arXiv:2301.13279v1 [cs.AI] 30 Jan 2023

通过循环神经调度传播，在异构图上为人机团队学习协调策略

Batuhan Altundas[1], Zheyuan Wang[1],Joshua Bishop[1] 和 Matthew Gombolay[1]

摘要——**随着人机协作在劳动力中日益普及，人机团队必须能够高效、直观地进行协调。传统的人机调度方法要么采用难以应用于大规模问题的精确方法，且难以考虑随机、随时间变化的人类任务表现；要么采用特定应用启发式方法，而开发此类方法需要专家领域知识。我们提出一种名为 HybridNet 的基于深度学习的框架，将异构图编码器与循环调度传播器相结合，用于在上、下界时间约束下调度随机人机团队。HybridNet 的编码器利用异构图注意力网络对初始环境和团队动态进行建模，同时考虑各项约束。通过将任务调度建模为序贯决策过程，HybridNet 的循环神经调度传播器利用长短期记忆网络（LSTM）模型传播动作的前向影响，从而快速生成调度，无需在每次选择任务—智能体对之间与环境交互。由此得到的调度策略网络计算开销低而表达能力强，可通过强化学习算法进行端到端训练。我们构建了一个用于多轮场景下混合人机团队的虚拟任务调度环境，能够对人类工作者的随机学习行为进行建模。实验结果表明，在人类表现确定和随机两种情况下，HybridNet 在不同问题规模上的表现均优于其他人机调度方案，且运行速度快于纯图神经网络调度器。**

I. 引言

随着协作机器人（cobots）在工业和制造环境中日益普及，机器人和人类越来越多地在同一工作空间中协作完成任务 [1]。通过移除传统机器人平台周围的围栏，并将机器人融入动态的最终装配作业，制造商能够缩小工厂占地面积、降低环境成本，同时提高生产力 [2]。本文聚焦于混合人机团队在同一任务分配问题的多次迭代中进行多智能体任务分配与调度的问题 [3]。我们的工作考虑并利用随时间变化的随机性人的任务表现，快速解决团队成员间的任务分配问题，从而根据特定应用的目标函数制定高质量调度，同时满足时间约束（即截止期限的上限和下限、等待约束以及任务持续时间约束）。

与多机器人系统中的任务调度相比，纳入人类工人会使调度更具挑战性，因为机器人可以被编程为以固定速率执行特定任务，而人类工人通常具有潜在的、动态变化且因任务而异的熟练度。人机团队要实现有效协作，就必须利用每位团队成员的独特能力，从而安全、有效且流畅地执行任务。对于这类问题，我们必须考虑人类随时间学习并提升任务表现的能力。要利用这一特性，调度算法必须推断人类潜在的表现特征，以决定是现在将任务分配给最优秀的工人，还是让一位速度较慢、但在该任务上有更大潜力达到流畅执行水平的人积累更多任务经验。然而，在确保团队满足必要的调度约束的同时推断人类的优势与劣势并非易事，因为不同个体在任务执行行为上的差异会带来不确定性，而实践中的学习效应也会使未来的任务表现难以预测 [4]。此外，若不考虑人类偏好和感知公平，从长远来看，高效行为与流畅协调之间可能会产生矛盾 [5]。

近年来，人机团队调度方法取得了显著进展，提升了在总装制造中动态协调大型团队的能力 [6], [7]。以往方法通常假设工人执行任务的能力是确定不变或静态的，并据此将调度问题建模为混合整数线性规划（MILP）；而该问题通常是 NP难的[8]。精确方法难以扩展规模，而且往往未能考虑多轮调度执行过程中人类工人随时间变化且具有随机性的任务能力，而这可能带来显著的生产力提升。启发式方法或许能够确定任务分配，但这类方法需要特定领域知识，而掌握这些知识往往需要数年时间。我们希望开发一种可扩展的算法方法，能够自动学习如何将人类行为纳入考量，从而实现快速、流畅的人机协作。

人工智能的进步催生了利用深度神经网络（DNNs）解决运筹学中大量问题的想法 [9]。深度神经网络可以通过训练，自动探索问题结构并发现高维数据中的有用表示，以构建高质量的解，而无需

*本研究部分经费由美国海军研究办公室资助（拨款号：N00014-19-1-2076），并由美国海军研究实验室资助（拨款号：N00173-21-1-G009）。[1]Batuhan Altundas、Zheyuan Wang、Joshua Bishop 和 Matthew Gom-bolay 均任职于美国佐治亚州亚特兰大市佐治亚理工学院机器人与智能机器研究所，邮编 30332 {baltundas3, pjohnwang, jbishop45, mgombolay3}@gatech.edu

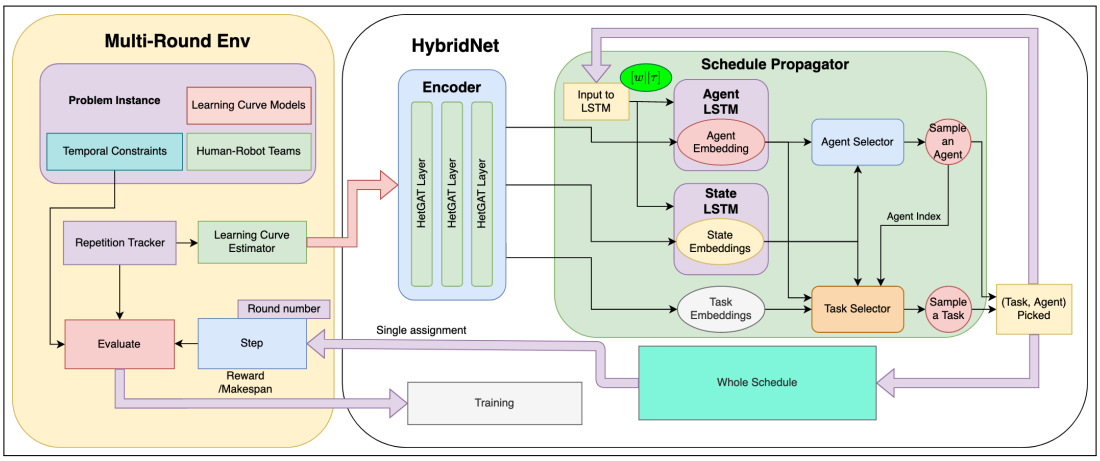


图1. HybridNet 调度器的多轮环境概览。左：多轮调度环境旨在模拟人机调度问题在多个迭代执行轮次中的情况，并考虑人的任务表现变化。右：HybridNet 由一个异构图编码器和一个循环调度传播器组成，前者用于提取问题的高层嵌入，后者用于快速生成调度。

手工设计特征工程 [10]。尤其是，通过模仿学习（IL）或强化学习（RL）利用图神经网络学习可扩展求解器，已取得令人瞩目的进展，其性能超过了最先进的近似方法 [11], [12], [13]。

为克服以往工作的局限性，我们提出了一种名为 HybridNet 的基于深度学习的框架，用于在时间约束下调度随机人机团队。图1展示了所提方法在多轮环境中运行时的整体框架。HybridNet 采用异构图编码器和循环调度传播器。编码器利用从简单时序网络（STN） [14] 扩展而来的问题异构图表示，提取调度问题的高层嵌入。通过将任务调度建模为一个序贯决策过程，循环传播器使用长短期记忆（LSTM）单元实现快速调度生成。由此得到的策略网络计算开销低、表达能力强，并且可通过强化学习算法进行端到端训练。

我们的工作主要贡献如下：

- 我们提出了一种基于深度学习的框架 Hybrid-Net，用于在时间约束下实现人机协调。HybridNet 由一个基于异构图的编码器和一个循环调度传播器组成。编码器提取有关初始环境的相关信息，而传播器则基于初始模型生成每个任务-智能体分配的后续模型。受大脑的感觉编码和循环处理机制启发，这种方法能够快速生成调度，无需在每次任务-智能体对选择之间与环境交互。

- 我们为多轮场景下的混合人机团队开发了一个虚拟任务调度环境，能够模拟人类工人的随机学习行为。我们使该环境兼容 OpenAI Gym，并希望将其作为测试平台，以促进人机调度算法的开发。该实现已公开提供。[1]

- 我们提出了一种新颖的策略模型，它能够联合学习如何选择智能体和任务，无需在中间调度决策之间与环境交互，并且只需在调度结束时获取一次奖励。通过将动作空间划分为智能体选择器和任务选择器，我们利用 HybridNet 实现了条件策略学习。选择智能体时，我们会考虑状态和智能体模型；进行任务分配时，则会综合任务、所选智能体和状态的相关信息。因此，HybridNet 可以通过策略梯度算法进行端到端训练。

- 我们针对一系列问题规模进行了大量实验，以验证 HybridNet 的性能。结果表明，在确定性和随机场景下，HybridNet 的表现始终优于以往的人机调度解决方案。

II. 相关工作

A. 多智能体调度问题

多智能体系统的任务分配与调度是一项优化问题，在现实应用中已有相关研究，既包括采用传统技术 [15] 和基于深度学习的技术 [16] 解决多机器人任务分配（MRTA）问题，也包括人机协作 [7]。任务分配可以通过混合整数线性规划（MILP）形式化，以表示其约束。求解 MILP 的指数级复杂度可以通过约束规划方法 [7], [17], [18] 或启发式调度器加以降低，从而提升可扩展性 [19], [20]。Zhang 等人将任务调度方案编码为遗传算法的染色体，通过反复交叉和变异这些解，为异构人机协作优化调度方案，以寻找最优调度方案。 [21]

[1]https://github.com/altundasbatu/HybridNet IROS 2022

Gombolay 等人提出了一种算法，通过调度策略捕捉领域知识，但需要领域专家示范 [22]。Wang 等人提出了 ScheduleNet，这是一种基于异构图神经网络的模型，用于在时空约束下进行任务分配，并通过模仿学习利用最优调度方案进行训练 [23]。ScheduleNet 依赖交互式调度方案，在生成完整调度方案之前持续更新环境。这些方法需要其他专家系统生成的最优调度方案作为训练数据，且计算复杂度较高，因此实施成本高昂。

B. 人机团队建模

随着机器人能力不断提升，机器人与人类协同完成专业工作时变得更加安全、有效。Liu等人提出了一个人类任务完成情况模型，表明学习能够提升任务效率。本文表明，对人类表现进行预测，可以增强调度系统明确推理智能体能力的能力 [4]。关于行为协作的先前研究，以及大规模调度优化在计算上的天然难解性表明，机器人可以通过为人类队友设计并调整调度方案提供宝贵服务。

在我们的系统中，我们借鉴Liu等人的研究成果，将人类随时间学习的过程纳入考虑：既在作为环境组成部分的问题生成中予以考虑，也通过作为调度策略组成部分的学习曲线预测器加以建模。正如公式l [4]所示，在多次迭代过程中，人类学习曲线遵循一种通用形式的指数函数：

$$y = c + ke^{-\beta i} \tag{1}$$

其中， i 表示人类此前执行某项任务的迭代次数， c、 k、 β 为参数。我们还在环境中考虑了人类学习的随机性。

C. 图神经网络

图神经网络（GNN）是一类深度神经网络，通过将对象表示为节点、将关系表示为边，并聚合相邻节点的信息，从非结构化数据中学习 [24]。GNN已广泛应用于节点分类、链接预测和聚类等基于图的问题，并展现出卓越的性能 [25]。Wang 等人提出的异构图注意力网络利用深度学习算法解决调度问题，与最早截止时间优先（EDF）[26] 和 Tercio [7] 等非深度学习调度器相比，性能有所提升，但代价是计算复杂度增加 [23]。

D. 基于 *LSTM* 的序列预测

长短期记忆网络的影响在语言建模 [27], 语音转文本[28],机器翻译 [29], 以及其他涉及预测建模 [30], [31], [32]方面尤为显著。通过神经网络的循环特性形成的这条较长路径，其优势在于有助于建立一定程度的直觉，这在流程的所有阶段都可能发挥作用 [30], [33]。

III. 人机调度问题 LEM

A. 问题概述

本文聚焦于具有时间约束的人机任务分配与调度问题 [15]。我们采用四元组 $\langle a, \tau, d, w\rangle$形式描述问题的组成部分。 a 表示人机团队中的所有智能体， τ 表示需要执行的所有任务。每个任务τ_i和智能体 a_j都有一个任务完成时长 dur（τ_i， a_j），且智能体能够按顺序完成一系列任务。 d 包含截止时间约束集合，其中 $d_i \in d$指定依赖于 τ_i [23]的任务。 w 是等待约束集合，其中 $w_{ij} \in w$ 表示任务 τ_i 与τ_j之间的等待时间。调度 S是由任务-智能体对 $\langle \tau_i, a_j\rangle$ 构成的序列，使得 S 包含 τ中的所有任务。

B. 多轮调度环境

多轮调度环境旨在模拟一个跨越多个迭代执行轮次的人机调度问题，并根据前几轮的情况考虑人类工人任务表现的变化。每一轮都是兼容 OpenAI Gym 的环境中的一个步骤，其输入为该调度问题的完整任务-智能体对集合，并模拟将任务顺序分配给智能体的过程。

当所有任务都已分配给某个智能体，或根据问题约束判定所提供的调度不可行时，即视为该轮执行结束。环境会根据问题约束检查所提供调度的可行性；如果调度可行，则计算完成该调度中所有任务所需的总时长。如果调度不满足约束，则判定其不可行，并返回无法调度的任务列表。

我们将多轮调度环境表述为一个部分可观测马尔可夫决策过程（POMDP），采用六元组 $\langle S, A, T, R, \Omega\ O,\ \gamma\rangle$ 如下：

- 状态：问题状态 S 是多轮环境的一个状态，由智能体的状态组成。
- 动作：多轮环境中第 t 轮的动作，是一个完整的任务分配集合，由按执行顺序排列的任务-智能体对列表组成，记作$A_t = [\langle \tau_{i_1}, a_{j_1}\rangle, \langle \tau_{i_2}, a_{j_2}\rangle, ...]$。
- 状态转移：T 对应于在多轮调度环境中执行该动作并进入下一时间步。
- 奖励： R_t 取决于用户希望优化的调度目标。在第 III-E 节中，我们将展示如何在优化完工时间时计算 R_t 。
- 观测：Ω是所有任务-智能体对的估计性能，以及可观测约束。

- 观测函数： O 由第*III-D*节中介绍的学习曲线估计器处理。
- 折扣因子， γ

C. 智能体模型

多轮环境存储智能体信息，使环境能够跟踪每个智能体及其先前完成的任务。环境在每轮结束时更新，使智能体能够根据其内部模型进行自我调整，并根据每轮选定的（任务-智能体）对更新模型。

1) 确定性机器人模型： 我们通过均匀分布随机生成机器人任务完成时间。

2) 随机人类模型： 我们根据公式1随机生成人类任务完成时间，使环境能够设置为针对人类学习提供确定性和随机性能。人类学习模型的任务持续时间参数 c、 k、 β在公式1中，根据第0轮随机选取的初始任务完成时间构建。对于随机性能，使用标准差从正态分布中进行采样，如Liu等人 [4]所述。

D. 学习曲线估计器

作为我们类似 OpenAI Gym 的环境的一部分，调度器通过学习曲线估计器 获取人类智能体在各项任务上的表现估计；该估计基于任务-智能体对先前执行该任务时的任务时长信息。在本文中，我们根据 Liu 等人提出的见解[4] 实现了一个黑箱模型，用于模拟随机人类学习估计器。当智能体在多轮中完成任务时，智能体模型会记录任务完成时长，使学习曲线估计器 能够更准确地预测下一次任务-智能体对的任务时长。为体现信息增加带来的准确度提升，我们实现了一个学习曲线估计器：它以实际任务表现作为高斯分布的均值来估计人类智能体的表现，并且该分布中的噪声会随着该智能体在先前轮次中重复执行同一任务的次数增加而呈指数级递减。

E. 奖励设计

多轮调度环境生成的调度方案的总奖励， R_t，根据任务分配的可行子集˜（ A'）和不可行子集 A'（˜）计算，满足 $A_t = A'_t \cup A'_t$。具体而言，奖励 R_t基于任务-智能体分配可行子集 R_t（A'_t）的预期奖励，以及任务-智能体分配不可行子集$R_t A'_t$中分配˜所获得的奖励.....该奖励根据将未完成任务分配给能够在最长时间内完成该任务的智能体所得奖励的点估计计算，并乘以不可行系数 C_i ，如公式 2 所示：

$$R_t = \sum_{i \in A'_t} R(\tau_i, a_i) + C_i \text{max}_{a_j} \left(\sum_{i \in \tilde{A}'} R(\tau_i, a_j) \right) \quad (2)$$

总调度奖励 R_S倾向于选择包含更多可行任务分配的调度方案，并使训练过程中能够从不可行探索中学习。

IV. HYBRIDNET调度策略

如图1所示，我们的 HybridNet 框架由一个异构图编码器和一个循环调度传播器组成：前者用于学习调度问题的高层嵌入，后者则按顺序生成团队日程。这种混合网络架构得以直接从问题结构中学习有用特征，这归功于异构图神经网络的表达能力；同时，它还借助基于LSTM的传播器高效地构建调度。因此，HybridNet 无需在每次任务-智能体对选择之间与环境交互，而这在先前工作中虽不可或缺，却会带来高昂的计算开销 [16], [23]。

我们将 HybridNet 学习到的策略表示为 $\pi_\theta(A|S)$，其中 θ 表示神经网络的参数。在第 t 轮，动作由一系列有序的调度决策构成， $A_t = \{d_1, d_2, ..., d_n\}$，$d_i = \langle \tau_i, a_j \rangle$，其中后续决策 d_i以先前的决策为条件，$d_{1:i-1}$。因此，策略可以分解为

$$p_\theta(A_t|S_t) = \prod_{i=1}^{n} p_\theta(d_i|S_t, d_{1:i-1}) \quad (3)$$

借助循环调度传播器，HybridNet 递归计算条件概率 $p_\theta(d_i|S_t, d_{1:i-1})$，以采样一个任务-智能体对。最后，网络收集所有决策，并发送到环境中执行。

A. 异构图编码器

我们使用 [23] 提出的异构图注意力（HetGAT）层构建编码器，该层已被证明在多智能体调度问题的表示学习中效果显著。对于给定的人机调度问题，在每一轮开始时，异构图表示由简单时序网络（STN）扩展而来；该网络对时间约束进行编码，并进一步纳入智能体节点和状态摘要节点。所得图的元图如图 2 所示，其中概述了所有节点类型和边类型。随后，HetGAT 层通过按边类型进行消息传递，再按节点类型进行特征归约，同时利用依赖于特征且不依赖结构的注意力机制，计算输出节点特征。有关 HetGAT 层实现的完整细节，请参阅 [23]。

通过依次堆叠多个 HetGAT层，我们构建了编码器，利用多层结构提取每个节点的高层嵌入，并将其发送给传播器以进行调度生成。我们遵循相同的

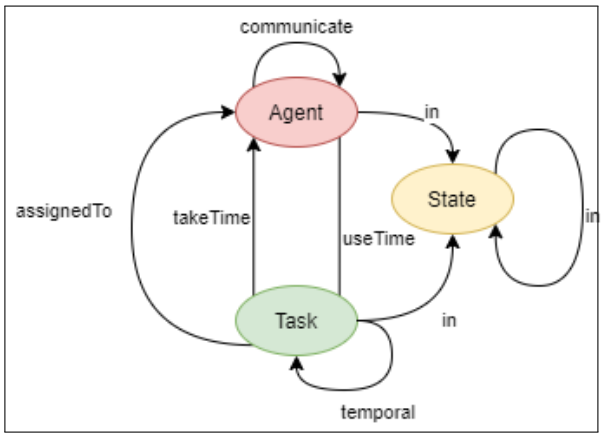


图2. 通过添加智能体和状态摘要节点，从STN构建的异构图的元图。

HetGAT层超参数设置，如 Wang 等人所提供的设置。 [23]

B. 循环调度传播器

HetGAT层的计算复杂度较高，并且需要交互式调度来生成初始模型。通过利用基于长短期记忆网络（LSTM）的循环预测器，我们可以向前传播每次任务-智能体分配所带来的后续影响，在不依赖初始HetGAT层的情况下重建环境的编码信息，从而显著降低调度器的计算复杂度。

循环调度传播器以异构图编码器生成的任务、状态和智能体嵌入为输入，并根据编码信息依次生成任务-智能体对。为了预测状态和智能体的后续编码，我们使用长短期记忆网络（LSTM）模型递归生成每次将任务分配给智能体之后的智能体和状态，无需与环境交互，并输出完整任务集的序列化任务-智能体分配。算法1给出了使用HybridNet生成调度的伪代码。

作为 $d_i = \langle \tau_i, a_j \rangle$，我们进一步将 $p_\theta(d_i|S_t, d_{1:i-1})$ 分解为智能体选择器和任务选择器。也就是说，$\pi_{factor}(d|\cdot) = \pi_{agent}(a_j|\cdot) \cdot \pi_{task}(\tau_i|a_j, \cdot)$。这种因式分解使策略能够捕捉调度决策内在的复合性和条件性，其中待调度的任务很大程度上取决于所选智能体。

智能体选择器根据状态和智能体信息，为下一次决策选择新的智能体 d。具体而言，将拼接后的状态-智能体嵌入输入前馈神经网络 f_a，并使用公式 4 计算为下一个任务-智能体对选择各智能体的可能性。随后执行 Softmax 操作，将原始预测转换为概率分布。选定智能体后，根据所选任务和状态嵌入更新该智能体的嵌入，因为只有被分配任务的智能体才会发生状态变化。这种方法使智能体选择器能够根据嵌入中固有的信息，考虑每个智能体的忙碌程度

算法1 调度生成伪代码**输入：** 图 g，特征 f，未调度任务 u**输出：** 调度

$schedule = [\,], i = 1$
$(h_{a_1}, c_{a_1}, h_{t_1}, c_{t_1}, h_{s_1}, c_{s_1}) \leftarrow Encoder(g, f)$
while $|u| \neq 0$ **do**
  $p_{a_i} \leftarrow AgentSelector(h_{s_i}, h_{a_i})$
  $a_i \leftarrow Sampling(p_{a_i})$
  $p_{t_i} \leftarrow TaskSelector(h_{t_i}, h_{s_i}, a_i)$
  $t_i \leftarrow Sampling(p_{t_{i-1}})$
  $schedule.append(\langle t_i, a_i \rangle)$
  $unscheduledTasks.remove(t_i)$
  if $|unscheduledTasks| == 0$ **then**
    return *schedule*
  end if
  $i \leftarrow i + 1$
  $h_{s_i}, c_{s_i} \leftarrow LSTM_s((h_{t_{i-1}}[t_i], h_{a_{i-1}}[a_i]), h_{a_{i-1}}, c_{a_{i-1}})$
  $h_{a_i}, c_{a_i} \leftarrow LSTM_a((h_{t_{i-1}}[t_i], h_{a_{i-1}}[a_i]), h_{a_{i-1}}, c_{a_{i-1}})$
end while

（这些信息）呈现于嵌入中。

$$\pi_{agent}(a_j|s) = softmax_i(f_a([h_{a_j}||h_s])) \tag{4}$$

接下来，调度传播器使用任务选择器，根据状态、智能体和未调度任务的嵌入，为所选智能体分配任务。如公式 5 所示，任务选择器将状态、所选智能体和未调度任务的嵌入进行拼接，并将组合后的信息输入前馈神经网络 f_τ，以计算该任务分配给所选智能体的可能性。将任务分配给智能体执行后，这些任务便会从未调度任务列表中移除。由于直到最后一次 Softmax 操作之前，每个任务的可能性计算彼此独立，因此该模型具有可扩展性，可用于不同规模的问题。

$$\pi_{task}(\tau_i|a_j, s) = softmax_i(f_\tau([h_{\tau_i}||h_{a_j}||h_s])) \tag{5}$$

调度传播器的关键组成部分是长短期记忆网络。如算法1第12行所示，每次进行任务-智能体对选择后，分别使用状态长短期记忆网络和智能体长短期记忆网络更新状态和智能体嵌入。长短期记忆网络单元会存储任务分配上一步的隐藏状态和单元数据，并根据输入使用公式 6 [33] 预测下一步。

$$\begin{aligned} f_t &= \sigma(W_f[h_{t-1}, x_t] + b_f) \\ i_t &= \sigma(W_i[h_t, x_t] + b_i) \\ \tilde{c}_t &= tanh(W_c[h_{t-1}, x_t] + b_c) \\ c_t &= f_t c_{t-1} + i_t \tilde{c}_t \\ o_t &= \sigma(W_o[h_{t-1}, x_t] + b_o \\ h_t &= o_t tanh(c_t) \end{aligned} \tag{6}$$

其中，编码器会以 $[h_1, c_1]$ 的形式生成初始隐藏状态 h_1 和初始单元状态 c_1。

在测试过程中，我们采用批量采样策略以进一步提升性能。具体而言，我们在每一轮针对同一任务分配问题生成多个调度方案。

我们利用学习曲线估计器计算预计完工时间，并据此选出表现最佳的调度方案，将其提供给多轮环境。增加采样次数可以提升解的质量，但也会增加计算量。

C. 随机策略学习

我们使用策略梯度方法，在多轮调度环境中训练HybridNet。该方法根据从环境收到的奖励 [34]，直接优化模型参数。具体而言，我们通过计算 Agent 和 Task Selector 对数似然之和来求取模型梯度，如公式 7 所示：

$$\nabla_\theta J(\theta) = \mathbb{E}_\pi(\sum_t^T A_t^{\pi_\theta}(s_t, \langle \tau_i, a_i \rangle) \nabla_\theta(log\pi_\theta(\tau_i|a_i, s_t) + log\pi_\theta(a_i|s_t)) \tag{7}$$

在公式 7 中，优势项 A_t 通过从公式 2 计算的未来总奖励中减去“基线”来估计。我们利用多轮环境中多个连续轮次执行的多个批次所产生的、针对同一任务分配问题的奖励来计算“基线”。批次中的每个元素都解决同一个调度问题，环境则会更新，以计入上一轮的任务分配，并更新智能体模型。我们根据公式 7 计算梯度，以更新模型权重。

由于任务调度问题具有组合性质，加之人类熟练度存在随机性，要学习一个有助于计算优势项的价值函数作为基线并非易事。因此，我们研究了两种更易实现且高效的替代方案：

- 基于步数的基线：在梯度估计过程中，所减去的基线值设为当前批次中训练 episode 的平均回报值。
- 贪心展开基线：贪心展开基线使用确定性的贪心版本 Hy- bridNet 调度器 $\pi_{greedy}(A|S)$，在环境中收集奖励。其权重 θ_{greedy} 会定期通过复制当前学习器 $\pi_\theta(A|S)$ 的权重来更新。

V. 实验结果

A. 数据生成

我们在不同规模下生成带有截止时间约束和等待约束的调度问题。对于所有规模，约 25% 的任务会从范围 $[1, 5N]$ 中随机生成截止时间约束，其中 N 为任务数。约 25% 的任务带有等待约束，非零等待约束的持续时间从 $U([1, 10])$ 中采样。任务时长被限制在 10 到 100 之间。

1) 小规模：小型数据集包含9至11项任务，由2名机器人和2名人类组成一个团队。我们生成了2000个训练问题和200个测试问题。

2）中规模：中型数据集包含 18 到 22 项任务，每个团队由 2 个机器人和 2 个人类组成。我们生成了 2000 个训练问题和 200 个测试问题，以检验所训练模型的可扩展性。

3）大规模：大型数据集定义为包含 36 到 44 项随机选取任务的问题，每个团队由 2 个机器人和 2 个人类组成。我们生成了 200 个测试问题，用于评估 HybridNet 在零训练问题条件下的性能（即从较小规模的数据集向大型数据集进行零样本迁移）。

为模拟人类智能体的随机学习，我们通过模拟公式 1中 c、 k、 β参数的自然分布，为每个数据集中的人类智能体模型引入噪声。这样，每个数据集都可以模拟确定性和随机性人类表现。随机模型会经过截断，以确保其落在指定的任务时长范围内。

B. 基准测试

我们将HybridNet与以下方法进行基准比较：

- *EDF*：一种广泛使用的启发式算法——最早截止期限优先（EDF）。该算法从可用任务列表中选取截止期限最早的任务，并将其分配给第一个可用的智能体。
- 遗传算法：一种进化优化算法，对EDF生成的调度方案进行后处理 [21]。遗传算法通过反复进行随机变异（交换任务分配和任务顺序），基于初始调度方案创建新的调度方案 [4]。每一代都会按照可行性和总调度完成时间对调度方案进行排序，并选出表现最佳的方案，作为生成新变异的基线。遗传算法运行10代，包含90个基线调度方案、10次任务分配变异和10次任务顺序交换变异。

此外，我们通过将循环调度传播器与以下HybridNet变体进行比较，评估其功能：

- *HetGAT*：我们基于 HybridNet 的编码器实现了一个HetGAT调度器。在每次进行任务-智能体配对分配后，它不再使用长短期记忆网络单元来更新任务、智能体和状态嵌入，而是直接与环境交互，通过新的异构图对动作的后果进行建模，并根据该图重新计算这些信息。

我们使用三个指标评估 HybridNet：1）问题求解比例；2）调整后完工时间：由可行调度的平均完工时间和不可行调度的最大可能完工时间共同确定；3）运行时间统计。我们比较 HybridNet 和 HetGAT调度器在训练和执行过程中的运行时间统计，以刻画它们的计算复杂度。由于 HetGAT调度器在每次进行任务-智能体配对分配后，都依赖通过环境进行交互式调度，因此我们仅针对确定性人类表现对其进行训练和评估。

C. 模型详情

我们使用 PyTorch [35]和深度图学习库 [36]实现 HybridNet 和 HetGAT。训练和测试中使用的 HybridNet 编码器由三个多头

表 I 评估结果：确定性人类任务熟练度下的调整后完工时间与可行性：比较基准与在小型和中型规模上训练的 HybridNet，调度方案从规模 8 和 16 中采样

训练	方法	小型		中型		大型	
		完工时间	可行性（%）	完工时间	可行性（%）	完工时间	可行性（%）
-	**EDF**	239.31	73.00	1109.85	15.00	2535.89	1.00
-	**遗传算法**	302.42 ± 0.77	74.10 ± 0.30	1180.07 ±2.54	16.60 ± 0.70	2542.79 ± 0.06	1.00 ± 0.00
基于步数	**HetGAT 8**	257.20 ± 0.18	86.29 ± 0.08	751.27 ± 1.29	50.17 ± 0.14	2123.96 ± 5.66	17.12 ± 0.27
	HetGAT 16	249.69 ± 0.30	86.51 ± 0.09	723.57 ± 0.94	50.29 ± 0.11	2081.65 ± 5.45	17.15 ± 0.16
贪心	**HetGAT 8**	261.15 ± 0.09	85.59 ± 0.10	784.32 ± 0.52	53.28 ± 0.17	2017年.25 ± 2.16	23.98 ± 0.14
	HetGAT 16	255.70 ± 0.23	86.05 ± 0.15	765.79 ± 0.96	53.41 ± 0.08	1983.73 ± 1.59	23.84 ± 0.01
基于步数	**HybridNet 小型 8**	260.22 ± 0.15	86.93 ± 0.10	770.48 ± 1.07	59.11 ± 0.35	2005年.80 ± 2.33	30.65 ± 0.39
	HybridNet 小型 16	**252.57 ± 0.49**	**87.08 ± 0.10**	746.35 ± 0.52	60.89 ± 0.36	1953.65 ± 3.76	33.24 ± 0.61
贪心	**HybridNet 小型 8**	266.74 ± 0.31	84.65 ± 0.32	758.96 ± 2.27	61.09 ± 0.43	2049.32 ± 3.73	28.74 ± 0.45
	HybridNet Small 16	258.17 ± 0.45	85.13 ± 0.20	723.35 ± 1.70	63.68 ± 0.49	1973.15 ± 2.91	32.46 ± 0.40
基于步数	**HybridNet 中型 8**	-	-	722.85 ± 0.61	64.69 ± 0.29	2010年.86 ± 1.97	30.86 ± 0.45
	HybridNet Medium 16	-	-	697.40 ± 2.04	66.25 ± 0.51	1944.72 ± 4.10	33.88 ± 0.49
贪心	**HybridNet 中型 8**	-	-	692.01 ± 3.69	68.33 ± 0.66	2011.78 ± 5.08	30.58 ± 0.87
	HybridNet Medium 16	-	-	**659.01 ± 0.89**	**71.00 ± 0.45**	**1936.97 ± 4.68**	**34.66 ± 0.74**

表 II 评估结果：随机人类任务熟练度下的调整后完工时间和可行性

方法	小型		中型		大型	
	完工时间	可行性（%）	完工时间	可行性（%）	完工时间	可行性（%）
EDF	227.81± 6.17	75.65 ± 1.21	1071.02± 20.65	17.30 ± 1.12	2524.92± 8.95	1.15 ± 0.23
遗传算法	283.79 ± 10.39	77.45 ± 2.05	1149.42 ± 12.14	19.55 ± 1.31	2541.20 ± 3.54	1.05 ± 0.15
HybridNet 小型	**298.81 ± 0.96**	**79.54 ± 0.52**	881.16 ± 2.89	48.89 ± 1.09	**2141.80 ± 5.12**	**23.51 ± 0.96**
HybridNet 中型	-	-	**859.99 ± 4.82**	**51.94 ± 1.32**	2174.57 ± 8.53	22.31 ± 0.94

表 III 评估结果：单个问题的运行时性能

	方法	HetGAT8	HybridNet8	HybridNet16
训练时间（秒）	**小型**	184.52 ± 18.00	**19.97 ± 0.91**	-
	中型	354.77 ± 38.31	**22.40 ± 6.52**	-
评估时间（秒）	**小型**	22.91 ± 5.85	**10.94 ± 0.99**	18.95 ± 3.53
	中型	70.12 ± 8.67	**14.77 ± 1.42**	22.30 ± 7.55
	大型	123.76 ± 32.32	**18.84 ± 7.38**	27.78 ± 16.52

HetGAT层堆叠而成（前两层采用拼接，最后一层采用平均）。隐藏层的特征维度= 64，头数为 = 8。循环传播器使用一个大小为 32 的 LSTM 单元，后接一个全连接层和一个 softmax 层。我们设置 $\gamma = 0.99$，批量大小为 =8，并使用 Adam 优化器 [37] ，学习率为2×10^{-3}，权重衰减为 5×10^{-4}。每经过 4000 个训练轮次，我们将学习率衰减为原来的 0.5 倍。我们使用批量大小为 8 和 16 对模型进行评估。对于多轮环境，不可行奖励系数为 $C_i = 2.0$ ，总轮数为 = 4。训练和评估均在 Quadro RTX 8000 GPU 上进行。

D. 评估结果

表 I 展示了不同规模下确定性人类熟练度条件下的评估性能。确定性人类熟练度是指，在训练和评估期间，人类学习曲线已知，且每个智能体的执行都是确定性的。表 I 中，模型名称后面的“小型”和“中型”表示模型训练所用的数据规模，其后的数字表示调度采样的批次大小。结果表明，HybridNet 在调整后完工时间和可行率方面均优于 EDF 和遗传算法。在小规模问题上训练的 HybridNet 能够泛化到中型和大型问题，其性能与在中型问题上训练的 HybridNet 相近或略差。HybridNet 和 HetGAT 在所有规模下的表现相近。这表明 HybridNet 能够学习高性能策略通过利用循环调度传播器，且无需与环境交互。

表 III 展示了 HetGAT 和 HybridNet 的训练与评估运行时间。与 HetGAT 模型相比，HybridNet 的训练速度约快 10 倍；在批量大小相同的情况下，评估速度至少快 2 倍。EDF 和遗传算法是在 CPU 上评估的，未使用 GPU 加速，因此无法准确比较深度学习模型与传统模型的性能。

我们展示了，对于 HybridNet，基于步数的训练优于贪心基线；而对于 HetGAT 模型，贪心基线训练效果更好。我们还观察到，贪心基线训练比基于步数的训练更快收敛（4500 轮训练次 vs. 19000 轮训练次）。值得进一步研究。

表 II 展示了不同规模下随机人类熟练度的评估性能。随机人类熟练度体现为多轮环境中实际人类执行过程的随机性，以及用于调度生成的学习曲线估计器所存在的不确定性。结果表明，在不同数据集规模下，HybridNet 的表现均优于 EDF 和遗传算法。在大型数据集上观察到的性能差距最大（23.51% vs. 1.15%）。此处未纳入 HetGAT 模型，因为它需要在每次任务-智能体分配后与环境交互以观察结果；而只有在生成完整调度并将其发送到随机环境中执行后，才能获得这些结果，从而模拟真实世界的情景。

VI. 结论

我们提出了一种名为 HybridNet 的基于深度学习的框架，它结合了基于异构图的编码器与循环式调度传播器，用于调度

具有时间约束的随机人机团队。由此得到的策略网络计算开销低，却具有很强的表达能力，可通过强化学习算法进行端到端训练。我们为随机人机团队开发了一个多轮任务调度环境，并开展了大量实验，结果表明，HybridNet 在不同问题规模下均优于其他人机调度方案。未来的研究方向包括将基于学习的人类估计器集成到 HybridNet 中、在优化不同目标函数之间进行迁移学习，以及将训练好的网络部署到现实场景中。

参考文献

[1] Z. Yan、N. Jouandeau 和 A. A. Cherif，《多机器人协调综述与分析》，International Journal of Advanced Robotic Systems，第10卷，第12期，第399页，2013年。 [在线]。获取地址：https://doi.org/10 5772 57313 。/ C. Heyer，《人机交互与未来工业机器人[2]应用》，载于2010 IEEE/RSJ International Conference on Intelligent Robots and Systems。IEEE，2010年，第4749–4754页。[3] E. Nunes、M. Manner、H. Mitiche 和 M. Gini，《具有时间和顺序约束的任务分配问题分类》，Robotics and Autonomous Systems，第90卷，第55–70页，2017年。[4] R. Liu、M. Natarajan 和 M. Gombolay，《动态且隐性的人员任务熟练度下的人机团队协调：结合学习曲线的调度》，2020年。[5] M. C. Gombolay、R. A. Gutierrez、S. G. Clarke、G. F. Sturla 和 J. A. Shah，《人机混合团队中的决策权、团队效率与人类工人满意度》，Autonomous Robots，第39卷，第3期，第293–312页，2015年。[6] E. Nunes 和 M. Gini，《通过多机器人拍卖分配具有时间约束的任务》，载于第二十九届 AAAI 人工智能会议，2015年，第2110–2116页。[7] M. C. Gombolay、R. J. Wilcox 和 J. A. Shah，《机器人团队执行具有时空约束任务的快速调度》，IEEE Transactions on Robotics，第34卷，第1期，第220–239页，2018年。[8] M. M. Solomon，《具有时间窗约束的车辆路径与调度问题中若干启发式算法的最坏情况性能》，Networks，第16卷，第2期，第161–174页，1986年。[9] Y. Bengio、A. Lodi 和 A. Prouvost，《组合优化的机器学习：方法论概览》，European Journal of Operational Research，第290卷，第2期，第405–421页，2021年。[10] Y. LeCun、Y. Bengio 和 G. Hinton，《深度学习》，Nature，第521卷，第7553期，第436–444页，2015年。[11] E. Khalil、H. Dai、Y. Zhang、B. Dilkina 和 L. Song，《在图上学习组合优化算法》，载于Advances in Neural Information Processing Systems，2017年，第6348–6358页。[12] W. Kool、H. van Hoof 和 M. Welling，《注意力：学习解决路径规划问题！》，载于International Conference on Learning Representations，2019年。[13] T. Ma、P. Ferber、S. Huo、J. Chen 和 M. Katz，《利用图神经网络和自适应调度进行在线规划器选择》，Proceedings of the AAAI Conference on Artificial Intelligence，第34卷，第04期，第5077–5084页，2020年4月。 [在线]。获取地址：https://ojs.aaai.org/index.php/AAAI/article/view/5949[14] R. Dechter、I. Meiri 和 J. Pearl，《时间约束网络》，Artificial intelligence，第49卷，第1-3期，第61–95页，1991年。[15] E. Nunes、M. Manner、H. Mitiche 和 M. Gini，《具有时间和顺序约束的任务分配问题分类》，Robotics and Autonomous Systems，第90卷，第55–70页，2017年4月，出版方版权：© 2016 Elsevier B.V.[16] Z. Wang 和 M. Gombolay，《利用图注意力网络学习多机器人协调的调度策略》，IEEE Robotics and Automation Letters，第5卷，第3期，第4509–4516页，2020年。[17] J. F. Benders，《求解混合变量规划问题的划分过程》，Numerische Mathematik，第4卷，第1期，第238–252页，1962年12月。 [在线]。获取地址：https://doi.org/10.1007/ bf01386316[18] H. Ren 和 L. Tang，《用于作业车间调度的改进型混合整数线性规划/约束规划算法框架》，2009 IEEE International Conference on Automation and Logistics，第890–894页，2009年。

[19] E. Castro 和 S. Petrovic，《放射治疗治疗前调度问题的数学规划与启发式算法结合方法》，《调度学报》第15卷，第3期，第333–346页，2011年5月。

[ ] 10 1007 10951 011 0239 8 在线。获[20]取地址：https://doi.org/。/s - - - J. Chen 和 R. Askin，《具有时变收益的项目选择、调度与资源分配》，European Journal of Operational Research，第193卷，第23–34页，2009年02月。[21] S. Zhang、Y. Chen、J. Zhang 和 Y. Jia，《基于人的能力的人机多机器人协作实时自适应装配调度*》，2020 IEEE International Conference on Robotics and Automation (ICRA)，第3860–3866页，2020年。[22] M. C. Gombolay，《人机团队的学徒制调度》，载于第三十届 AAAI 人工智能会议论文集，系列：AAAI'16。AAAI出版社，2016年，第2497–2498页。[23] Z. Wang、C. Liu 和 M. Gombolay，《用于具有时空约束的可扩展多机器人调度的异构图注意力网络》，Autonomous Robots，第46卷，第1期，第249–268页，2022年。[24] F. Scarselli、M. Gori、A. C. Tsoi、M. Hagenbuchner 和 G. Monfardini，《图神经网络模型》，Trans. Neur. Netw.，第20卷，第1期，第61–80页，2009年1月。 [在线]。获取地址：https://doi.org/10 1109 2008 2005605 。/[25] TNN.。K. Xu、W. Hu、J. Leskovec 和 S. Jegelka，《图神经网络的能力有多强？》，载于International Conference on Learning Representations，2019年。[26] J. Singh，《一种降低实时系统中最早截止期限优先调度算法时间复杂度的算法》，2010年。[27] M. Sundermeyer、R. Schl¨uter 和 H. Ney，《用于语言建模的LSTM神经网络》，载于第十三届国际语音通信协会年会，2012年。[28] A. Graves、S. Fern´andez 和 J. Schmidhuber，《用于改进音素分类与识别的双向LSTM网络》，载于人工神经网络国际会议。施普林格，2005年，第799–804页。[29] J. Zhou、G. Cui、S. Hu、Z. Zhang、C. Yang、Z. Liu、L. Wang、C. Li 和 M. Sun，《图神经网络：方法与应用综述》，2021年。[30] A. Sherstinsky，《循环神经网络（RNN）和长短期记忆网络（LSTM）的基础》，CoRR，第abs/1808.03314卷，2018年。 [在线]。获取地址：http://arxiv.org/abs/1808. 03314[31] P. Malhotra、A. Ramakrishnan、G. Anand、L. Vig、P. Agarwal 和 G. Shroff，《用于多传感器异常检测的基于LSTM的编码器-解码器》，arXiv预印本 arXiv:1607.00148，2016年。[32] A. Ycart、E. Benetos 等，《复调音乐序列建模中的LSTM网络研究》。ISMIR，2017年。[33] Y. Yu、X. Si、C. Hu 和 J. Zhang，《循环神经网络综述：LSTM单元与网络架构》，《神经计算》第31卷，第7期，第1235–1270页，2019年。[34] R. S. Sutton、S. Singh 和 D. McAllester，《策略梯度算法比较》，《IEEE系统、人类与控制论汇刊》，2000年。[35] A. Paszke、S. Gross、F. Massa、A. Lerer、J. Bradbury、G. Chanan、T. Killeen、Z. Lin、N. Gimelshein、L. Antiga、A. Desmaison、A. Kopf、E. Yang、Z. DeVito、M. Raison、A. Tejani、S. Chilamkurthy、B. Steiner、L. Fang、J. Bai 和 S. Chintala，《PyTorch：一种命令式风格的高性能深度学习库》，载于Advances in Neural Information Processing Systems，H. Wallach、H. Larochelle、A. Beygelzimer、F. d'Alch´e-Buc、E. Fox 和 R. Garnett 编，第 32 卷。Curran Associates, Inc.，2019 年。[36] M. Wang、D. Zheng、Z. Ye、Q. Gan、M. Li、X. Song、J. Zhou、C. Ma、L. Yu、Y. Gai、T. Xiao、T. He、G. Karypis、J. Li 和 Z. Zhang，《深度图学习库：一种以图为核心、高性能的图神经网络软件包》，2020年。[37] D. P. Kingma 和 J. Ba，《Adam：一种随机优化方法》，2017年。