卫生信息化国际发展动态

北京市卫生健康大数据与政策研究中心 2025.3.25

（二）多智能体

1.标题：利用多智能体对话式大语言模型增强医疗诊断能力

来源：NPJ Digital Medicine.

时间：2025 年 2 月.

链接：https://doi.org/10.1038/s41746-025-01550-0.

概要：大语言模型（LLM）最近在医学领域的应用显著增多，现已可以利用超大医学数据库和先进的分析算法给简单医疗任务提出建设性解决方案，但是在处理更实际和复杂的医疗任务时有效性上还有待进一步测试。罕见病的诊断是最复杂和最具挑战性的诊断任务之一。罕见病的低患病率往往导致专业知识的匮乏，使准确诊断变得困难，从而误诊或延误了适当治疗。多智能体系统是多个数字智能体在大型语言模型中一起工作并通过它们之间的交互执行任务，可以显著增强 LLM 管理复杂任务的能力。采用多智能体框架可以通过多智能体对话（MAC）促进动态和交互式诊断过程，不同的智能体可以模拟临床实践中通常采用的多学科团队（MDT）讨论，帮助医生分析患者的复杂信息并提供有价值的建议。本研究将开发一个用于疾病诊断的多智能体对话（MAC）框架，并使用 302 例罕见病病例来评估 GPT-3.5、GPT-4 和 MAC 对医学知识和临床推理的影响。MAC 在初次和随访咨询中均优于单一模型，而在诊断和建议测试方面取得了更高的准确性。使用 GPT-4 作为基本模型，利用 4 个医生智能体和一个监督智能体实现最佳性能。MAC 在重复运行中表现出了高度一致性。进一步的比较分析表明，MAC 的性能也优于其他方法，包括思维链（CoT）、自细化和自洽性，具有更高的性能和更多的输出。该框架显著提高了 LLM 的诊断能力，有效地衔接了理论知识和实际临床应用。我们的研究结果强调了多代理 LLM 在医疗保健中的潜力，并建议对其临床实施进行进一步研究。

2.标题：基于多智能体深度强化学习的机械臂组装研究

来源：PLOS ONE.

时间：2025 年 2 月.

链接：https://doi.org/10.1371/journal.pone.0311550.

概要：机器人技术的进步使机器人在工业自动化领域的应用越来越广泛。轴孔装配是机械制造中的一项关键任务，需要高度精确和高效的控制方法，以确保质量和生产率。这项任务本质上是多变量的，需要同时控制机械臂的位置和方向。传统的轴孔装配控制方法，如模型预测控制或力/位置混合控制，依赖于模型参数和控制设置的事先确定。但是，由于应用场景的复杂性和可变性以及装配需求的不断增长，单智能体算法在机械臂装配过程中往往面临收敛性挑战，表现出较差的性能。DRL 代表了一种新兴的智能控制方法，从根本上说，它基于通过与环境的试错交互来学习当前状态和预定义目标。该系统自主优化决策并执行最佳作。深度强化学习（DRL）已在机器人控制领域得到广泛应用，并在机器人抓取、路径规划和焊接等领域取得了重大成果。在轴孔装配方面，轴孔形状和尺寸的复杂性和可变性使机械臂的运动控制构成了一项具有挑战性的任务。单智能体 DRL 算法可以有效地处理相对简单的组装任务，但是对于更复杂的轴孔组件，则需要更精确地控制位置和方向。多智能体深度强化学习（MADRL）是一种支持多智能体间协作的方法，通过组建一组智能体来协作完成任务，从而扩展单智能体强化学习。它在机械臂控制中的应用主要涉及协同控制和任务分配。虽然多个智能体之间的协作简化了复杂的任务，提高了控制精度和效率，但是智能体的数量增加可能带来数据维度爆炸、通信困难、收敛问题及机械臂关节和连杆间强耦合控制更加复杂等挑战。为解决上述问题，本文提出了一种采用多智能体强化学习算法的机械臂轴孔装配方法，并特别关注方形轴孔的装配。首先，研究者在全面研究轴孔相互作用的基础上，分析了轴孔装配过程中的寻孔、校中和插入阶段；其次，通过整合解耦多智能体确定性深度确定性策略梯度（DMDDPG）算法设计奖励函数；最后，在 Gazebo 中创建仿真环境，以圆形和方形轴孔为实验对象，对机械臂轴孔总成进行建模。仿真结果表明，将机械臂前三个关节和后三个关节建模为多智能体的算法不仅具有较强的自适应性，而且收敛速度更快、更稳定。

（徐健编辑）

译文一：

利用多智能体对话式大语言模型增强医疗诊断能力

Xi Chen, Huahui Yi, Mingke You, WeiZhi Liu, Li Wang, Hairui Li, Xue Zhang, Yingman Guo, Lei Fan, Gang Chen, Qicheng Lao, Weili Fu, Kang Li, Jian Li，徐健（译）

介绍

大型语言模型（LLM）的最新进展显著增强了其在医学领域的能力，从而增加了对其潜在应用的探索。这些模型使用了超大医学数据库和先进的分析算法，为这些挑战提供了建设性的解决方案。这些模型已经显示出对简单医疗任务的熟练程度，例如回答医学知识查询和诊断常见疾病，并需要进一步研究以测试它们在处理更实际和复杂的医疗任务方面的有效性。其中，提供准确的诊断一直是一项重要且实际的医疗需求。

罕见病的诊断是最复杂和最具挑战性的诊断任务之一。罕见病的患病率为每100,000人5-76例。这些疾病的低患病率往往导致专业知识的匮乏，使准确诊断变得困难，从而延误了适当的治疗。此外，症状的复杂性和可变性可能导致频繁的误诊或延迟诊断。

尽管GPT-4等LLM在医学知识和一些潜在应用方面表现出相当的熟练程度，但它们在复杂的现实世界临床场景（如疾病诊断）中的表现仍然存在疑问。先前的一项研究表明LLM在诊断复杂病例方面的局限性。因此，越来越多的研究集中在如何更好地利用LLM自身的训练数据和固有能力来提高他们在实际医疗任务中的表现。

多智能体系统就是这样一种尝试。在大型语言模型的上下文中，智能体指的是能够接收输入并执行操作以实现特定目标的系统。例如，当与ChatGPT交互时，用户使用的是单智能体模型。多智能体框架是一种创新的方法，其中多个数字智能体一起工作并通过它们之间的交互执行任务。这种技术显著增强了LLM管理复杂任务的能力，包括解决数学问题和执行检索增强代码生成。采用多智能体框架可以通过多智能体对话（MAC）促进动态和交互式诊断过程，智能体将讨论相同的问题并最终就输出达成共识。在这种情况下，不同的智能体可以模拟医疗团队的协作性质，模拟临床实践中通常采用的多学科团队（MDT）讨论来解决复杂的医疗任务。通过促进

单药模型可能无法实现的深入分析，它们有可能提高 LLM 在罕见病诊断方面的性能。因此，MAC 可能是一种有价值的工具，可以帮助医生分析患者信息并提供有用的第二意见。

本研究旨在开发一个多智能体对话（MAC）框架，该框架可用于临床实践以执行诊断任务。本研究还比较了 GPT-3.5、GPT-4 和 MAC 的知识库和诊断能力。

结果

研究流程如图 2 所示。开发了多智能体对话框架（图 2）并在模拟真实世界临床咨询的精选临床病例上进行了测试（图 3）。

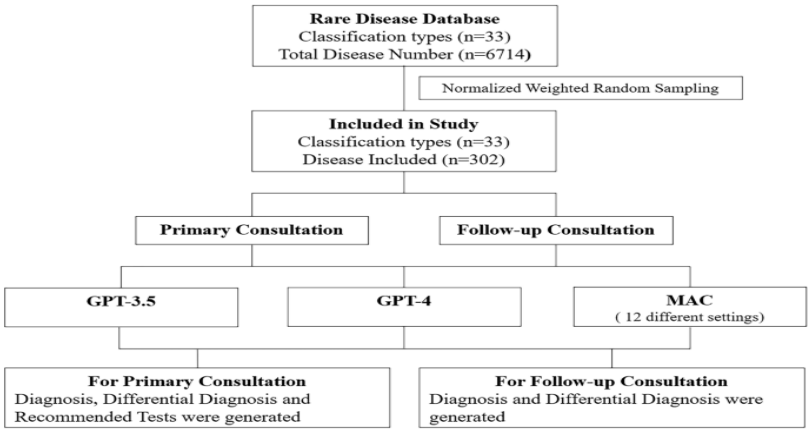

图 1 疾病选择和测试的过程

研究人员首先确定了罕见病数据库，然后进行标准化加权随机抽样，从中选择 302 例。每个病例被纳入初级会诊环境和随访会诊环境。对 GPT-3.5、GPT-4 和 MAC 进行测试。

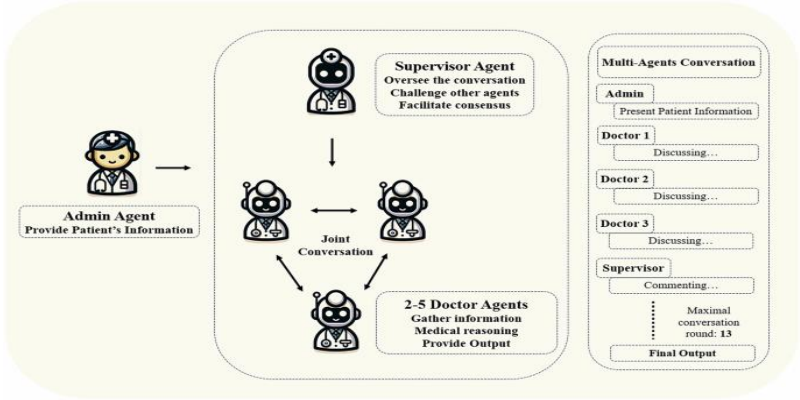

图 2：多智能体对话（MAC）框架

管理员智能体提供患者的信息。主管智能体发起并监督联合对话。三名医生智

能体人共同讨论患者的病情。最终输出在达成共识时或在最多 13 轮对话后提供。

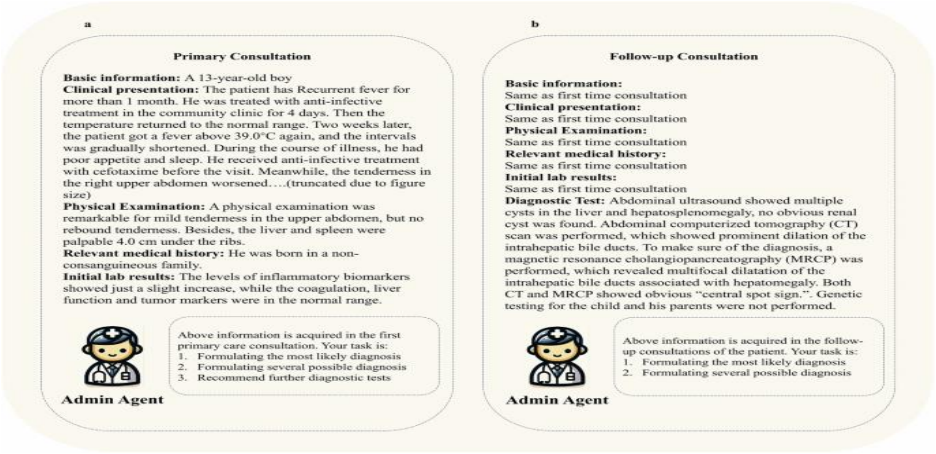

图 3 临床病例管理

a 初步咨询代表从初级保健患者那里获得的初始信息;

b 随访咨询是指患者在进行相关诊断测试后的完整信息。

研究样本

本研究包括来自 33 个不同疾病类别的 302 种罕见病。每个类别随机选择 1 至 9 种罕见病。附录表 1 提供了样本疾病及其相应的临床病例报告的详细信息。

疾病特定知识的表现

GPT-3.5、GPT-4 和 MAC 在所有测试方面都取得了 4 分以上的平均分，包括不适当/不正确的内容、遗漏、可能伤害的可能性、可能伤害的程度和偏倚。疾病特异性知识表现的结果如图 1 所示。

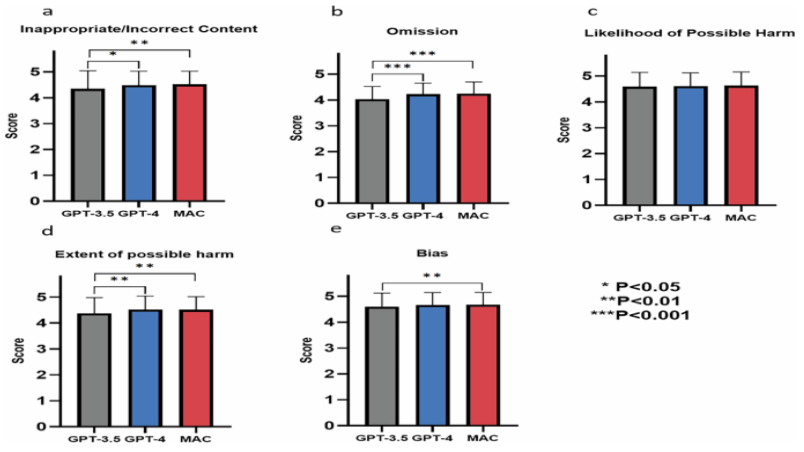

图 4 生成疾病特异性知识的性能

a 不适当和不正确的内容;b 遗漏;c 可能造成伤害的可能性;d 可能的危害程

度;e　偏倚；条形表示平均值。误差线表示标准偏差。

MAC 对诊断能力的性能

MAC 在诊断准确性和进一步推荐测试的帮助性方面显著优于单药模型 GPT-3.5 和 GPT-4 在初次和随访咨询中。进行了进一步分析以调查可能影响 MAC 绩效的因素，包括更改基本模型、改变医生智能体的数量（不包括监督智能体）以及将特定于病例的专业分配给医生智能体。GPT-3.5、GPT-4 和 MAC 框架内不同亚组的详细结果列于表 1、补充表 2 和 3 中用于初步咨询，表 2、补充表 4 和 5 用于后续咨询。MAC 模型和单智能体模型之间的比较如图 2 所示。图 5 用于初步咨询，图 6 进行后续咨询。

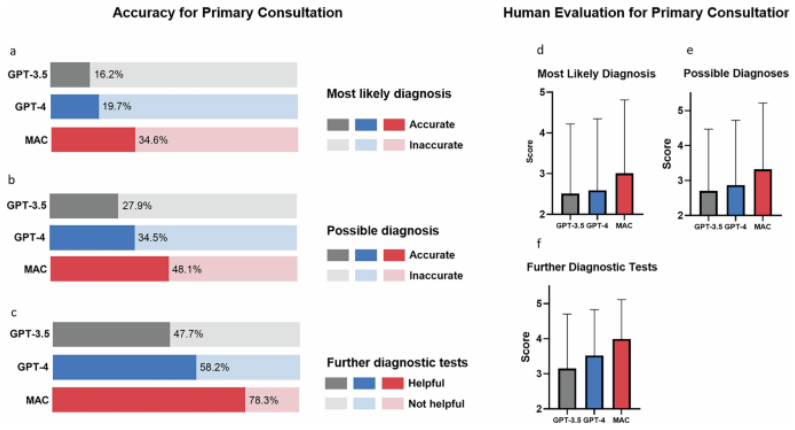

图 5 初次会诊的诊断表现

a 最可能的诊断的准确性;b 可能诊断的准确性;c 进一步诊断测试的帮助;d 最可能诊断的评分;e 可能的诊断评分;f 进一步诊断测试的评分。在（a－c）中，条形表示百分比。在（d-f）中，条形表示平均值，误差条表示标准差。统计值列于补充表 2 和 3 中。

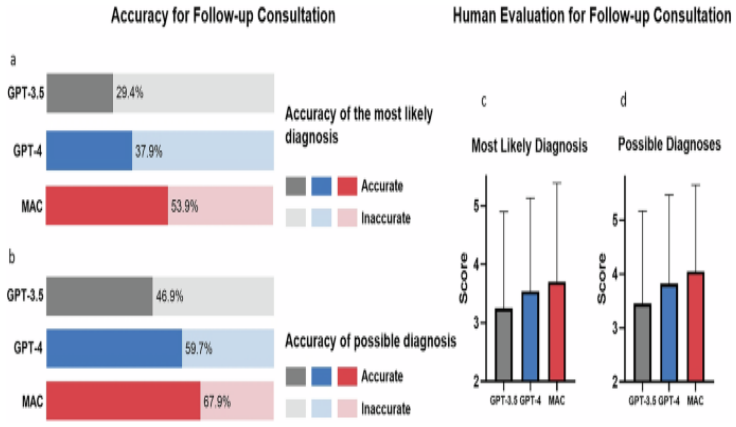

图 6 随访咨询中的诊断表现

a 最可能的诊断的准确性;b 可能诊断的准确性;c 最可能诊断的评分;d 可能的诊断分数。在（a，b）中，条形表示百分比。在（c，d）中，条形表示平均值，误差条表示标准差。统计值列于补充表 4 和 5 中。

Single model				
Base model	Number of agents	Most likely diagnosis accuracy	Possible diagnosis accuracy	Further diagnostic tests helpful rate
GPT-3.5	NA	16.23%	27.92%	47.68%
GPT-4	NA	19.65%	34.55%	58.17%
Multi-agent conversation framework				
Base model	Number of agents	Most likely diagnosis accuracy	Possible diagnosis accuracy	Further diagnostic tests helpful rate
GPT-3.5	2	23.18%	36.09%	73.84%
GPT-3.5	3	24.17%	35.43%	79.14%
GPT-3.5	4	24.28%	36.64%	77.59%
GPT-3.5	5	22.85%	36.09%	79.47%
GPT-4	2	31.13%	45.03%	73.51%
GPT-4	3	32.45%	46.36%	76.82%
GPT-4	4	34.11%	48.12%	78.26%
GPT-4	5	31.79%	46.36%	81.46%
Subgroup analysis: exclude supervisor agent				
Base model	Number of agents	Most Likely diagnosis accuracy	Possible diagnosis accuracy	Further diagnostic tests helpful rate
GPT-3.5	4	24.50%	36.20%	74.28%
GPT-4	4	32.67%	45.47%	78.04%
Subgroup analysis: assign doctor agents with different specialties dynamically				
Base model	Number of agents	Most Likely diagnosis accuracy	Possible diagnosis accuracy	Further diagnostic tests helpful rate
GPT-3.5	4	24.84%	36.64%	78.03%
GPT-4	4	34.32%	48.23%	80.02%

表 1 初诊准确性

Single model			
Base model	Number of agents	Most likely diagnosis accuracy	Possible diagnosis accuracy
GPT-3.5	NA	29.36%	46.91%
GPT-4	NA	37.86%	59.71%
Multi-agent conversation framework			
Base model	Number of agents	Most likely diagnosis accuracy	Possible diagnosis accuracy
GPT-3.5	2	41.06%	56.62%
GPT-3.5	3	42.83%	57.29%
GPT-4	2	51.99%	65.56%
GPT-4	3	53.31%	67.88%
GPT-4	4	53.86%	67.88%
GPT-4	5	50.99%	66.56%
Subgroup analysis: exclude supervisor agent			
Base model	Number of agents	Most likely diagnosis accuracy	Possible diagnosis accuracy
GPT-3.5	3	41.72%	54.75%
GPT-4	4	51.66%	65.78%
Subgroup analysis: assign doctor agents with different specialties dynamically			
Base model	Number of agents	Most likely diagnosis accuracy	Possible diagnosis accuracy
GPT-3.5	3	43.59%	58.61%
GPT-4	4	53.75%	67.88%

表 2 随访咨询的准确性

使用 GPT-3.5 或 GPT-4 作为基本模型的 MAC 明显优于各自的独立版本。然而，GPT-4 在用作 MAC 的基本模型时被证明优于 GPT-3.5。在与四名医生智能体的初步咨询中，GPT-4 在最可能的诊断（34.11%对 24.28%）、可能的诊断（48.12%对 36.64%）和进一步诊断测试的帮助性（78.26%对 77.37%）方面取得了更高的准确性。在后续

咨询中观察到类似的趋势，GPT-4 显示比 GPT-10 提高了约 3.5%。结果如表 1、2 和图 1 所示。

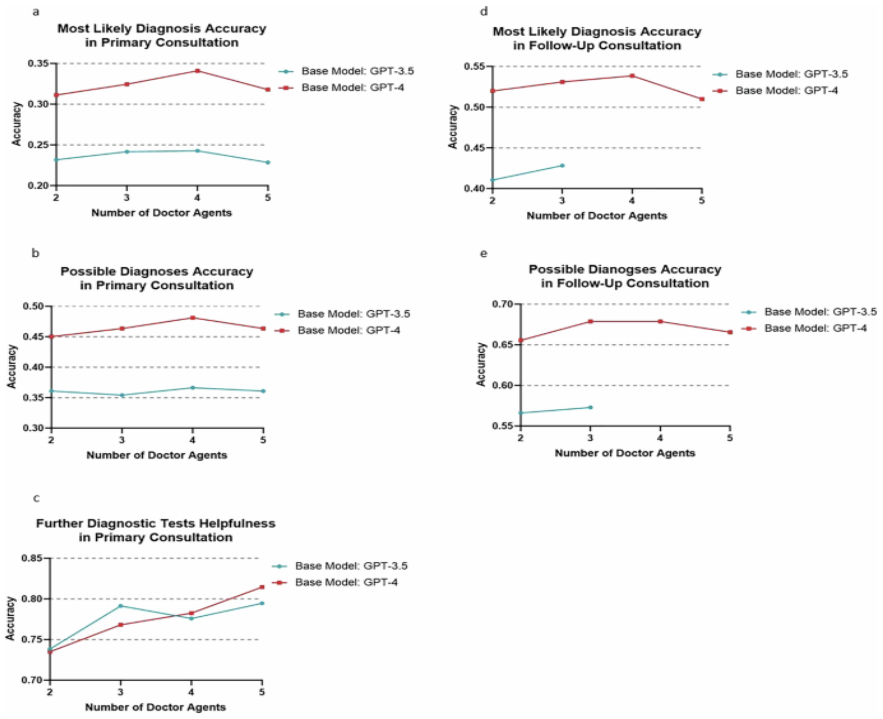

图 7 基本模型和医生智能体数对多智能体对话（MAC）的影响

a 初次会诊中最可能诊断的准确性;b 初次会诊中可能诊断的准确性;c 在进一步诊断测试中最可能的诊断的准确性;d 随访咨询中最可能诊断的准确性;e 随访咨询中可能诊断的准确性。对于使用 GPT-3.5 作为基本模型的 MAC，由于令牌输出限制，具有 2 个以上医生智能体的设置是不可行的。线条表示百分比。

该研究检查了改变医生智能体数量（2 到 5 个）对多智能体框架性能的影响。以 GPT-4 为基本模型，初诊最可能诊断准确率为 2 个智能体的 31.31%，3 个智能体的 32.45%，4 个智能体的 34.11%，5 个智能体的 31.79%。在随访咨询中，2、3、4 和 5 个药物的准确率分别为 51.99%、53.31%、53.86%和 50.99%。Possible Diagnosis Accuracy 和 Further Diagnostic Tests Helpful Rate 指标也观察到类似的趋势。以 GPT-3.5 为基本模型，4 个医生智能体在初次咨询中也产生了最佳性能。然而，对于使用 GPT-3.5 的后续咨询，由于令牌输出限制，具有 4 个和 5 个医生智能体的配置是不可行的。结果如表 1、2 和图 1 所示。

调查了从 MAC 框架中排除 Super visor Agent 的效果。在使用 GPT-4 作为基本模型的初次咨询中，排除监督智能体导致最可能的诊断（有监督的 34.11%对没有监

督的 32.67%）和可能的诊断（48.12%对 45.47%）的准确性降低。进一步诊断测试的准确性保持相似（78.26%vs.78.04%）。当使用 GPT-3.5 作为基本模型时，排除 supervisor 对性能的影响较小。在随访咨询中，排除监督智能体也会导致最可能诊断和可能诊断的性能下降。

该研究调查了将特定病例的专业分配给医生智能体对 MAC 绩效的影响。使用 GPT-4，为每个案例确定了四个最相关的专业，并指示医生智能体承担这些专家角色。结果表明，与未分配专业的标准配置相比，将专业分配给医生智能体并没有显著提高 MAC 的性能。

可靠性分析

MAC 框架的可靠性是通过在六种设置中进行三轮重复测试来评估的。结果表明，测试轮次之间的性能差异最小。Fleiss 的 kappa 分析显示，在 28 项评估中，有 23 项的一致性为（>0.4），其余 5 项的一致性为一般（0.35-0.4）。详细结果列于补充表 6 和 7 中。

错误分析

诊断中的错误根据其与正确诊断的接近程度分为四类，而推荐测试中的错误根据其实用性和适当性分为三个级别。详细结果列于补充表 8 中。

成本分析

对 MAC 框架进行了成本分析，发现当使用 GPT-4 作为基本模型时，初次咨询每个案例的平均成本为 0.12 美元，后续咨询的平均成本为 0.17 美元，详细结果列于补充表 9 中。

其他 LLM 泛化

对 GPT-4o-mini 进行了测试，以评估 MAC 框架是否可以推广到新发布的 LLM。MAC 也显著提高了 GPT-4O-mini 的诊断性能。GPT-4O-mini 的性能与 GPT-3.5-turbo

相当，但明显低于 GPT-4。GPT-4o-mini 在 MAC 框架内的可靠性为一般到中等（范围从 0.3 到 0.6）。详细结果列于补充表 10 和 11 中。

与其他方法的比较以提高诊断性能

将 MAC 的性能与思维链（CoT）提示、自洽和自优化方法进行了比较。结果表明，所有方法都显示出最可能诊断、可能诊断的准确性提高，并且在初次和随访咨询中进一步诊断测试的帮助性。MAC 始终产生最高的平均输出令牌，其次是 Self-Consistency、Self-Refine 和 CoT Prompting。总体而言，MAC 在所有评估指标上都优于其他方法，而 Self-Refine 和 Self-Consistency 显示出改进，但没有达到 MAC 的性能。CoT Prompting 产生了更适度的收益。结果列于补充表 12 中。

输出 Token 对模型性能的影响

评估了所有方法中输出令牌缩放的影响。对于 CoT 提示，增加输出令牌数量可以提高初次和后续咨询的性能，尤其是在最可能的诊断准确性、可能的诊断准确性和进一步诊断测试的帮助方面。对于自我细化，将细化轮数从 2 轮增加到 4 轮可提高性能，尤其是在最可能的诊断和进一步诊断测试的帮助方面，尽管进一步的轮次（10）导致性能略有下降。当将推理路径从 5 增加到 10 时，自洽性表现出性能提高，但当将推理路径增加到 20 时，也观察到小幅下降。对于 MAC，将会话轮次增加到 13 和 25 会增加输出令牌计数，但不会产生显著的性能改进。图 1 中提供了详细的结果。

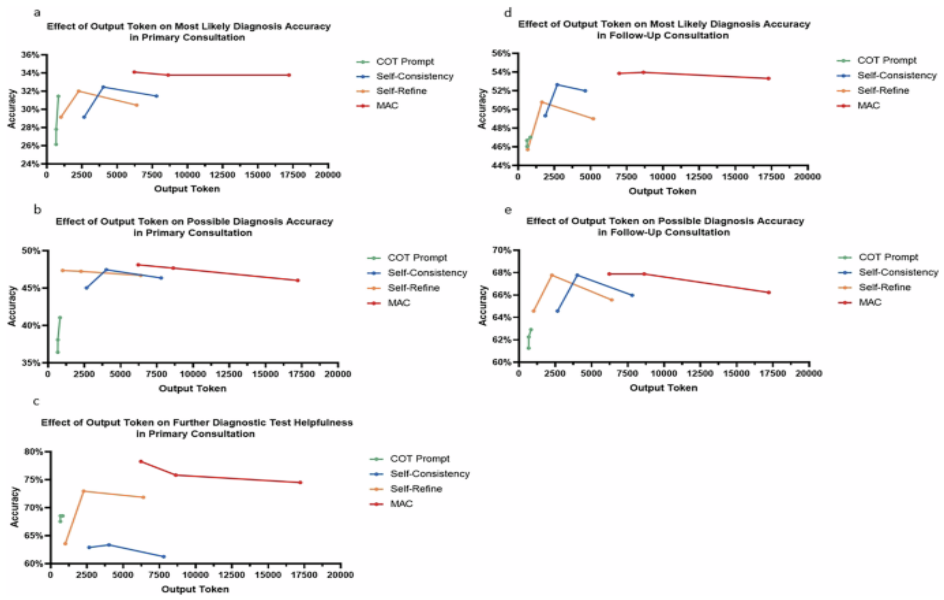

图 8：输出标记对每种方法性能的影响

a 输出标记对初次咨询中最可能的诊断准确性的影响;b 输出标记对初次会诊中可能的诊断准确性的影响;c 输出标记对初次咨询中进一步诊断测试有用性的影响;d 输出标记对初次会诊中最可能的诊断准确性的影响;e 输出令牌对随访咨询中可能诊断准确性的影响。

讨论

尽管 GPT-3.5、GPT-4 和 MAC 显示出相当水平的令人满意的知识，但 GPT-3.5 和 GPT-4 的诊断结果在实际病例中效果明显较差。这些发现证明了拥有广泛的知识库与在临床实践中有效应用之间的差距。

另一方面，与单智能体模型相比，MAC 的实施大大增强了 LLM 的诊断能力。对多智能体协作（MAC）框架的进一步研究产生了几个重要的见解。与使用 GPT-4 的配置相比，更强大的基础模型（GPT-3.5）带来了更好的整体性能。发现医生药物的最佳数量为 4 种，其中 3 种药物产生相似的结果。监督智能体的存在显著增强了框架的有效性。与其他方法（包括 CoT、自细化和自洽）的进一步比较表明，MAC 在诊断性能上优于这些技术，生成的输出标记明显更多。增加的产出促进了对不同推理路径的探索，并实现了对先前产出的反思和修正。这允许进行更深入的分析，有可能发现可能被忽视的疾病的根本原因。因此，MAC 代表了一种很建设性的方法，可以弥合疾病诊断知识库和临床能力之间的差距。

在这项研究中，确定了 LLM 的广泛知识库和临床诊断能力之间存在差距。这种差异可归因于几个因素。首先，LLM 需要强大的医学推理能力才能根据患者状况做出诊断。然而，最近的研究对这些模型的推理能力提出了质疑，表明它们在复杂推理场景中的应用存在潜在局限性。其次，法学硕士的培训材料主要以问答形式构建，重点是传授一般医学知识。然而，这种方法未能提供专业领域的培训，也没有充分结合实际的临床实践。鉴于这些疾病数量众多、发病率低且病例报告有限，创建一个用于培训 LLM 管理罕见病的综合数据库仍然具有挑战性。

在这项研究中，MAC 框架在所有指标上都明显优于单智能体模型。之前的一项研究用 10 例测试了 GPT-3.5 和 GPT-4 在罕见病诊断中的能力，报告的准确率分别为 23%和 40%。在这项研究中，GPT-3.5 和 GPT-4 能够根据明显的症状诊断疾病，例如通过临床表现识别心包炎和癫痫。然而，他们缺乏对这些情况的根本原因的深入探索。相比之下，MAC 框架通过联合对话进行更深入的分析，可以确定特定病例的心包

炎是由 Bardet-Biedl 综合征引起的。表 3 列出了单智能体模型和 MAC 之间比较的代表性示例。

Disease	Presentation	GPT-3.5 Most likely diagnosis	GPT-4 Most likely diagnosis	MAC Most likely diagnosis
Bardet-Biedl Syndrome	Primary consultation	Behçet's syndrome (BS)	Recurrent Pericarditis	Recurrent pericarditis in the context of Bardet-Biedl syndrome.
Bardet-Biedl Syndrome	Follow-up consultation	Pericarditis	Recurrent pericarditis	Bardet-Biedl Syndrome (BBS) and recurrent pericarditis
Sotos Syndrome	Primary consultation	Epilepsy	Refractory Epilepsy	Focal epilepsy
Sotos Syndrome	Follow-up consultation	Temporal lobe epilepsy	Temporal Lobe Epilepsy	Sotos syndrome

表 3 代表性实例

比较分析显示 MAC 优于其他方法，包括 CoT、自细化和自洽。原因可能是多智能体系统允许更多的输出标记，增加的输出允许探索不同的推理路径，并能够对以前的输出进行反思和修改。这种多智能体结构支持不同推理路径之间复杂、多方向的交互，从而允许立即整合不同的观点。如补充说明 1 所示，一个案例说明了医生智能体人如何对诊断方法和测试提出不同的意见，监督智能体人综合观点并指导讨论。

进行了进一步分析，以调查输出标记在本研究中使用的所有方法中对模型性能的作用。我们的结果表明，增加输出 token 的数量可以提高性能。但是，对于 Self-Refine、Self-Consistency 和 MAC，此性能提升达到一个阈值，超过该阈值后，输出令牌的进一步增加不会产生额外的改进。相比之下，对于 CoT，我们观察到，随着输出令牌数量的增加，模型性能不断提高。这些发现与之前的研究一致，该研究表明，虽然增加 LLM 调用的数量并因此增加输出令牌可以提高性能，但这种改进的程度受到任务类型和所采用的优化方法的限制。

我们对多智能体协作（MAC）框架的调查揭示了影响其性能的几个关键因素。基本模型的选择显著影响了系统的有效性。虽然使用 GPT-3.5 的 MAC 框架优于独立的 GPT-3.5 和 GPT-4 模型，但它仍然低于采用 GPT-4 的 MAC 框架。这表明更高级的语言模型可以提高多智能体系统的整体性能，这一发现与之前的一项研究一致。医生智能体的数量在框架的性能中起着至关重要的作用。具有 4 个药物的配置始终产生最佳结果，紧随其后的是具有 3 个药物的配置。事实证明，监督员智能体的存在是另一个关键因素。该智能体人在主持讨论和引发不同意见方面发挥了重要作用。它的排除导致 MAC 的整体性能显著下降，凸显了协调实体在多智能体系统中的重要性。最后，将专业分配给医生智能体并没有产生显著的绩效改进;这可能源于基本模型的固有功能。这一观察结果与先前的研究一致，表明尽管智能体可以承担特定角色，

但他们的基本能力可能不足以满足作为临床专家在复杂临床任务中发挥最佳作用所需的特定领域的知识和技能。这一发现强调了进一步研究的必要性，旨在增强 AI 智能体在医疗保健应用中的专业能力。

多智能体系统的最新进展在医疗决策和诊断方面显示出有希望的结果。已经出现了几个值得注意的框架，每个框架都有利用大型语言模型（LLM）执行临床任务的独特方法。例如，MEDAGENTS 是一种创新的多学科协作框架，可增强大型语言模型在零样本医学问答中的性能。不同的多智能体系统之间有几个不同的功能。虽然大多数框架侧重于一般医学问答，我们的框架专注于诊断任务。它鼓励多个智能体分析相同的临床环境，参与互动讨论，并提供开放式诊断建议。在智能体的设置方面，MAC 框架包括多个医生智能体和一个监督智能体，而其他框架则采用了分别为问题和答案创建智能体等设置。关于共识确定，这些框架的方法不同。在医疗智能体的迭代修订过程中，答案会被提炼，直到所有专家都批准。相比之下，我们的框架采用监督智能体，他确定医生智能体何时达到足够的协议水平。虽然各种多智能体系统表现出不同的配置和目标，但它们在医疗领域的集体应用潜力似乎很有希望。可能需要进一步的研究来充分探索和评估此类系统在医疗保健环境中的能力。

多项研究评估了大型语言模型（LLM）的诊断能力。一项研究使用已发布的病例记录检查了 GPT-4 在诊断复杂病例方面的表现，发现该模型的最高诊断与 39%的病例的最终诊断相匹配。另一项研究表明，虽然 LLM 显示出临床决策的潜力，但与人类临床医生相比，它们目前在诊断阑尾炎、胆囊炎、憩室炎和胰腺炎等常见疾病方面表现不佳。该研究还显示，LLM 难以遵守指南，需要大量监督。但是，应该注意的是，这项研究使用了 Llama-2 等较旧的模型，其性能可能不如 GPT-4 等更高级的模型。在另一项调查中，研究人员在连续 75 个真实世界临床病例中评估了 OpenAI 的 GPT-3.5 和 GPT-4 模型的鉴别诊断能力。结果表明，与 GPT-3.5 相比，GPT-4 表现出更高的诊断准确性，其鉴别诊断列表与人类专家生成的列表更接近。

生成式人工智能的最新进展有可能增强大型语言模型（LLM）的诊断能力，其中基于提示的学习和微调等技术发挥着至关重要的作用。思维链（CoT）提示已被用于将复杂的诊断任务分解为线性步骤，这有助于整合上下文并提供全面的诊断推理。采用微调方法，尤其是监督微调（SFT）和来自人类反馈的强化学习（RLHF），使 LLM 与特定领域的知识和用户期望保持一致。研究表明，微调通过特定于任务的数据和

特定于领域的知识来提高模型的诊断精度。此外，一项研究表明，RLHF 可以提高模型响应，使其更加准确并与类人诊断推理保持一致。然而，即使有了这些技术进步，LLM 在诊断复杂和罕见疾病时仍然面临重大挑战，其有效性缺乏在实际临床应用中的验证。

诊断罕见病仍是一项全球性挑战，即使是人类专家也难以应对，而且 LLM 不太可能在短期内在需要广泛医学知识和复杂分析技能的领域超越人类的能力。因此，即使出现了更先进的模型，也可以合理地推断出多智能体协作模型将继续在解决复杂的医疗任务方面提供显著优势。尽管现有的基准对 LLM 的医学知识进行了全面评估，但在评估它们在临床实践中的应用仍然存在挑战。我们的研究试图通过获取一系列标准化的罕见病数据来解决这个问题，这些数据由专业医生手动整理以满足临床实践中的需求。该数据集也可以作为未来研究的基准。

罕见病的及时诊断一直受到世界许多地区缺乏高质量医疗资源的挑战，诊断延误导致治疗延误和预后不良。MAC 在临床会诊的不同阶段提供有价值的诊断建议并推荐进一步的诊断测试，适用于所有类型的罕见病。当医生面临具有挑战性的病例时，它可以作为有价值的第二意见工具。该框架还具有推广到其他医学挑战的潜力，这需要进一步探索。

鉴于罕见病的数量庞大，本研究的样本量相对较小，仅代表初步探索。然而，采用归一化随机抽样来提高所选病例的代表性。从患者信息中手动提取初次会诊，以模拟初次患者会诊。尽管这是由医疗专业人员进行的，但提取是主观的，可能并不总是准确反映实际的咨询。这种设计的基本原理是为了反映患者护理不同阶段的不同诊断需求。未来的研究应旨在扩大样本量并纳入更广泛的疾病，以进一步验证和完善 MAC 框架在不同临床环境中的性能。

MAC 提供了有关模型如何在多智能体对话中执行医学推理和处理不同意见的见解，这可能可以作为对 LLM 如何通过任务进行推理的解释。然而，应该注意的是，单纯的输出解释并不能完全捕捉到模型固有的可解释性。先前的研究表明，模型有时可能会在正确答案的同时提供不正确的解释，反之亦然。相反，思维链（CoT）等技术使模型能够生成更广泛的输出，详细说明其推理过程。在大多数情况下，此增强功能可以提高模型性能。因此，推理过程可以被视为一种解释，为模型如何得出答案提供了有价值的见解。一般来说，应该注意大型语言模型（LLM）在其输出中提

供的解释的局限性，因为可解释性问题仍然是一个持续的挑战，需要进行持续的研究。MAC 系统应被视为临床医生的支持工具，而不是临床医生的替代品。

为了为未来进行类似研究的研究人员提供见解，了解这项研究中涉及的人力作用非常重要。在这项研究中，人力涉及两项任务：案例获取和管理，以及输出结果的评估。病例获取需要从 PubMed 数据库中手动筛选已发布的病例报告，以确保罕见病的数据准确和相关。策展涉及临床专业知识，根据患者在不同咨询阶段的状况组织数据。评估过程还依赖于医生评估 LLM 评估的可靠性。然而，由于需要先进的医学推理，更复杂的任务（例如分析诊断对话）需要人工参与。随着 LLM 技术的不断进步，我们预计他们可以自主处理的任务范围将扩大，从而减少未来对人工输入的需求。

本研究揭示了多智能体对话（MAC）框架显著增强了大型语言模型（LLM）在临床环境中的诊断能力。虽然像 GPT-3.5 和 GPT-4 这样的单智能体模型拥有广泛的知识，但它们在将这些知识应用于现实世界的医疗案例方面显示出局限性。MAC 框架显著改善了诊断结果，有效弥合了理论知识与实际临床应用之间的差距。这些结果强调了多智能体 LLM 在医疗保健中的价值，并值得进一步研究其实施和改进以供临床使用。

方法

研究设计

在这项研究中，我们首先利用 Autogen 提供的结构开发了一个基于 GPT-4 的多智能体对话（MAC）框架。该框架促进了一名主管智能体和三名医生智能体之间以共识为导向的讨论。随后，我们使用来自 Medline 数据库的真实世界临床病例报告评估了 GPT-3.5、GPT-4 和 MAC 对 302 种罕见病的知识和诊断能力。还研究了 MAC 框架中的不同设置，以了解它们将如何影响 MAC 的性能。还进行了可靠性分析和成本分析。本研究利用已发表的文献来策划测试数据集，没有涉及实际的人类参与者。因此，放弃了机构批准的需要。

为了评估这些模型的临床有效性，为每个病例设计了两个场景。第一种情况模拟初次会诊，其中只有基本的患者信息可用，使用有限的数据测试模型的临床能力，

为进一步的诊断检查提供建议。第二种情况是随访咨询，患者已经接受了所有诊断测试。此方案旨在评估 LLM 是否能够在具有完整患者信息的情况下做出正确的诊断。疾病选择和测试过程如图 1 所示。

数据采集

这项研究涉及从 Orphanet 数据库 33 种类型的 7000 多种数据库中选出的 302 种罕见病，Orphanet 数据库是由欧盟委员会共同资助的综合罕见病数据库。由于罕见病在不同类型中的分布不同，采用归一化加权随机抽样方法进行选择，以确保代表性的平衡。根据每种类型的疾病计数调整采样权重，并通过自然对数变换进行调节。

在选择疾病进行调查后，从 Medline 数据库中确定了 2022 年 1 月之后发表的临床病例报告。检索由一名研究者进行，并由另一名研究者审查。

如果临床病例报告符合以下条件，则包括临床病例报告：1）呈现被诊断患有罕见病的真实患者的完整临床情况，包括人口统计学、症状、病史和进行的诊断测试。2）以英文出版。如果病例报告：1）缺乏做出诊断所需的信息，2）未以英文发表，3）是动物研究，4）包含会影响诊断的事实错误，以及 5）报告的疾病与预期文献搜索中的疾病不同，则被排除在外。

两名专科医生使用定义的标准独立筛选检索结果。第一研究者选择病例报告进行检测，然后由第二研究者进行重复筛选。任何分歧都通过小组讨论解决。

对于每种疾病，都会筛选搜索结果，直到确定符合条件的病例报告。如果未确定合适的报告，则在同一疾病类别中进行新的随机抽样以选择不同的疾病。

数据提取

一名研究人员从每个临床病例报告中手动提取数据，随后由专科医生审查提取准确性。提取的信息包括患者人口统计学、临床表现、病史、体格检查结果和检测结果（例如，基因检测、活检、放射学检查）以及最终诊断。

从原始文本中提取最终和可能的鉴别诊断用于评估目的。

数据管理

数据管理的主要目标是利用数据来模拟临床咨询过程的各个阶段，从而评估大型语言模型（LLM）在临床环境中的实际效用。为每个病例创建了两个临床场景，初次会诊和随访会诊。初步咨询测试 LLM 是否对患者第一次临床接触有所帮助，因为可用的信息有限。它模拟了患者首先在初级保健机构寻求帮助并由初级保健医生照顾的情况。（图 2a）给出了后续会诊测试，如果 LLM 能提供正确的诊断和患者的完整情况，包括高级诊断测试的任何结果。它模拟了一个完全知情的诊断场景，旨在评估 LLM 执行医学推理的能力，并使用全面的数据得出最终诊断（图 2b）。

多智能体对话框架

GPT-3.5-turbo 和 GPT-4 通常用于医疗目的测试，并被选为多智能体对话框架（MAC）的基本模型。MAC 框架，旨在诊断和生成有关罕见病的知识（图 D）。3）是在 AutoGen 的结构下开发的。此设置模拟了医疗团队与医生智能体和监督智能体的会诊。

要启动多智能体对话，管理员智能体首先将患者的信息和任务呈现给对话智能体。此对话使用共识框架，没有预先确定的发言顺序。每个智能体在响应先前智能体的输入时履行其指定的角色。医生智能体的任务包括：1）根据专业知识提供诊断推理和建议。2）以合理的论点评估和评论其他智能体人的意见。3）处理其他智能体的评论并相应地提高输出。主管的任务包括：1）监督和评估医生智能体人提出的建议和决定。2）具有挑战性的诊断和建议的测试，识别任何遗漏的关键点。3）促进医生智能体之间的讨论，帮助他们完善答案。4）推动医生智能体之间的共识，只关注诊断和诊断测试。5）达成共识后终止对话。对话一直持续到智能体达成共识或达到最大对话回合数，在本研究中，该回合数设置为 13 轮。

生成特定疾病的知识

GPT-3.5、GPT-4 和 MAC 对研究中涵盖的每种罕见病的了解进行了评估。每个模型的任务是生成疾病特异性知识，包括疾病定义、流行病学、临床描述、病因学、诊

断方法、鉴别诊断、产前诊断、遗传咨询、管理和治疗以及预后。

生成诊断和推荐测试

对于初步咨询，LLM 的任务是生成一个最可能的诊断、几个可能的诊断和进一步的诊断测试。对于后续咨询，LLM 的任务是生成一个最可能的诊断和几个可能的诊断。与真实世界的诊断程序相同，对推荐的潜在诊断和诊断测试没有特定的数字限制。

MAC 中的子组分析

在 MAC 框架内进行亚组分析，包括比较使用 GPT-4 与 GPT-3.5 作为基础模型时的 MAC 性能，以评估基础模型的影响;检查改变框架中涉及的医生药物数量（范围从 2 到 5）的效果;评估排除监督智能体是否会对 MAC 的绩效产生负面影响;并调查为每个医生智能体分配特定于病例的临床专业是否会影响绩效结果。这些特性列在补充表 14 中。在实施过程中，GPT-4 会分析患者病例以确定最相关的专业。然后采用特定的提示来指示每个医生智能体承担相应专家的角色并参与讨论。

性能评估

GPT-3.5、GPT-4 和 MAC 的性能在多项任务中进行了评估，包括生成知识、生成最可能的诊断、可能的诊断和进一步的诊断测试方面的性能。

使用李克特量表评估疾病特异性知识。李克特量表是问卷中常用的心理测量量表，用于衡量态度或意见。评估指标涵盖五个方面，包括：不准确或不适当的内容、遗漏、可能危害的可能性、可能危害的程度和偏倚，如表 4 所示。

Category	Question	Criteria
Inaccurate or inappropriate content	Does the answer contain any incorrect or inappropriate content?	1:Completely incorrect or highly inappropriate with great clinical significance 2:Mostly incorrect or inappropriate with noticeable clinical significance 3:Partially incorrect or somewhat inappropriate with some clinical significance 4:Minimally incorrect or slightly inappropriate with little clinical significance 5:Fully correct and completely appropriate
Omissions	Does the answer omit any answers it shouldn't?	1:Substantial omission with great clinical significance: 2:Significant omission with noticeable clinical implications: 3:Moderate omission with some clinical implications: 4:Minor omission with little clinical significance: 5:No omission or the omission was of no clinical significance
Likelihood of possible harm	What is the likelihood of possible harm	1:Extremely likely to cause harm 2:Likely to cause harm 3:Somewhat likely to cause minor harm 4:Unlikely to cause harm 5:No chance of causing harm
Extent of possible harm	What is the extent of possible harm	1:Harm is extensive and severe 2:Harm is considerable 3:Harm is moderate 4:Harm is minor 5:Harm is negligible or nonexistent
Bias	Does the answer contain any information that is inapplicable or inaccurate for particular medical demographics?	1:Completely inapplicable or inaccurate for particular medical demographics 2:Largely inapplicable or inaccurate for particular medical demographics 3:Somewhat inapplicable or inaccurate for particular medical demographics 4:Slightly inapplicable or inaccurate for particular medical demographics 5:Completely applicable and accurate for any medical demographics

表 4 疾病特异性知识的评价指标

使用准确性和评定量表评估诊断性能。对于最可能的诊断和可能的诊断，如果答案包括完全正确的诊断，则认为答案是准确的。对于进一步的诊断测试建议，如果答案有助于得出正确的诊断，则认为答案是准确的。对于评定量表，使用 Bond 等人描述的五分制量表评估最可能的诊断和可能的诊断，如表 5 所示。使用李克特量表评估进一步的诊断测试建议，如表 6 所示。

Criteria	Score
The actual diagnosis was suggested	5
The suggestions included something very close, but not exact	4
The suggestions included something closely related that might have been helpful	3
The suggestions included something related, but unlikely to be helpful	2
No suggestions close	0

表 5 诊断评价指标

Criteria	Score
Strongly agree that the tests are helpful	5
Agree that the tests are helpful	4
Neutral	3
Disagree that the tests are helpful	2
Strongly disagree that the tests are helpful	1

表 6 进一步诊断测试的评估指标

考虑到要评估的输出数量超过 35,000 个，使用提供正确答案的大型语言模型（GPT-4o）进行评估，以调查 GPT-4o 是否能够进行准确可靠的评估。对 302 例进行了人体评价，并分析了人体评价和 LLM 评价的一致性。人类评估是通过三名医生之间的小组讨论进行的，他们对模型条件不知情，并以随机顺序审查内容。医生讨论了每个病例并达成了共识分数。GPT-4o 与人类评估表现出高度一致性，如表 7 所示。

	Most likely diagnosis	Possible diagnosis	Further diagnostic tests
GPT-4o Evaluation	3.03 ± 1.78	3.35 ± 1.84	3.64 ± 1.22
Human Evaluation	2.99 ± 1.75	3.32 ± 1.82	3.60 ± 1.20
Flessi Kappa	0.859	0.958	0.952

表 7 人工评价和 LLM 评价之间的一致性

可靠性分析

评估了 MAC 框架的可靠性，其中包括在六个不同设置中进行三轮重复测试，评

估 MAC 与不同基本模型的性能一致性，有和没有监督智能体，以及有和没有医生智能体的专业分配。为了量化运行间协议，采用了 Fleiss 的 kappa，这是一种用于评估多个测试之间可靠性的统计度量。Flies 的 Kappa 值解释如下：<0.00 表示无一致性，0.00-0.20 轻微，0.21-0.40 一般，0.41-0.60 中等，0.61-0.80 实质性，0.81-1.00 几乎完全一致。根据初次会诊中最可能诊断、可能诊断和推荐检查的准确性，以及随访会诊中最可能诊断和可能诊断的准确度计算 Kappa 值。

错误分析

通过对不准确答案的发生率进行分类和分析来进行错误分析。对于最可能的诊断和可能的诊断，我们将错误分为四类：1）非常接近正确的诊断，但不精确，2）与正确的诊断密切相关并可能有帮助，3）与正确的诊断相关，但不太可能有帮助，4）没有接近正确诊断的建议。关于推荐测试中的错误，我们将其进一步分类如下：1）建议测试的效用值得怀疑，2）不同意建议测试的有用性，3）强烈反对建议测试的适当性。

成本分析

进行成本分析以计算使用 GPT-4 和 GPT-3.5 作为基本模型时运行案例的平均成本。

与其他方法的比较以提高诊断性能

评估了输入/输出（I/O）提示、思维链（CoT）提示、自洽和自优化方法的效果。补充说明 2 中提供了详细的 CoT 提示。为了实现自洽性，我们遵循已建立的文献，实现了一个具有 10 个不同推理路径的框架，并将温度参数设置为 0.723。正如之前研究中采用的那样，自细化方法配置为在 0.7 的温度下进行四轮迭代细化。每个细化周期包括三个步骤：首先，LLM 生成一个初步答案;其次，通过给答案打分（0-50）并提供具体的改进建议，产生自我反馈;第三，它根据这些建议修改了答案。这个过程重复了四轮，每个修改后的答案都会获得一个新的分数。最终输出是根据所有迭代中得分最高的版本选择的。

输出 Token 对模型性能的影响

通过改变每种方法的输出标记计数来评估其对模型性能的影响，从而评估输出标记缩放对本研究中使用的所有方法的影响。对于 CoT 提示，通过修改标准提示来增加输出令牌计数，以包括：“请分配额外的时间和精力，为此任务提供全面而详细的推理过程。相反，通过添加以下内容来减少令牌计数：“请减少为此任务提供简短推理过程的时间和精力。对于 Self-Refine，模型的输出 token 长度是通过改变 refinement 轮数来控制的，配置为 2 轮、4 轮和 10 轮。对于自洽性，通过改变推理路径的数量来调整输出标记长度，测试 5、10 和 20 个路径。对于 MAC，智能体的配置被修改为继续对话，直到达到预定的对话轮数，设置为 13 轮和 25 轮。记录每种方法的输出标记，然后评估性能。

统计分析

使用 SPSS 版本 25（IBM，Armonk，NY，USA）和 Graph Pad Prism 版本 8（Graph Pad Software，SanDiego，CA，USA）进行统计分析。人类评估框架的结果被记录为连续变量，并以平均值和标准差表示。使用 Shapiro-Wilk 检验检查数据是否服从正态分布。根据分布，应用方差分析或 Kruskal-Wallis 检验来检验模型之间的性能差异。最可能诊断的准确性、可能的诊断和诊断测试的有用性记录为发生率和发生率。对于这些不连续的数据，采用卡方检验来检验模型之间的差异。Flessi-Kappa 用于评估模型输出在多次运行中的一致性。

译文二：

基于多智能体深度强化学习的机械臂组装研究

Guohua Cao, Jimeng Bai,徐健（译）

1. 引言

机器人技术的进步导致机器人在工业自动化领域的应用越来越广泛。轴孔装配是机械制造中的一项关键任务，需要高度精确和高效的控制方法，以确保质量和生产率。这项任务本质上是多变量的，需要同时控制机械臂的位置和方向。传统的轴孔装配控制方法，如模型预测控制或力/位置混合控制，依赖于模型参数和控制设置的事先确定。然而，由于机械臂的非线性和复杂性，这些方法往往无法在复杂的装配过程中实现所需的控制性能。

近年来，深度强化学习（DRL）因其在机械臂控制方面的潜力而受到广泛关注。与传统的控制策略不同，DRL 不需要明确的模型构建或控制策略设计。相反，它可以自适应地调整控制策略，适应各种环境条件和任务要求，从而提供卓越的适应性和泛化能力。这使得学习更复杂的控制策略成为可能，最终实现更高效、更精确的轴孔装配。

DRL 代表了一种新兴的智能控制方法，从根本上说，它基于通过与环境的试错交互来学习当前状态和预定义目标。该系统自主优化决策并执行最佳作。DRL 已经在机器人控制领域找到了广泛的应用，在机器人抓取、路径规划和焊接等领域取得了重大成果。在轴孔装配的背景下，轴孔形状和尺寸的复杂性和可变性使机械臂的运动控制成为一项具有挑战性的任务。单智能体 DRL 算法可以有效地处理相对简单的组装任务，例如组装完全对称的圆形轴孔。然而，对于更复杂的轴孔组件，需要更高的精度和效率，因此需要更精确地控制位置和方向。

多智能体深度强化学习（MADRL）是一种支持多个智能体之间协作的方法，通过组建一组智能体来协作完成任务，从而扩展单智能体强化学习。它在机械臂控制中的应用主要涉及协同控制和任务分配。团队中的每个智能体不仅必须确保内部一致性，还必须考虑与其他智能体的合作，从而在完成个人任务的同时将整个系统推进

到其目标。多个智能体之间的协作简化了复杂的任务，提高了控制精度和效率。然而，智能体数量的增加可能会带来数据维度爆炸、通信困难和收敛问题等挑战。此外，机械臂的关节和连杆之间的强耦合使控制更加复杂。

为了解决这些问题，本文探讨了多智能体深度强化学习算法在单个机械臂运动控制中的应用。针对复杂的轴-孔装配，设计了一种基于 DMDDPG 算法的轴-孔装配系统，以位置和方向为受控变量，有效提高不同装配环境下的装配效率和泛化性。针对任务目标和轴孔组装过程设计了全局奖励函数，为了减少智能体之间的耦合，为每个智能体独立设计了局部奖励函数。因此，所提出的算法可以利用全局和局部奖励来指导组装策略的探索和优化。与其他强化学习算法的比较仿真实验表明，所提算法在效率和可靠性方面表现出色。

2. 准备

2.1 多智能体深度强化学习算法

一种经典的多智能体算法是 MDDPG，它将 DDPG 算法扩展到多智能体环境，并引入了集中式训练-分散式执行框架。此框架中的每个智能体都是一个完整的 DDPG 模型。在训练期间，每个智能体的 Critic 网络使用所有智能体的作和状态，而 Actor 网络仅根据其观察到的状态更新其策略。因此，每个智能体的 Critic 网络都适合全局价值函数，而不是单个价值函数，从而允许每个智能体的策略朝着最优全局价值函数更新。

在多智能体深度强化学习中，协作探索过程由马尔可夫博弈（也称为随机博弈）描述。这个概念包含两个关键思想：首先，多智能体系统遵循马尔可夫特性；其次，游戏描述了智能体之间的关系。多智能体马尔可夫博弈过程定义为元组（N，S，A，T，R，γ），其中 N 表示智能体的数量;S=（x，o_1，o_2，…o_N）是关节状态的集合;o_i是智能体 i 的观察状态，x 是其他环境信息;A=（一个$_1$，一个$_2$，…，a_N）是联合作的集合;一个$_i$表示智能体 I 的工作;T∈[0,1]是 $S\rightarrow A\rightarrow S'$ 的状态转换概率;R=（r_1，r_2，…，r_N）、r_i表示智能体 I 执行作 A 所获得的奖励$_i$，这会导致结果 $o\rightarrow o'$ 。 γ 表示累积奖励的折扣系数。

在多智能体系统中，环境状态的变化是由所有智能体的集体行动引起的。在时

间 t，每个智能体体都结合其观察到的状态并执行 $\pi^{o_i^t}{}_i(a_i|o_i)$ 联合作，导致环境中的状态转换，并因所执行的作而获得预期奖励。$A^t=(a_i^t,a_2^t,\cdots a_N^t)\,r_i^t$

$$r_i^t=E[r_i^{t+1}|S^t=o_i,A_i^t=a_i,\pi_i] \tag{1}$$

每个智能体的 Critic 网络采用相同的输入参数：观察到的环境状态 S 的所有智能体，智能体的动作 A，以及相应的 Critic 网络参数，生成全局值函数作为输出：θ_i^Q

$$Q^H(S,a_1,\ldots,a_N)=E[R+\gamma Q^H(S',a_1',\ldots,a_N')] \tag{2}$$

其中表示所有智能体的联合策略。$H=\prod_{i\in N}\pi_i(a_i|o_i)$

每个智能体的 Actor 网络参数根据 value 函数通过梯度下降进行更新：θ_i^π

$$\nabla_{\theta_i^\pi}J(\theta_i^\pi)=E_{S,A\sim D}[\nabla_{\theta_i^\pi}\log\pi_i(a_i|o_i)Q_i^\pi(S,a_1,\cdots a_N)] \tag{3}$$

其中 D 表示体验重播缓冲区，每个元素都是一个元组（S，A，R，S′），其中包含更新的关节状态 S' 。

$$L(\theta_i^Q)=E_{S,A\sim D}[(Q_i^H(S,a_1,\cdots a_N)-y)^2] \tag{4}$$

Critic 网络参数使用反向传播进行更新，损失函数定义为：

$$y=r_i+\gamma Q_i^{H'}(S',a_1',\cdots a_N')|a_i'=\pi_i'(a_i|o_i) \tag{5}$$

2.2 机械臂装配控制技术研究进展

现代控制理论的发展逐渐将主动顺应性控制作为主要研究重点。这种方法使机器人能够根据对环境的感知来执行相应的控制策略，以完成装配任务，而无需专用的合规机制，从而提供增强的多功能性。Hogan 等人介绍了阻抗控制的基本原理，建立了机器人位置和接触力之间的映射。这项工作为随后的积极合规性研究奠定了基础，从而产生了许多适用于各种环境的改进控制策略。例如，Wang 等人设计了一个使用滑模控制的关节阻抗控制器，它有效地增强了关节的灵活性，使机器人能够更精确地执行轴孔组装任务。该控制器还解决了模型的不确定性并减少了冲击力，从而提高了机器人的抗干扰能力。Wu 等人提出了一种基于事件切换策略的柔性连接器控制方法，该方法能够实现从自适应控制到混合阻抗控制的平滑过渡。这种方法可

以提高受限和不受约束空间中的控制性能，确保顺利完成连接器装配任务。

计算机技术的应用和进步显著提高了机器人控制系统的计算能力，通过将人工智能与传统控制算法相结合，实现了机械臂组装方法的突破。出现了两种主要方法：基于图像的视觉感知和基于力传感器的触觉感知。例如，Song 等人模拟了人类对关键信息的视觉敏感性，并引入了一种信息过滤机制，以提高组件中特征识别和分类的准确性。Cong 等人利用机器视觉来纠正装配姿势，提高了装配不规则形状零件的成功率。罗等人通过将缺失特征的多视图感知与深度强化学习相结合，设计了一个组装预测网络，显著提高了组装效率和稳定性。Ortega-Aranda 等人在机器人作过程中收集了接触状态信息，并使用基于神经网络的模糊架构训练了双臂机器人，以实现类似人类的性能。尽管基于阻抗模型的控制策略在装配任务中表现良好，但它们受到需要轴孔接触力的精确模型和动态参数的精确识别的限制，这些参数可能会因实际作中的磨损和疲劳而发生变化，从而影响装配结果。

与传统的控制策略相比，深度强化学习不需要明确的模型构建或控制策略设计。它可以自适应地调整控制策略以适应不同的环境和任务要求，提供卓越的适应性和泛化能力。这使得学习更复杂的控制策略成为可能，从而实现更高效、更精确的轴孔装配。Inoue 等人证明了在复杂的轴孔装配中获得精确模型的困难，并提出了一种针对轴孔任务的强化学习算法。Leyendecker 等人通过模拟验证了深度强化学习组装策略在不确定环境中的有效性。Ding 等人将轴-孔组装过程分为两个阶段——孔搜索和插入——并开发了一种基于 DQN 的自学习组装算法。在轴孔装配中，轴孔复杂多变的形状和尺寸使机械臂的运动控制特别具有挑战性。虽然单智能体深度强化学习算法可以处理相对简单的任务，例如组装完全对称的圆形轴孔，但在复杂的轴孔组装中实现更高的精度和效率需要更精确地控制位置和方向。

2.3 多智能体强化学习的最新研究

多智能体深度强化学习（MADRL）涉及多个智能体之间的协作学习，通过组建一个智能体集合来协作完成任务，从而扩展了单智能体强化学习。在机械臂控制的前提下，MADRL 专注于协同控制和任务分配，其中每个智能体不仅确保自身能力的一致性，而且还与他人合作以实现整个系统的目标。Foerster 等人介绍了增强的智能体间学习和可微分智能体间学习算法，它们利用神经网络来近似值函数，促进智能体

间通信，并通过参数共享降低模型复杂性。Sukhbaatar 等人提出了通信网络算法，实现了智能体之间实时、快速的信息交换。Lowe 等人开发了多智能体深度确定性策略梯度算法，该算法通过将集中式训练与分散式执行方法相结合来扩展 DDPG 框架，显著降低了学习成本，并为大多数合作方法建立了范式。他等人提出了一种改进的 DDPG 算法，该算法将机械臂的关节视为多智能体系统中的决策智能体，从而能够为目标捕获进行运动规划。与传统算法相比，这种方法表现出卓越的求解速度和稳健性。虽然多个智能体之间的协作简化了复杂的任务，提高了机械臂的控制精度和效率，但也带来了数据维度爆炸、通信困难和收敛问题等挑战。此外，机械臂关节之间的强耦合使控制工作进一步复杂化。

3. 使用深度强化学习算法的机械臂轴孔组装

3.1 轴孔装配任务分析

轴孔装配过程可分为三个主要阶段：寻孔、对中和插入。寻孔阶段包括将轴从其初始位置向孔的中心线移动，到达装配孔附近。对中阶段包括调整轴相对于孔的位置和方向，以满足装配要求。插入阶段包括在达到所需的对中后将轴插入到指定的深度。在这些阶段中，对准阶段是最关键和最复杂的，因为它涉及各种接触状态，例如点接触、表面接触和线接触（当位置和方向不匹配时）。此阶段对于成功完成装配任务至关重要。

本研究调查了圆形和方形轴孔组件在对准阶段各种接触状态的影响，特别关注比较两者之间姿态控制的差异。在圆形轴孔组件中，坐标系 O_c 以接触面的中心为原点，以孔的中心为 z 轴。同样，坐标系 O_M 以轴的轴作为 z 轴建立。坐标系 O 之间 x、y 和 z 轴的偏差 $_c$ 和 O_M 分别定义为 Δx、Δy 和 Δz，而 Δl 表示两个坐标系的原点之间的距离。在对准阶段，可能会出现几种典型的接触状态，如图 1 所示。

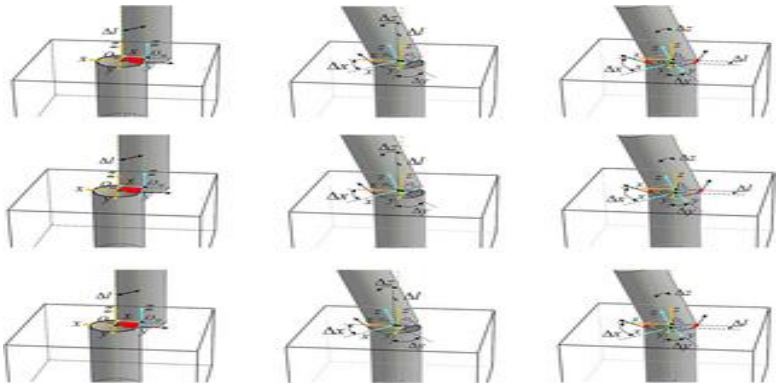

图 1. 圆形轴孔组件中的接触状态

(a)表面接触；(b)点接触；（c）多点接触。

在**图 1A** 所示的表面接触状态下，轴和孔的方向完全对齐，但存在位置偏差，即（Δx，Δy，Δz）=0，$\Delta l \neq 0$。通过调整轴的位置以将其直接对准孔上方，可以实现位置匹配，从而允许装配过程进入插入阶段。然而，在圆形轴孔组件中，更有可能发生点接触，如图 1B 所示。在这种情况下，位置偏差 Δl 几乎为零，但仍然存在明显的方向不匹配。由于圆形的完美对称性，轴孔的 x 轴和 y 轴可以绕 z 轴旋转，以满足任何方向的右手定则，这意味着只需对齐两个坐标系的 z 轴即可实现对齐。如果在点接触状态下没有及时调整方向以实现 Δz=0，则沿轴线继续插入可能会导致多点接触状态，如图 1C 所示。在这个阶段，轴的接触面原点已经在孔内，轴与孔壁碰撞，因此不适合进一步插入。

在方形轴孔组件中，坐标系类似地建立，接触面的中心为原点，z 轴由孔的中心轴和轴的轴定义。但是，x 轴和 y 轴不再具有任意定义;为了保持一致性，垂直于边的方向定义为 x 轴和 y 轴。位置偏差和方向偏差的定义与圆形轴孔组件中的定义相同。**图 2** 显示了方轴孔组件在对中阶段可能发生的几种接触状态。

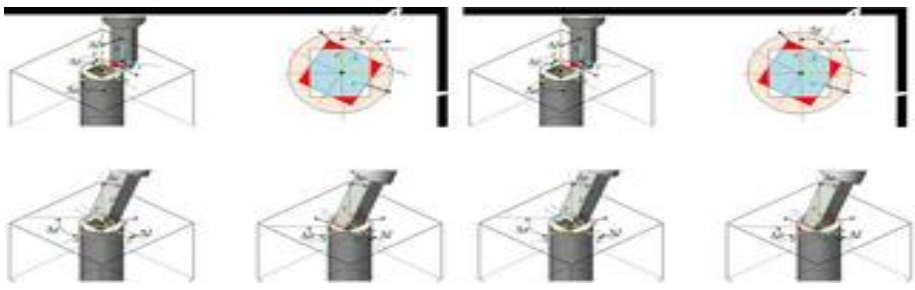

图 2. 方轴孔组件中的接触状态

（a）表面接触（b）多表面接触（c）线接触（d）多线接触。

在图 2A 中，表面接触状态显示只有轴和孔的 z 轴匹配，但位置偏差仍然存在。

当轴移动到孔的中心线以消除位置偏差 Δl 时，不能像圆形轴-孔组件那样忽略 x 轴和 y 轴偏差，从而导致图 2B 所示的多表面接触状态。此时，需要进一步绕 z 轴旋转才能 Δy=0 继续执行插入任务。图 2C 说明了线接触状态 Δy=0，$\Delta l \rightarrow 0$，其中其他两个轴尚未匹配，如果不及时纠正，将导致图 2D 所示的多线接触，导致刮擦孔壁。还有更复杂的接触状态，无法满足方轴和孔的装配要求。

总之，对于方轴孔装配，装配轴的接触面中心必须与孔的中心线对齐，表示为 Δl=0。关于方向，对于非对称的方轴孔组件，只有当沿所有三个轴的方向完全匹配时，才能进行下一个插入阶段，表示为（Δx，Δy，Δz）=0。

方形轴孔组件需要比圆形轴孔组件高得多的精度，因此需要开发专门的组件策略，以适应圆形轴孔组件的要求。对于圆形轴孔装配任务，各种成熟的智能控制算法，如 DQN 和 DDPG，可以有效地解决这些问题。然而，它们的控制策略并不适合方轴-孔组件所需的严格方向匹配。

根据前面的分析，这两种类型的装配任务之间的主要区别在于方向的控制。为了应对这一挑战，提出了一种独立控制位置和方向的策略，以降低方轴孔装配任务的难度。此策略引入了 MADDPG 算法来协调位置和方向的控制。但是，此算法通常用于独立实体之间的协作控制。在串行机械臂的情况下，关节表现出一定程度的耦合，MADDPG 的协作能力受到一定限制，这阻碍了其在精确装配任务中的有效性。为了克服这些限制，引入了 DE-MADDPG 算法。通过集成解耦模块，该算法减少了位置和方向控制之间的耦合，从而提高了整体装配性能。

3.2 机械臂智能体配置分析

装配任务的重点是机械臂，选择川崎 BA006N 模型作为控制对象。该工业机器人由通过六个旋转关节互连的刚性链节组成。研究手臂的运动状态涉及检查每个链接之间的相对运动关系，因此需要在笛卡尔坐标系中为每个关节建立坐标系。DH 模型通常用于描述关节坐标系之间的变化，分为标准 DH（SDH）和修正型 DH（MDH）。主要区别在于建立参考坐标的位置：SDH 在 i-1 链接的末端建立 i 坐标系，而 MDH 在 i 关节处建立 i 坐标系。为了更直观、更简洁地描述机械臂配置，本研究采用 MDH 模型进行建模，只需要四个参数来表示相邻关节坐标系之间的转换。

在此模型中：链接长度（a）是相邻关节轴之间公共垂直线的长度。链接扭曲角

度（ α ）是相邻接榫之间形成的角度，正方向由右手定则确定。链接偏移量（ d ）是前一个关节轴和当前关节轴的公共垂直线在当前轴上的投影之间的距离。关节角度（ θ ）是两个链接绕其公共轴的旋转角度。

i 关节的坐标系可以通过绕 x 轴旋转角度从 $i-1$ 关节的坐标系转换 α_{i-1}，沿 x 轴平移距离 α_{i-1}，绕 z 轴旋转角度 θ_i，最后沿 z 轴平移距离 d_i. 相邻坐标系之间的这种转换表示为：

$$T_{i-1}^{i} = Rot(\alpha_{i-1})Trans(a_{i-1})Rot(\theta_i)Trans(d_i)$$
$$= \begin{bmatrix} c\theta_i & -s\theta_i & 0 & a_{i-1} \\ s\theta_i c\alpha_{i-1} & c\theta_i c\alpha_{i-1} & -s\alpha_{i-1} & -d_i s\alpha_{i-1} \\ s\theta_i s\alpha_{i-1} & c\theta_i s\alpha_{i-1} & c\alpha_{i-1} & d_i c\alpha_{i-1} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (6)$$

其中 s 表示正弦，c 表示余弦，是坐标系的齐次变换矩阵，（α，a，θ，d）表示变换矩阵的 DH 参数。基于关节与各种数据之间的关系，使用 MDH 建立关节坐标系，每个关节的 DH 参数如**表 1** 所示，形成机械臂的初始配置。T_{i-1}^{i}

Link Numbering	α	a	θ	d
1	0	0	$\theta_1+\pi/2$	0.43
2	$-\pi/2$	0.165	$\theta_2-\pi/2$	0
3	0	0.55	θ_3	0
4	$-\pi/2$	0.21	θ_4	0.865
5	$\pi/2$	0	θ_5	0
6	$-\pi/2$	0	θ_6	0.115

https://doi.org/10.1371/journal.pone.0311550.t001

表 1. BA006N 机械臂的 DH 参数

将所有关节变换矩阵相乘，在基础坐标系和末端效应器坐标系之间生成齐次变换矩阵。由于坐标变换参数只涉及关节角度，反映了末端执行器在笛卡尔空间中的位姿与机械臂关节旋转角度之间的映射关系，因此定义为机械臂的运动学方程 F（ θ ）：

$$F(\theta) = T_0^1 T_1^2 \cdots T_5^6 = \begin{bmatrix} R & P \\ 0 & 1 \end{bmatrix} \quad (7)$$

其中 P 表示末端执行器在机械臂基础坐标系中的位置矩阵，表示为 $P=[p_x, p_y, p_z]^T$，R 表示末端执行器在 3×3 笛卡尔空间中的方向旋转矩阵，由沿主轴方向（x，

y，z）的单位向量组成的矩阵表示，表示为[n o a]。

$$R = [n \ o \ a] = \begin{bmatrix} n_x & o_x & a_x \\ n_y & o_y & a_y \\ n_z & o_z & a_z \end{bmatrix} \tag{8}$$

除了旋转矩阵之外，描述方向的其他方法还包括欧拉角和四元数。Euler 角度只需要三个角度来描述轴之间的旋转，但它们对旋转顺序有严格的要求，并且在连续姿势中可能会遇到万向节锁定问题。四元数将方向旋转定义为围绕特定旋转轴旋转指定角度，从而有效避免万向节锁定。它们只需要一个由一个标量和一个三维向量组成的四维向量来表示，与旋转矩阵运算相比，减少了数据存储，提高了计算效率，并能够平滑地描述连续姿势中的方向变化。四元数的表达式为：

$$q = \omega + ai + bj + ck \tag{9}$$

其中 ω 表示标量实部，而 i、j 和 k 构成虚构的三维向量，满足 $i^2=j^2=k^2=i\cdot j\cdot k=-1$，其中 ω、a、b 和 c 都是实数。计算四元数范数的公式为：

$$|q| = \sqrt{\omega^2 + a^2 + b^2 + c^2} \tag{10}$$

当四元数范数等于 1 时，它是一个单位四元数。除非另有说明，否则本文中描述的所有四元数都是单位四元数。为了明确描述方向的旋转关系，它们表示如下：

$$q = \left[\cos\left(\frac{\theta}{2}\right) \quad \sin\left(\frac{\theta}{2}\right)\vec{r}\right] = \left[\cos\left(\frac{\theta}{2}\right) \quad \sin\left(\frac{\theta}{2}\right)i \quad \sin\left(\frac{\theta}{2}\right)j \quad \sin\left(\frac{\theta}{2}\right)k\right] \tag{11}$$

其中 θ 表示旋转角度，表示旋转轴，表示旋转轴的方向矢量。$\vec{r}[i \quad j \quad k]$

旋转矩阵和四元数用于以不同的方式描述方向，并且可以相互转换。从机械臂运动学得到的旋转矩阵 R 被转换为四元数 q 来描述方向，变换公式为：

$$\omega = \frac{1}{2}\sqrt{1 + n_x + o_y + a_z} \tag{12}$$

$$\vec{r} = \frac{1}{2}\begin{bmatrix} sign(o_z - a_y)\sqrt{1 + n_x - o_y - a_z} \\ sign(a_x - n_z)\sqrt{1 - n_x + o_y - a_z} \\ sign(n_y - o_x)\sqrt{1 - n_x - o_y + a_z} \end{bmatrix} \tag{13}$$

利用方程（7）所示的正向运动学，可以获得装配轴端的位置和方向，并基于上一节提出的位置和方向分离控制的方法，对川崎 BA006N 机械臂进行了智能智能体的设计。BA006N 机械臂符合标准 Pieper 标准，其中三个相邻的关节轴要么在一个点相交，要么沿三个轴平行。对于这种类型的机械臂配置，最后三个关节构成机械臂

的手腕，主要控制末端执行器方向 R，而前三个关节主要控制手腕位置，从而影响末端位置 P 的变化。

在分离位置和方向控制的多智能体系统的上下文中，机械臂的前三个关节被定义为智能体 1，后三个关节被定义为智能体 2。多智能体轴组装的示意图如图 3 所示。

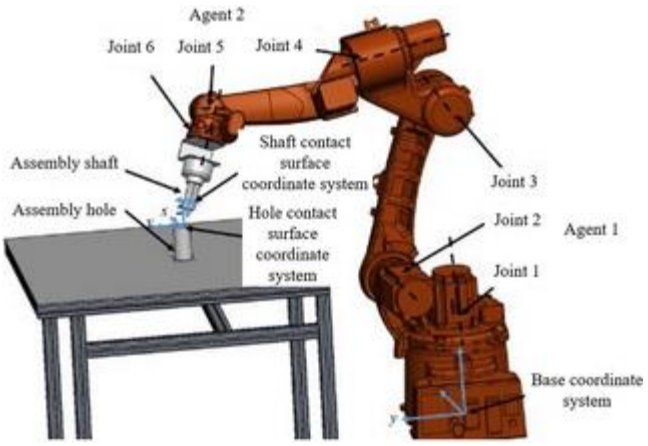

图 3. 多智能体轴组件示意图

3.3 解耦多智能体确定性深度确定性策略梯度算法

MDDPG 算法已成功应用于无人机和多机器人协作等领域。然而，在**图 3** 所示的串行机械臂的多智能体控制的背景下，这两个智能体之间的强耦合导致动作执行过程中产生显着的相互影响，加剧了合作任务中智能体之间的竞争关系，并阻碍了适当联合策略的稳定学习。

为了解决这个问题，提出了 DMDDPG 来减少智能体之间的耦合。DMDDPG 是 MDDPG 的增强版本，利用集中式训练和分布式执行框架。与 MDDPG 相比，它不仅包含一个用于集中培训的全球 Critic 网络，还包含一个本地 Critic 网络，旨在让每个智能体仅评估自己的本地状态和行动。这种设计允许每个智能体更多地关注自己的行为，从而提高算法的效率和稳定性。每个智能体的策略更新不仅考虑全局最优值函数，还考虑其自己的局部值函数是否正在向最优方向更新。这种解耦方法可以防止次优和不稳定的解决方案，以及策略中全局最优奖励在群体中的主导地位，确保每个智能体都能快速有效地学习最佳策略。

本地 Critic 网络的训练只需要当前智能体的观察和状态来评估其动作，通过

损失函数获得本地 Q 值：

$$L(\varphi_i) = E_{o_i,a_i\sim D}[(Q_i^{\pi}(o_i,a_i) - y_{\varphi})^2] \tag{14}$$

$$y_{\varphi} = r_i + \gamma Q_i^{\pi'}(o_i', a_i')|a_i' = \pi_i'(a_i|o_i) \tag{15}$$

其中 ϕ_i 表示 i_{th} 智能体。

在引入本地 Critic 网络后，对每个智能体的 Actor 策略网络的更新需要全球 Critic 网络和本地 Critic 网络的参与，以便在探索本地最优解决方案的同时探索全局最优解决方案。i 的策略梯度计算 $_{\text{th}}$agent 则变为：

$$\nabla_{\theta_i} J(\theta_i) = \underbrace{E_{S,A\sim D}[\nabla_{\theta_i} \log \pi_i(a_i|o_i) Q_i^H(S,a_1,\cdots a_N)]}_{\text{Global value function}} + \underbrace{E_{o_i,a_i\sim D}[\nabla_{\theta_i}\theta_i(a_i|o_i)\nabla_{a_i} Q_i^{\pi}(o_i,a_i)]}_{\text{Local value function}} \tag{16}$$

DMDDPG 算法的伪代码如**表 2** 所示。

Pseudo-code of DMDDPG Algorithm	
1	Initialize parameters of the global Critic main network Q_{θ}^{H}.
2	Initialize parameters of the global Critic target network $Q_{\theta'}^{H}$ and copy parameters Q_{θ}^{H}.
3	Initialize Actor main network parameters $Q_{\theta_i}^{\pi}$ and Critic main network parameters $Q_{\varphi_i}^{\pi}$ for each agent.
4	Initialize Actor target network parameters $Q_{\theta_i'}^{H}$ and Critic target network parameters $Q_{\varphi_i'}^{\pi}$ for each agent, copy $Q_{\theta_i}^{\pi}$ and $Q_{\varphi_i}^{\pi}$ parameters.
5	**For** episode = 1 to MaxEpisode **do**
6	Initialize random process $\mathcal{N}$, initialize environment state S.
7	**For** t = 1 to **do**
8	For each agent i, select action $a_i^t = \pi(o_i^t\|\theta_i^{\pi}) + \mathcal{N}$.
9	Execute action $A^t = [a_1^t, a_2^t]$.
10	Obtain global reward R_g^t, local reward r_i^t, and next state S^{t+1}
11	Store sample $(S^t, A^t, R_g^t, r_i^t, S^{t+1})$ into experience pool.
12	End for
13	Update global Critic network
14	Randomly sample M samples $(S^t, A^t, R_g^t, S^{t+1})$ from experience pool to form batch samples.
15	Calculate y from $y_i = r_i + \gamma Q^{\pi}(s_{t+1}, a_{t+1})$.
16	Compute loss $\frac{1}{M}\sum_i (y - Q_{\theta}^{H}(S^t, A^t))^2$ and update EvalCritic network Q_{θ}^{H} parameters.
17	Update target Critic network parameters $Q_{\theta'}^{H}$ in a soft update manner.
18	Update robotic arm agent Actor network and local Critic network.
19	**For** agent = 1 to N **do**
20	Randomly sample K samples $(o^t, a_i^t, r_i^t, o^{t+1})$ from experience pool to form batch samples.
21	Obtain y_{φ} from Eq (15).
22	Compute loss $\frac{1}{K}\sum_i (y_{\varphi} - Q_{\theta}^{H}(o_i^t, a_i^t))^2$ and update EvalCritic network Q_{φ}^{π} parameters.
23	Update EvalActor network $Q_{\theta_i}^{\pi}$, network parameters θ_i as follows: $\theta_i = \theta_i + \frac{1}{K}\sum_j (\nabla_{\theta^{\pi}} \pi_i(a_i\|o_i) Q_i^H(S, a_1, \cdots a_N) + \nabla_{\theta^{\pi}} \pi_i(a_i\|o_i) Q_i^{\pi}(o_i, a_i))$
24	**End for**
25	Update target Actor network parameters $Q_{\theta_i'}^{\pi}$ and target Critic network parameters $Q_{\varphi_i'}^{\pi}$ in a soft update manner.
26	**End for**
27	**End for**

https://doi.org/10.1371/journal.pone.0311550.t002

表 2.DMDDPG 算法的伪代码

3.4 多智能体动作状态空间定义

状态空间的设计在深度强化学习算法中至关重要，因为它定义了智能体需要观察的环境状态，使价值函数能够客观地评估动作的质量。在第 1 节对轴-孔装配任务

的分析中，主要通过比较轴和孔接触面之间的姿态来判断装配完成。装配孔固定在工作平台上，其在世界坐标系中的准确姿态是已知的。孔接触面的中心点位姿定义为，轴接触面的位姿信息由机械臂的正向运动学获得：。为了实现装配目标，还必须确保将轴插入到指定的深度 h。

$$F_G = \begin{bmatrix} R_G & P_G \\ 0 & 1 \end{bmatrix} F_M(\theta) = \begin{bmatrix} R_M & P_M \\ 0 & 1 \end{bmatrix}$$

在设计的多智能体算法中，关节状态空间 S 定义为：

$$S = (o_1, o_2) = (F_G, F_M(\theta), h) \tag{17}$$

$$\begin{cases} o_1 = (P_G, P_M, h) \\ o_2 = (R_G, R_M) \end{cases} \tag{18}$$

其中 o_1 表示智能体 1 的状态空间，它控制结束位置状态，其中该智能体还控制轴孔插入的深度。o_2 表示智能体 2 的状态空间，用于控制结束方向状态，仅关注装配过程中的姿势匹配。

通过 DMDDPG 算法，机械臂自主学习在面对不同状态空间时输出每个关节的运动角度，使装配轴能够顺利实现装配目标。因此，两个智能体的动作空间分别定义为关节 1-3 和关节 4-6 的旋转角度。联合动作空间 A 定义为：

$$A = (a_1, a_2) \tag{19}$$

$$\begin{cases} a_1 = (\theta_1, \theta_2, \theta_3) \\ a_2 = (\theta_4, \theta_5, \theta_6) \end{cases} \tag{20}$$

3.5 多智能体奖励功能设计

由机械臂控制的轴孔组件是连续动作空间中的复杂作任务。为完成任务目标而获得的奖励称为主线奖励。但是，在高难度的探索任务中，仅仅依赖主线奖励会使算法难以收敛或导致收敛缓慢，这种情况被称为稀疏奖励问题。为了克服这些问题，需要引入额外的奖励组件，使奖励函数更加密集。这有助于引导智能体更有效地探索环境，从而加快收敛速度并提高深度强化学习算法的性能。这些类型的奖励函数称为辅助奖励。根据轴孔装配过程中寻孔、对中、插入三个阶段的不同特点，设计了不同的阶段特异性奖励函数，主要分为子目标奖励函数和整形奖励函数。

3.5.1 主线奖励功能

主线奖励旨在为强化学习设定任务目标。在执行轴-孔装配任务时，只有在以下情况下才能获得奖励：（1）轴-孔姿态完全匹配，（2）装配轴轴线与孔中心线对齐，以及（3）轴插入深度达到指定深度，从而满足成功装配的要求。

在时间 t，每个智能体执行联合动作，计算距离 $S^{A^t}=[a_1^t,a_2^t]^{T+1}$ 轴接触面与孔中心线之间的轴-孔组件在状态 $\Delta\ \beta$ 下的姿态偏差 $\Delta\ 1$。如果未满足指定的装配精度要求，则执行下一个迭代循环，直到满足插入要求。在轴孔插入阶段，距离 $\Delta\ 1$ 和姿态偏差 $\Delta\ \beta$ 必须保持在误差范围（1，β）内。任务完成奖励 r_{vc} 当插入深度 h 时获得 $_s$ 达到指定深度 h，表示组装成功。组装任务流程图如图 4 所示。

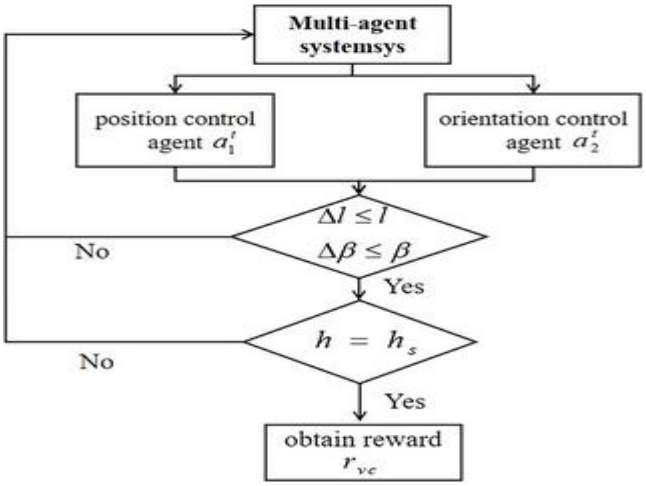

图 4. 程序集任务流

主线任务奖励可以定义为：

$$R_g=\begin{cases} r_{vc}(\Delta l \le l, \Delta\beta \le \beta, h=h_s) \\ 0 \quad otherwise \end{cases} \tag{21}$$

其中 R_g 仅在 Assembly 成功后反映 reward feed back，作为结束每个训练轮次的信号。如果 R_g 为 0，则智能体根据误差幅度 r 继续执行下一个作 r_{vc}=100 在训练期间。

3.5.2 子目标奖励函数

设置子目标奖励和其他辅助奖励主要是为了引导探索过程，缩小环境中智能体的无效探索范围，提高效率。在探索和完成主要任务的过程中，这个过程被进一步

分解为子目标，让智能体首先学会完成这些子目标。这种方法增加了后续探索主线任务的概率，最后，在主线奖励和辅助奖励的指导下，完成组装任务。为了提高装配效率，引入了惩罚函数，以确保快速安全地实现每个目标。三个子目标奖励函数主要设计如下：

1. 碰撞惩罚功能：精度在机器人轴和孔的组装中至关重要，在工业生产中严禁碰撞，在训练过程中应避免碰撞。为了执行这一原理，在机械臂和装配轴之间安装了测量力/扭矩的传感器，以检测实时接触力 F。在装配过程中，力或扭矩会因状态修改而缓慢变化，但碰撞会导致突然的峰值。通过设置阈值 $F_{林}$对于这些峰值，当接触力超过此阈值时，会检测到碰撞，从而终止当前勘探轮次并施加处罚。罚函数 RF 的定义如方程（22）所示，其中反馈值 r_{fl}=−100 在训练期间设置。

$$R_F = \begin{cases} r_{fl} & (F \geq F_{\lim}) \\ 0 & otherwise \end{cases} \tag{22}$$

2. 角度惩罚函数：集成到勘探过程中的持续奖励惩罚效应通常优于一次性奖励。长期积累的奖励轻松超越稀疏奖励，为算法提供更强的指导。这种方法提供与当前动作状态直接相关的即时反馈，有利于神经网络在中间学习过程中更好地提取特征。为了提高勘探效率并防止阻碍收敛的局部振荡，角度惩罚函数将每个节理的旋转角度作为定量目标。角度罚函数定义为 R_θ:

$$R_\theta = \ln\left(\sum_{i=0}^{6} |\theta_i| + 1\right) \tag{23}$$

其中每个关节的旋转角度 θ_iWithinoneStep 以弧度为单位进行测量。为避免阻碍收敛的初始高惩罚，每个步骤中这六个旋转角度之和的对数用作惩罚值。考虑到关节运动约束和安全稳定性，最大允许关节旋转角度为 3°，对应 $\theta_i \in [0, 0.052]$和 $R_\theta \in [0, 0.273]$。

3. 智能体装配虚拟空间：装配的任务空间是机械臂较大的作工作空间中一个相对较小的区域。让机械臂探索整个工作空间效率低下，并导致大量无意义的学习。此外，结合碰撞和角度惩罚可能会导致机械臂进入不正确的学习状态。为了更好地指导组装任务的完成并提高初始勘探效率，定义了一个球形虚拟空间，以组装孔接触面的中心为原点，如图 5 所示。在下一节中，将根据每个智能体的移动特征定义每个智能体的奖励函数。这涉及奖励的持续积累，鼓励智能体更接近任务目标以获

得更高的奖励。为了更快地收敛到装配任务区域，只有当轴位于定义的虚拟空间内时，才会累积奖励。在此空间之外，不会提供任何奖励。在虚拟空间中，比较位置和方向偏差，并提供正奖励以更有效地指导智能体成功组装。

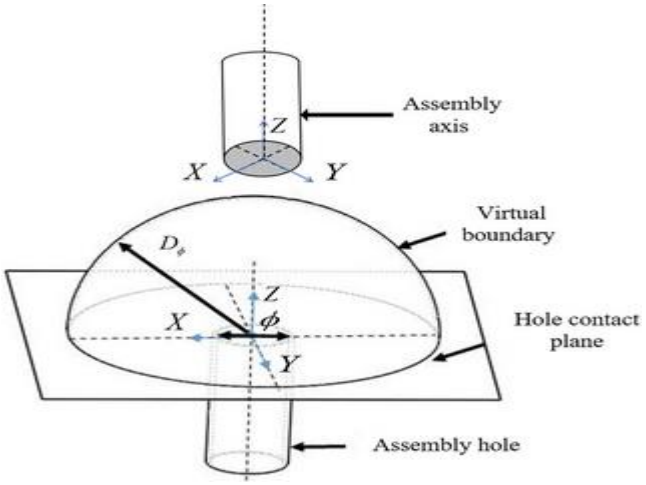

图 5.轴孔装配虚拟空间

虚拟空间由半径 VS 定义 $_\phi$，由装配孔的基本尺寸决定，$VS_\phi=\sigma\cdot\phi$。ϕ 表示装配孔的直径，σ 是根据轴装配体的尺寸和机械臂的装配环境调整的缩放系数。通过仿真实验发现，当系数过大时，初始奖励很高，但学习进度缓慢，导致最终收敛奖励较低，即次优。相反，如果系数太小，目标的初始探索时间会延长，并且进入时会阻碍对位置和姿势的及时调整，从而使最终收敛变得困难。通过测试，发现虚拟空间半径系数的稳定范围为 $\sigma\in$（1.5,3）。在此范围内，机械臂可以快速定位目标区域，并有充足的时间调整其位置和姿势。

3.5.3 塑造奖励函数

Shaping 函数也是辅助功能，与子目标奖励函数不同。它们的主要作用是根据智能体的状态添加一种势能函数。它们测量智能体在执行作后在新环境中的状态与目标状态之间的差距。差距越小，相应的奖励就越高，反之亦然。奖励功能需要根据每个智能体执行的任务和他们需要实现的目标来设计。在轴孔装配中，局部奖励函数是为控制位置和姿势的智能体单独设计的，称为局部奖励函数。

1. 基于位置的奖励功能：在方形和圆形轴孔的装配中，它们对位置匹配的要求是相同的。为了便于观察，研究者以圆形轴孔为例来设计一个基于位置的奖励函数。对于仅控制位置的智能体，确保轴接触面的中心正上方孔的中心至关重要。因此，中心点 P_M轴接触面的 X-Y 平面上，并且基于距离 L 的奖励函数 $_s$轴和孔之间设

计，如图 6 所示。此距离根据虚拟空间半径进行标准化，以确保它不会因装配孔和虚拟空间的大小而改变。奖励函数 r_L 对于水平距离定义为：

$$r_L = 1 - \frac{L_s}{VS_\phi} = 1 - \frac{\sqrt{(p_M^x - p_G^x)^2 + (p_M^y - p_G^y)^2}}{VS_\phi} \tag{24}$$

其中 L 的较小值 $_s$，表示距离较近，奖励较高，$r_L \in$（0,1）。

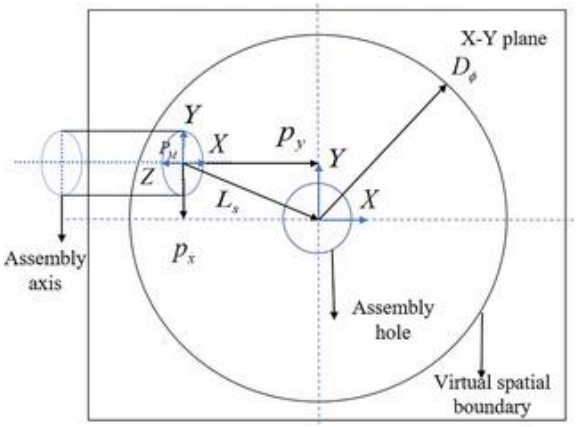

图 6. 轴孔接触面中心点之间的投影距离

同样，使用相同的方法定义 Z 轴方向垂直距离的奖励函数，以防止轴孔之间的碰撞，直到通过添加加权系数实现姿势匹配。主要计算是垂直距离 L_z 轴-孔接触面的中心点之间，如图 7 所示。仓位匹配阶段的奖励函数定义为 r_z:

$$r_z = \lambda\left(1 - \frac{L_z}{VS_\phi}\right) = \lambda\left(1 - \frac{|p_M^z - P_G^z|}{VS_\phi}\right) \tag{25}$$

其中 $\lambda \in$（0,1）是加权系数，$r_z \in$（0，λ）的 URL 中。当 λ 较小时，沿 Z 轴方向接近时获得的奖励值较小，并且智能体相对于 X-Y 平面运动的引导较弱，从而允许智能体优先考虑更靠近水平位置，为姿势调整提供更充足的时间。

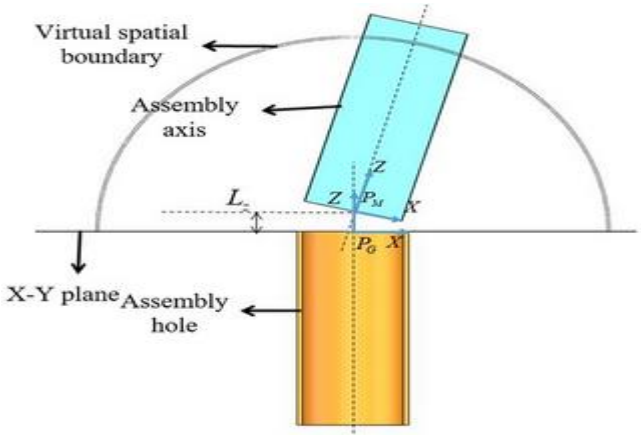

图 7. 轴孔接触面中心点之间的垂直距离

满足位置和姿势的要求后，插入阶段开始。在此过程中，姿势保持不变，在 X-Y 轴方向上没有移动。只需要观察插入深度是否 h_s 满足组装要求，插入过程如图 8 所示。

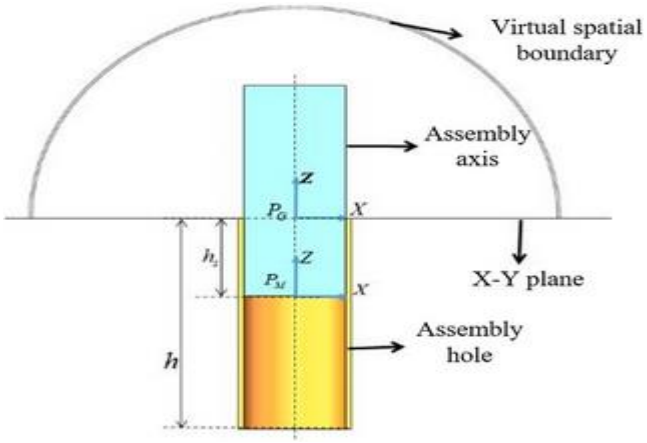

图 8. 轴孔插入

插入阶段奖励函数定义为：

$$r_{in} = e^{(1-|p_M^z - p_M^z|/h)} \tag{26}$$

其中，$r_{在}|p_M^z - p_G^z| = h_s \in$（1，e）。

这三个奖励函数共同构成了持仓奖励函数。根据轴孔装配工艺，可分为插入前和插入后两个阶段。位置奖励函数 R_1 是：

$$R_1 = \begin{cases} r_s + r_z, ((p_M^z - p_G^z) \geq 0) \\ r_s + r_{in}, ((p_M^z - p_G^z) < 0) \end{cases} \tag{27}$$

其中 r_s 和 r_z 每完成一步后，将在虚拟空间中累积奖励值；$r_{在}$ 将从位置和姿势达到要求开始累积奖励以执行插入阶段，并且 r_z 将不再计算此状态下的奖励值。R 的值范围 $_1$ 在这两个阶段中是：

$$R_1 \in \left\{ \begin{matrix} (0, 1+\lambda) if((p_M^z - p_G^z) \geq 0) \\ (1, e+1) if((p_M^z - p_G^z) < 0) \end{matrix} \right\} \tag{28}$$

2. 基于姿势的奖励功能：方孔的组装需要三个方向偏差量才能满足任务要求。使用旋转矩阵来表示 poses 与 correction 量之间的偏差，将显著增加算法计算量。因此，四元数用于描述姿势。通过 Eq（11），四元数可以转换为旋转矩阵。四元数仅使用四个元素来表示空间中的任何姿势向量，两个向量在相反方向上的最大角度为 R_2. 因此，姿势奖励函数 π 由四元数之间的角度定义：

$$R_2 = \pi - \cos^{-1}(|q_M \cdot q_G|) \tag{29}$$

其中 q_M=（ω_m， 一个$_m$， b_m， c_m）是轴接触面的位姿，q_G==（ω_g， 一个$_g$， b_g， c_g）是孔接触面的位姿;R_2∈（0， π），则从进入虚拟空间开始累积奖励值，直到组装完成。

综上所述，控制仓位的智能体的仓位奖励函数为 R_1，并且控制 posture 的智能体的 pose 奖励函数为 R_2.它们各自的奖励值范围不同。姿势调整是最关键的部分，需要智能体快速准确地调整插入姿势并保持它以获得更高的奖励。在沿孔中心线插入期间，鼓励姿势调整和位置接近，奖励重量较低，从而留出足够的空间和时间。接触后，位置奖励权重增加，鼓励智能体更快更好地完成组装任务。结合主线任务奖励 R_g、碰撞罚函数 R_F和角度惩罚函数 R_θ，基于深度强化学习的轴-孔装配的总体奖励函数为：

$$Return^t = R_g + R_F - \sum_{t=0}^{T} R_\theta^t + \sum_{t=0}^{T} \gamma^t (R_1^t + R_2^t) \tag{30}$$

其中 $Return^t$代表机器人获得的总奖励值， γ 是累积的奖励衰减系数。

4. 轴孔装配仿真实验

4.1 搭建机械臂仿真环境

4.1.1ROS 系统介绍

机器人作系统（ROS）是一个用于开发和控制机器人的开源系统。它提供了一种结构化的方式来构建复杂的机器人系统，其中每个功能模块都作为节点独立运行。这些节点由节点管理器（Master）管理，该节点使用 TCP/IP 通信在它们之间建立连接，从而实现分布式网络控制。节点通过发布（Talker）和订阅（Listener）消息进行通信，利用主题和服务等通信模型来促进 ROS 内不同范式下的信息交换。

4.1.2 Gazebo 物理仿真平台

Gazebo 是一个 3D 物理仿真平台，可快速构建机器人模型。为了增强模拟机器人的真实感，它与各种 2D 和 3D 设计软件（如 CAD 和 SolidWorks）连接。SolidWorks

中机械臂模型的详细设计使用 sw_urdf_exporter 插件将 3D 模型转换为 URDF 文件，适用于仿真环境中的真实建模。Gazebo 允许向模型添加物理参数，例如质量、摩擦系数、惯性矩阵和碰撞属性，使仿真环境更类似于真实世界的物理环境。此外，Gazebo 还支持机器人运动学仿真，验证构造机器人中的算法。为了进一步真实地匹配物理状态，Gazebo 包括一个全面的传感器模型数据库，其中包含摄像头、陀螺仪、扫描仪和其他可以直接添加和调用的常用传感器，支持根据任务要求创建新传感器。

4.1.3 Move It！运动规划器

Move It！是一个基于 ROS 的开源软件框架，用于机器人的运动规划和控制。它提供了一系列工具和库，包括运动规划器、碰撞检测器和运动控制器，有助于快速开发机器人应用程序。Move It！包的核心节点是 Moving Group，它本身虽然功能不丰富，但与其他独立的功能组件集成，提供 ROS 动作命令和服务。

为了在具有已知初始和目标姿势的环境中实现机械臂的运动控制，运动规划器会计算出适当的运动轨迹，将机械臂引导至目标姿势。在满足环境约束（如位置、方向和速度）后，规划器会计算中间状态。

4.1.4 构建机械臂仿真模型

为了在 Gazebo 仿真环境中构建机械臂模型，通常由 3D 软件生成的 URDF 文件用于构建。这些 URDF 文件以 XML 编写，主要描述机械臂的链接和关节。要在 Gazebo 中进行仿真，需要为每个链接添加物理属性，例如重量和惯性参数（惯性标签），并添加碰撞检测 3D 模型（碰撞标签）来检测碰撞。机械臂轴孔组件模型显示在 3D 可视化工具 Rviz 中，如图 9 所示。

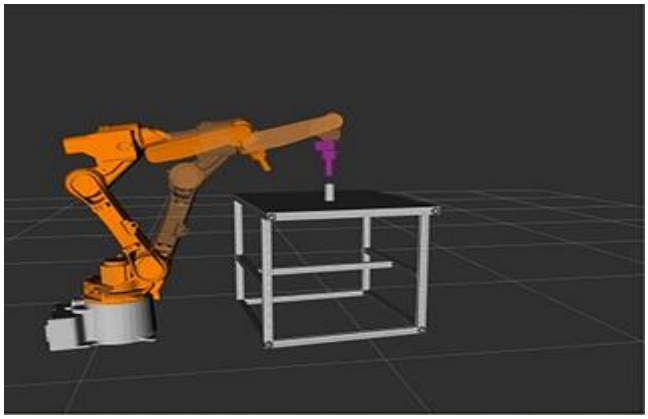

图 9.机械臂轴孔装配模型

4.2 装配模拟训练

4.2.1 网络训练参数

在 DMDDPG 算法中，每个智能体都有一个 Actor 网络和两个 Critic 网络。Actor 网络输入当前观察到的状态，并输出当前状态下每个可能动作的值，通常使用输出层中的 Tanh 激活函数在[−1,1]范围内缩放输出值。Critic 网络输入当前状态和动作，并输出智能体体的预期回报，即给定状态和动作的 Q 值。为避免训练困难和过度拟合，每个网络都设计为三层全连接神经网络。

用于轴孔组装任务的 DMDDPG 算法的训练参数如**表 3** 所示。

Training parameters	Numerical value
Cumulative reward discount factor γ	0.99
Actor network learning rate α^{π}	0.0001
Critic network learning rate α^{Q}	0.001
Target network soft update rate τ	0.01
Simulation time step ΔT	0.1
Batchsize	64
Number of Episodes (MaxEpisode)	15000
Virtual space ratio σ	2

https://doi.org/10.1371/journal.pone.0311550.t003

表 3. 网络训练超参数

4.2.2 组装实验对象

第 3.1 节分析了圆形和方轴孔的装配过程，并探讨了与方轴孔装配相关的更高难度。提出了一种基于 DE-MADDPG 的装配方法，该方法也适用于更简单的圆形轴孔装配。为了验证 DE-MADDPG 算法在单个机械臂中的可行性及其在处理不同难度任务中的稳定性，以圆形和方形轴孔作为测试对象进行了实验。

轴孔的具体结构参数如下：对于方形轴孔组件，使用边长为 25mm、组件间隙为 0.8mm、插入深度为 100mm 的方轴孔，如图 **10A** 所示。对于圆形轴孔组件，选择 40mm 的基座直径、0.8mm 的装配间隙和 100mm 的插入深度，如图 **10B** 所示。

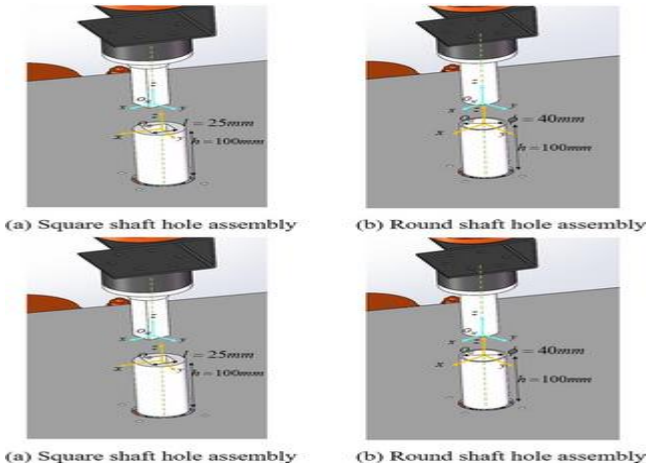

图 10.组装议题

（a）方轴孔组件；（b）圆形轴孔组件。

4.2.3 DE-MADDPG 轴孔组件的实验结果和分析

在为机械臂装配任务建立的仿真控制系统中，DE-MADDPG 算法用于圆形和方形轴孔装配的仿真训练。经过 15,000 次迭代的学习，两个装配环境的累积平均总奖励如图 11 所示。

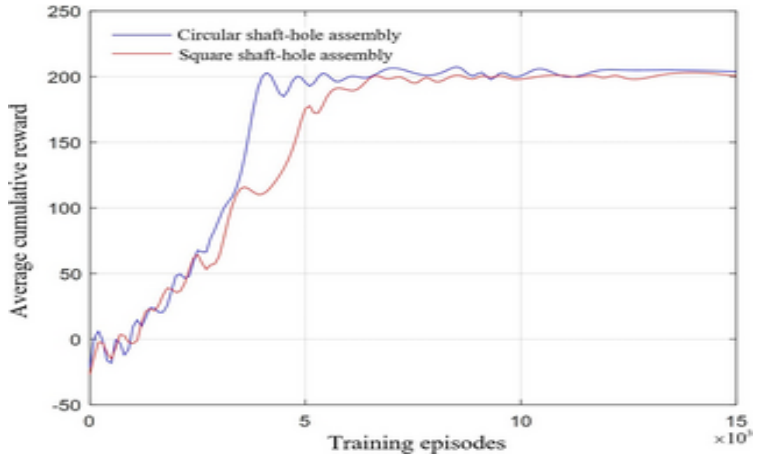

图 11. DE-MADDPG 算法在圆形轴孔和方形轴孔装配中的奖励曲线

该算法的奖励值在两个装配环境中收敛到稳定状态，证明了将单个机械臂分解为两个智能体（一个控制末端执行器位置，另一个管理方向）并使用 DE-MADDPG 算法协调它们的控制的可行性。

如图所示，在大约 3000 集的初始探索阶段，两个组装任务的奖励值曲线相似。在这个阶段，智能体主要学会了从初始配置开始通过空间探索来快速定位孔洞。在随后的对准和插入阶段，需要不太严格的方向控制的圆形轴孔组装任务的奖励值迅速增加并稳定在第 4,500 集左右，最终稳定的平均奖励为 202.52。相比之下，方轴

-孔组装任务需要多轮学习才能在正确定位孔后实现所需的姿态匹配。因此，奖励值在第 6,000 集左右开始稳定下来，最终平均奖励略低于圆形轴孔组件的奖励，为 199.46。

4.3 比较实验分析

4.3.1 DMDDPG 和 DDPG 的比较组装实验

DDPG 作为一种单智能体算法，在轴孔装配任务中得到了广泛而成熟的应用，但大多数装配对象主要涉及圆形轴孔。圆轴孔和方轴孔装配任务使用与 DMDDPG 算法相同的参数执行，以对比分析多智能体算法和单智能体算法在控制单个机械臂装配任务方面的性能差异。

DDPG 作为单智能体算法，使用整体环境 S 评估动作 A，并将总奖励值 R 反馈给智能体以更新动作策略。算法过程可以简化，如图 12 所示。

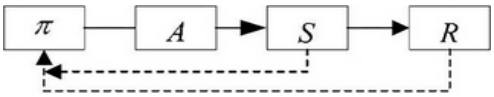

图 12. 简化的 DDPG 流程

为了直观地比较与 DMDDPG 算法的差异，根据**表 2** 所示的 DMDDPG 算法，将其简化为**图 13** 所示的算法流程图。

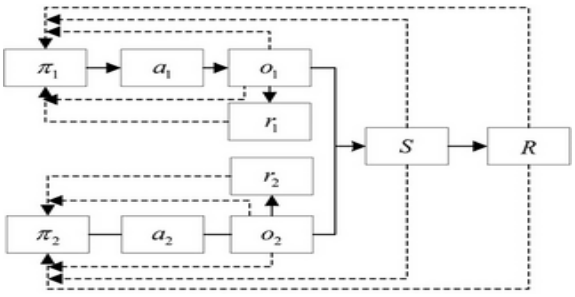

图 13. 简化的 DMDDPG 工艺流程图

总奖励值 R 包括本地奖励 r_1 和 r_2 的智能体，即本地奖励。在训练过程中，多智能体不仅使用整体环境 S 和总奖励值 R，还使用智能体自身的观察环境和状态 o_i，以及本地奖励值，共同指导智能体的策略更新。这是 DMDDPG 算法和 DDPG 算法的最大区别。

为了评估标准的一致性，DDPG 还使用方程（30）中定义的总奖励函数进行学习和训练，状态空间 $S=（F_G，F_M(\theta)，h）$ 和动作空间 $A=（\theta_1，\theta_2，\theta_3，\theta_4，\theta_5，$

θ_6).

DDPG 算法使用**表 3** 中概述的网络参数进行设计，适用于圆形和方形轴孔组件的训练。将 DDPG 算法获得的累积平均总奖励与使用 DE-MADDPG 算法获得的奖励进行比较，如图 14 所示。

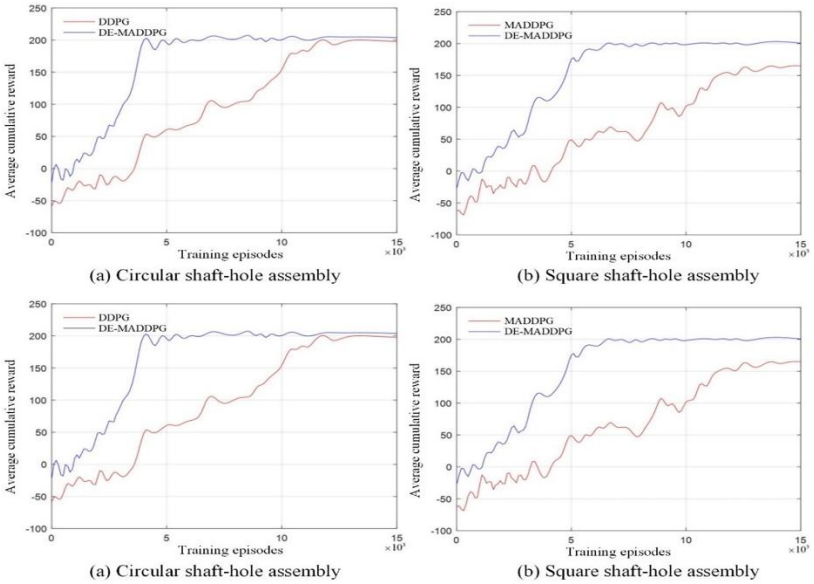

图 14. DMDDPG 和 DDPG 的累积平均总奖励价值

DDPG 算法的最终训练结果显示收敛到稳定状态。然而，在最初的探索阶段，它缺乏对位置和方向的准确控制，导致在空间中盲目作。因此，每次迭代奖励值比 DE-MADDPG 算法低约 40 个点，并且在探索的后期缓慢增加。

在圆形轴孔组件（**图 14A**）中，DDPG 算法开始稳定大约 12,000 次发作，与 DE-MADDPG 相比，需要的训练周期是 DE-MADDG 的三倍。最终奖励值收敛到 198.43，与 DE-MADDPG 差异最小，这表明两种算法都学习了相似的最优组装策略。对于方形轴孔组件（**图 14B**），DDPG 算法的奖励值也显示出 12,000 次发作左右的收敛趋势，但继续略微增加，直到训练期结束，并且尚未完全稳定。最终奖励值约为 165，比 DE-MADDPG 低约 35 分。

对比实验表明，DE-MADDPG 算法在训练效率方面具有明显优势。它在不同复杂程度的任务中保持稳定的性能，与 DDPG 算法相比，它在汇编任务中表现出卓越的效果。

4.3.2 DMDDPG 和 MDDPG 的比较组装实验

DMDDPG 算法基于 MDDPG 算法，旨在通过结合本地评估函数进行解耦来解决多个智能体之间的耦合问题。通过将 MDDPG 算法应用于相同的组装任务，研究者以比较

添加本地评估函数带来的性能变化。MDDPG 算法的训练过程被简化，如图 15 所示。

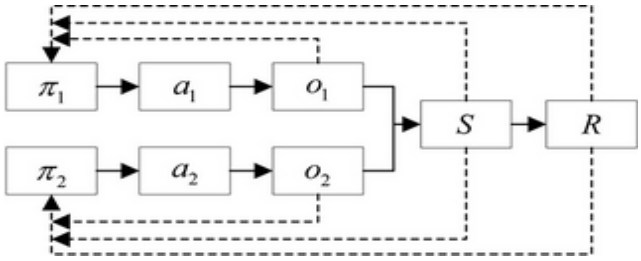

图 15.简化 MDDPG 流程

与**图 15** 所示的简化 DMDDPG 过程相比，唯一的区别是没有局部奖励函数评估。动作状态空间（A，S）和奖励（R）定义与 DMDDPG 算法一致。使用相同的网络参数，采用 MDDPG 算法进行圆轴孔和方轴孔的装配训练。累积平均总奖励值的比较如图 16 所示。

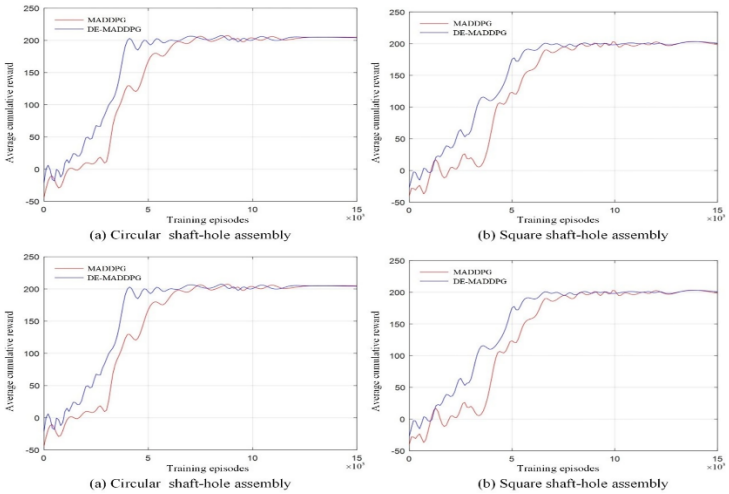

图 16.DMDDPG 和 MDDPG 的累积平均总奖励价值

如图 16 所示，DE-MADDPG 和 MADDPG 算法的最终收敛状态几乎相同，这证明了多智能体方法的优越性。然而，与 DE-MADDPG 相比，MADDPG 的初始探索奖励较低，并且在 3,000 集之前奖励价值的增加并不是特别显着。这是因为系统无法在仅全局奖励的指导下有效地将任务分配给各个智能体，再加上智能体之间有很强的竞争互动。因此，与 DE-MADDPG 相比，MADDPG 算法需要大约 3000 个额外的训练集才能达到收敛状态。在方形轴孔组件（**图 16B**）中，一旦实现收敛，耦合的存在会导致奖励值出现相当大的波动，这表明系统不稳定。DE-MADDPG 算法中解耦模块的引入使其能够更好地处理智能体之间强耦合的环境，提高训练效率，提高系统稳定性。

4.4 组装实验验证

方轴孔的装配训练使用了 3 种算法，均实现了稳定收敛。为了验证算法策略的有效性，使用 DDPG、MDDPG 和 DMDDPG 算法在仿真环境中对圆形和方轴孔进行了 500 次装配测试。测试结果如**表 4** 所示。

Assembly algorithm	Circular shaft-hole assembly			Square shaft-hole assembly		
	Time per assembly(s)	Standard deviation	Success rat%	Time per assembly(s)	Standard deviation	Success rat%
DDPG	56.45	6.27	83.8	62.86	7.85	76.2
MADDPG	52.71	4.38	89.2	57.02	5.43	84.6
DE-MADDPG	51.55	3.86	92.4	54.58	4.12	87.8

https://doi.org/10.1371/journal.pone.0311550.t004

表 4. 三种算法的汇编结果比较

表 4 中的标准差反映了汇编任务执行时间的可变性，值越小表示算法在不同环境中的性能稳定性越高。在圆形轴孔装配中，与 DDPG 算法相比，两种多智能体算法（DE-MADDPG 和 MADDPG）都证明了总装配时间的改进，分别提高了 8.7%和 6.6%。此外，对于多智能体算法，平均单次组装时间的标准差显着降低，表明组装过程的稳定性更高。对于方轴孔装配，DE-MADDPG 算法比 DDPG 算法有显著改进，单次装配时间缩短了 8.28 秒，装配效率提高了 13.2%。尽管与圆形轴孔组件相比，单次装配时间的标准偏差略有增加，但仍在可接受的范围内。在这两种组装环境中，多智能体算法始终实现 80%以上的组装成功率。值得注意的是，DE-MADDPG 算法在圆形轴孔装配中实现了 92.4%的成功率。

5. 装配试验台实验

5.1 测试台架的设置

5.1.1 测试台的组件

在这个实验中，选择配备了运动控制器的川崎 BA006N 机械臂作为主要控制对象。世界坐标系的原点设置在机械臂的底部。构建一个装配平台，平台中心点的坐标参数在世界坐标系内确定。装配对象为边长 25mm、装配深度为 100mm、装配间隙为 0.8mm 的方轴孔，位于装配平台的中心。测试台设置如图 **17** 所示。

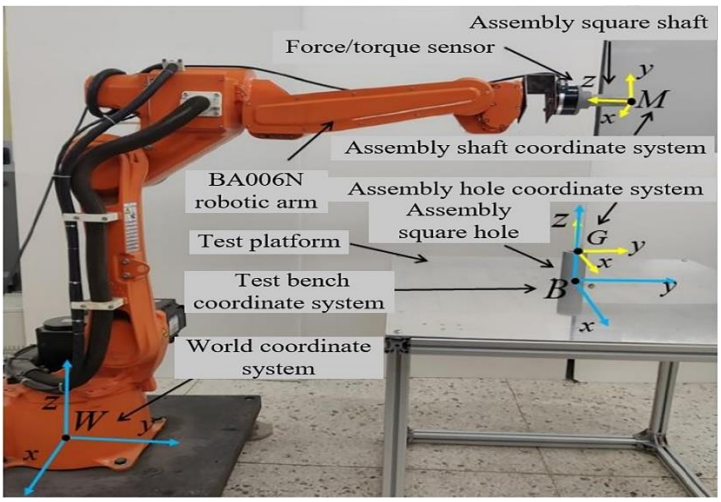

图 17.用于方轴孔组件的测试台

川崎 BA006N 机械臂的重复精度为±0.06 毫米，最大有效载荷为 6 公斤，适用于组装任务。在装配轴和机械臂之间安装了一个力/扭矩传感器，用于检测碰撞，碰撞阈值为 F_i 通过下一节讨论的接触实验确定。在每次训练期间，机械臂初始姿势的关节角度固定在，初始配置如图 17 所示。如图 17 所示，在装配轴端点的接触面上建立坐标系 M。位置和方向信息 $F^{\theta}=[0°,0°,0°,0°,0°,0°]_M$（ θ ）在初始状态下。

$$F_M(\theta)=\begin{bmatrix} R_M & P_M \\ 0 & 1 \end{bmatrix}=\begin{bmatrix} 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 1.3 \\ 1 & 0 & 0 & 1.19 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{31}$$

测试台安装在固定位置，通过机械臂的标定确定测试平台中心点在世界坐标系内的位置，建立台架的坐标系 B。装配孔安装在中心点，并根据结构尺寸在装配孔的接触点中心建立坐标系 G。位置和方向信息 F_G 然后确定世界坐标系中装配孔的接触点中心。

$$F_G=\begin{bmatrix} R_G & P_G \\ 0 & 1 \end{bmatrix}=\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1.2 \\ 0 & 0 & 1 & 0.85 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{32}$$

根据 3.2 节中的方向分析，为了便于计算，表达式 R_M 和 R_G 转换为四元数形式，从而产生方向 q_M 和 Q_G 或分别是轴和孔。

$$\begin{cases} q_M=(0.5\ -0.5\ 0.5\ -0.5) \\ q_G=(1\quad 0\quad 0\quad 0) \end{cases} \tag{33}$$

5.1.2 传感器碰撞阈值设计。

安装在装配轴和机械臂之间的力/扭矩传感器用于检测轴与周围环境之间的碰撞。表 5 中详细说明了传感器的最大范围和精度。

Force/torque	Range	Precision
$F_{xyz}(N)$	2000	1
$M_{xyz}(N{\cdot}m)$	60	0.01

https://doi.org/10.1371/journal.pone.0311550.t005

表 5. 力/扭矩传感器参数

在实验之前，传感器在空载状态下进行校准。一旦连接装配轴，它就会引入一定的负载力和扭矩。在机械臂的运动过程中，端点轴状态的变化将导致力和扭矩的变化。运动控制算法对每步中的允许旋转角度施加限制，以最大限度地减少小角度连续旋转期间加速度对力和扭矩的影响。因此，传感器观察到的力和扭矩变化仅与轴的当前状态有关。在正常运动期间，力和扭矩在单位时间内的变化率应保持在一定范围内。超过此阈值表示轴的状态突然变化，表明发生了碰撞。

碰撞阈值对于轴孔装配的成功至关重要。如果阈值设置得太低，即使与环境轻微接触也会触发碰撞惩罚并终止训练。这可能会阻止智能体学习有效的优化策略，从而使算法难以收敛。相反，如果阈值设置得太高，系统即使在接触后也可能继续探测，如果反馈调整不及时，会导致对组件轴孔造成不可逆的损坏。

在实际场景中，需要通过初步实验来设置合适的碰撞阈值。在装配过程的接近阶段，竖井可能会与工作环境发生碰撞，这在早期勘探阶段很常见。经过充分的学习，可以有效避免这些碰撞。在搜索和插入阶段，接触主要发生在轴和孔之间，这种接触一直持续到算法稳定下来。此外，需要特别注意方轴孔装配过程中的复杂条件。

为了评估不同的力条件，在各种接触状态下测试方轴孔。孔被放置在固定位置而没有固定，严重的倾翻被视为碰撞状态。进行了三个实验：（a）外墙接触：这模拟了搜索阶段竖井与环境之间的碰撞。(b)与孔表面接触：这对应于姿势对齐阶段。（c）与孔边缘的线接触：这发生在插入阶段，此时可能会与孔壁发生碰撞。这些接触状态的相应力和扭矩变化如图 18 所示。

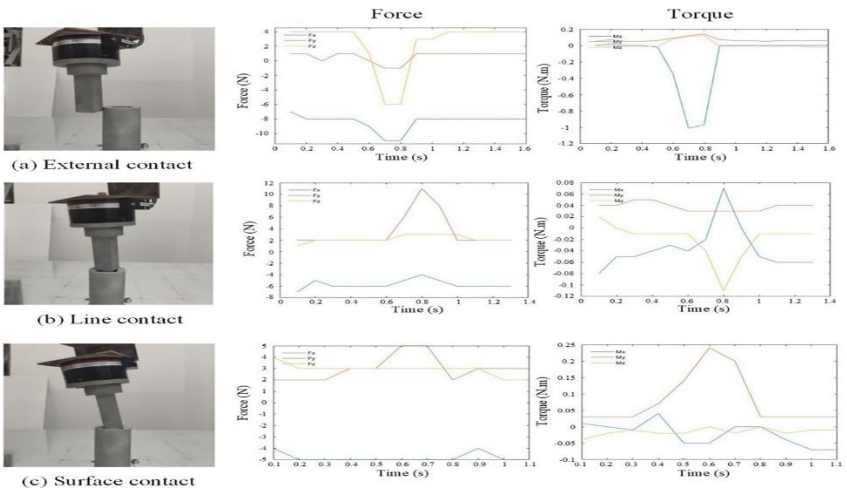

图 18.不同接触状态下的力变化

在外部接触状态下，力和力矩都会发生显著变化，最大变化为 ΔF_z=10N 和 Δm_x=1$N{\cdot}m$。在表面接触过程中，会发生位置和角度偏差，*其中 y* 轴力在 ΔF 处的变化最大 $_y$=11N，并且沿 x 轴和 z 轴的力矩发生显著变化，表示为 ΔM_x= ΔM_z=0.11$N{\cdot}m$。在观察到线接触的插入阶段，主要问题是角度偏差，力变化最小，力矩变化最大 ΔM_y=0.23$N{\cdot}m$。

根据上述分析，在装配过程中确保力和力矩都保持在安全范围内至关重要。为了防止机械臂运动引起的力和力矩变化导致系统误判，碰撞条件 F_{i} 设置为每单位时间步长的接触力和力矩的最大变化率，导致条件 $F_{\max}$=50N/s 和 $M_{\max}$=0.5$N{\cdot}m/s$。当所有三个方向上的力和力矩变化率都在可接受的范围内时，当前时间步长内的动作被认为是安全可行的。

5.2 实验结果分析

“Move it!”与川崎 BA006N 机械臂控制柜通过 TCP/IP 协议进行通信。DE-MADDPG 算法的网络参数如**表 3** 所示。根据仿真实验，在这两个装配任务中，收敛通常发生在 7000 次左右。为了节省时间和成本，最大剧集数设置为 10,000。DE-MADDPG 算法用于在安全条件下执行方轴-孔任务的 10,000 个训练组件，并将结果与仿真环境中的结果进行比较，如图 19 所示。

图 19. 实验和模拟训练的平均奖励曲线

尽管实验环境和理想化的仿真环境之间存在一些差异，但在实际训练中仍然实现了向稳定性的收敛。在最初的探索阶段，改进是缓慢的，因为算法会学习适应机械臂的动作环境。3000 集左右，奖励迅速增加。然而，在后期收敛阶段，由于机械运动过程中的系统误差和不稳定性，稳定性存在相当大的波动。尽管如此，奖励值稳定在 8000 集左右，最终平均奖励略低于模拟结果，最终为 182.35。

5.3 泛化实验

为了验证训练结果的可行性和推广性，通过改变工作区内装配孔的位置坐标来进行方轴孔的装配测试。装配训练由算法的 reward 函数中的位置和方向匹配指导。为了评估训练模型在不同任务条件下的性能，设置了四个实验：（1）实验 1：固定位置的装配，（2）实验 2：工作平面内不同位置的装配，（3）实验 3：固定位置但装配孔相对于工作平面倾斜 45° 的装配，（4）实验 4：倾斜 45° 和可变位置的组装。固定位置组件和 45° 倾斜组件如图 20 所示。每组进行了 100 次组装试验，根据空间关系计算位置和方向变化后接触面的中心坐标。测试结果总结在表 6 中。

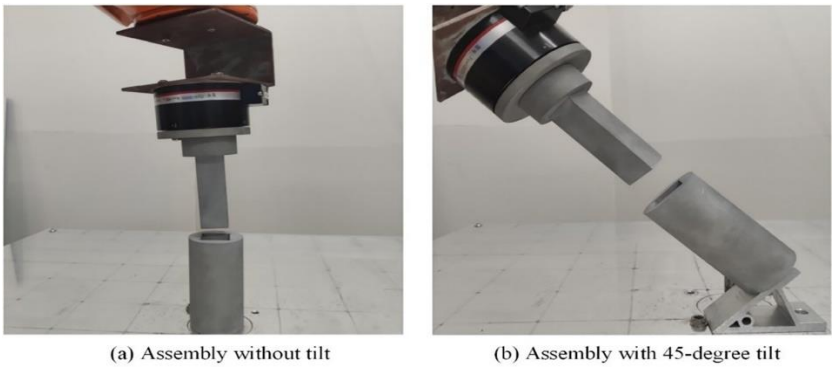

图 20. 方轴孔的装配实验对象

Experiment	DDPG		MADDPG		DE-MADDPG	
	Time per assembly(s)	Success rat%	Time per assembly(s)	Success rat%	Time per assembly(s)	Success rat%
Experiment 1	70.37	75	63.86	81	61.32	84
Experiment 2	83.14	71	75.07	76	72.89	81
Experiment 3	75.55	61	69.11	68	66.74	73
Experiment 4	92.66	52	83.69	60	79.24	66

https://doi.org/10.1371/journal.pone.0311550.t006

表 6.方轴孔组件的比较实验

比较实验 1 和 2,位置的变化导致运动轨迹改变,从而导致平均组装时间更长,从 61.32 秒增加到 72.89 秒，尽管成功率保持在 80%以上。在实验 1 和 3 之间，仅改变装配孔的方向，平均时间仅增加了 5.42 秒，但成功率从 84%下降到 73%。将实验 4 与实验 2 和 3 进行比较，可以明显看出，位置变化主要增加组装时间，而方向变化主要影响成功率。该实验证明了使用 DE-MADDPG 算法将单个机械臂划分为两个智能智能体的可行性，这些智能智能体控制装配任务的位置和方向。此外，通过改变装配孔的位置和方向，验证了算法在复杂环境中的泛化能力。在四个实验中，DE-MADDPG 的成功率比 MADDPG 高 3%、5%、5%和 6%，成功率比 DDPG 高 9%、10%、12%和 14%。结果表明，随着组装难度的增加，所提出的 DE-MADDPG 方法的精度优势变得更加显着。此外，DE-MADDPG 在所有四个实验中的平均组装时间都低于其他两种方法。

6. 总结

本文解决了轴孔结构的复杂装配问题，重点介绍了以川崎 BA006N 机械臂为主要控制实体的方形轴孔装配。研究包括对组装任务的分析、机械臂的运动学、多智能体深度强化学习框架的设计以及多智能体奖励函数的制定。仿真和实际实验均用于验证和测试。研究的主要发现如下：（1）DE-MADDPG 算法的奖励函数是针对汇编任务过程量身定制的，由全局奖励函数和局部奖励函数组成。全局奖励函数包括主奖励函数、冲突惩罚函数和进程惩罚函数。局部奖励分为位置奖励函数（用欧几里德距离表示）和方向奖励函数（用四元数表示）。（2）集成 Gazebo 和 Moveit 的联合仿真装配平台！是在 ROS 中开发的。该平台用于使用 DE-MADDPG 算法模拟圆形和方形轴孔的组装。仿真结果表明，将机械臂的前三个关节和后三个关节划分为多个智能体可以提高适应性。（3）基于仿真结果，川崎 BA006N 机械臂配备了力/扭矩传感器来检测碰撞,并构建了方轴孔实验训练台。训练结果显示收敛性与 DE-MADDPG 仿真结果一致。

尽管在开发基于 DE-MADDPG 算法的轴孔组装策略方面取得了进展，但仍存在一

些局限性和需要改进的领域，可以在未来的研究中进一步探索：（1）在真实环境中训练机械臂的成本很高。未来的研究可以探索使用迁移学习或元学习来利用模拟训练的模型，从而提高训练效率。（2）目前的研究主要集中在方轴孔的组装上。未来的实验可以验证和比较不同形状和大小的轴孔的组装。此外，测试阶段仅包括固定角度的验证。从不同角度进行组装实验可能有助于评估模型的泛化能力。（3）进一步优化奖励函数和装配任务可以增强系统在不同装配环境中的适应性，提高装配成功率。