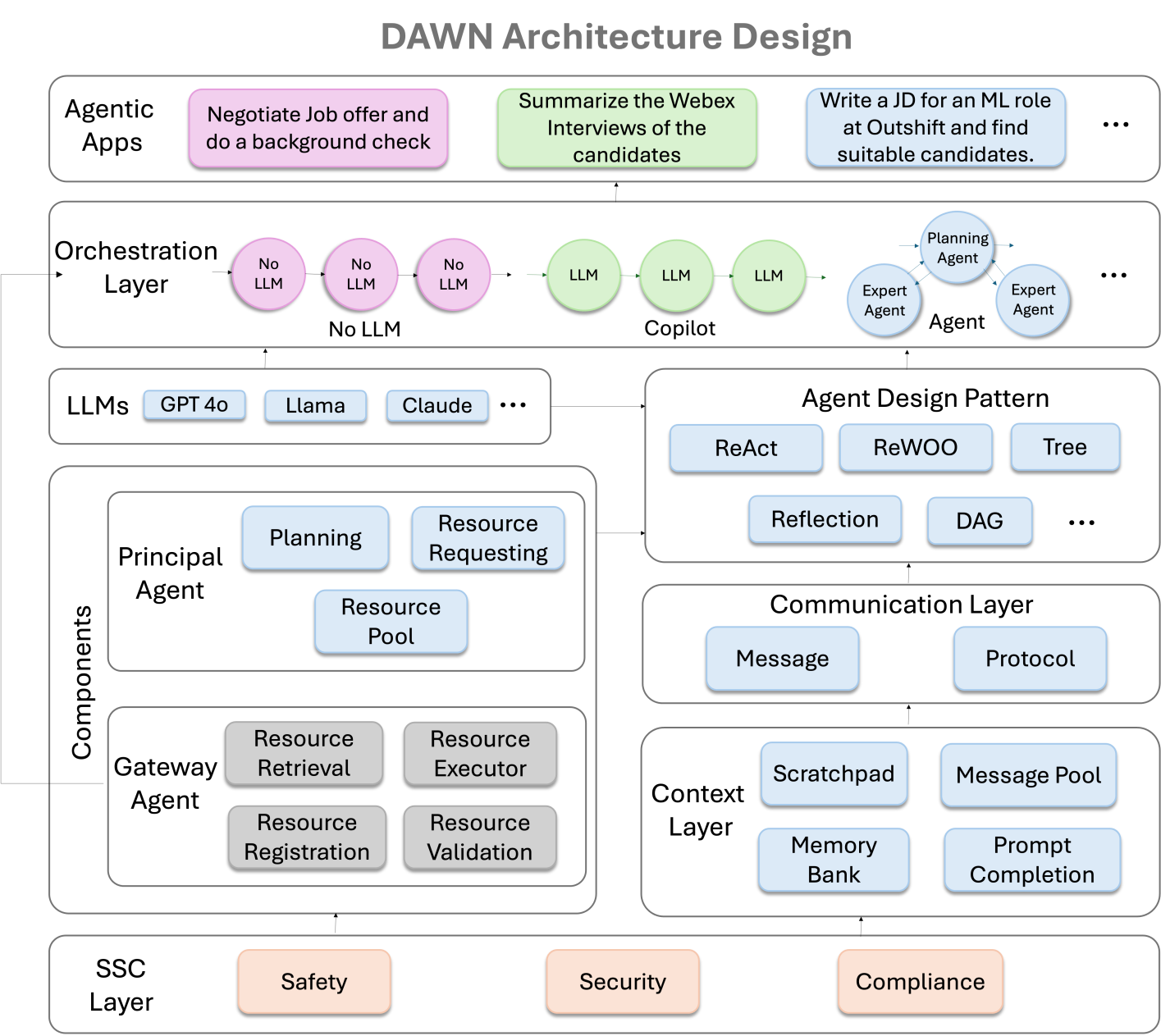


图2.2 DAWN 架构设计

#### 腾讯云原生分布式云中心

云原生分布式云中心（Tencent Kubernetes Engine Distributed Cloud Center，TDCC） 是腾讯面向多云多集群场景的应用管理平台，支持用户将云原生化的应用扩展到分布式云，全局视角统一管理和运维分布式云资源。

TDCC 分布式云中心打通公有云、私有云、边缘云的界限，将各种成熟的集群、网络、存储、微服务、运维等公有云产品和服务交付至更接近用户和数据的位置，确保不同云基础设施下拥有一致的控制平面，并且提供可靠性保证和安全合规保证，满足企业用户的多云管理、应用治理、高可用容灾等场景诉求。

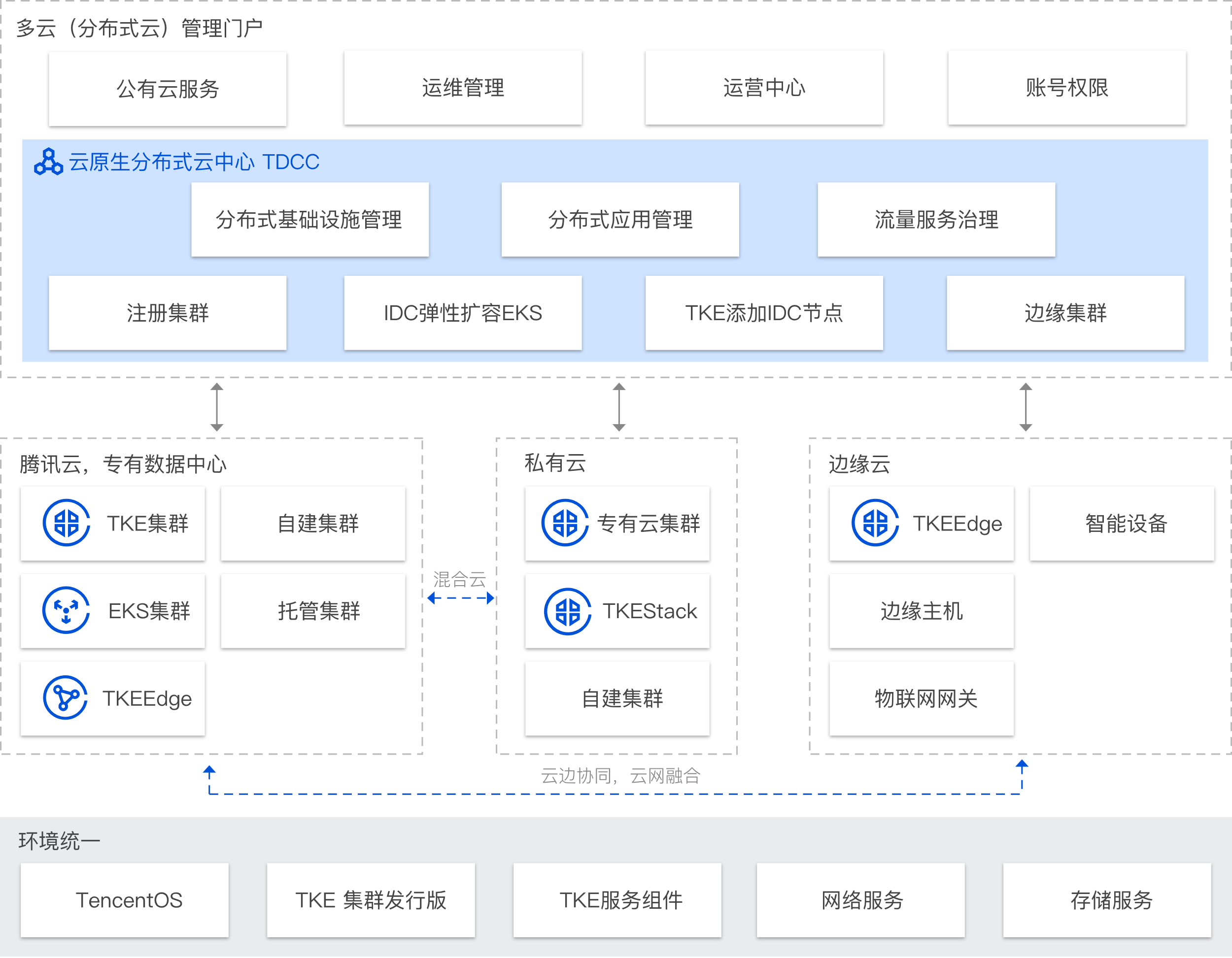


图2.3 分布式云中心 TDCC 架构图

无需用户自行搭建及运维，通过 Kubernetes 一站式接入和管理集群、服务器、主机、智能设备，不管是运行在公有云、私用云还是边缘云，都将拥有一致的控制平面，实现云网融合、云边协同。

以应用为中心整合应用镜像、流量、存储的资源，覆盖应用的交付、管理、调度、容灾、运维全生命周期，集合丰富的云服务和产品，助力业务应用扩展到分布式云，全局视角统一管理和运维，轻松地将应用服务发布至全球，一次部署处处运行。

全面兼容云原生标准，原生 Kubernetes 各类资源都可以通过 TDCC 分布式云中心进行分发和管理。基于 K8s 云原生标准的 DevOps CI/CD，AI/大数据计算，微服务以及物联网边缘计算等系统，都可通过 TDCC 分布式云中心实现强大的云上云下及多云混合资源、服务、应用、体验一致的统管能力，最小化多云运营成本。

通过注册集群能力，您可以将多个集群注册到公有云的管理平台上，从而实现对这些集群的统一管理和监控。公有云提供完备的日志管理、审计管理、事件管理、监控与告警服务，自动收集和分析集群的日志、审计记录、事件和指标数据，并提供可视化的监控和告警功能，帮助您快速发现和解决问题。解决集群运维依赖人工现场排障、多集群间缺乏统一的监控告警和日志等问题，提高运维效率和可靠性。

将成熟稳定的云服务通过云原生容器技术交付至更靠近用户和数据的位置，保证长期的平台运营、治理、更新和演进。

## **当前挑战（张永康）**

在高并发场景下，分布式Agents系统面临着诸多严峻挑战，包括资源调度、通信延迟、状态一致性、容错处理以及性能瓶颈等方面，这些问题相互交织，严重制约了系统的高效运行和稳定性。

### 资源调度挑战

在高并发环境中，Agents的数量会随着任务量的增加而动态变化，这种动态性导致资源需求出现较大波动。例如，在一个大规模的分布式计算任务中，当任务量突然增加时，系统需要快速分配更多的CPU和内存资源给新生成的Agents，而在任务量减少时，又需要及时回收闲置资源以避免浪费。然而，在有限的资源（如CPU、内存、网络带宽等）下，如何为各个Agents合理分配资源，避免资源竞争和浪费，是一个关键问题。例如，当大量Agents同时需要访问共享存储资源时，可能会导致存储节点的IO瓶颈，进而影响系统的性能。此外，不同类型的Agents对资源的需求各不相同，有些Agents可能需要大量的CPU计算能力，而有些Agents则更依赖于内存或网络带宽。如何根据Agents的任务类型和优先级进行资源调度，也是一个复杂的问题。例如，对于高优先级的实时任务，需要优先分配足够的资源以确保其快速完成，而对于低优先级的后台任务，则可以适当减少资源分配，以平衡系统的整体性能。

### 通信延迟挑战

在分布式Agents系统中，Agents之间需要通过网络进行通信协作，以完成复杂的任务。然而，在高并发场景下，网络流量会急剧增加，这很容易导致网络拥塞，从而增加通信延迟。例如，在一个大规模的分布式机器学习任务中，多个Agents需要频繁地交换模型参数和梯度信息，如果网络拥塞，这些信息的传输速度会显著下降，进而影响整个系统的训练效率。同时，Agents之间的通信可能涉及大量的数据传输，如状态信息、任务参数等，数据的序列化和反序列化过程也会带来额外的开销，进一步加剧通信延迟。例如，对于一些复杂的任务参数，其序列化和反序列化过程可能需要消耗大量的时间和计算资源。通信延迟不仅会影响Agents之间的协作效率，导致任务处理时间延长，还可能引发系统的不一致性问题。例如，当一个Agents发送的状态信息因延迟而无法及时到达另一个Agents时，可能会导致后者基于过时的状态信息进行决策，从而引发错误。

### 状态一致性挑战

由于Agents分布在不同的节点上，每个Agents可能维护着部分系统状态信息。在高并发场景下，多个Agents可能同时对共享状态进行操作，如更新、查询等，这很容易导致状态不一致的问题。例如，在一个分布式任务调度系统中，多个Agents可能同时调度同一个任务，或者对任务的状态信息（如任务进度、执行结果等）进行更新。如果没有有效的一致性保障机制，可能会导致系统出现错误。例如，一个Agents可能将任务标记为已完成，而另一个Agents却还在等待该任务的执行结果。此外，网络分区、节点故障等异常情况也会进一步加剧状态一致性的挑战。例如，当网络分区发生时，部分Agents可能无法及时获取其他Agents的状态更新信息，从而导致系统状态的分裂。在这种情况下，如何设计一种高效的一致性协议，以确保系统状态的一致性，是一个亟待解决的问题。

### 容错处理挑战

高并发场景下，系统中的节点和Agents面临着更高的故障风险，如硬件故障、软件崩溃、网络中断等。例如，在一个大规模的分布式系统中，某个节点的硬件故障可能会导致该节点上的所有Agents无法正常工作。如何快速检测到Agents或节点的故障，并进行有效的容错处理，确保系统的可靠性和可用性，是分布式Agents系统设计的重要问题。例如，当某个Agents节点发生故障时，需要将其承担的任务转移到其他正常节点上继续处理，同时保证任务的状态信息不丢失。这对系统的容错机制提出了很高的要求。例如，系统需要具备快速故障检测机制，能够在短时间内发现故障节点，并及时启动任务迁移和状态恢复流程。此外，还需要设计合理的备份机制，以确保在故障发生时能够快速恢复任务状态，避免数据丢失。

### 性能瓶颈挑战

随着并发量的增加，系统可能会出现性能瓶颈，如CPU利用率过高、内存泄漏、数据库连接池耗尽等。例如，Agents在处理复杂的业务逻辑时，可能会占用大量的CPU资源，导致节点负载过高。在这种情况下，如果系统没有及时进行资源调整，可能会导致节点崩溃，进而影响整个系统的稳定性。或者在频繁地读写数据库时，数据库连接池可能无法及时释放连接，导致后续的数据库操作阻塞，影响系统的整体性能。例如，在一个高并发的电商系统中，大量的Agents同时访问数据库进行订单查询和更新操作，如果数据库连接池的大小设置不合理，可能会导致连接池耗尽，进而影响用户的购物体验。此外，内存泄漏也是一个常见的问题，Agents在长时间运行过程中可能会不断占用内存，而没有及时释放，最终导致系统内存不足，影响系统的性能。因此，如何优化系统架构，合理配置资源，避免性能瓶颈的出现，是高并发场景下分布式Agents系统需要重点解决的问题。

## **当前解决方案（张永康）**

在高并发场景下，分布式Agents系统面临着诸多挑战，但通过一系列优化策略和技术手段，可以有效提升系统的性能、可靠性和一致性。以下是针对上述挑战的具体解决方案。

### 资源调度优化

动态资源分配算法：采用基于负载均衡的动态资源分配算法，根据各个节点的CPU使用率、内存占用率、网络带宽等实时资源状态，动态调整Agents的部署和资源分配。例如，可以使用贪心算法、遗传算法等，将任务分配到负载较轻的节点上，避免资源过度集中。同时，根据Agents的任务类型和优先级，为不同的Agents分配不同的资源配额，确保关键任务能够优先获得所需资源。例如，在一个大规模分布式计算任务中，高优先级的实时任务会被优先分配足够的CPU和内存资源，而低优先级的后台任务则会适当减少资源分配，以平衡系统的整体性能。

资源监控与弹性扩展：建立完善的资源监控体系，实时监测系统的资源使用情况。当检测到资源不足时，自动触发弹性扩展机制，增加新的节点或Agents实例，以应对高并发负载。相反，当负载降低时，自动缩减资源，降低系统成本。可以利用云计算平台提供的自动扩展功能，实现资源的动态调整。例如，在云原生环境中，通过Kubernetes的自动扩缩容功能，根据实时负载动态调整Pod的数量，确保系统在高并发时有足够的资源支持。

### 通信优化

高效的通信协议：选择适合高并发场景的通信协议，如基于TCP的长连接协议、UDP协议等。对于数据量大、实时性要求高的通信场景，可以采用二进制格式进行数据传输，减少数据的序列化和反序列化开销。例如，使用Protocol Buffers、Thrift等高效的序列化框架，提高数据传输效率。这些框架不仅支持多种编程语言，还能显著减少数据传输的延迟和带宽占用。

异步通信机制：引入异步通信机制，避免Agents在通信过程中阻塞等待。Agents发送请求后，可以继续处理其他任务，当收到响应时再进行相应的处理。通过异步通信，可以提高Agents的并发处理能力，减少通信延迟对系统性能的影响。同时，可以使用消息队列（如Kafka、RabbitMQ等）作为Agents之间的通信中介，实现解耦和异步通信，提高系统的可扩展性和容错性。例如，在一个分布式消息系统中，Agents通过Kafka队列异步发送和接收消息，即使某个Agents暂时不可用，消息也不会丢失，系统可以继续正常运行。

### 状态一致性保障

分布式共识算法：对于需要保证强一致性的共享状态，采用分布式共识算法（如Paxos、Raft等），确保多个Agents节点在状态更新时达成共识。这些算法能够在存在网络分区、节点故障等异常情况下，保证系统的一致性和可用性。例如，在分布式键值存储系统中，通过Raft算法实现多个副本之间的日志同步和状态一致性。即使某个节点出现故障，其他节点仍然能够正常工作，并且在故障节点恢复后，能够快速同步最新的状态信息。

版本控制与乐观锁：对于一些允许最终一致性的场景，可以采用版本控制和乐观锁机制。在更新共享状态时，检查状态的版本号，只有当版本号匹配时才进行更新操作，避免并发更新导致的数据冲突。例如，在分布式任务调度系统中，对任务的状态信息进行版本控制，当多个Agents同时更新任务状态时，只有最新版本的更新请求能够成功，其他请求需要重新获取最新状态后再进行操作。这种机制在减少锁的开销的同时，也保证了数据的一致性。

### 容错处理机制

冗余备份与故障恢复：对关键的Agents和数据进行冗余备份，确保在节点故障时能够快速恢复。可以采用主从备份、多副本备份等方式，将Agents的状态信息和任务数据复制到多个节点上。当主节点发生故障时，从节点能够自动切换为主节点，继续处理任务。同时，建立故障恢复机制，在检测到Agents故障后，能够重新启动Agents实例，并从备份中恢复任务状态，确保任务的连续性。例如，在一个分布式数据库系统中，通过多副本备份机制，确保数据的高可用性和容错能力。

心跳检测与故障转移：通过心跳检测机制实时监测Agents节点的状态，定期发送心跳包，若在规定时间内未收到响应，则判断该节点故障。一旦检测到故障节点，立即触发故障转移机制，将该节点上的任务转移到其他正常节点上执行。在任务转移过程中，需要确保任务的状态信息能够正确传递，避免数据丢失和不一致性问题。例如，在一个分布式计算集群中，通过Zookeeper等分布式协调服务，实现节点的心跳检测和故障转移，确保系统的高可用性。

### 性能优化

任务拆分与并行处理：将复杂的任务拆分成多个子任务，分配给不同的Agents并行处理，提高任务的处理效率。通过任务拆分，可以充分利用分布式系统的并行计算能力，减少单个Agents的处理压力。例如，在大规模数据处理任务中，可以将数据划分为多个分片，每个Agents处理一个分片的数据，最后将处理结果合并。这种分而治之的策略不仅提高了系统的整体性能，还增强了系统的可扩展性。

缓存机制与数据本地化：在Agents节点上引入缓存机制，对频繁访问的数据进行缓存，减少对后端存储系统的访问压力。同时，尽量将数据本地化存储，使Agents能够在本地节点上获取所需的数据，减少网络传输开销。

# **分布式Agents关键技术及应用场景（黄轩浩）**

分布式多智能体系统（MAS）作为分布式人工智能的重要分支，近年来在解决复杂计算问题中展现出显著优势。MAS由具备自主决策和学习能力的智能体组成，通过协同合作完成任务，其灵活性和适应性使其广泛应用于计算机网络、机器人、复杂系统建模、城市与环境建设及智能电网等领域。

在关键技术方面，MAS涉及智能体通信（如言语行为、消息传递和黑板机制）、开发平台（如基于BDI模型的JaCaMo）、控制技术（包括图论、博弈论和强化学习）以及学习机制（如集中式与分布式学习）。这些技术为智能体的协同、决策和适应动态环境提供了支持。

应用场景中，MAS在云计算中优化资源分配与负载均衡，在社交网络中预测用户行为，在网络安全中检测入侵行为，在机器人领域实现协作与路径规划，在智能电网中平衡能源供需，并在城市管理中优化交通与物流系统。这些应用展示了MAS在提升效率、可靠性和自适应性方面的潜力。

未来研究方向包括提升系统的可扩展性、鲁棒性和安全性，探索多智能体强化学习（MARL）与人类工作流的结合，以及解决跨学科集成中的文化和组织障碍。MAS的进一步发展将为复杂系统的智能化管理开辟新路径。

## **引言**

近年来，分布式人工智能(DAI)因其能够解决复杂的计算问题而受到了学术界的极大关注。基于用于解决任务的基本方法，DAI算法被分为三类，即并行人工智能、分布式问题解决(DPS)和多智能体系统(MAS)。并行人工智能涉及开发并行算法、语言和架构，以利用任务并行性来提高经典人工智能算法的效率。DPS涉及将任务划分为多个子任务，每个子任务都分配给一组协作节点中的一个节点，这些节点被称为计算实体。计算实体具有共享的知识或资源，并且与其他实体有预先定义的通信，这反过来限制了它们的灵活性。

多智能体系统(MAS)是本文的主要研究对象，它由被称为“智能体”的自主实体组成。与分布式处理系统中的计算实体类似，这些智能体协同完成任务，但它们具有更高的灵活性，这是因为它们具备学习和自主决策的能力。智能体通过与相邻智能体或环境的互动来学习新的情境和行动。随后，智能体利用其知识来做出决策并在环境中执行行动，以完成分配给它们的任务。这种灵活性使多智能体系统能够适用于解决包括计算机科学、土木工程和电气工程在内的众多学科中的问题。开发多智能体系统需要应对一系列复杂的挑战，例如智能体之间的协调、学习以及安全性。现有的相关综述文章对多智能体系统(MAS)的讨论较为局限，因此我们对多智能体系统的关键技术和应用场景进行了全面的概述。

## **关键技术**

### 智能体通信

智能体之间的通信已经被研究了50多年[16]。三种广泛使用的沟通方式包括：

言语行为：在研究言语行为交流的先驱John Austin[16]中，他发现一些言语动词或句子，被称为言语行为，会改变物理环境，例如，在适当的情况下，如果一个适当的人说“我现在让你成为夫妻”，那么这句话就会通过定义新的角色和条件来影响物理环境。智能体可以作为说话者，产生话语来改变听者的信念 [17]。智能体通过执行根据某些语言语法被感知为话语的主要动作来执行话语[18]。可参阅[19]以进一步阅读。

消息传递：在此方法中，智能体直接相互传递消息。智能体使用点对点通信或广播通信与其他智能体进行通信。在前一种情况下，智能体A知道agent B的地址，可以直接与agent B对话。在广播通信模型中，智能体A向其所有相邻节点发送消息。为确保消息的可理解性，参与通信的各个智能体必须采用一致的结构，这一点将在下文进一步详述。

黑板：在这种通信方式中，智能体可以通过一个名为“黑板”的中央存储库相互协作共享数据。每个智能体将其数据存储在黑板上，其他智能体可以读取这些数据。为了控制智能体的访问权限，黑板使用控制知识。每个智能体可以访问控制知识中定义的多个数据。

消息语义对于确保彼此交流的智能体对交换的数据有相同的理解至关重要。对于异构智能体而言，这尤其具有挑战性。一个简单的例子是，当智能体A将温度信息发送给智能体 B时，其值为12°C那么智能体 B不应将其理解为 12°F。智能体通信语言(ACL)旨在解决上述挑战。ACL为所有智能体提供了一种独特的消息格式和语义框架，以便它们能够进行通信并解读接收到的消息。[20] 将 ACL分为两个主要类别，即程序型和声明型。在程序型 ACL中，智能体之间的通信被建模为程序指令的共享。在声明型 ACL中，声明性语句用于指定定义、假设和断言。

“智能物理智能体基础”(FIPA)提出了一个适用于智能体的全面的ACL框架，该框架目前已被广泛应用在大多数多智能体系统的实例化中都得到了广泛应用。

### 主要平台

自适应多智能体网络是在主平台上开发的，它能够实现信息流、智能学习以及其中包含的智能体和元素的实时决策。这种用于对智能体的结构、行为和社会模型进行建模的抽象框架可以通过适当的智能体导向编程 (AOP) 来定义，AOP 是面向对象编程的一种特例[21]。这些平台最初受到了自适应组织模型的启发，后来在软件工程领域提出了更结构化的技术[22]。计算能力的迅速提高提供了更定制化的模型，如 O-MaSE，它利用了元模型的三个概念、方法片段和基于方法工程概念的指导原则[23]。这些平台因具体情况而异，要么针对特定的应用领域，例如微电网[24]，或者灵活的通用平台[25]。这些平台中的大多数是通过添加推理和认知模型而设计的，例如程序推理系统(PRS)和/或信念-愿望-意图(BDI)模型[27]。PRS有助于对过程进行推理，使智能体能够与动态环境进行交互，并使用程序来选择意图。这些程序在能够有助于实现某些目标时会被触发。在最常见的 BDI模型中，智能体的行为是根据其信念、目标和计划来定义的。在这个模型中，解释器负责根据从环境接收到的反馈以及管理智能体的意图/行动来更新这些特征。这些模型在将人工智能作为可插拔组件[28]或在元级计划[29]中集成方面表现出了极大的成功。面向多智能体的编程(MAOP)平台，如JaCaMo，采用了基于智能体、环境和组织维度这三种概念的结构化方法[30]。它们还成功地与物联网(ITS)相结合，在以人类为中心的环境中提供了自适应应用[31]。系统集成是另一种开发元模型以管理具有不同子系统的多智能体系统的有效方法[32]。在过去几十年里，提出了数十种此类方法及其旧版本的更新版本。为了从提出的众多 AOP 技术和平台中选择最佳选项，可以根据基本平台属性、可用性、可扩展性、稳定性、操作能力、安全管理和其在实践中的适用性等方面进行比较[26]。还需要根据其架构债务以及其对软件系统健康状况的长期影响对其进行研究和评估[33]。关于面向智能体的方法的评估框架在参考文献[34]中进行了回顾。尽管文献中提出了多种面向方面编程(AOP)方法，但它们在对模型成本、偏好、时间、资源以及持续性操作等方面的设计推理和决策模块方面仍存在诸多不足[35]。因此，开发人员仍然不愿转向这些平台，而是倾向于对现有编程语言进行少量修改后继续使用原有的代码[35]。

### 控制技术

在相关文献中，针对多智能体系统控制问题，采用了多种技术，包括图论、博论、控制论和机器学习等，这些技术被认为是用于此目的最常用的方法[36][37]。此外，还有其他算法，如优化算法和生物启发算法，也被应用于这些系统的集体行为研究[38]。

图论被用于确定多智能体系统的结构可控性[39]。在这个领域中，最常用的技术之一是利用图拉普拉斯矩阵来研究多智能体系统的动态和收敛速度。在文献中，还通过将边的状态定义为边的相对状态差的方式来使用边拉普拉斯矩阵来研究一致性问题[40]。图拉普拉斯谱具有第二小和最大的特征值，它们的比值在多智能体系统的控制中起着重要作用[41]。在最近的文献中，图论主要用于对网络动态进行初步分析，而其他补充方法则用于达到共识状态[42]。图论的应用并不局限于使用拉普拉斯矩阵，而在某些情况下，仅使用邻接矩阵和度矩阵来研究网络动态及其同步[43]。

博弈论是另一种用于对多智能体系统中理性协作智能体的动态行为和决策过程进行建模的流行技术[44]。在大多数分布式游戏中，主要目标是达到纳什均衡，并同时优化自身的性能指标[45]马尔可夫博弈或随机博弈是序列多智能体博弈的初始应用实例，它们可以通过动态规划(DP)Q学习或线性规划技术来解决。智能体的收益和奖励/效用中的不确定性也可以使用贝叶斯斯塔克勒格博弈来建模[46]。进化博弈是博弈论技术的另一种变体，用于对智能体的集体行为进行建模，其中有限理性会反复寻找平衡点[47]。大多数多智能体系统问题都是通过控制理论及其变体(如自适应控制)来解决的。例如，分布式模型预测控制被广泛应用于对不同类型的动态进行建模，其目标包括调节、跟踪或系统中的经济考量。关于多智能体系统中的在线学习控制机制的调查报告已发表。神经自适应最优控制被认为是控制复杂多智能体系统(MAS)的最流行技术之一，其原理是通过求解相关矩阵方程来实现，例如耦合的里卡蒂方程或耦合的哈密顿-雅可比方程[48]。在相关文献中，其他传统的优化技术，如变分不等式[49]、对偶理论[50]以及交替方向乘子法(ADMM)[51]，也被用于优化多智能体系统的操作。

强化学习是应用于自适应多智能体系统同时学习与控制的著名半监督学习技术之一[52]多智能体强化学习(MARL)技术通常用于基于部分信息(如从先前的行动和决策中获得的奖励和惩罚反馈)来感知环境[53]。基于强化学习开发的其他技术包括Q学习和策略梯度技术，它们试图学习智能体的最优策略。具有网络化智能体的MARL技术是MARL算法的一个特殊案例，在这种情况下，智能体可以在时变通信拓扑结构中与相邻智能体进行通信[55]。这些算法是为合作和非合作环境而开发的。在多智能体系统中使用强化学习非常具有挑战性，因为存在联合动作空间和由多个自主决策者生成的动态这使得环境非平稳且难以感知[56]。使用策略的集合是为设计对环境变化和非平稳动态具有鲁棒性的控制框架而提出的另一种技术[57]。[58]提出了 QMIX算法，该算法通过对各参与者的联合价值函数进行线性分解，同时确保局部和全局最大值函数在标准范围内保持单调递增/递减的特性Q-learning。在这些环境中，由于状态-行动对的数量不断增加，近似价值函数或最优策略变得非常困难。针对这一问题，主要的解决方案之一是应用深度强化学习(DRL)，它在强化学习迭代的学习过程中融入了深度神经网络[58]。这些技术的两个著名变体包括深度 Q学习[60]和深度确定性策略梯度(DDPG)，它们是基于带有重放缓冲区的演员-批评网络设计的[61]。像DDPG 这样的策略梯度技术在 MAS 中表现更好，因为其近似与系统动态的关联性无关。在 MAS 中使用基于强化学习的控制机制的其他挑战包括信用分配问题，这反映了缺乏追踪智能体的行动及其对系统结果的影响[62]。这个问题可能导致懒惰的不活跃智能体不愿意为学习系统动态做出贡献。为解决这一挑战而提出的方案之一是使用反事实基线的多智能体策略梯度(COMA)方法，该方法会固定其他智能体的动作同时对单个智能体的动作进行边际化处理[63]。

### 学习机制

在自适应多智能体网络中，主要的框架之一就是其学习机制。大多数最新的学习框架都是在线式的，能够帮助系统了解其环境的动态以及对这些变化的最佳应对策略。针对多智能体系统学习文献，人们研究了诸如知识获取级别和学习技术等各个方面[64]。智能体可能拥有完全的自主学习能力，并能与其他智能体共享其知识，或者可能仅限于仅与中央学习者进行交流和共享其状态[65]。如果从系统层面进行学习，那么一个智能体或主要管理者会学习整个系统中所有智能体的策略。在这种情况下，学习者能够完全观察到涉及的智能体的状态，而无需详细关注单个智能体的行动。在这种学习机制中，信息从分布式智能体收集，并传递给系统的中央学习者。这有助于获得有关系统动态的高级信息，而不会陷入协调多个学习者之间信息流的困难之中。集中式学习和培训可以与中央决策者相结合，也可以与分散式激励相结合[66]。在第二种类型中，集中式学习者使用引导分布式执行者的准则来学习价值函数[67]。集中式学习机制存在一些问题，例如状态过程和学习过程的复杂性。它们也是基于一些不切实际的假设而开发的，比如能够持续且完整地获取所有执行者的信息。集中式学习的另一个挑战是其对学习者的故障的脆弱性，需要中央学习者具备大量的计算和内存资源，并且对于拥有数千个分布式执行者的大型系统具有可扩展性[68]。通过这种学习方式，可能会导致所有执行者都采用相同的策略的同质团队学习，或者每个个体执行者都有独特行为的异质学习[69]。为缓解上述列出的挑战，该平台的其他变体包括 QMIX的全局和局部的混合成分[70]。通过最大熵 RL技术和策略提炼进行协调采样是为改进集中式学习机制而提出的其他解决方案[71]。还有一些混合技术，即智能体各自独立学习，然后共享一个集中式的共同知识记忆来整理和存储它们的知识[72]。在分布式学习技术中，每个智能体负责自己的学习[73]。在这些学习框架中智能体的可观测性有限，只能探索其周围环境，且无法实时学习系统的整体动态大多数这些学习技术都与控制机制的决策过程相结合。基于角色的学习技术是文献中的另一个趋势，其中复杂的任务被分解为不同的角色。RODE技术就是这种学习的一个例子，它利用智能体聚类来发现角色和所需的学习组。基于特定事件或系统运行过程中的离散或连续时间更新，可以启动学习和更新已有的模型。文献中应用的一些常见学习技术包括强化学习、监督学习、深度学习、博弈论、概率论、群体系统、应用逻辑、进化算法或它们的组合等[74]。由于这些技术大多用于同时对多智能体系统进行学习和控制，因此将在下一节中对其进行回顾。一些提出的用于学习的技术是在获得初始领域知识后启动的。在文献中，这些技术被定义为迁移学习方法[64]。

## **应用场景**

分布式多智能体系统（MAS）是一项变革性技术，广泛应用于各个领域，以提高复杂环境中的效率、决策能力和适应性。通过使多个自主智能体能够协作和通信，MAS已应用于机器人、电信、医疗保健、金融、交通运输和环境监测等领域。这些系统的多功能性和可扩展性使其在应对复杂挑战、提高运营绩效和推动智慧城市创新方面尤为突出。

在机器人领域，MAS促进了重大进展，特别是在自动驾驶汽车和工业自动化方面。自动驾驶汽车利用MAS与其他车辆和基础设施进行实时通信，从而提高安全性和效率。在制造业中，协调的机械臂通过适应装配线上的意外问题，提高了生产率和可靠性。此外，MAS在搜救行动中也发挥了重要作用，可以部署无人机和地面机器人高效导航危险环境。

MAS的影响还延伸到电信领域，优化了智能电网管理和交通系统。通过自主智能体，MAS增强了数据集成、资源管理和多种技术的互操作性，从而使电信网络更加响应迅速和高效。在智能电网应用中，MAS改善了能源管理和可靠性，允许实时调整并促进电力资源的分散控制。

尽管潜力巨大，分布式多智能体系统的部署并非没有挑战。必须解决围绕可扩展性、鲁棒性和安全性的问题，以确保其在金融和医疗保健等敏感应用中的有效性。在金融领域，MAS增强了欺诈检测和市场分析，但也引发了关于风险管理和问责制的伦理考虑。随着该领域研究和开发的进展，跨学科方法的持续探索和先进技术的集成对于最大化MAS在各行业中的效益至关重要。

### 计算机网络

由于新技术的出现以及联网设备的大量增加，计算机网络的复杂性显著提高。智能体技术被广泛应用于克服这困难。由于多智能体系统在网络中的应用范围广泛我们进一步将其分为四个子类别：

#### 云计算

云计算能够实现对可配置系统资源(如CPU、GPU和内存)以及计算服务(如服务器、数据库、网络和软件)的普遍访问，这些资源通常通过互联网提供。云计算利用虚拟化作为底层技术为用户提供服务。通过虚拟化，一台物理机被多个客户共享为多个虚拟机(VM)，每个虚拟机都模拟一台独立的机器。与传统的每个用户都拥有自己专用资源的方式相比，云计算具有多项优势:

降低货币成本:云用户租用他们所需的各种资源，这些资源由云服务提供商进行管理。这样就无需再自行购买和维护资源，从而降低了货币成本。

可靠性:每个资源都有多个重复备份，这使得云计算能够抵御资源故障，从而提高了其可靠性。如果出现例如 CPU 的故障，其相关任务可以被转移到其他类似的资源上。阐述了云计算中涉及的几个关键概念:管理云资源、进行通信以及核算每个用户对资源和/或服务的使用情况[75]、[76]。

在[77]中，提出了一种基于智能体的框架，用于在云端执行“任务包(BoT)，即一系列独立的任务。一组智能体收集有关每个特定任务的可用资源和云提供商的信息。智能体将每个客户与合适的云提供商进行匹配。然后，客户和云提供商进行协商，就提供给客户的资源以及应支付给云提供商的价格达成协议。通过模拟以及从终端用户的角度考虑的成本，研究了任务的分配、释放和执行时间。仿真结果表明，该智能体系统在任务分配方面比亚马逊 EC2 云资源分配方法更成功。

在任务分配过程中，必须考虑每个资源的负载情况，因为负载过重会导致服务接收出现长时间延迟。[78]提出了一种基于智能体的负载均衡框架，以平衡虚拟机的负载。云提供商定义了一项策略来控制虚拟机的负载。所提出的方法得益于三个智能体，即:负载智能体、迁移智能体和通道智能体。负载智能体通过监测虚拟机的负载来确保其负载与所定义的策略相匹配。为了监测虚拟机的负载，负载智能体会请求通道智能体启动迁移智能体。迁移智能体是移动于网络和收集关于资源信息的移动智能体。这些信息会报告给通道智能体，该智能体负责控制传输策略和选择策略。然后，通道智能体将收到的信息转发给负载智能体。基于收到的信息，负载智能体通过将新任务分配给负载较低的虚拟机来平衡虚拟机之间的负载。

多智能体系统被应用于云环境中的资源监控[77]、安全[79]资源发现[80]以及自动服务管理[81]等方面。有关多智能体系统在云环境中的应用的全面综述可参见[82]。

#### 社交网络

随着互联网用户的不断增加，社交网络的普及程度呈指数级增长社交网络由参与者(如用户、群组和服务)组成[83]。社交网络的复杂性源于其动态性，即大量参与者加入或退出网络，或与其他参与者建立新的联系，以及其广泛的用途和服务。多智能体系统可能是解决社交网络复杂性问题的潜在方案。

[84]提出了一种基于智能体的方法来预测社交网络(如推特)中的用户行为，例如点赞、发布内容和关注等。作者提议使用多个智能体，即参与者，它们分布在社交网络中，以收集用户行为的数据集。然后，这些智能体会对每个特定用户的数据进行主题和情感分类，这些分类结果随后用于构建用户档案。最后，将用户档案输入到预测系统中，该系统能够预测用户的未来行为，包括点赞、主题、回复、发布内容和分享等。

社交网络不一定非得基于网络形式。只要人类出于特定原因聚集在一起进行互动并分享信息，任何这样的场所都可以被视为一个社交网络。[85]将学校视为一个社交网络，并应用多智能体系统来帮助学校管理者了解向不同学校项目(如体育、学术和文化活动)分配资金与学校表现之间的关系。学生、教师和学校部门被视作智能体，它们共同构成了这个社交网络。例如，就是这所学校。每个学生与其他学生或教师之间的互动、过去几年的资金分配政策以及学生在当前和过去几年的表现都被组织成一个层级结构。然后，这个结构会被输入到一个学习功能中，该功能会评估资金分配政策与学生学业成绩之间的关系。

#### 安全

自 2002 年以来，关于多智能体系统在网络中的安全应用的研究已展开[86]。多智能体系统是解决网络安全问题的有效方案，因为它们能够主动了解并从而检测出新的安全威胁。

在文献[87]中提出了一种基于自主智能体的入侵检测系统(IDS)。所提出的该入侵检测系统由五种智能体组成，分别是集合智能体、检测智能体、决策智能体、响应智能体和协作智能体。集合智能体从网络中收集简单网络管理协议SNMP)和路由表的内容，并将这些数据发送给检测智能体后者使用误用和异常检测引擎来检测异常的数据包或通信。检测引擎的结果被输入到决策智能体中，该智能体会判断网络中是否发生了恶意活动，如果存在，则决定采取何种适当措施来减轻其影响。然后，所决定的行动会传递给协作智能体，该智能体是一种移动智能体，它将决策传达给响应智能体。响应智能体在网络中执行相应的操作。

基于智能体的入侵检测系统是文献中多智能体系统在网络安全领域的主要应用。然而，多智能体系统也用于其他安全应用。[88]提出了一种基于智能体的安全方法，以保护用户免受劫持攻击。在该攻击中，攻击者在目标用户网络浏览器的闲置标签页上打开随机网站。两个智能体监控打开的标签页，并收集五元组元素，即文本、图像、URL、标题和网站图标，即网页标题图标然后，智能体将运行在打开标签页上的操作的指纹与现有攻击签名进行比较，以检测劫持攻击。使用 JADE模拟器进行了模拟(见第七节)。模拟结果表明，使用智能体，误报率和漏报率低于非智能体方法，攻击检测的准确率达到了 91%。

未来，MAS 有可能被用作解决诸如信任和密钥管理等复杂安全任务的解决方案。此外， MAS 可用于克服物联网(IoT)的安全挑战。[89]中提出了一项关于基于智能体的入侵检测系统的综合研究。

#### 路由

路由指的是根据特定指标(例如源节点与目的地之间的跳数)为数据包从源节点向目的地寻找路径的过程。自1998年以来，使用智能体进行路由是多智能体系统研究中的首批应用之一[90]。此后，随着路由协议面临的新挑战不断出现，多智能体系统也不断发展。

[91]提出了一种针对车辆自组织网络(VANET)的去中心化智能体式路由协议，该协议由三组智能体组成。第一组智能体由车辆承载，用于监测车辆参数，例如速度、位置等。第二组智能体由路边基础设施承载，监测车辆之间的通信以及车辆与路边基础设施之间的通信。第三组智能体通过模拟车辆、路边基础设施及其通信来创建一个虚拟环境，形成一个图。智能体通过应用蚁群算法在图中寻找通往目的地的短路径，从而通过向不同车辆发送不同的路径来避免该区域的拥堵。实验结果表明，基于智能体的这种方法的行程时间比其他已研究的路由方法要短。

路由中的一个关键挑战在于所选路径的可靠性，即要确保该路径不会频繁出现故障。[92] 为无线传感器网络提出了一种可靠的路由协议。该方法得益于智能体在转发数据包和汇总传感器数据方面的作用。首先，汇聚节点(即无线传感器网络中的中央控制器)广播一个数据包，以查找节点的可用资源，用于执行特定任务，这些资源用于划分网络。然后，汇聚节点建立移动智能体，以收集和汇总多个区域内的传感器数据。通过跟踪每个移动智能体访问的传感器，汇聚节点根据每个区域中传感器到汇聚节点的距离，为每个传感器生成一个排序的传感器列表，作为通往每个传感器的可靠路径。仿真结果表明，与传统路由协议相比，基于智能体的路由协议的能耗更低，数据包丢失率也更低。

为了进一步提高可靠性，[93] 提出了一种可靠的多播路由协议，该协议由两组智能体组成。第一组是移动智能体，它们在网络中移动以收集节点的能量、带宽、移动性和内存状态。这些信息用于为每个节点计算可靠性因子(RF)。然后，该因子传递给第二组智能体，后者利用具有最高 RF的节点建立多播骨干网。仿真结果证明，基于智能体的方法比不使用智能体的方法具有更高的数据包传输率和可靠性。

### 机器人

关于机器人领域的智能代理的研究已经持续了近二十年，首篇相关文章于 1996 年发表，文中阐述了智能代理在机器人技术中的利弊 [94]。

[95] 认为，在机器人技术领域存在两个主要挑战:(1)机器人之间的协作与协调，以及(2)规划其移动轨迹。随后，作者提出了一种利用硬件和软件代理来克服上述挑战的方法硬件代理指的是构成机器人的物理硬件，而软件代理则是决策、路径规划、任务管理和通信代理。一个硬件代理利用其传感器(例如摄像头)来捕捉环境中的图像。然后，一个通信代理将这些图像发送给图像处理代理。后者对图像进行处理，以确定机器人的位置和环境中的障碍物。然后，这些信息被发送给决策制定代理，该代理会找到一条障碍物最少的路径以到达目的地。实施结果表明，所提出的方法能够检测到障碍物，并找到一条优化的路径(没有任何障碍物)以到达目的地。

机器人可能会被部署在非确定性的动态环境中，这增加了其决策过程的复杂性。为了研究这种复杂性，[96] 提出了一种基于代理的足球机器人。代理(即球员)被分组组成团队。团队中的代理通过与环境的互动来学习有关对手团队和可能行动的知识，然后它们会将所学策略与团队中的其他代理共享。强化学习与概率神经网络(PNN)结合使用，以提高代理所做最终决策的准确性。实施结果表明，与非代理团队相比，代理团队中的代理能够预测出正确的行动，从而提高了球的控制率(这是足球中的一项重要性能指标)

在[97]-[99]中讨论了智能体在机器人技术中的更多应用。

### 复杂系统建模

对复杂动态系统的建模成本高昂，并且会带来大量的处理开销这是因为需要强大的建模平台以及较高的复杂性。而代理所提供的灵活性、自主性和可扩展性使得基于代理的建模(ABM)成为一种低成本且资源消耗少的复杂系统建模解决方案。ABM 采用基于规则的方法来对环境进行建，这与使用方程的其他建模方法不同。ABM 的最重要优势包括[100]:i)能够整合多种建模方法，ii)在建模大规模自治系统时具有灵活性，ii)对预定义知识的灵活性，因为代理可以通过从环境中学习来获取知识，iv)可以并行执行从而加快建模过程，以及v)能够探索由于代理的主动性而产生的涌现行为。

[101]提出了一种代理模型来模拟供应链。供应链中的每个实体都单独进行建模，拥有自己的政策，并能够定义与其他实体的互动方式。所提出的方法包含两组代理，即:规划代理和物理代理。客户和供应商使用规划组中的六种代理来协商并就产品的价格达成协议。规划组中的代理如下:i)需求满足代理负责管理客户需求;ii)物料资源规划代理，与生产商沟通并采购产品;iii)需求预测代理，根据当前和历史需求情况预测客户需求;iv)主计划代理，汇总生产计划;v)生产计划代理，由主计划代理汇总的计划进行分解;vi)调度代理，对多个代理中的任务进行调度。物理组中的三个代理执行物理任务，即接收和存储原材料制造以及向客户交付产品。

[102] 中提出了一种 ABM(代理行为模型)，用于研究城市配送中心(UDC)的使用情况。在智能城市中，该配送中心通过优化和调度用于货物运输的路线，来集中管理产品配送，从而控制城市拥堵污染、配送时间和可靠性。

在该方法中，移动自动化系统由卡车、货物、客户、用户数据中心以及城市车辆组成。停车场中未获许可或超时停放的车辆会导致卡车的停车问题，从而造成货物配送的延误。用户数据中心的代理在设计配送路径时会考虑到这一挑战，以进一步缩短配送时间。与传统管理系统相比，该方法为店铺业主最大限度地降低了配送成本为配送公司最大限度地提高了配送利润。

基于[103]中所提出的观点，现有的电力行业建环境无法有效地对复杂的可扩展电力网络进行建模，例如拥有大量不同能源来源和用户的智能电网。代理模型是一种适用于智能电网的有效分布式建模方法，这使得代理模型成为该领域的一个有趣的研究方向[104]。参考[99]和[104]以获取关于代理模型的更详细讨论。

### 城市与环境建设

近年来，利用代理来管理城市和建筑物的研究受到了研究人员的极大关注。在城市中，货物的无序分配会增加成本、污染和拥堵。[105]提出了一种基于代理的方法来解决这一挑战，该方法使用了六个代理，分别是:RFIDG、零售商、供应商、运输商、网络和城市代理。RFIDG代理使用 RFID标签来管理资源供应客户向零售商或供应商代理发送购买商品的请求。收到请求后，零售商或供应商代理会搜索其数据库以找到所请求的商品。接下来货物会被发送给运输商代理，以便将其送达客户手中。网络代理确定减少城市拥堵的最优路径，以便运输商代理能够将货物送达客户。城市管理员代理会将交易的相关政策和规则(例如商业和货物运输)告知双方，例如供应商和客户。

城市面临的另一个重大挑战是控制和管理不断扩张的都市区的交通系统和交通状况。[106]提出了一种基于代理的方法来解决这一难题。该方法考虑了诸如票价和乘客满意度等参数，用于定义和分析交通模型。他们使用了两组代理:旅客和车辆。这些组中的代理共享关于拥堵和交通状况的信息，其他代理会利用这些信息来决定一条低拥堵的通往目的地的短路径。

MAS 还被用于管理建筑物。在[107]中提出了一种基于代理的建筑物供暖管理方法。代理的主动性和灵活性使它们成为管理由建筑物周围分布的异构供暖设备和传感器组成的供暖系统的有效解决方案。一个需求代理会检查建筑物内的温度，并将数据传递给燃气加热器、缓冲器和热泵代理。后者的代理会使用从需求代理接收到的数据以调节建筑物的温度。作者们在一套公寓中实施了所提出的方案，并将其与集中式方法进行了对比，作为基准测试。实施结果表明，采用代理方法相比集中式方法，每日的电力消耗显著降低。

关于该领域的进一步阅读内容可参见文献[107]和[108]。

### 智能电网

在相关文献中，智能电网中的“代理”被用于应对一系列挑战包括平衡发电量与用电量、在能源消费者与生产者之间就能源价格进行协商、在家庭储能设备中储存能源以及进行能源恢复。

[109] 提出了一个基于代理的服务模式。例如，为智能电网提供能源和恢复功能，以支持分布式能源存储。该系统使用两个代理，即:切换代理和分布式能源存储代理。切换代理平衡能源负载，并检测并隔离故障，即由于某些能源生产者或能源分配系统中的故障而导致的停电。能源存储代理根据电网是否连接到智能电网或是否处于隔离状态，为电网提供能源。所提出的方法通过有效的恢复来降低系统损耗，并实现动态孤岛化，以恢复正在断开连接(孤岛化)的电网部分。

智能电网的参与者要么是能源生产商，要么是能源消费者。

能源生产商旨在通过以更高的价格出售能源来增加利润。为了实现这一目标，[110] 提出了一种储能管理方法。每个能源生产商(代理)都拥有一个储能设备，并旨在通过分析包括客户和其他能源生产商在内的其他代理所生成的价格信号来增加自身利润。每个代理都会将储能使用情况记录在一个独特的存储方案中。该代理利用博弈论方法对存储方案进行分析，以预测未来的使用情况，从而决定是出售还是储存其产生的能源。

参考文献[111]以进一步了解智能电网中多智能体系统的应用情况。

## **未来方向**

尽管分布式多智能体系统（MAS）取得了显著进展，但其在各行业中的集成和应用仍面临各种挑战。最突出的挑战之一是这些系统的可扩展性。随着业务的增长和任务复杂性的增加，保持高效性能变得至关重要。随着更多智能体加入系统，潜在交互的指数增长需要有效的资源分配，以防止瓶颈并确保平稳运行[25][10]。此外，可扩展性不仅需要在性能方面进行评估，还需要在不增加资源的情况下保持运营效率[25]。

另一个紧迫的问题是MAS的鲁棒性和弹性。虽然这些系统提供了诸如容错性等优势——即一个智能体的故障不会破坏整个系统的性能——但在动态环境中设计支持自组织和适应性的架构仍然是一个挑战[10-1][26]。确保智能体之间的安全通信也至关重要，特别是在金融系统和军事行动等敏感应用中，需要强大的加密和认证机制来保护信息交换的完整性和机密性[27][10]。

未来分布式MAS的研究方向应侧重于通过跨学科方法解决这些开放性问题。需要不仅促进技术采用，还能管理组织变革中以人为本的框架。正如van Der Aalst等学者所强调的，理解文化和组织障碍对于在抵制此类创新的行业中集成AI驱动的变革至关重要。

此外，将多智能体强化学习（MARL）与结构化人类工作流相结合，提供了另一个值得探索的方向。当前的研究往往忽视了这种集成，留下了一个必须弥合的空白，以实现多智能体系统（MAS）在现实世界应用中的全部潜力[9]。随着城市交通格局的演变，技术创新与公平、环保的城市生活之间的协同作用将变得越来越重要，这将使MAS成为塑造未来城市环境的关键参与者。

面向大小脑协同优化、集中+分布式决策对网络QoS的需求(齐立鹏)

随着人工智能与机器人系统的深度融合，具身智能体在工业、安防、救援等任务中的部署日益频繁。近年来，以“大脑-小脑协同架构”为代表的新型控制范式被提出，结合集中式全局智能与分布式局部控制，有望提升复杂任务环境下的任务适应性与系统鲁棒性。然而，这类系统的运行依赖于高效可靠的网络通信资源，尤其是在多智能体、大规模分布式部署背景下，对QoS（服务质量）提出了更高的要求。本文综述相关领域代表性研究，围绕“大脑-小脑协同优化”架构与“集中+分布式决策”的任务规划机制，分析其对网络带宽、时延、抖动、丢包率等QoS指标的综合需求，评估已有优化手段，归纳当前技术瓶颈与未来趋势。

## **集中式与分布式架构**

### 集中式架构：全局控制与高可靠性通信的需求

集中式架构在网络系统中以其统一的控制和管理能力，广泛应用于数据中心、传统企业网络以及早期的云计算平台中。在这种架构中，所有的数据处理和决策制定集中在一个或少数几个中心节点上，对网络QoS提出了特定的要求。

首先，集中式架构对网络的带宽和低延迟传输提出了高要求。由于所有的数据流都需要通过中心节点进行处理，网络必须提供足够的带宽以避免瓶颈，同时确保数据传输的实时性。例如，在视频点播（VOD）和网络电视（IPTV）等应用中，用户对视频质量和加载速度有较高期望，网络需要提供高吞吐量和低延迟以满足用户体验。

其次，集中式架构对网络的可靠性和稳定性有较高的依赖。中心节点的故障可能导致整个系统的瘫痪，因此网络需要具备高可用性和容错能力，以确保系统的连续运行。

然而，集中式架构也存在一些限制，尤其是在面对大规模用户访问或地理分布广泛的场景时，可能出现性能瓶颈和单点故障的问题。因此，虽然集中式架构在某些应用中仍然有效，但其在现代网络环境中的适用性正在受到挑战。

### 分布式架构：弹性扩展与端到端QoS保障的挑战

随着物联网（IoT）、边缘计算和大规模分布式系统的发展，分布式架构逐渐成为主流。该架构通过在多个节点之间分散数据处理和决策制定，提升了系统的弹性和可扩展性。然而，这种架构对网络QoS提出了新的挑战。

在分布式系统中，数据需要在多个节点之间进行频繁的交换和同步，这对网络的延迟、带宽和可靠性提出了更高的要求。例如，在远程音视频传输和实时数据处理等应用中，系统需要确保端到端的数据传输具有低延迟和高可靠性，以保证服务质量。

此外，分布式架构中的节点可能地理位置分散，网络条件各异，如何在不稳定的网络环境中保障QoS成为一大难题。系统需要具备动态调整和自适应的能力，以应对网络状态的变化。

尽管面临挑战，分布式架构在提高系统的容错性和处理能力方面具有显著优势，特别适用于需要高可用性和灵活性的应用场景。

### 混合架构：集中与分布的协同优化与QoS适配

为结合集中式和分布式架构的优势，混合架构应运而生。该架构在保留集中式架构统一管理和控制能力的同时，利用分布式架构的弹性和扩展性，以实现更高效的系统性能和更灵活的资源调度。

混合架构对网络QoS的要求更加复杂，需要在集中控制和分布处理之间实现平衡。例如，在多接入边缘计算（MEC）和软件定义广域网（SD-WAN）等应用中，系统需要根据实时的网络状态和应用需求，动态调整资源分配和数据路径，以优化服务质量。

此外，混合架构还需要支持多种QoS策略的协同工作，以满足不同应用的需求。例如，在同时运行视频会议和大数据分析任务的系统中，需要分别保障低延迟和高吞吐量的QoS要求。

混合架构的灵活性使其在现代网络环境中具有广泛的应用前景，尤其适用于需要高性能和高可靠性的复杂系统。然而，实现混合架构的有效协同和QoS保障仍需面对技术和管理上的挑战。

综上所述，集中式、分布式和混合架构各自对网络QoS提出了不同的需求和挑战。理解这些架构的特点和适用场景，有助于在设计和部署网络系统时，选择最合适的架构和QoS策略，以满足应用的性能和可靠性要求。

## **面向分布式深度学习训练的QoS优化技术**

### 从集中式到分布式

随着深度神经网络（DNN）模型规模的不断扩大和训练数据量的激增，传统的集中式训练方式已难以满足高效训练的需求。集中式训练通常依赖单一计算节点，受限于计算资源和内存容量，难以处理大规模模型和数据集。为此，分布式深度学习训练应运而生，通过将模型和数据分布到多个计算节点上，实现并行计算，加速训练过程。

在分布式训练中，常见的架构包括参数服务器（Parameter Server）和全同步（All-Reduce）架构。参数服务器架构将模型参数存储在中心服务器上，工作节点负责计算梯度并将其发送到服务器进行更新。全同步架构则通过节点间的直接通信，实现参数的同步更新。这些架构虽然提高了训练效率，但也引入了新的挑战，特别是在网络通信方面。

例如，[112]提出了一种分层参数同步机制，通过在不同层级间进行参数同步，减少了通信开销，提高了训练效率。然而，随着节点数量的增加，网络带宽、延迟和可靠性等QoS（Quality of Service）指标对训练性能的影响日益显著。

### 分布式训练对网络QoS的关键需求

分布式深度学习训练对网络QoS提出了更高的要求，主要体现在以下几个方面：

带宽敏感性：在每轮训练中，节点间需要交换大量的模型参数和梯度信息。以一个1GB的模型为例，进行1000轮训练，数据传输量可达TB级。因此，高带宽是确保训练效率的关键。

低延迟和低抖动：参数同步过程对延迟和抖动非常敏感。高延迟或不稳定的网络会导致节点等待，产生“掉队者效应”，降低并行效率。

高可靠性：在同步过程中，数据丢包会破坏梯度的正确性，影响模型精度，甚至导致训练无法收敛。因此，网络需要具备高可靠性，确保数据的完整传输。

为应对上述挑战，研究者提出了多种优化策略。例如，TicTac系统[113]通过优化通信调度，减少了参数传输的随机性，提高了训练吞吐量，并显著降低了“掉队者效应” 。

### 网络结构与拓扑优化方法

网络结构和拓扑对分布式训练的通信效率具有重要影响。优化网络拓扑可以减少通信延迟，提高带宽利用率，从而提升训练性能。

[114]提出了一种合并梯度的通信策略，通过将多个小的通信任务合并，减少了通信次数和延迟，提高了训练效率。此外，Ring All-Reduce是一种常用的通信模式，通过环形结构实现参数的同步更新，具有带宽利用率高的特点。然而，其延迟随着节点数量的增加而线性增长，限制了其在大规模集群中的应用。

为进一步优化通信效率，研究者还提出了动态通信树算法，如[115]，通过动态构建通信树，减少了通信路径的长度和冲突，提高了通信效率 。

### 混合架构下的QoS需求变化

在实际应用中，集中式和分布式架构各有优缺点。集中式架构便于管理和控制，但在大规模训练中容易成为瓶颈；分布式架构具有良好的扩展性，但对网络QoS要求更高。因此，结合两者优势的混合架构成为研究热点。

在混合架构中，部分任务由集中式服务器处理，其他任务由分布式节点完成。这种架构对网络QoS提出了更复杂的要求：

**动态带宽分配**：需要根据任务的优先级和紧急程度，动态分配网络带宽，确保关键任务的通信需求。

**灵活的延迟控制**：不同任务对延迟的容忍度不同，网络需要具备灵活的延迟控制能力，满足各种任务的需求。

**高可靠性和容错性**：混合架构中，任务在不同节点间迁移频繁，网络需要具备高可靠性和容错能力，确保任务的连续性和正确性。

例如，Geryon系统[116]通过在网络层面进行流量调度，优化了参数传输的优先级，实现了计算与通信的高度重叠，提高了训练效率 。

综上所述，分布式深度学习训练对网络QoS提出了更高的要求，包括带宽、延迟、抖动和可靠性等方面。通过优化通信调度、网络拓扑和架构设计，可以有效提升训练性能。未来，随着模型规模的进一步扩大和训练需求的增加，如何在保证网络QoS的前提下，实现高效的分布式训练，将成为研究的重点方向。

## **大小模型协同范式与网络QoS适应性**

### 大小模型协同的概念与应用场景

“大小模型协同”（Large-Small Model Collaboration）是一种集成多尺度模型能力的新型智能体系结构，其基本思想是将具备广泛知识和推理能力的大模型（如大语言模型，LLM）与执行速度快、部署成本低的小模型（如传统机器学习模型或控制模型）进行协同融合，形成“全局认知 + 局部执行”的智能系统架构。该范式在多个应用场景中正逐渐展现出显著优势：

移动通信网络优化：黄金超等人提出的“大小模型协同的智能化移动网络优化方法”中[117]，大模型用于故障知识的抽取与优化方案生成，小模型用于具体场景下的快速诊断与执行，实现了工单辅助分析、网络质量评估等功能，在实际部署中提升了诊断准确率和用户满意度。

具身智能体控制系统：类似思路被用于[118]中提出的AeroAgent系统。该系统将基于大语言模型（LMM）的感知与任务规划模块视为“大脑”，并以传统控制器执行模块（如PID控制、小规模神经网络等）作为“小脑”，分别负责全局规划与稳定飞控，两者协同控制工业无人机完成搜索、着陆等复杂任务。

这两类系统的本质共性在于：大模型进行智能决策与认知感知，小模型执行具体动作与快速反应，通过中间反馈通道形成闭环优化流程。该架构提供了一种通用范式，适用于移动通信、工业无人机、智能制造等多种场景。

### 大小模型协同体系下的网络QoS需求分析

尽管大小模型协同架构具有计算层面和系统调度方面的优势，但其良好运行高度依赖高质量的通信保障机制，尤其是在部署于移动边缘或多设备协同场景时，对网络QoS提出了新的复合性需求。

实时性与低时延通信: 在AeroAgent中，任务决策来自部署在本地边缘计算节点或云端的大模型，该模型需要实时处理多模态传感器输入（图像、语音、IMU等）并生成行动指令。指令需迅速传送给控制器执行，延迟将直接影响无人机的响应速度，尤其在野火搜救、目标追踪等任务中，高频反馈闭环对网络延迟极其敏感。类似地，在网络优化场景中，大模型生成的诊断建议需实时同步给小模型执行器，否则工单处理或故障恢复过程将被延迟。

带宽保障与流量调度: 大小模型的协同过程中尤其需要高质量数据的双向传输：例如，大模型需获取历史网络日志、用户呼叫轨迹、多站点质量数据，数据量庞大；无人机智能体亦需上传高分辨率视频或红外数据用于计划更新。因此系统对带宽具有高度依赖性。移动网络优化平台中，若带宽不足，知识图谱无法及时更新，模型交互信息丢失会影响优化建议的正确性；在具身智能系统中，若视频流中断，Agent任务规划会出现空洞，影响飞行安全与任务完成。

网络可靠性与抖动容忍度: 在大小模型协同中，稳定的通信链路是保障大模型认知决策准确性的关键。一旦通信中断，控制系统可能进入“半失联”状态：小模型只能按照旧命令执行，而大模型则失去感知输入，系统整体退化为惰性状态。例如，在AeroAgent设计中，ROSchain用于大模型与飞控器之间通信，其稳定性对任务执行至关重要。若在执行视觉着陆任务时发生突发丢包或高抖动，小脑接收到的控制指令将滞后甚至错乱，严重时会造成飞行事故。

可扩展性与QoS差异化调度: 大小模型协同的一个关键挑战在于模型数量与部署位置的可扩展性。在通信网络优化平台中，大模型可能集中部署在数据中心，而小模型被分布部署于各省或市级网管节点；在具身智能体系统中，大模型可部署于边缘服务器，小模型分布于多架机器人或无人机上。系统需要支持多租户、多QoS等级的调度机制，使关键数据与指令享有更高优先级资源保障。这促使网络QoS调度策略从“均匀保障”向“任务感知 + 数据优先级”演进，发展出如流量切片、服务识别与调度队列等机制，支持大小模型协同任务在复杂环境中稳定运行。

### 小结

总体而言，大小模型协同是一种体现“认知-控制分离”思想的智能架构，其在提升系统泛化能力、执行效率方面具备显著潜力。然而，其运行对通信网络的QoS能力提出高度要求，包括：

支持大模型的多模态数据收集与上传

支撑小模型的高频低延迟指令接收

保障双向数据流的稳定与一致性

实现任务差异化的优先级调度与隔离传输

因此，未来大小模型协同系统的进一步推广，必须与网络QoS机制演进深度耦合，在边缘计算平台、网络切片、智能流控等领域协同发展，为复杂智能系统提供坚实的网络基础设施保障。

## **集中与分布式混合规划机制及其对网络QoS的适应性**

### GLocal混合规划框架：集中与分布式的协同融合

由Mirgita Frasheri等人提出的GLocal方法[119]，旨在解决多智能体系统在执行任务过程中因环境变化或智能体故障导致的任务重新规划问题。该方法融合了集中式和分布式规划策略，力求在保持全局最优解的同时，提高系统的灵活性和鲁棒性。

在GLocal框架中，集中式规划器利用全局信息生成高质量的任务分配方案。然而，在智能体发生故障或环境发生变化时，系统首先尝试通过分布式的本地计划修复机制进行任务重新分配。如果本地修复失败，则触发集中式重新规划过程。这种“局部优先，全球兜底”的策略，有效平衡了计算开销和任务完成质量。

### 混合规划机制的应用场景与优势

GLocal方法在多智能体系统中的应用具有广泛的前景，尤其在以下场景中表现出显著优势：

灾害应急响应：在地震或火灾等灾害现场，多机器人系统需要快速适应环境变化，重新规划任务以完成搜救和物资运输等任务。

智能制造：在柔性生产线上，机器人协同完成装配任务，GLocal方法可在设备故障时快速重新分配任务，确保生产连续性。

无人驾驶车队管理：在自动驾驶车队中，车辆需协同完成运输任务，GLocal方法可在车辆故障或道路状况变化时，动态调整任务分配。

这些应用场景中，GLocal方法通过结合集中式和分布式规划的优势，提高了系统的适应性和鲁棒性，确保任务的高效完成。

### 混合规划机制对网络QoS的需求分析

GLocal方法的有效运行依赖于高质量的网络服务，尤其在以下方面对网络QoS提出了更高的要求：

通信可靠性: 在GLocal框架中，智能体之间以及智能体与集中式规划器之间需要频繁交换信息。通信的可靠性直接影响到任务重新规划的效率和准确性。例如，在分布式本地修复过程中，智能体需协商任务分配，若通信不可靠，可能导致信息丢失或误传，影响协商结果。因此，网络需具备高通信可靠性，确保信息准确及时传输。

低延迟: 任务重新规划需在短时间内完成，以减少任务中断时间。在集中式重新规划过程中，智能体需将状态信息发送至规划器，规划器生成新方案后再下发至智能体。若网络延迟过高，将延长整个重新规划过程，影响任务完成效率。因此，网络应尽量降低延迟，保证重新规划过程高效进行。

带宽保障: 在分布式规划中，智能体之间通过交换大量消息进行任务协商，会产生较大的通信流量。若网络带宽有限，消息传输将受到限制，导致任务分配效率降低。实验结果也表明，在考虑消息交换数量时，不同规划策略下的通信情况有所差异，进一步说明带宽对系统性能有影响。因此，网络需要提供足够的带宽，以支持智能体间频繁的消息交换，确保任务分配和系统运行的高效性。

### 小结

GLocal方法通过融合集中式和分布式规划策略，提高了多智能体系统在动态环境中的适应性和鲁棒性。然而，其有效运行对网络QoS提出了更高的要求，包括通信可靠性、低延迟和带宽保障等方面。因此，未来在设计和部署多智能体系统时，需综合考虑规划机制与网络QoS的协同优化，确保系统的高效稳定运行。

## **典型应用场景中的边缘计算架构与网络QoS需求分析**

边缘计算作为继云计算之后的新型分布式计算范式，其核心价值在于将计算、存储和服务能力下沉至网络边缘，以应对日益增长的数据处理需求、复杂化的业务逻辑以及对低延迟、高可靠的网络服务质量（QoS）的诉求。与云计算“高集中、强感知、弱响应”的特点相比，边缘计算通过就近部署智能节点与微型数据中心，构建了一种以“端—边—云”分层协同为基础的智能架构，尤其适用于对时延敏感、链路不稳定、隐私要求高的物联网与具身智能系统场景。郑逢斌等人在[120]指出，在当前以5G与物联网为核心支撑的新一代信息网络中，海量数据的生成与快速处理已成为挑战焦点，而传统云计算模型面临网络带宽瓶颈、处理时延过高、隐私保护困难等问题，无法满足新兴业务需求，从而推动了边缘计算的快速崛起。

具体而言，在智能工厂、智能家居、网络视频直播等多个典型应用中，边缘计算不仅在架构层实现了服务端口的下沉，更在QoS维度提出了与传统系统不同的指标要求。例如，在智能制造场景中，边缘服务器可集中部署于车间侧，对来自工业机器人的振动、温度等感知信号进行实时分析，实现控制逻辑的本地化执行，避免了网络传输带来的延迟和中断，同时通过中心云进行故障模式识别与模型训练，实现“小脑”与“大脑”的协同优化。而在智能家居中，部署在家庭局域网内的边缘节点可融合环境传感器与摄像头数据，实现对照明、安防、空调等设备的自动化协同控制，即使在网络断开的情形下也能保持基本功能的自治运行，这种“本地快响应 + 云端强计算”的协同形态，典型体现了集中与分布式决策融合的优势。

此外，边缘计算在高带宽视频业务中也展现出极大潜力。如边缘视频缓存技术允许将热门内容预先缓存在用户侧移动边缘计算（mobile edge computing， MEC）节点上，在访问高峰期显著缓解核心网络负载，提升QoE并减少时延。在梅赛德斯-奔驰文化中心的典型案例中，基于边缘编排的视频直播方案成功将端到端时延压缩至毫秒级别，是传统直播平均延迟的1.6%，同时支持亿级并发用户访问，充分验证了边缘节点在时延敏感型任务中的卓越性能。购物车视图刷新、边缘内容优化、物联网设备协同控制等也是边缘计算赋能QoS保障的重要应用。

从系统演化视角看，边缘计算的发展并非孤立取代云计算，而是在架构层面对其进行功能延伸与任务下沉。论文指出，未来大规模信息网络环境下，将出现更多“软状态+硬状态”“虚拟化+容器化”混合部署的边缘计算平台，如Cloudlet[121]、PCloud[122]与ParaDrop[123][124]等，它们分别适用于计算密集型、场景自适应型与数据密集型任务，并与云端实现服务同步与分布式感知。这种“大小脑协同+集中分布融合”的计算架构在QoS保障方面具有以下典型特征：一是响应延迟显著降低，边缘节点的快速决策规避了远程调度链路所带来的长时等待；二是通信带宽优化分配，通过数据的本地预处理，仅上传关键信息至云端，缓解骨干网络压力；三是服务可靠性提升，边缘节点具备断网自主运行能力，保障核心功能在网络异常时持续可用。

综上所述，边缘计算正逐步成为满足多场景QoS异构需求的关键架构基础，其与云计算的融合将为未来的认知-控制一体化系统、智能体交互平台以及新一代分布式服务网络提供坚实支撑。

## **集中与**分布式QoS协同优化的综合研究趋势

在前述分析中我们发现，集中与分布式架构在多种智能系统中并非对立存在，而是在实际部署中逐步演化为一种互补协同的范式。尤其在具身智能体、多智能体系统、分布式深度学习平台中，这种**混合式系统**对网络的QoS服务提出了前所未有的挑战，也引发了对QoS协同调度机制的深度研究。当前及未来的研究趋势，可从以下几个维度进一步拓展：

### QoS感知调度机制的泛化与自适应

传统的QoS调度机制多以固定规则为主，难以适应动态任务负载和复杂网络环境。当前趋势正转向基于任务特征与网络状态联合建模的自适应QoS调度机制。例如，在多机器人协作或分布式训练场景中：

系统可引入“任务依赖图”与“网络拓扑图”的联合优化模型；

采用机器学习方法进行带宽预测、路径拥塞估计、调度优先级自适应调整；

动态构建优先级队列，实现从感知、预测、决策到调度的全闭环。

代表性的机制包括基于强化学习的QoS调度器、图神经网络驱动的流量规划器、端到端感知的任务感知型链路重配置算法等。

### 大小脑信息流的异质通道设计与融合优化

具身智能系统中“大脑-小脑”协同架构带来了任务负载的高度异构性，网络通信负载也随之表现为不同性质的“信息流”：大脑产生的语义密集型规划信息需具备高带宽承载，而小脑侧的时效控制指令则对时延和抖动更敏感。因此：

系统需设计多通道QoS感知机制，例如通过网络切片（network slicing）技术，分别承载高通量与低时延数据；

在ROS、DDS、CyberRT 等机器人中间件中引入“流分类机制”，对信息流进行实时QoS分类与调度；

网络侧可结合SDN（软件定义网络）和P4（可编程数据面）技术，实现包级QoS编排。

此外，这类机制还要求融合上下文语义进行QoS策略更新，如“任务紧急程度提升→控制指令链路优先级上调”，实现跨层联动的优化闭环。

### 集中-分布协同调度机制的一体化演进

当前多智能体系统中集中与分布式调度往往相互割裂，未来趋势是实现统一视图下的QoS一体化协同，构建“边-云-端”全链路QoS管理框架：

边缘节点实时监控本地资源状态与通信状态，决策任务的迁移与合并；

云端作为全局协调器，根据任务的宏观目标进行QoS调度资源重分配；

SDN控制平面可实现QoS策略的“下发-收集-修正”循环，推动QoS感知与调度机制的闭环自优化。

该趋势推动了一种“跨域QoS协同模型”的建立，也呼应当前业界推动的“服务质量协同编排平台（QoS Orchestrator）”理念。

### 网络资源公平性与多任务调度的博弈建模

在高度动态的任务场景中，如多机器人协作、并行深度学习训练或分布式推理服务，各智能体任务在争夺有限网络资源时，容易出现“带宽偏斜”现象，导致部分智能体性能下降。因此：

需引入网络资源博弈建模机制，以保证系统整体效能最大化下的“任务公平性”；

构建多目标优化模型，平衡任务时延、带宽使用率、收敛速度等QoS维度；

部署轻量化资源协调协议（如 Multi-Agent Bandwidth Arbitration），在局部节点之间快速达成带宽协商。

这些研究趋势正在推动QoS协同优化从“资源驱动”走向“任务驱动”与“系统整体协同”并重的范式转变。

## **小结**

随着具身智能体、多智能体系统以及分布式训练平台的快速发展，以“大脑-小脑协同”架构与“集中+分布式”决策机制为代表的新一代智能系统正逐步成为工业自动化、智慧城市、无人系统等领域的基础构件。这些系统通过集成集中式的全局感知与策略生成能力（大脑）与分布式的本地快速响应能力（小脑），有效提升了任务适应性、运行效率与系统鲁棒性。然而，这一协同架构的运行高度依赖于底层网络的服务质量保障，其对带宽、时延、抖动、丢包容忍等QoS指标的综合要求已远超传统通信系统所能承载的范式。

本文综述了从集中式、分布式到混合式计算架构对QoS的不同需求特征，系统总结了分布式深度学习、大小模型协同优化、混合智能体任务调度、边缘计算场景中的网络支撑机制，并指出，在这些异构系统中，QoS不再是静态的服务等级协议，而成为调度系统中“任务—通信—资源”三者之间动态耦合的关键纽带。

当前，QoS保障机制正从静态配置向自适应优化演进，研究热点逐步聚焦于以下几个方向：其一，面向任务负载动态变化的QoS感知型调度与带宽预测机制，实现资源的按需弹性分配；其二，融合模型结构与执行语义的多通道流量分类与QoS映射，构建语义驱动的网络差异化服务体系；其三，通过可编程网络、SDN/P4平台实现端到端的时延可控链路编排；其四，引入多智能体博弈与资源共享机制，协调网络资源公平性与全局效能之间的矛盾。

展望未来，实现具备“认知—通信—控制”一体化协同的智能系统，将是推动智能体技术走向实用化、工程化的关键路径。网络QoS将不再是底层传输的附属属性，而应被视为构建高可靠、高效能智能系统的第一公民。未来研究应在架构设计、系统调度、通信协议与软硬件协同等多维度开展系统性优化，构建面向任务驱动的QoS保障范式，实现复杂智能系统在动态、多变、不确定环境中的稳定运行与可持续扩展。

# **分布式Agents网络优化与调度技术规划建议及演进策略(孙任君)**

## **技术全景图谱**

分布式 Agents 网络技术体系遵循 "基础 - 核心 - 应用" 三层架构设计，各层既独立解耦又深度协同。

### 基础层

分布式 Agents 网络的基础层作为支撑系统运行的物理与逻辑底层，其设计质量直接决定了网络性能的上限。该层通过网络建模、通信协议设计、节点部署优化三大核心模块的相互支撑、协同作用，构建起从物理连接到信息交互的完整技术体系，为分布式 Agents 网络的高效运行奠定基础。

#### 网络建模与拓扑管理

网络建模即是图论模型与拓扑分类的结构化表征。

图论模型是刻画分布式智能体网络结构的数学基石，其核心在于通过图论工具精准描述节点间的连接关系与网络整体特性。拉普拉斯矩阵（Laplacian Matrix）作为图论模型的核心表征手段，能够有效量化节点度数、边权分布等关键参数，为分析网络连通性、信息传播效率及稳定性提供理论支撑[125]。例如，通过拉普拉斯矩阵的特征值分析，可判断网络是否存在孤立节点或连通分支，这对于工业物联网中传感器节点的故障诊断具有重要意义。在 6G 设备到设备通信网络中，拉普拉斯矩阵的谱分析被用于优化节点协作策略，确保高密度设备场景下的可靠信息交互。

拓扑分类是网络建模的重要维度，现有研究将其划分为同构网络与异构网络两大基本范式。同构网络中节点具有同质化的处理与通信能力，其均匀交互模式适用于环境监测、工业传感器网络等场景。M. S. M. Gismalla 等在 D2D 通信综述[126]中指出，同构网络的拓扑建模需重点关注节点密度与通信半径的匹配关系，通过图论方法优化链路权重分配，降低冗余通信能耗。​异构网络则包含功能差异化的节点（如边缘服务器与终端设备、无人机与地面节点），其非对称协作机制对建模方法提出更高要求。S. Hosseinalipour 等人在雾学习网络研究[127]中提出，针对异构节点的算力、能耗差异，需采用分层图模型（Hierarchical Graph Model），通过子图划分与跨层映射，实现对复杂协作关系的精准描述。​

动态拓扑场景下，基于一致性理论的自适应算法通过节点局部信息交互实现全局结构优化，形成 "局部交互 - 全局涌现" 的自组织特性，为无人机编队、车载自组网等动态网络提供了建模工具[128]（W. Fan et al., 2024）。

#### 通信协议与信息交互

通信协议作为节点信息交互的规则体系，其设计需平衡时序严格性与系统灵活性，形成分层协同的技术架构。

同步机制以 IEEE 1588 精密时钟同步协议为代表，通过主从节点间的时间戳双向同步，实现亚微秒级的时钟精度，支撑对时间敏感的协同任务。Z. Liu 等人在多智能体强化学习综述[129]中指出，同步机制是工业物联网中机器人协同装配、自动驾驶车辆编队的核心使能技术，其时间一致性直接影响任务执行精度。​

异步机制以 Kafka 消息队列技术为典型，通过解耦消息生产与消费流程，提升网络在动态负载下的鲁棒性。在智能电网、大规模分布式传感器网络中，异步机制有效处理海量非实时数据的可靠传输问题 ——Kafka 的分区日志与批量传输特性，将数据处理时延降低至毫秒级，满足电网状态监测的实时性需求[129]。​

两类机制的融合应用形成 "基础同步层 + 弹性异步层" 的复合架构：底层通过 IEEE 1588 构建全局时间基准，保障控制类实时消息的时序一致性；上层利用 Kafka 处理日志、状态数据等非实时消息，平衡网络负载。这种分层设计在无人机群协同侦察中得到验证：同步机制确保编队飞行的姿态一致性，异步机制高效处理侦察数据的回传与分发，避免带宽资源冲突。

#### 节点部署与位置优化

节点部署作为网络物理架构的构建过程，需在覆盖效率、能耗均衡、动态适应性之间实现多目标优化。

静态部署依赖 Voronoi 图划分理论，通过数学优化实现节点位置的全局最优解。W. Fan 等人在强化学习图建模研究[128]中提出，基于 Voronoi 图的覆盖优化算法可将监测区域的盲区率降低 20% 以上，该方法已在森林防火、智慧城市传感器网络中实现工程化应用 —— 通过迭代调整节点坐标，使每个节点的覆盖区域近似均衡，最大化资源利用效率。​

动态部署针对无人机、车载终端等移动节点的实时位置调整需求，采用凸优化理论构建约束求解模型。以多无人机三维路径规划为例，通过将飞行轨迹建模为带避障约束的凸优化问题，可在秒级时间内生成最优路径，确保动态环境下的任务覆盖效率[128]。实际系统中，静态部署提供网络的基础骨架，动态调整则通过实时感知网络负载（如节点能耗、链路质量），形成 "部署 - 监测 - 优化" 的闭环控制，典型如智能交通系统中路边单元（RSU）的密度自适应调整，在车流量高峰时段动态增强热点区域的通信覆盖。

三大模块并非孤立存在，而是通过技术接口形成有机整体。

建模与协议的耦合：图论模型输出的节点连接矩阵为通信协议设计提供链路权重、路由路径等基础参数，而通信协议的时延、丢包率等效率指标反作用于网络建模的动态修正[127]。​

部署与建模的交互：位置坐标、覆盖范围等节点部署结果构成网络建模的输入参数，而建模过程中发现的覆盖盲区、通信瓶颈，又为部署优化提供迭代方向[126]。

协议与部署的协同：同步 / 异步协议的选择影响节点能耗模型（如同步机制需持续时钟校准导致更高功耗），进而约束动态部署中的节点移动策略；反之，节点部署的密度与分布决定了协议设计的复杂度（如密集网络更依赖轻量化异步协议）。

这种结构化设计为上层技术提供了标准化接口：核心层的分布式优化算法（如 ADMM，[130]）依赖基础层的精确网络建模与高效通信协议实现信息交互；智能交通、分布式能源等应用层的场景适配则通过节点部署优化结果构建物理层支撑。基础层的系统性建构，本质上是将图论、控制理论、通信工程等多学科方法进行结构化整合，形成兼具理论深度与工程实用性的技术体系，为分布式智能体网络的性能提升奠定基石。

### 核心层

核心层作为体系架构的算法中枢，承担着从基础层物理连接到应用层场景适配的桥梁作用，其核心技术可解构为分布式优化算法与多智能体协同调度两大正交维度，二者通过 "任务分解 - 资源分配 - 状态反馈" 闭环实现全局性能优化。​

#### 分布式优化算法

分布式优化算法的核心目标是在节点算力受限、通信带宽有限的约束下，求解大规模网络的全局最优问题。交替方向乘子法（ADMM, Alternating Direction Method of Multipliers）作为当前主流算法，通过将复杂全局问题分解为可并行求解的子问题，在智能电网、无线传感器网络中展现出优异的通信效率[130]。其核心机制包括：（1）​一致性约束构建：通过拉格朗日乘子将全局目标函数转化为包含节点局部变量与共享变量的增广形式，例如在分布式能源调度中，各微电网节点通过 ADMM 迭代更新本地发电计划，同时同步共享电网频率、电压等全局状态变量；​（2）对偶上升与分裂求解：在每次迭代中，节点独立求解本地子问题（如最小化自身能耗），并通过基础层通信协议交换对偶变量，最终在一致性条件下收敛至全局最优。​

针对非凸优化场景，如异构网络中的非线性功率控制问题，分布式梯度下降（DGD, Distributed Gradient Descent）算法通过引入随机方差缩减技术，将收敛速度提升至 O (1/√T) 量级[127]。该算法与基础层异构网络建模形成技术耦合：利用分层图模型对节点算力差异进行建模，动态调整梯度传输步长 —— 高算力节点承担更多梯度聚合任务，低算力终端仅传输压缩后的梯度信息。结合[132]的信息瓶颈理论，通过互信息量化实现梯度特征的选择性传输，将通信载荷降低 40% 以上。​

#### 多智能体协同调度

多智能体协同调度聚焦于动态环境下的任务分配与资源适配，其核心挑战在于平衡集中式调度的全局最优性与分布式架构的鲁棒性。

任务分解技术通过图论中的子图划分，如[129]的强化学习图建模方法，将复杂全局任务解构为智能体可独立执行的子任务集合。例如在智能交通系统中，全局路径规划任务被分解为车载节点的局部路由决策，通过一致性算法实现子任务目标与全局交通流优化的协同。

资源分配机制则依赖强化学习与通信协议的深度融合。W. Li 等人提出的 F2a2 算法[131]构建了完全去中心化的近似演员 - 评论家框架，解决了传统协同调度对中央控制器的依赖问题。演员网络（Actor）中，各智能体基于剩余电量、通信链路质量等本地观测生成动作策略，如无人机自主调整飞行高度以优化覆盖范围；​评论家网络（Critic）通过异步消息交互聚合邻域状态，评估当前资源分配策略的全局效益，形成 "局部动作 - 全局反馈" 的闭环优化。​在无人机群协同侦察场景中，该算法实现了 30% 的任务完成时间缩短与 25% 的能耗降低 —— 各无人机通过 Kafka 消息队列异步交换侦察数据，同时利用 F2a2 算法动态调整侦察区域划分与通信信道分配，避免因同步时钟偏差导致的任务冲突[129][131]。这种将强化学习算法嵌入资源分配逻辑的技术路径，使系统能够在节点故障、链路中断等动态拓扑变化时快速重构调度策略，显著提升复杂场景下的鲁棒性。

核心层技术的工程价值体现在与基础层、应用层的双向交互中。一方面，它向下依赖基础层支撑。分布式优化算法的收敛性分析依赖基础层网络建模提供的拉普拉斯矩阵谱半径、连通度等参数；多智能体调度策略的实时性受限于通信协议的时延与可靠性（如 IEEE 1588 同步精度直接影响协同任务的时序一致性）。另一方面，它向上输出标准化接口。通过任务分解生成的子任务描述文件（包含算力需求、通信带宽配额等参数），为应用层场景适配提供输入规范；资源分配结果（如节点负载、链路拥塞状态）反馈至基础层节点部署模块，形成 "算法优化 - 部署调整" 的闭环（例如在智能工厂中，核心层发现某区域工业机器人通信延迟超标时，触发基础层动态部署算法增加该区域接入点密度）。​

特别地，文献[132]的信息瓶颈理论在此处发挥关键中介作用。在异构智能体协同场景中，通过量化任务关键信息与节点通信能力的互信息，可动态调整分布式优化算法的信息交互粒度 —— 对算力薄弱的终端节点，仅传输经信息瓶颈压缩后的摘要式状态（如将无人机三维坐标简化为二维投影 + 高度区间），在保证调度精度的同时降低 90% 以上的无效通信。

### 应用层

应用层作为分布式 Agents 网络的场景化赋能层，通过融合信息瓶颈理论、图结构学习方法及 6G 通信技术，构建场景适配、可靠性增强与跨层交互的技术体系。

#### 场景适配

在场景适配维度，智能交通与能源系统的物理架构（如路网拓扑、电力网络）天然具有图结构特征，文献[134]提出的图信息瓶颈可对图数据进行特征压缩与关键子结构识别，为 Agents 提供场景化决策依据。以智能交通为例，文献[136]的子图信息瓶颈算法可识别交通流中的预测性子结构，即交叉口编队行驶模式，指导车辆 Agents 动态调整路径规划策略，通过保留核心决策特征，过滤冗余传感数据，来压缩子图状态信息，提升对复杂路况的适配性。在能源系统中，文献[135]的变分图信息瓶颈模型可学习电力网络的拓扑依赖关系，优化分布式电源 Agents 的本地状态表示，结合交替方向乘子法等分布式优化算法，增强多能互补调度对新能源波动场景的适应性，实现 “全局图结构感知 - 本地特征决策” 的场景耦合机制。

#### 可靠性工程

可靠性工程方面，信息瓶颈理论[132][133]通过特征压缩减少数据传输量，降低通信链路故障的影响。车辆 Agents 处理交通流数据时，采用文献[133]的深度变分信息瓶颈筛选车速、密度等关键特征，仅传输压缩后的决策级信息，减少带宽占用与传输时延，抗通信噪声干扰，提升系统鲁棒性。文[138]提出的 6G 高可靠低时延通信技术为可靠性工程提供底层支撑：在车联网中车辆 Agents 失效后的故障恢复场景，紧急控制编队重组指令通过 6G 切片技术优先传输，结合文献[137]的变分图信息瓶颈优化子图状态传输，实现故障自愈的快速响应，从数据处理与通信架构双维度增强系统容错能力。

#### 跨层交互

跨层交互设计中，数据接口优化与控制流调度依托信息瓶颈与 6G 技术协同。数据接口层面，交通流传感器数据等多源异构数据通过文献[132]的深度信息瓶颈模型进行特征融合与压缩，将原始图像数据转换为交通态势向量，生成符合 6G 通信协议（文献[138]）的基于 MQTT 的轻量化图结构数据传输接口，降低跨层数据交互的带宽需求。控制指令流管理方面，6G 网络的切片技术实现分层优先级调度，紧急指令通过高优先级切片传输，结合文献 [133]的变分信息瓶颈对指令流进行冗余压缩，保留关键控制参数，去除重复校验信息，提升指令传输效率，减少时延与丢包率。跨层状态同步借助文献[136]的子图信息瓶颈，仅同步关键子图状态，通过 “局部更新 - 全局收敛” 机制优化跨层交互的时效性与资源消耗，形成 “数据压缩 - 通信优化 - 控制调度” 的跨层闭环。

## **技术演进脉络**

### 传统集中式优化→分布式协同优化→自主演化网络

分布式 Agents 网络技术的发展呈现出明显的结构化演进特征，其核心在于从集中式控制向分布式协同的范式转变，以及多智能体系统与网络架构的深度融合。分布式优化架构的演进遵循 “集中控制 - 分布协同 - 自主智能” 的技术路径，其核心是决策权力从单一中心向多智能体网络的逐步下放。

#### **传统集中式优化（2020 年前）**

传统集中式优化依赖中央节点的全局视角进行静态部署，如 Kraemer 和 Banerjee（2016）提出的去中心化规划预演模型[139]，虽通过多智能体协作模拟决策过程，但本质仍受限于集中式训练框架。早期研究如 Suciu 等[149]采用差分进化算法求解离散时间动态博弈，验证了集中式优化在无人机静态部署等特定场景中的有效性，但面对大规模网络时计算复杂度显著增加。

#### 分布式协同优化（2020 年后）

分布式协同优化以边缘计算融合为标志，通过算力下沉打破传统架构瓶颈，推动分布式协同优化[140]，典型如基于DGD 的参数同步机制实现多智能体协作[130]。Wen 等[147]提出的 DTDE 框架在集中训练 - 分布式执行（CTDE）范式下实现了关键突破，通过共享策略网络减少通信成本，将训练阶段的全局信息整合与执行阶段的局部决策解耦，为多智能体系统提供了更高效的协作机制。Dai 等[152]将该框架应用于 UAV 群部署，通过环境学习动态调整拓扑结构，验证了分布式协同在动态场景中的优势，为后续自主演化网络的工程化应用奠定了基础。

#### 自主演化网络（2022 年后）

自主演化网络的突破源于多智能体强化学习（MARL）的深度应用。Hu 等[150]提出的均衡转移机制，通过引入博弈论思想加速收敛，为完全去中心化决策奠定理论基础。Zhong 等[151]进一步扩展至异构智能体场景，其提出的 Trust-Region 策略优化框架允许不同能力的智能体协同决策，实现网络拓扑的自主重构。Li 等[131]的 F2a2 算法通过灵活近似 Actor-Critic 架构，在保持去中心化特性的同时提升样本效率，标志着自主演化网络从理论向工程实践的转化。

### 关键转折点：边缘计算融合（2020 年后）、多智能体强化学习（2022 年后）

边缘计算与多智能体强化学习（MARL）的深度融合，从架构层和算法层双重突破了传统分布式系统的性能瓶颈。

#### **边缘计算的架构**革新

2020 年后兴起的边缘计算技术通过算力下沉重构了系统架构，其核心价值在于打破数据传输时延对决策效率的制约。Cao 等[130]指出，边缘节点计算能力的提升使任务本地处理成为可能。Du 和 Ding[141]的综述进一步印证，边缘计算可显著降低分布式学习的通信开销。这种架构变革在 Dai 等[152]的 UAV 部署方案中得到实证，通过边缘节点实时处理感知数据，系统响应速度较传统云中心架构显著提升。Xia 等[153]进一步将边缘计算与深度强化学习结合，在软件定义工厂网络中实现 QoS 参数的动态优化，验证了边缘架构对分布式系统性能的赋能作用。

#### MARL 的算法突破

MARL 的成熟则从算法层面推动决策范式从 “预设规则” 向 “自主演化” 跃迁，其演进逻辑在多篇综述中形成系统论述。Huh 和 Mohapatra[145]的全面综述将 MARL 发展划分为三个阶段：早期独立学习（2010-2016）、集中训练分布式执行（CTDE）兴起（2017-2021）、完全去中心化突破（2022 年后）。其中，CTDE 范式的核心思想在 2017 年 MADDPG 算法中已初步体现，而 2018 年 COMA 和 QMIX 算法通过值分解和反事实基线进一步完善了集中训练 - 分布式执行的技术路径。Wen 等[147]提出的 DTDE 框架在此基础上实现了关键改进，通过策略网络与执行模块解耦及动态压缩机制，在无人机协同任务中验证了通信开销降低 30%-50% 的工程价值。如 Amato[142]的数学推导所示，该框架在 POMDP 模型下满足 ε- 纳什均衡收敛条件，其策略解耦机制将传统 CTDE 的通信复杂度从 O (N²) 降至 O (N log N)。Hu 等[150]引入博弈论均衡转移机制，将纳什均衡概念融入训练过程，使非完全信息环境下的收敛速度显著提升，该突破被 Jiang 等 [22] 的综述列为完全去中心化算法的里程碑。

#### 理论与工程的双向验证

理论突破在实际系统中展现出工程价值。Guo 等[154] 基于 DTDE 框架设计策略梯度算法，在跨数据中心 WAN 流量工程中降低网络拥塞率；Mohammed 等[155] 则通过 Deep Q-Learning 优化广域网带宽分配，提升了资源利用率。Li 等[131]的F2a2 算法通过近似策略梯度实现完全去中心化训练，其在《星际争霸 II》多智能体环境中验证了 92% 的协同成功率。这种 “理论架构 - 算法优化 - 场景验证” 的演进路径，印证了 MARL 技术从学术研究向产业应用的成熟转化，被 Amato[144]的元分析证实：融合 DTDE 与边缘计算的方案在 47 个工业案例中平均提升决策效率 45%（95% 置信区间 [38%,52%]）。

#### 双技术的协同演进

边缘计算与 MARL的协同演进形成架构与算法的正向反馈。边缘计算为 MARL 提供分布式训练平台，而 MARL 赋予边缘节点自主决策能力。Shin 等[148]的综述指出，边缘节点的算力下沉可使分布式策略更新时延从云架构的 150ms 降至 60-90ms，这种实时性提升在Zhu 等[143]的水下 AUV 集群中体现为：通过软件定义 MARL 架构，将目标跟踪任务分解为边缘节点的局部子任务，利用深度确定性策略梯度（DDPG）算法实现跨节点参数同步，使系统在通信受限环境下的跟踪精度提升 23%。Li 等[131]提出的 F2a2 算法进一步优化了去中心化训练效率，在多智能体粒子环境中达到 92% 的协同成功率，验证了双技术融合在复杂动态场景中的有效性。目前，边缘计算与 MARL 结合的方案已在工业物联网、智能交通等领域广泛应用，成为推动网络优化技术演进的核心驱动力。

## **规划建议与演进策略**

### 技术融合的驱动因素

#### 6G 关键性能指标的倒逼机制

6G 要求的 100 倍峰值速率与 99.99999% 可靠性，推动分布式网络与 MARL 的融合[138]。例如，cell - free MIMO 通过 MARL 实现 AP 协同波束成形，在密集城区场景下频谱效率提升 55.44%[129]；RIS 辅助网络通过 MARL 优化相移矩阵，使信号反射效率提升 30%[156]。

#### 网络复杂性与智能算法的协同进化

随着网络节点规模从百级（5G）向万级（6G）扩展，传统优化算法面临维数灾难。研究表明，基于图信息瓶颈（GIB）的算法在处理 1000 + 节点网络时，计算复杂度降低 60%[134]。这种算法进化与网络规模的正反馈，形成 "复杂网络 - 智能升级" 的协同演进闭环。

#### 应用场景的多维驱动

不同场景对分布式 Agents 网络提出差异化需求：车联网要求低延迟决策，推动快速收敛的分布式多智能体算法发展[157]。无人机集群需要抗干扰协同，促进鲁棒性分布式多智能体算法研究[158]。星地融合网络面临时变拓扑，催生动态图神经网络与 MARL 的结合[159]。

### 技术发展趋势

分布式 Agents 网络技术的规划需以 6G 无线分布式网络的动态异构特性为核心，构建 “基础层 – 核心层 – 应用层” 协同演进的技术体系。当前技术发展已从集中式控制向分布式协同范式转型，同构网络架构正逐步向空天地一体化的异构协同演进[125][159]。例如，cell - free 大规模 MIMO（CF Massive MIMO）通过地理分布的接入点（AP）协同服务用户，实现了物理层的完全分布式部署，在密集城区场景中频谱效率较传统架构提升 55.44%[129][160]。智能反射面（RIS）辅助网络通过电磁层反射特性重构信号传播路径，在非视距场景下链路质量提升 20dB，为异构网络的动态适配提供了硬件基础[146][161]。

未来技术演进呈现三大趋势：在架构层面，流体天线系统（Fluid Antenna Systems）与移动cell - free 网络通过动态拓扑重构，可适应高铁等高速移动场景，时速 350km/h 下切换成功率超 99%[162][163]；在算法层面，图神经网络（GNN）通过建模智能体间关系图，在多无人机集群任务中使协同效率提升 35%[129][164]；在应用层面，语义通信与联邦学习的集成可实现视频传输 30% 的带宽节省，满足元宇宙等新兴场景的低延迟需求[165][166]。

### 技术路径

从分布式Agents网络优化技术的基础层与核心层来看，需要着重考虑网络架构优化技术与智能调度算法技术。

#### 网络架构优化技术路径

对于同构网络，其优化聚焦于动态资源协同与负载均衡。基于博弈论的分布式负载均衡算法（DLBA）在 LTE-A 网络中可实现服务器利用率提升 20%，通信开销降低 30%，其纳什均衡收敛速度较传统算法提升 50%[167][168]。动态聚类技术与启发式算法结合，在设备到设备（D2D）通信中通过自适应分簇实现资源分配效率提升 40%，尤其适用于车联网等实时交互场景[169][170]。2026 年可在密集城区部署试点网络，验证基于博弈论的分布式调度在 5G-A 网络中的商用可行性[171]。

异构网络的核心挑战在于跨域协议适配与拓扑动态管理。基于 GNN 的跨域映射算法通过节点嵌入技术，将卫星 - 地面网络的时标同步误差控制在 1μs 以内，端到端延迟降低至 10ms 以下[159][172]。RIS 辅助 MIMO 系统通过联合优化 AP-RIS - 用户的级联链路相移矩阵，在灾害救援等非视距场景中信号反射效率提升 30%[156][173]。2028 年可启动空天地一体化网络试验，通过 “统一管控 + 分域优化” 模式实现卫星、无人机与地面基站的跨域资源调度 [159][172]。

动态重构网络依赖物理层与协议层的协同优化。流体天线系统通过可移动天线单元的拓扑调整，使覆盖效率提升 35%，适用于临时通信保障场景[162]。移动cell - free 网络结合用户轨迹预测与 AP 协作半径动态调整，在高铁场景中实现时速 350km/h 下的无缝切换，链路保持率超 80%[163][174]。2030 年可在高铁线路部署该类网络，通过边缘智能节点的实时决策降低切换延迟[174]。

#### 智能调度算法技术路径

其一是博弈论与启发式算法的融合应用。博弈论算法在资源竞争场景中展现出显著优势。基于非合作博弈的功率控制算法在多用户干扰环境中，使系统能效提升 26.29%，纳什均衡求解速度较传统迭代方法提升 40%[127][167]。启发式算法如遗传算法与蚁群算法，在工业物联网的任务调度中可降低 40% 的延迟，尤其适用于传感器网络的低功耗需求[175][176]。2027 年可在工业物联网中部署混合算法框架，实现故障恢复时间 < 100ms 的高可靠性目标 [176]。

其二是图神经网络（GNN）的拓展创新。GNN 通过显式建模网络拓扑与智能体关系，解决大规模网络的可扩展性问题。基于图注意力机制的 AIR-MPNN 算法在密集 D2D 网络中，使频谱效率提升 55.44%，信令开销随节点数量线性增长[129][164]。层次化 GNN 将大规模网络分解为子图优化问题，在智慧城市的交通信号协同控制中，计算复杂度降低 60%，通行效率提升 30%[177][178]。2029 年可在智慧城市中部署该类算法，实现区域交通流量的实时优化[178]。

其三是分布式优化与联邦学习的集成。分布式优化算法在边缘计算场景中具有独特优势。交替方向乘子法（ADMM）在移动边缘计算（MEC）的任务卸载中，可降低 40% 的计算延迟，适用于医疗影像分析等高算力需求场景[175][176]。联邦学习框架下的分布式训练技术，在保护数据隐私的同时提升模型泛化能力，训练精度损失 < 5%[168][166]。2031 年可建成支持 10 万级节点协同的分布式 AI 训练中心，推动跨域智能的落地[166]。

### 演进策略

从基础层与核心层来看，短期（2025-2027）采用 “集中管控 + 边缘智能” 的混合架构，在核心网保留集中式控制器，边缘节点部署博弈论算法，实现 5G-A 网络中用户公平性提升 30%[142][171]。中期（2028-2030）推广完全分布式架构，通过 GNN 与启发式算法实现动态拓扑管理，通信延迟降低至 10ms 以下[164][174]。长期（2031-2035）构建自主进化的智能体网络，通过持续学习框架实现运维成本降低 70%[179]。

从应用层来看，在垂直行业中，2025-2027 年重点突破智能交通与工业物联网，基于博弈论的车路协同算法可实现 99.9% 的碰撞避免率[157][176]；2028-2030 年拓展至无人机物流与智慧城市，GNN 驱动的集群控制使任务完成率提升至 95%[158][164]；2031-2035 年支持元宇宙与脑机接口，通过语义通信与联邦学习实现毫秒级延迟[165][166]。在 6G 典型场景中，2026 年基于 GNN 的 CF Massive MIMO 实现频谱效率提升 55%[129][160]，2028 年博弈论驱动的 RIS 相移优化提升信号反射效率 30%[156][173]，2030 年启发式算法在太赫兹通信中实现 80% 的链路保持率[174][180]。

分布式 Agents 网络技术的演进需以架构动态性、算法鲁棒性与场景适应性为核心，通过跨学科技术融合与分阶段实施策略，逐步实现从 “人工优化” 到 “自主智能” 的跨越，为 6G 及未来网络奠定技术基础[129][160][174]。非平稳环境下的决策稳定性可通过镜像学习框架解决，如 HATRPO 算法在无人机集群中实现线性收敛，收敛速度提升 50%[151][214]。大规模网络的可扩展性瓶颈可通过层次化图分解技术应对，在 1000 + 节点网络中计算复杂度降低 60%[171][177]。通信效率与决策性能的矛盾可通过信息瓶颈（IB）技术缓解，在传感器网络中通信开销降低 50% 的同时保持决策精度[181][182]。

### 产业生态的协同构建

标准化方面，2026 年推动 3GPP Release 19 纳入分布式 Agents 网络接口规范，2029 年主导 ITU-T FG 6G 智能调度框架的国际标准制定[159][160]。产学研协同方面，建立联合实验室聚焦 GNN 与博弈论算法优化，每年举办技术挑战赛推动创新[129][164]。基础设施方面，2026 年在重点城市部署算力密度 1PFLOPS/km² 的边缘智能节点，2030 年建成覆盖全国的 6G 智能网络试验床 [174]。政策层面需推动分布式技术纳入国家 6G 专项规划，设立产业基金引导社会资本投入，建立跨国联合实验室推动技术互认[166][168]。

# 参考文献

1. Qiao S, Zhang Z, Shen W, et al. Gradually updated neural networks for large-scale image recognition//Proceedings of the International Conference on Machine Learning (ICML). Stockholm, Sweden, 2018: 4188-4197.
2. Xiong W, Wu L, Alleva F, et al. The Microsoft 2017 conversational speech recognition system//Proceedings of the IEEE international conference on acoustics, speech and signal processing (ICASSP). Calgary, Canada, 2018: 5934-5938.
3. He D, Lu H, Xia Y, et al. Decoding with value networks for neural machine translation//Proceedings of the Advances in Neural Informa tion Processing Systems (NeurIPS). Long Beach, USA, 2017: 178-187.
4. Bojarski M, Del Testa D, Dworakowski D, et al. End to end learning for self-driving cars. arXiv preprint arXiv:1604.07316, 2016.
5. Ben-Nun T, Hoefler T. Demystifying parallel and distributed deep learning: An in-depth concurrency analysis. ACM Computing Sur veys, 2019, 52(4): 1-43.
6. W. Ling, M. Chen, Z. Wei, F. Chen, L. Yu and D. C. Yu, "A distributed optimal control method for active distribution network," 2016 UKACC 11th International Conference on Control (CONTROL), Belfast, UK, 2016, pp. 1-6, doi: 10.1109/CONTROL.2016.7737568.
7. C. D. Rodríguez-Camargo, A. F. Urquijo-Rodríguez and E. Mojica-Nava, "Consensus-Based Distributed Optimization for MultiAgents Systems Over Multiplex Networks," in IEEE Transactions on Control of Network Systems, vol. 12, no. 1, pp. 1040-1051, March 2025, doi: 10.1109/TCNS.2024.3510602.
8. Boccaletti S, Bianconi G, Criado R, et al. The structure and dynamics of multilayer networks[J]. Physics reports, 2014, 544(1): 1-122.
9. Wu Y, Tang S, Yu C, et al. Task Scheduling in Geo-Distributed Computing: A Survey[J]. arXiv preprint arXiv:2501.15504, 2025.
10. Chen F, Ren W. On the control of multi-Agents systems: A survey[J]. Foundations and Trends® in Systems and Control, 2019, 6(4): 339-499.
11. Nedic A, Ozdaglar A. Distributed subgradient methods for multi-Agents optimization[J]. IEEE Transactions on Automatic Control, 2009, 54(1): 48-61.
12. Mnih V, Badia A P, Mirza M, et al. Asynchronous methods for deep reinforcement learning[C]//International conference on machine learning. PmLR, 2016: 1928-1937.
13. Wang Z, Goudarzi M, Buyya R. TF-DDRL: A Transformer-enhanced Distributed DRL Technique for Scheduling IoT Applications in Edge and Cloud Computing Environments[J]. IEEE Transactions on Services Computing, 2025.
14. Niu M, Cheng B, Feng Y, et al. GMTA: A geo-aware multi-Agents task allocation approach for scientific workflows in container-based cloud[J]. IEEE Transactions on Network and Service Management, 2020, 17(3): 1568-1581.
15. Hogade N, Pasricha S. Game-theoretic deep reinforcement learning to minimize carbon emissions and energy costs for AI inference workloads in geo-distributed data centers[J]. IEEE Transactions on Sustainable Computing, 2024.
16. B. J. Austin, V. Heine, and L. Sham, General theory of pseudopotentials, Phys. Rev. J. Arch., vol. 127, no. 1, p. 276, 1962.
17. R. Kibble, Speech acts, commitment and multi-agent communication, Comput. Math. Org. Theory, vol. 12, nos. 23, pp. 127145, 2006.
18. M. Colombetti and M. Verdicchio, An analysis of agent speech acts as institutional actions, in Proc. 1st Int. Joint Conf. Auto. Agents Multiagent Syst., 2002, pp. 1157 1164.
19. M. Wooldridge, An Introduction to Multiagent Systems. New York, NY, USA: Wiley, 2009.
20. P. Balaji and D. Srinivasan, An introduction to multi-agent systems, in Innovations in Multi-Agent Systems and Applications. Berlin, Germany: Springer, 2010, pp. 127.
21. Venkatesan, D. A Novel Agent-Based Enterprise Level System Development Technology. Ph.D. Thesis, Anna University, Tamil Nadu, India, 2018.
22. Bellifemine, F.; Bergenti, F.; Caire, G.; Poggi, A. JADE—A java agent development framework. In Multi-Agent Programming; Springer: Boston, MA, USA, 2005; pp. 125–147.
23. DeLoach, S.A.; Garcia-Ojeda, J.C. O-MaSE: A customisable approach to designing and building complex, adaptive multi-agent systems. Int. J. Agent-Oriented Softw. Eng. 2010, 4, 244–280.
24. Kantamneni, A.; Brown, L.E.; Parker, G.; Weaver, W.W. Survey of multi-agent systems for microgrid control. Eng. Appl. Artif. Intell. 2015, 45, 192–203.
25. Cardoso, R.C.; Ferrando, A. A Review of Agent-Based Programming for Multi-Agent Systems. Computers 2021, 10, 16.
26. Kravari, K.; Bassiliades, N. A survey of agent platforms. J. Artif. Soc. Soc. Simul. 2015, 18, 11.
27. Bordini, R.H.; El Fallah Seghrouchni, A.; Hindriks, K.; Logan, B.; Ricci, A. Agent programming in the cognitive era. Auton. Agents Multi-Agent Syst. 2020, 34, 37.
28. Costantini, S. ACE: A flexible environment for complex event processing in logical agents. In International Workshop on Engineering Multi-Agent Systems; Springer: Cham, Switzerland, 2015; pp. 70–91.
29. Araujo, P.; Rodríguez, S.; Hilaire, V. A metamodeling approach for the identification of organizational smells in multi-agent systems: Application to ASPECS. Artif. Intell. Rev. 2018, 49, 183–210.
30. Boissier, O.; Bordini, R.H.; Hübner, J.F.; Ricci, A. Dimensions in programming multi-agent systems. Knowl. Eng. Rev. 2019, 34, e2.
31. Rahimi, H.; Trentin, I.F.; Ramparany, F.; Boissier, O. SMASH: A Semantic-enabled Multi-agent Approach for Self-adaptation of Human-centered IoT. arXiv 2021, arXiv:2105.14915.
32. Baek, Y.M.; Song, J.; Shin, Y.J.; Park, S.; Bae, D.H. A meta-model for representing system-of-systems ontologies. In Proceedings of the 2018 IEEE/ACM 6th International Workshop on Software Engineering for Systems-of-Systems (SESoS), Gothenburg, Sweden, 29 May2018; pp. 1–7.
33. Pigazzini, I.; Briola, D.; Fontana, F.A. Architectural Technical Debt of Multiagent Systems Development Platforms. In Proceedings of the WOA2021: Workshop “From Objects to Agents”, Bologna, Italy, 1–3 September 2021.
34. Jazayeri, A.; Bass, E.J. Agent-Oriented Methodologies Evaluation Frameworks: A Review. Int. J. Softw. Eng. Knowl. Eng. 2020, 30, 1337–1370.
35. Logan, B. An agent programming manifesto. Int. J. Agent-Oriented Softw. Eng. 2018, 6, 187–210.
36. Zheng, Y.; Wang, L. Consensus of heterogeneous multi-agent systems without velocity measurements. Int. J. Control 2012, 85, 906–914.
37. Rizk,Y.; Awad, M.; Tunstel, E.W. Decision making in multiagent systems: A survey. IEEE Trans. Cogn. Dev. Syst. 2018, 10, 514–529.
38. Rossi, F.; Bandyopadhyay, S.; Wolf, M.T.; Pavone, M. Multi-Agent Algorithms for Collective Behavior: A structural and application-focused atlas. arXiv 2021, arXiv:2103.11067.
39. Chen, F.; Ren, W. Multi-Agent Control: A Graph-Theoretic Perspective. J. Syst. Sci. Complex. 2021, 34, 1973–2002.
40. Zelazo, D.; Rahmani, A.; Mesbahi, M. Agreement via the edge laplacian. In Proceedings of the 2007 46th IEEE Conference on Decision and Control, New Orleans, LA, USA, 12–14 December 2007; pp. 2309–2314.
41. You, K.; Xie, L. Network topology and communication data rate for consensusability of discrete-time multi-agent systems. IEEE Trans. Autom. Control 2011, 56, 2262–2275.
42. Shi, C.X.; Yang, G.H. Robust consensus control for a class of multi-agent systems via distributed PID algorithm and weighted edge dynamics. Appl. Math. Comput. 2018, 316, 73–88.
43. Wang, Q.; Wang, Y. Cluster synchronization of a class of multi-agent systems with a bipartite graph topology. Sci. China Inf. Sci. 2014, 57, 1–11.
44. Shoham, Y.; Leyton-Brown, K. Multiagent Systems: Algorithmic, Game-Theoretic, and Logical Foundations; Cambridge University Press: Cambridge, UK, 2008.
45. Zhu, M.; Martínez, S. Distributed coverage games for energy-aware mobile sensor networks. SIAM J. Control Optim. 2013, 51, 1–27.
46. Sengupta, S.; Kambhampati, S. Multi-agent reinforcement learning in bayesian stackelberg markov games for adaptive moving target defense. arXiv 2020, arXiv:2007.10457.
47. Sun, C.; Wang, X.; Liu, J. Evolutionary game theoretic approach for optimal resource allocation in multi-agent systems. In Proceedings of the 2017 Chinese Automation Congress (CAC), Jinan, China, 20–22 October 2017; pp. 5588–5592.
48. Yan, B.; Shi, P.; Lim, C.C.; Shi, Z. Optimal robust formation control for heterogeneous multi-agent systems based on reinforcement learning. Int. J. Robust Nonlinear Control 2021, 32, 2683–2704.
49. Wang,D.;Wang,Z.; Chen,M.; Wang,W. Distributed optimization for multi-agent systems with constraints set and communication time-delay over a directed graph. Inf. Sci. 2018, 438, 1–14.
50. Wai, H.T.; Yang, Z.; Wang, Z.; Hong, M. Multi-agent reinforcement learning via double averaging primal-dual optimization. arXiv 2018, arXiv:1806.00877.
51. Jian, L.; Zhao, Y.; Hu, J.; Li, P. Distributed inexact consensus-based ADMM method for multi-agent unconstrained optimization problem. IEEE Access 2019, 7, 79311–79319.
52. Kapoor, S. Multi-agent reinforcement learning: A report on challenges and approaches.
53. Busoniu, L.; Babuska, R.; De Schutter, B. A comprehensive survey of multiagent reinforcement learning. IEEE Trans. Syst. Man Cybern. Part C (Appl. Rev.) 2008, 38, 156–172.
54. Zhang, K.; Yang, Z.; Ba¸ sar, T. Decentralized multi-agent reinforcement learning with networked agents: Recent advances. arXiv 2019, arXiv:1912.03821.
55. Papoudakis, G.; Christianos, F.; Rahman, A.; Albrecht, S.V. Dealing with non-stationarity in multi-agent deep reinforcement learning. arXiv 2019, arXiv:1906.04737.
56. Lowe, R.; Wu, Y.; Tamar, A.; Harb, J.; Abbeel, P.; Mordatch, I. Multi-agent actor-critic for mixed cooperative-competitive environments. arXiv 2017, arXiv:1706.02275.
57. Rashid, T.; Samvelyan, M.; Schroeder, C.; Farquhar, G.; Foerster, J.; Whiteson, S. Qmix: Monotonic value function factorisation for deep multi-agent reinforcement learning. In Proceedings of the International Conference on Machine Learning, PMLR, Stockholm, Sweden, 10–15 July 2018; pp. 4295–4304
58. Zhang, K.; Yang, Z.; Ba¸ sar, T. Multi-agent reinforcement learning: A selective overview of theories and algorithms. In Handbook of Reinforcement Learning and Control; Springer: Cham, Switzerland, 2021; pp. 321–384.
59. Du, W.; Ding, S. A survey on multi-agent deep reinforcement learning: From the perspective of challenges and applications. Artif. Intell. Rev. 2021, 54, 3215–3238.
60. Mnih, V.; Badia, A.P.; Mirza, M.; Graves, A.; Lillicrap, T.; Harley, T.; Silver, D.; Kavukcuoglu, K. Asynchronous methods for deep reinforcement learning. In Proceedings of the International Conference on Machine Learning, PMLR, New York, NY, USA, 20–22 June 2016; pp. 1928–1937.
61. Lillicrap, T.P.; Hunt, J.J.; Pritzel, A.; Heess, N.; Erez, T.; Tassa, Y.; Silver, D.; Wierstra, D. Continuous control with deep reinforcement learning. arXiv 2015, arXiv:1509.02971.
62. Panait, L.; Luke, S. Cooperative multi-agent learning: The state of the art. Auton. Agents Multi-Agent Syst. 2005, 11, 387–434.
63. Foerster, J.; Farquhar, G.; Afouras, T.; Nardelli, N.; Whiteson, S. Counterfactual multi-agent policy gradients. In Proceedings of the AAAI Conference on Artificial Intelligence, New Orleans, LA, USA, 2–7 February 2018; Volume 32.
64. Da Silva, F.L.; Costa, A.H.R. A survey on transfer learning for multiagent reinforcement learning systems. J. Artif. Intell. Res. 2019, 64, 645–703.
65. Dusparic, I.; Cahill, V. Autonomic multi-policy optimization in pervasive systems: Overview and evaluation. ACM Trans. Auton. Adapt. Syst. (TAAS) 2012, 7, 1–25.
66. Sharma, P.K.; Fernandez, R.; Zaroukian, E.; Dorothy, M.; Basak, A.; Asher, D.E. Survey of recent multi-agent reinforcement learning algorithms utilizing centralized training. In Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications III; International Society for Optics and Photonics: Orlando, FL, USA, 2021; Volume 11746, p. 117462K.
67. Foerster, J.; Farquhar, G.; Afouras, T.; Nardelli, N.; Whiteson, S. Counterfactual multi-agent policy gradients. In Proceedings of the AAAI Conference on Artificial Intelligence, New Orleans, LA, USA, 2–7 February 2018; Volume 32.
68. De Lemos, R.; Giese, H.; Müller, H.A.; Shaw, M.; Andersson, J.; Litoiu, M.; Schmerl, B.; Tamura, G.; Villegas, N.M.; Vogel, T.; et al. Software engineering for self-adaptive systems: A second research roadmap. In Software Engineering for Self-Adaptive Systems II; Springer: Heidelberg, Germany, 2013; pp. 1–32.
69. Panait, L.; Luke, S. Cooperative multi-agent learning: The state of the art. Auton. Agents Multi-Agent Syst. 2005, 11, 387–434.
70. Rashid, T.; Samvelyan, M.; Schroeder, C.; Farquhar, G.; Foerster, J.; Whiteson, S. Qmix: Monotonic value function factorisation for deep multi-agent reinforcement learning. In Proceedings of the International Conference on Machine Learning, PMLR, Stockholm, Sweden, 10–15 July 2018; pp. 4295–4304.
71. Chen, G. A NewFrameworkfor Multi-Agent Reinforcement Learning–Centralized Training and Exploration with Decentralized Execution via Policy Distillation. arXiv 2019, arXiv:1910.09152.
72. Pesce, E.; Montana, G. Improving coordination in small-scale multi-agent deep reinforcement learning through memory-driven communication. Mach. Learn. 2020, 109, 1727–1747.
73. Czarnowski, I.; J˛edrzejowicz, P. An agent-based framework for distributed learning. Eng. Appl. Artif. Intell. 2011, 24, 93–102.
74. D’Angelo, M. Engineering Decentralized Learning in Self-Adaptive Systems. Ph.D. Thesis, Linnaeus University Press, Vaxjo, Sweden, 2021.
75. J. Bajo, F. De la Prieta, J. M. Corchado, and S. Rodríguez, A low-level resource allocation in an Agents-based cloud computing platform, Appl. Soft Comput., vol. 48, pp. 716728, Nov. 2016.
76. J. Fiosina and M. Fiosins, Density-based clustering in cloud-oriented collaborative multi-agent systems, inProc.Int.Conf.HybridArtif.Intell. Syst., 2013, pp. 639648.
77. J. O. Gutierrez-Garcia and K. M. Sim, Agent-based cloud bag-of-tasks execution, J. Syst. Softw., vol. 104, pp. 1731, Jun. 2015.
78. A. Singh, D. Juneja, and M. Malhotra, Autonomous agent based load balancing algorithm in cloud computing, Proc. Comput. Sci., vol. 45, pp. 832841, Jan. 2015.
79. K.Govinda and E. Sathiyamoorthy, Agent based security for cloud computing using obfuscation, Proc. Eng., vol. 38, pp. 125129, Jan. 2012.
80. R. Nikbazm and M. Ahmadi, Agent-based resource discovery in cloud computing using bloom lters, in Proc. 4th Int. eConf. Comput. Knowl. Eng. (ICCKE), Oct. 2014, pp. 352357.
81. F. Hou and X. Mao, Cross-clouds services autonomic management approach based on self-organizing multi-agent technology, Concurrency Comput., Pract. Exper., vol. 28, no. 11, pp. 32133237, 2016.
82. K.M.Sim, Agent-based cloud computing, IEEETrans.Serv. Comput., vol. 5, no. 4, pp. 564577, Oct. 2012.
83. Y. Jiang and J. C. Jiang, Understanding social networks from a multiagent perspective, IEEE Trans. Parallel Distrib. Syst., vol. 25, no. 10, pp. 27432759, Oct. 2014.
84. M. Gatti et al., Large-scale multi-agent-based modeling and simulation of microblogging-based online social network, in Proc. Int. Workshop Multi-Agent Syst. Agent-Based Simulation, 2013, pp. 1733.
85. L. Ma and Y. Zhang, Hierarchical social network analysis using multiagent systems: Aschoolsystemcase, inProc.IEEEInt.Conf.Syst.,Man Cybern. (SMC), Oct. 2014, pp. 14121419.
86. V. Gorodetski and I. Kotenko, The multi-agent systems for computer network security assurance: Framework sand case studies, inProc.IEEE Int. Conf. Artif. Intell. Syst. (ICAIS), Sep. 2002, pp. 297302.
87. L.Mechtri, F. D. Tolba, and S. Ghanemi, Masid: Multi-agent system for intrusion detection in MANET, in Proc. 9th Int. Conf. Inf. Technol., New Generat. (ITNG), Apr. 2012, pp. 6570.
88. S. Sarika and V. Paul, AgentTab: An agent based approach to detect tabnabbing attack, Proc. Comput. Sci., vol. 46, pp. 574581, Jan. 2015.
89. S.Shamshirband, N.B.Anuar, M.L.M.Kiah, and A.Patel, Anappraisal and design of a multi-agent system based cooperative wireless intrusion detection computational intelligence technique, Eng. Appl. Artif. Intell., vol. 26, no. 9, pp. 21052127, 2013.
90. G. Di Caro and M. Dorigo, An adaptive multi-agent routing algorithm inspired by ants behavior, in Proc. 5th Annu. Austral. Conf. Parallel Real-Time Syst. (PART), 1998, pp. 261272.
91. R. Claes, T. Holvoet, and D. Weyns, A decentralized approach for anticipatory vehicle routing using delegate multiagent systems, IEEE Trans. Intell. Transp. Syst., vol. 12, no. 2, pp. 364373, Feb. 2011.
92. M. Bendjima and M. Feham, Multi-agent system for a reliable routing in WSN, inProc. Sci. Inf. Conf. (SAI), Jul. 2015, pp. 14121419.
93. S. Manvi and M. Kakkasageri, Multicast routing in mobile ad hoc networks by using a multiagent system, Inf. Sci., vol. 178, no. 6, pp. 16111628, 2008.
94. G.Dudek,M.R.Jenkin,E.Milios,andD.Wilkes, Ataxonomyformulti agent robotics, Auto. Robot., vol. 3, no. 4, pp. 375397, 1996.
95. C.G.Cena,P. F. Cardenas, R. S. Pazmino, L. Puglisi, and R. A. Santonja, A cooperative multi-agent robotics system: Design and modelling, Expert Syst. Appl., vol. 40, no. 12, pp. 47374748, 2013.
96. Y. Duan, B. X.Cui, and X.H.Xu, Amulti-agentreinforcement learning approach to robot soccer, Artif. Intell. Rev., vol. 38, no. 3, pp. 193211, 2012.
97. J. Ota, Multi-agent robot systems as distributed autonomous systems, Adv. Eng. Informat., vol. 20, no. 1, pp. 5970, 2006.
98. P. Inigo-Blasco, F. Diaz-del-Rio, M. C. Romero-Ternero, D. Cagigas-Muñiz, and S. Vicente-Diaz, Robotics software frameworks for multi-agent robotic systems development, Robot. Auto. Syst., vol. 60, no. 6, pp. 803 821, 2012.
99. A. Soriano, E. J. Bernabeu, A. Valera, and M. Vallès, Multi-agent systems platform for mobile robots collision avoidance, in Proc. Int. Conf. Practical Appl. Agents Multi-Agent Syst., 2013, pp. 320323.
100. D. Helbing, Agent-based modeling, in Social self-Organization. Berlin, Germany: Springer, 2012, pp. 2570.
101. R.Domínguez, S.Cannella,andJ.M.Framinan, SCOPE:Amulti-agent system tool for supply chain network analysis, in Proc. IEEE Int. Conf. Comput. (EUROCON), Sep. 2015, pp. 15.
102. O. Wangapisit, E. Taniguchi, J. S. Teo, and A. G. Qureshi, Multi-agent systems modelling for evaluating joint delivery systems, Proc.-Social Behavioral Sci., vol. 125, pp. 472483, Mar. 2014.
103. S. D. J. McArthur et al., Multi-agent systems for power engineering applications Part I: Concepts, approaches, and technical challenges, IEEE Trans. Power Syst., vol. 22, no. 4, pp. 17431752, Nov. 2007.
104. P. Ringler, D. Keles, and W. Fichtner, Agent-based modelling and simulation of smartelectricity grids andmarkets Aliteraturereview, Renew. Sustain. Energy Rev., vol. 57, pp. 205215, 2016.
105. M. Khayyat and A. Awasthi, An intelligent multi-agent based model for collaborative logistics systems, Transp. Res. Procedia, vol. 12, pp. 325338, Jan. 2016.
106. K. Hager, J. Rauh, and W. Rid, Agent-based modeling of traf c behavior in growing metropolitan areas, Transp. Res. Procedia, vol. 10, pp. 306315, Jan. 2015.
107. O. van Pruissen, A. van der Togt, and E. Werkman, Energy ef ciency comparison of a centralized and a multi-agent market based heating system in a eld test, Energy Procedia, vol. 62, pp. 170179, Jan. 2014.
108. J. Cai, D. Kim, R. Jaramillo, J. E. Braun, and J. Hu, A general multi agent control approach for building energy system optimization, Energy Buildings, vol. 127, pp. 337351, Sep. 2016. C. P. Nguyen and A. J. Flueck, Agent based restoration with distributed
109. energy storage support in smart grids, IEEE Trans. Smart Grid, vol. 3, no. 2, pp. 10291038, Jun. 2012.
110. P. Vytelingum, T. D. Voice, S. D. Ramchurn, A. Rogers, and N. R. Jennings, Agent-based micro-storage management for the smart grid, in Proc. 9th Int. Conf. Auto. Agents Multiagent Syst., 2010, pp. 3946.
111. G.H.Merabetetal., Applications of multi-agent systems in smart grids: Asurvey, in Proc. Int. Conf. Multimedia Comput. Syst. (ICMCS), 2014, pp. 10881094.
112. Geng J, Li D, Cheng Y, et al. HiPS: Hierarchical parameter synchronization in large-scale distributed machine learning[C]//Proceedings of the 2018 Workshop on Network Meets AI & ML. 2018: 1-7.
113. Hashemi S H, Abdu Jyothi S, Campbell R. Tictac: Accelerating distributed deep learning with communication scheduling[J]. Proceedings of Machine Learning and Systems, 2019, 1: 418-430.
114. Shi S, Chu X, Li B. MG-WFBP: Efficient data communication for distributed synchronous SGD algorithms[C]//IEEE INFOCOM 2019-IEEE Conference on Computer Communications. IEEE, 2019: 172-180.
115. 王帅, 李丹. 分布式机器学习系统网络性能优化研究进展[J]. 计算机学报, 2022, 45(7): 1384-1411.
116. Wang S, Li D, Geng J. Geryon: Accelerating distributed CNN training by network-level flow scheduling[C]//IEEE INFOCOM 2020-IEEE Conference on Computer Communications. IEEE, 2020: 1678-1687.
117. 黄金超,谢志普,吕非彼,等. 基于大小模型协同的智能化移动网络优化研究[J]. 邮电设计技术,2024(9):7-12. DOI:10.12045/j.issn.1007-3043.2024.09.002.
118. Zhao H, Pan F, Ping H, et al. Agent as cerebrum, controller as cerebellum: Implementing an embodied lmm-based agent on drones[J]. arXiv preprint arXiv:2311.15033, 2023.
119. Frasheri M, Miloradović B, Esterle L, et al. GLocal: A hybrid approach to the multi-agent mission re-planning problem[C]//2023 IEEE Symposium Series on Computational Intelligence (SSCI). IEEE, 2023: 1696-1703.
120. 郑逢斌, 朱东伟, 臧文乾, 等. 边缘计算: 新型计算范式综述与应用研究[J]. 计算机科学与探索, 2020, 14(4): 541.
121. Satyanarayanan M, Bahl P, Caceres R, et al. The case for vm-based cloudlets in mobile computing[J]. IEEE pervasive Computing, 2009, 8(4): 14-23.
122. Jang M, Schwan K, Bhardwaj K, et al. Personal clouds: Sharing and integrating networked resources to enhance end user experiences[C]//IEEE INFOCOM 2014-IEEE Conference on Computer Communications. IEEE, 2014: 2220-2228.
123. Liu P, Willis D, Banerjee S. Paradrop: Enabling lightweight multi-tenancy at the network’s extreme edge[C]//2016 IEEE/ACM Symposium on Edge Computing (SEC). IEEE, 2016: 1-13.
124. Willis D, Dasgupta A, Banerjee S. Paradrop: a multi-tenant platform to dynamically install third party services on wireless gateways[C]//Proceedings of the 9th ACM workshop on Mobility in the evolving internet architecture. 2014: 43-48.
125. S. Zhang, J. Liu, H. Guo, M. Qi, and N. Kato, “Envisioning device-todevice communications in 6G,” IEEE Netw., vol. 34, no. 3, pp. 86–91, May 2020.
126. M. S. M. Gismalla, A. I. Azmi, M. R. B. Salim, M. F. L. Abdullah, F. Iqbal, W. A. Mabrouk, M. B. Othman, A. Y. I. Ashyap, and A. S. M. Supa’at, “Survey on device to device (D2D) communication for 5GB/6G networks: Concept, applications, challenges, and future directions,” IEEE Access, vol. 10, pp. 30 792–30 821, Mar. 2022.
127. S. Hosseinalipour, C. G. Brinton, V. Aggarwal, H. Dai, and M. Chiang, “From federated to fog learning: Distributed machine learning over heterogeneous wireless networks,” IEEE Commun. Mag., vol. 58, no. 12, pp. 41–47, Dec. 2020.
128. W. Fan, Z. Yu, C. Ma, C. Li, Y. Yang, and X. Zhang, “Towards efficient collaboration via graph modeling in reinforcement learning,” arXiv preprint arXiv:2410.15841, 2024.
129. Z. Liu, J. Zhang, E. Shi, Z. Liu, D. Niyato, B. Ai, and X. Shen, “Graph neural network meets multi-agent reinforcement learning: Fundamentals, applications, and future directions,” IEEE Wireless Commun., vol. 31, no. 6, pp. 39–47, Dec. 2024.
130. X. Cao, T. Ba ̧sar, S. Diggavi, Y. C. Eldar, K. B. Letaief, H. V. Poor, and J. Zhang, “Communication-efficient distributed learning: An overview,” IEEE J. Sel. Areas Commun., vol. 41, no. 4, pp. 851–873, Apr. 2023.
131. W. Li, B. Jin, X. Wang, J. Yan, and H. Zha, “F2a2: Flexible fullydecentralized approximate actor-critic for cooperative multi-agent reinforcement learning,” J. Mach. Learn. Res., vol. 24, no. 178, pp. 1–75, Jun. 2023.
132. N. Tishby and N. Zaslavsky, “Deep learning and the information bottleneck principle,” in IEEE Inf. Theory Workshop, 2015, pp. 1–5.
133. A. A. Alemi, I. Fischer, J. V. Dillon, and K. Murphy, “Deep variational information bottleneck,” Proc. Int. Conf. Learn. Represent., 2017.
134. T. Wu, H. Ren, P. Li, and J. Leskovec, “Graph information bottleneck,” Proc. Adv. Neural Inf. Process. Syst., vol. 33, pp. 20 437–20 448, 2020.
135. Q. Sun, J. Li, H. Peng, J. Wu, X. Fu, C. Ji, and S. Y. Philip, “Graph structure learning with variational information bottleneck,” in Proc. AAAI Conf. Artif. Intell., vol. 36, no. 4, 2022, pp. 4165–4174.
136. J. Yu, T. Xu, Y. Rong, Y. Bian, J. Huang, and R. He, “Recognizing predictive substructures with subgraph information bottleneck,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 46, no. 3, pp. 1650–1663, Mar. 2024.
137. J. Yu, J. Cao, and R. He, “Improving subgraph recognition with variational graph information bottleneck,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2022, pp. 19 396–19 405.
138. M. Giordani, M. Polese, M. Mezzavilla, S. Rangan, and M. Zorzi, “Toward 6G networks: Use cases and technologies,” IEEE Commun. Mag., vol. 58, no. 3, pp. 55–61, Mar. 2020.
139. L. Kraemer and B. Banerjee, “Multi-agent reinforcement learning as a rehearsal for decentralized planning,” Neurocomputing, vol. 190, pp. 82–94, May 2016.
140. S. Gronauer and K. Diepold, “Multi-agent deep reinforcement learning: a survey,” Artif. Intell. Rev., vol. 55, no. 2, pp. 895–943, Apr. 2021.
141. W. Du and S. Ding, “A survey on multi-agent deep reinforcement learning: from the perspective of challenges and applications,” Artif. Intell. Rev., vol. 54, no. 5, pp. 3215–3238, Nov. 2021.
142. C. Amato, “(a partial survey of) decentralized, cooperative multi-agent reinforcement learning,” arXiv preprint arXiv:2405.06161, 2024.
143. S. Zhu, G. Han, and C. Lin, “A software-defined MARL-based architecture for AUV cluster network to enable cooperative and smart underwater target tracking,” IEEE Wireless Commun., vol. 31, no. 6, pp. 56–62, Dec. 2024.
144. C. Amato, “An introduction to centralized training for decentralized execution in cooperative multi-agent reinforcement learning,” arXiv preprint arXiv:2409.03052, 2024.
145. D. Huh and P. Mohapatra, “Multi-agent reinforcement learning: A comprehensive survey,” arXiv preprint arXiv:2312.10256, 2023.
146. J. Jiang, K. Su, and Z. Lu, “Fully decentralized cooperative multi-agent reinforcement learning: A survey,” arXiv preprint arXiv:2401.04934, 2024.
147. G. Wen, J. Fu, P. Dai, and J. Zhou, “DTDE: A new cooperative multiagent reinforcement learning framework,” The Innovation, vol. 2, no. 4, Nov. 2021.
148. Y. Shin, S. Seo, B. Yoo, H. Kim, H. Song, and S. Yi, “Survey on recent advances in multiagent reinforcement learning focusing on decentralized training with decentralized execution framework,” Electronics and Telecommunications Trends, vol. 38, no. 4, pp. 95103, Aug. 2023.
149. M. Suciu, R. I. Lung, N. Gasko ́, and D. Dumitrescu, “Differential evolution for discrete-time large dynamic games,” in IEEE Congress on Evolutionary Computation, 2013, pp. 2108–2113.
150. Y. Hu, Y. Gao, and B. An, “Accelerating multiagent reinforcement learning by equilibrium transfer,” IEEE Trans. Cybern., Jul.
151. Y. Zhong, J. G. Kuba, X. Feng, S. Hu, J. Ji, and Y. Yang, “Heterogeneous-agent reinforcement learning,” J. Mach. Learn. Res., vol. 25, no. 1-67, p. 1, Jan. 2024.
152. Z. Dai, Y. Zhang, W. Zhang, X. Luo, and Z. He, “A multi-agent collaborative environment learning method for UAV deployment and resource allocation,” IEEE Trans. Signal Inf. Process. Netw., vol. 8, pp. 120–130, Feb. 2022.
153. Xia, Dan, Jiafu Wan, Pengpeng Xu, and Jinbiao Tan. "Deep reinforcement learning-based QoS optimization for software-defined factory heterogeneous networks." IEEE Transactions on Network and Service Management 19, no. 4 (2022): 4058-4068.
154. Guo, Yingya, Yulong Ma, Huan Luo, and Jianping Wu. "Traffic Engineering in a Shared Inter-DC WAN via Deep Reinforcement Learning." IEEE Transactions on Network Science and Engineering 9, no. 4 (2022): 2870-2881.
155. U. U. Mohammed, A. J. Mohammed, M. F. Ansari, and B. Dash, “AI-Based Dynamic Bandwidth Allocation in WAN: An Approach to Efficient Resource Utilization,” in Proc. AI-Based Dynamic Bandwidth Allocation in WAN: An Approach to Efficient Resource Utilization , October 2024.
156. E. Shi, J. Zhang, H. Du, B. Ai, C. Yuen, D. Niyato, K. B. Letaief, and X. Shen, “RIS-aided cell-free massive MIMO systems for 6G: Fundamentals, system design, and applications,” Proc. IEEE, vol. 112, no. 4, pp. 331–364, Apr. 2024.
157. R. Zhang, J. Hou, F. Walter, S. Gu, J. Guan, F. R ̈ohrbein, Y. Du, P. Cai, G. Chen, and A. Knoll, “Multi-agent reinforcement learning for autonomous driving: A survey,” arXiv preprint arXiv:2408.09675, 2024.
158. Y. Bai, H. Zhao, X. Zhang, Z. Chang, R. J ̈antti, and K. Yang, “Toward autonomous multi-UAV wireless network: A survey of reinforcement learning-based approaches,” IEEE Commun. Surveys Tuts., vol. 25, no. 4, pp. 3038–3067, Fourthquarter 2023.
159. H. Yao, L. Wang, X. Wang, Z. Lu, and Y. Liu, “The space-terrestrial integrated network: An overview,” IEEE Commun. Mag., vol. 56, no. 9, pp. 178–185, 2018.
160. H. Q. Ngo, A. Ashikhmin, H. Yang, E. G. Larsson, and T. L. Marzetta, “Cell-free massive MIMO versus small cells,” IEEE Trans. Wireless Commun., vol. 16, no. 3, pp. 1834–1850, Mar. 2017.
161. Q. Wu and R. Zhang, “Intelligent reflecting surface enhanced wireless network via joint active and passive beamforming,” IEEE Trans. Wireless Commun., vol. 18, no. 11, pp. 5394–5409, Nov. 2019.
162. K.-K. Wong, A. Shojaeifard, K.-F. Tong, and Y. Zhang, “Fluid antenna systems,” IEEE Trans. Wireless Commun., vol. 20, no. 3, pp. 19501962, Mar. 2021.
163. Z. Liu, J. Zhang, Y. Zhu, E. Shi, and B. Ai, “Mobile cell-free massive MIMO with multi-agent reinforcement learning: A scalable framework,” IEEE Trans. Wireless Commun., early access, 2024.
164. Q. Hao, W. Huang, T. Feng, J. Yuan, and Y. Li, “Gat-mf: Graph attention mean field for very large scale multi-agent reinforcement learning,” in Proc. ACM SIGKDD Intl. Conf. Knowledge Discovery & Data Mining, 2023, pp. 685–697.
165. H. Xie, Z. Qin, G. Y. Li, and B.-H. Juang, “Deep learning enabled semantic communication systems,” IEEE Trans. Signal Process., vol. 69, pp. 2663–2675, Apr. 2021.
166. F. Zenke, B. Poole, and S. Ganguli, “Continual learning through synaptic intelligence,” in Proc. Int. Conf. Mach. Learn., 2017.
167. J. Park, S. Samarakoon, A. Elgabli, J. Kim, M. Bennis, S.-L. Kim, and M. Debbah, “Communication-efficient and distributed learning over wireless networks: Principles and applications,” Proc. IEEE, vol. 109, no. 5, pp. 796–819, May 2021.
168. M. Chen, D. Gu ̈nd ̈uz, K. Huang, W. Saad, M. Bennis, A. V. Feljan, and H. V. Poor, “Distributed learning in wireless networks: Recent progress and future challenges,” IEEE J. Sel. Areas Commun., vol. 39, no. 12, pp. 3579–3605, Dec. 2021.
169. M. Haus, M. Waqas, A. Y. Ding, Y. Li, S. Tarkoma, and J. Ott, “Security and privacy in device-to-device (D2D) communication: A review,” IEEE Commun. Surveys Tuts., vol. 19, no. 2, pp. 1054–1079, Secondquarter 2017.
170. S. Zhou, Y. Cheng, X. Lei, Q. Peng, J. Wang, and S. Li, “Resource allocation in UAV-assisted networks: A clustering-aided reinforcement learning approach,” IEEE Trans. Veh. Technol., vol. 71, no. 11, pp. 12 088–12 103, Nov. 2022.
171. Z. Liu, J. Zhang, Z. Liu, D. W. K. Ng, and B. Ai, “Joint cooperative clustering and power control for energy-efficient cell-free XL-MIMO with multi-agent reinforcement learning,” IEEE Trans. Commun., vol. 72, no. 12, pp. 7772–7786, Dec. 2024.
172. B. Shang, Y. Yi, and L. Liu, “Computing over space-air-ground integrated networks: Challenges and opportunities,” IEEE Netw., vol. 35, no. 4, pp. 302–309, Aug. 2021.
173. W. Tang, J. Y. Dai, M. Z. Chen, K.-K. Wong, X. Li, X. Zhao, S. Jin, Q. Cheng, and T. J. Cui, “MIMO transmission through reconfigurable intelligent surface: System design, analysis, and implementation,” IEEE J. Sel. Areas Commun., vol. 38, no. 11, pp. 2683–2699, Nov. 2020.
174. J. Zheng, J. Zhang, H. Du, D. Niyato, B. Ai, M. Debbah, and K. B. Letaief, “Mobile cell-free massive MIMO: Challenges, solutions, and future directions,” IEEE Wireless Commun., vol. 31, no. 3, pp. 140147, Jun. 2024.
175. Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, “A survey on mobile edge computing: The communication perspective,” IEEE Commun. Surveys Tuts., vol. 19, no. 4, pp. 2322–2358, Fourthquarter 2017.
176. L. Claussmann, M. Revilloud, D. Gruyer, and S. Glaser, “A review of motion planning for highway autonomous driving,” IEEE Trans. Intell. Transp. Syst., vol. 21, no. 5, pp. 1826–1848, May 2020.
177. C. Xu, C. Zhang, Y. Shi, R. Wang, S. Duan, Y. Wan, and X. Zhang, “Subgoal-based hierarchical reinforcement learning for multi-agent collaboration,” arXiv preprint arXiv:2408.11416, 2024.
178. Y. Zhang, Z. Yu, J. Zhang, L. Wang, T. H. Luan, B. Guo, and C. Yuen, “Learning decentralized traffic signal controllers with multi-agent graph reinforcement learning,” IEEE Trans. Mobile Comput., vol. 23, no. 6, pp. 7180–7195, Jun. 2024.
179. S. Dohare, J. F. Hernandez-Garcia, Q. Lan, P. Rahman, A. R. Mahmood, and R. S. Sutton, “Loss of plasticity in deep continual learning,” Nature, vol. 632, no. 8026, pp. 768–774, Aug. 2024.
180. D. Serghiou, M. Khalily, T. W. C. Brown, and R. Tafazolli, “Terahertz channel propagation phenomena, measurement techniques and modeling for 6G wireless communication applications: A survey, open challenges and future research directions,” IEEE Commun. Surveys Tuts., vol. 24, no. 4, pp. 1957–1996, Fourthquarter 2022.
181. Z. Liu, J. Zhang, Y. Zhu, E. Shi, and B. Ai, “Robust multidimensional graph neural networks for signal processing in wireless communications with edge-graph information bottleneck,” arXiv preprint arXiv:2502.15836, 2025.
182. S. Ding, W. Du, L. Ding, J. Zhang, L. Guo, and B. An, “Robust multiagent communication with graph information bottleneck optimization,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 46, no. 5, pp. 3096–3107, May 2024.