用于多智能体导航的图增强指挥者-执行者学习

Xinyi Yang[1], Shiyu Huang[2], Yiwen Sun[3], Yuxiang Yang[1], Chao Yu[1,4], Wei-Wei Tu[2], Huazhong Yang[1†], Yu Wang[1†]1 清华大学, [2] 4Paradigm Inc., [3] 复旦大学, [4] 上海人工智能实验室

† 通讯作者
yang-xy20@mails.tsinghua.edu.cn

arXiv:2302.04094v1 [cs.RO] 8 Feb 2023

摘要

本文研究多智能体导航问题，该问题要求多个智能体在有限时间内到达目标。多智能体强化学习（MARL）在解决这一问题上展现出良好前景。然而，直接在大规模搜索空间中探索（近似）最优策略对 MARL 来说效率低下；随着智能体数量增加（例如 10+ 个智能体）或环境变得更加复杂（例如 3*D* 模拟器），这一问题会更加突出。目标条件分层强化学习（HRL）通过引入分层结构来分解搜索空间，为应对这一挑战提供了一个有前景的方向：低层策略在高层策略所导出目标的指引下预测原始动作。本文提出了多智能体图增强指挥官-执行器（MAGE-X），一种面向多智能体导航任务的基于图的目标条件分层方法。MAGE-X由高层目标指挥器和低层动作执行器组成。目标指挥器预测目标的概率分布，并利用这些目标为每个智能体分配最合适的最终目标。动作执行器利用图神经网络（GNN）为每个智能体构建一个仅包含关键协作对象的子图，以提升协作能力。此外，动作执行器中的目标编码器捕捉智能体与指定目标之间的关系，以促使智能体到达最终目标。结果表明，在包含 50 个智能体的多智能体粒子环境（MPE）中，MAGE-X 仅用 300 万训练步数便以 100% 的成功率超越了最先进的 MARL 基线；在更复杂的四旋翼无人机导航任务中，其成功率至少提高了 12%， 2× 数据效率也更高。

关键词

多智能体强化学习，目标条件强化学习，多智能体导航，图神经网络

ACM 引用格式：
Xinyi Yang[1], Shiyu Huang[2], Yiwen Sun[3], Yuxiang Yang[1], Chao Yu[1,4], Wei-Wei Tu[2]„ Huazhong Yang[1†], Yu Wang[1†], . 2023. 用于多智能体导航的图增强指挥者-执行者学习。载于第*22*届自主智能体与多智能体系统国际会议论文集（*AAMAS 2023*），英国伦敦，*2023*年*5*月*29*日至*6*月*2*日，IFAAMAS，共12页。

第*22*届自主智能体与多智能体系统国际会议论文集（*AAMAS 2023*），*A. Ricci*、*W. Yeoh*、*N. Agmon*、*B. An*（编），*2023*年*5*月*29*日至*6*月*2*日，伦敦，英国。 © 2023 国际自主智能体与多智能体系统基金会（www.ifaamas.org）。版权所有。

1 引言

导航是智能体系统中的一项典型任务，广泛应用于自动驾驶 [1, 8],物流与运输 [7, 23], 以及灾害搜救 [2, 17]等领域。本文研究一种多智能体导航任务，其中多个智能体以协作方式同时移动至目标点。多智能体强化学习（MARL）因其在多智能体导航任务中的强大表达能力而受到广泛关注 [13, 27, 42, 45]。

在多智能体强化学习中，寻找近似最优解的常见方法是直接训练一个策略，为智能体生成环境动作。然而，直接从大规模搜索空间中学习策略会导致数据效率低下；随着智能体数量或环境复杂度增加，这一问题会更加严重。因此，现有导航任务方法针对的都是智能体较少的简单场景。目标条件分层强化学习已被认为是解决这一问题的有效范式，它包含一个高层策略，将原始任务分解为一系列子目标，以及一个旨在抵达这些子目标的低层策略。近期目标条件分层强化学习研究主要聚焦于开发高层策略，为智能体提供合适的子目标。然而，指定子目标可能会使智能体难以确定应抵达哪些目标点。

图神经网络（GNN）因能够有效学习智能体的图表示，已广泛应用于多智能体协作任务。关于 MARL 中 GNN 的研究构建了一张图，其中的节点表示智能体的信息，以此对智能体之间的交互进行建模，并鼓励智能体开展协作。为增强智能体之间的协作，我们可以在分层强化学习（HRL）的低层策略中利用 GNN 捕捉智能体之间的关系，并表达对不同队友的偏好。

为提升数据效率和协作能力，我们提出多智能体图增强指挥器-执行器（MAGE-X），这是一种面向多智能体导航任务、基于图的目标条件分层框架。MAGE-X由两个组件构成：高层**目标指挥器**和低层**动作执行器**。在目标分配方面，**目标指挥器**会推断目标点的概率分布，为每个智能体分配最合适的目标点，而不是子目标。因此，多智能体导航被转化为多智能体环境中的多个单智能体导航任务，其中每个智能体都需要在避开碰撞的同时到达指定目标。在**动作执行器**中，我们利用图神经网络感知智能体之间的关系，并为每个

智能体生成一个子图，以决定其应关注哪些对象。之后，状态提取器接收来自目标编码器的智能体与其目标之间的关联性，以及该智能体在子图中的嵌入特征，从而赋予团队表示强大的目标引导能力，推动智能体到达目标点。我们在多智能体粒子环境（MPE）[24]（其中智能体数量庞大）以及更复杂的四旋翼导航任务 [31]中，将所提出的方法与多智能体强化学习基线进行比较。实验结果表明，MAGE-X优于多智能体强化学习基线，在包含50个智能体的MPE中仅用300万训练步数便达到100%的成功率。此外，在四旋翼导航任务中，MAGE-X的成功率至少提高了12%，数据效率也有所提升 2× 。

我们的贡献总结如下：

- 我们在多智能体导航任务中引入了一种基于图的目标条件框架——多智能体图增强型指挥器-执行器 （MAGE-X），以解决大规模搜索空间中的数据效率和协作问题。
- 我们开发了高层目标指挥器，该指挥器利用目标的概率分布，将每个智能体分配至最合适的目标。
- 我们提出了低层动作执行器，该执行器采用图神经网络（GNN）来提升协同能力，并通过目标编码器和状态提取器鼓励智能体完成任务。
- MAGE-X收敛速度更快，并且在具有大量智能体的多智能体粒子环境（MPE）[24] 和四旋翼无人机导航任务 [31] 中显著优于多智能体强化学习算法。

2 相关工作

2.1 导航

近年来，导航已得到广泛研究，强化学习（RL）展现出解决各种应用问题的能力 [18, 26, 43, 47]。例如，Rao [33] 提出了一种面向多目标视觉导航任务的嵌入式模型行动者-评论家架构。该嵌入式模型包含两个辅助任务模块：一个路径闭环检测模块，用于判断是否经历过某一状态；另一个状态-目标匹配模块，用于区分状态与目标之间的差异。此外，Zhu [47] 提出了一种目标驱动的行动者-评论家模型，以提升其在目标驱动视觉导航任务中的适应性和灵活性。

对于 MARL 中的多智能体导航任务 [13, 27, 42, 45]，其难点在于大空间中的数据效率和协作能力；随着智能体数量增加以及环境变得更加复杂，这些难点会进一步加剧。EPS [27] 被引入以提升多机器人无地图导航中的探索效率并改善样本效率。该方法利用由机器人策略定期生成的进化种群，搜索不同且新颖的状态。此外，Xia [42] 提出了一种基于推理的分层强化学习框架（IHRL），通过高层推理与低层动作之间的交互来解决多智能体协同导航问题。所提出的 MAGE-X 专注于导航任务，并凭借较高的样本效率优于多智能体强化学习基线。

2.2 目标条件分层强化学习

目标条件分层强化学习（HRL）已在广泛的任务中展现出其能力 [3, 29, 32]，其框架由高层策略和低层策略组成。高层策略在每个全局时间步生成中间子目标，这些子目标被视为对低层策略的目标引导。

Kreidieh [19] 通过引入层间协作来应对高层智能体与低层智能体之间交互所带来的挑战。这种层间协作通过修改高层策略的目标函数及其后续梯度来实现。另一个代表性方法是 LGA [4],，它将子目标分配参数化为待训练的潜变量。LGA 根据潜变量直接为智能体提供原始动作，以完成多智能体任务。MASER [11] 依托经验回放缓冲区中的数据，结合个体Q值和总Q值自动为智能体生成子目标。此外，该方法为每个智能体采用个体内在奖励，使其达到分配的子目标并最大化联合动作价值。然而，由于子目标与最终目标之间的对应关系是隐式的，子目标的生成可能效率低下。本文中，MAGE-X 在多智能体导航任务中采用目标条件分层框架来提高样本效率，其中高层策略负责目标分配，低层策略负责每个智能体的动作执行。

2.3 图神经网络

图神经网络（GNN）[37] 因能够有效学习图表示并捕捉不同图之间的关系而得到广泛应用。近年来，一些研究将图神经网络应用于多智能体系统，以对智能体之间的交互进行建模。HAMA [36] 提出了一种分层图注意力网络，用于捕捉智能体层级和群体层级的内在关系，从而增强泛化能力和可扩展性。MAGIC [30]是一种新颖的图通信协议，它隐含地表示智能体交互的拓扑结构，帮助智能体决定何时与谁通信。DICG [22] 利用动态协调图隐式推断联合动作和价值。GCS [35] 通过将联合团队策略分解为图生成器和基于图的协调策略来学习协调行为。图生成器捕捉智能体之间潜在的动态相关性，再由基于图的协调策略加以利用。MAGE-X 的动作执行器借助图神经网络捕捉团队表示，并引入目标编码器和状态提取器，以加强对目标点的表达。

3 预备知识

3.1 去中心化马尔可夫决策过程

本文考虑一种马尔可夫决策过程的变体，即去中心化马尔可夫决策过程（Dec-MDPs），以解决随机环境中的去中心化控制问题。

在此，多智能体去中心化马尔可夫决策过程问题的形式化定义如下：

$$< \mathcal{N}, \mathcal{S}, \mathcal{A}, \mathcal{T}, R, G, O, \gamma >, \tag{1}$$

其中，N ≡{1、...、 n} 是由 $N = |\mathrm{N}|$ 个智能体组成的集合。请注意，在假设 S 可联合观测的情况下，S 是全局状态的集合。A 是每个智能体的动作空间，而 A ≡ A^N 是联合

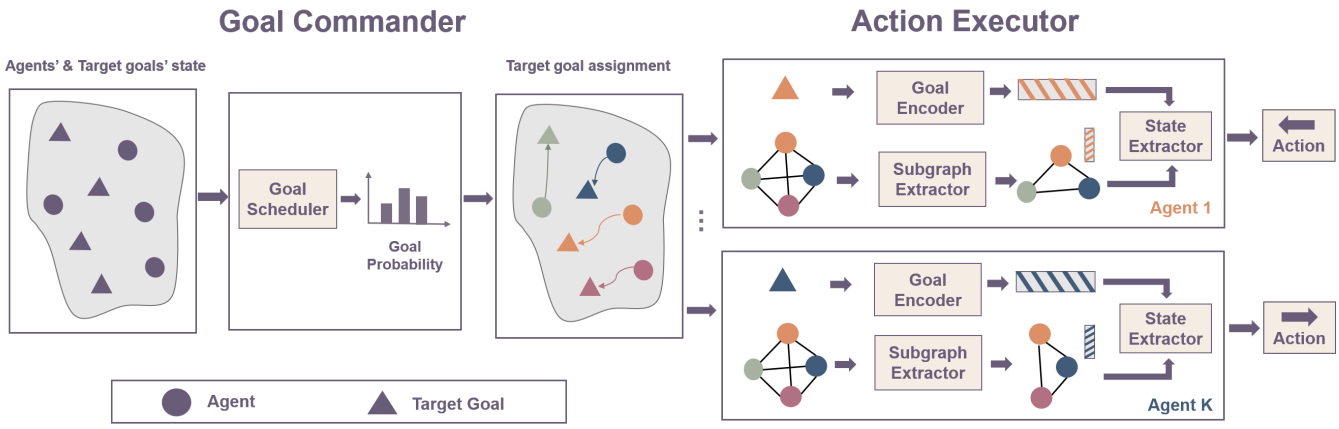


图 1：概述多智能体图增强指挥官-执行器 （**MAGE-X**）。

动作空间。O 是观测空间。 R 表示奖励函数， $R(s, \mathrm{a}\ s')$ 表示联合动作 $\mathrm{a}\ \in \mathrm{A}$ 将状态 $s \in \mathrm{S}$ 转移到状态 $s' \in \mathrm{S}$ 所获得的奖励。$\gamma \in [0, 1)$ 是折扣因子。$\mathrm{T}(s, \mathrm{a}\ s') : \mathrm{S} \times \mathrm{A} \times \mathrm{S} \mapsto [0, 1]$ 是表示转移概率的动力学函数。 G是观测函数， $G(s, \mathrm{a}\ s'\ o_i)$ 是智能体 $i \in \mathrm{N}$ 观测到观测值 $o_i \in \mathrm{O}$ 的概率。每个智能体都有一个策略 $\pi^i(a_t^i|o_{1:t}^i)$，用于根据步骤 t时的观测 o_t^i生成动作 a_t^i。智能体需要最大化期望折扣回报 $\mathrm{E}\ \sum \inf_{l=0} \gamma^l r_{t+l}$，其中 $r_t = R(s_t, \mathrm{a}_t, s_{t+1})$ 是步骤 t时的联合奖励。

3.2 图神经网络

图神经网络（GNN）是一种特殊类型的神经网络，能够处理图结构数据。图神经网络的关键在于成对消息传递，即图节点通过与邻居交换信息，迭代更新其表示。其一般公式如下：

$$h_i^{(l)} = \sigma(\sum_{j \in N_i} \frac{1}{\sqrt{d_i d_j}} (h_j^{l-1} W^{(l)})), \tag{2}$$

其中， h_i是节点 i 的特征向量。 N_i表示 i 的一组邻居节点，而$W_{(l)}$是第 l 层中的可学习权重。σ是激活函数。 d_i是节点 i 中存储的特征维度。公式 2 表明，节点 i 的特征会受到其邻居的影响。本文使用图卷积网络（GCN） [16] 来建模智能体之间的交互。图卷积网络是卷积神经网络（CNN）的一种变体，适用于处理图结构数据。

4 方法

本节介绍我们提出的框架 MAGE-X，旨在提升多智能体导航任务中的样本效率和协作能力。图 1 展示了该框架的整体概览。MAGE-X 由两个组件构成：目标指挥器和动作执行器。高层目标指挥器遵循集中式训练-集中式执行（CTCE）原则。相比之下，低层动作执行器采用去中心化设置，其中智能体 N 仅能获得部分观测，智能体 k 在步骤 t接收局部观测 o_t^k。智能体 k 学习策略 π_k，以在每个时间步 t生成动作分布 $a_t^k \sim \pi_k$ （$a_t^k|o_t^k$）。导航任务始于目标指挥器中的目标调度器接收所有智能体的生成位置和目标点。调度器预测目标点的概率分布，并为每个智能体分配最合适的目标点，而非子目标。因此，多智能体导航被转化为多智能体环境中的多个单智能体导航任务；每个智能体都需要尽快到达指定目标，同时避免碰撞。以智能体 k 为例。子图提取器接收智能体 k 及其他智能体的观测，并生成仅包含其关键邻居的智能体 k 子图，以提升协作能力。目标编码器提取智能体 k 与其目标之间的关系，并将其与子图中智能体 k 的特征一同发送给状态提取器，从而赋予团队表示强有力的目标引导。最后，智能体 k 根据动作生成器提供的初步动作完成导航任务。

4.1 目标指挥器

目标分配是一个经过长期研究的最大匹配问题，尤其是在具有大规模智能体的场景中。目标指挥器基于目标调度器模块进行目标分配，该模块生成目标点的概率分布。所设计的奖励与匈牙利算法的距离代价相关， [14], 匈牙利算法是一种用于解决图论中组合优化算法的先进经典方法。

目标调度器：该模块由一个多层感知器（MLP）层组成，f_{sche}，它将所有智能体的位置P_a以及目标点的位置 P_g作为输入。通过对 f_{sche}的输出计算softmax算子，我们得到目标点的概率 P_{goal}：

$$P_{goal} = Softmax(f_{sche}(P_a, P_g)). \tag{3}$$

此后， P_{goal}按降序排列，重新排序后的目标 $1...N$ 依次分配给智能体 $1...N$。目标调度器的奖励 R_c表示我们的分配策略 C_c相对于匈牙利

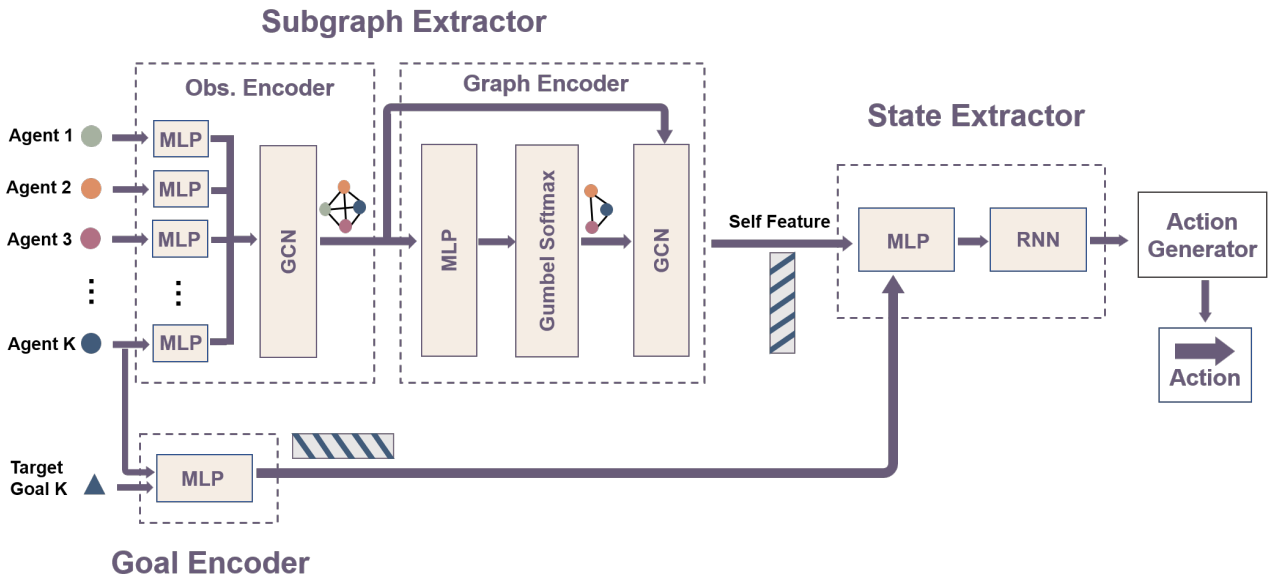


图 2：动作执行器的工作流程，包括子图提取器、目标编码器、用于表征学习的状态提取器和动作生成器。

算法的距离代价：

$$R_c = 1 - \frac{C_c}{C_h}. \tag{4}$$

4.2 动作执行器

动作执行器旨在实现高效协作，使智能体在多智能体环境中快速抵达指定目标，同时尽量减少碰撞。动作执行器的工作流程如图 2 所示。它由子图提取器、目标编码器、状态提取器和动作生成器组成。子图提取器对所有智能体的观测进行编码，并为智能体 k 生成一个仅包含关键队友的子图。目标编码器提取智能体 k 与其分配目标之间的关联性，然后将其与子图中智能体 k 的特征一同输入状态提取器，从而使团队表示具备目标引导。最后，动作生成器为智能体 k生成动作。

每个智能体的环境奖励 R_e 是完整奖励 R_b、距离惩罚 R_d 和碰撞惩罚 R_c的线性组合：

$$R_e = \alpha R_b + \beta R_d + \gamma R_c, \tag{5}$$

其中， α、 β和 γ分别是 R_b、 R_d和 R_c的系数 .

子图提取器： 子图提取器由观测编码器和图编码器组成。在观测编码器中，我们使用 MLP 层 f_o 对所有智能体的观测进行编码，并通过图卷积网络生成智能体的全连接图 G_a。这可以表示为：

$$G_a = GCN\left(\left(f_o(o^1), f_o(o^2), ..f_o(o^N)\right), A_f\right), \tag{6}$$

其中， A_f表示全连接图的邻接矩阵。在图编码器中，我们计算 gumbel softmax [10] 在由 MLP 层更新的 G_a 特征上计算 f_g。因此，我们可以通过它获得A_s^k的邻接矩阵：

$$A_s = Gumbel_Softmax(f_g(G_a)). \tag{7}$$

随后，我们使用图卷积网络生成 G_s^k：

$$G_s^k = GCN(G_a, A_s^k). \tag{8}$$

智能体的特征 k 在 G_s^k中随后被送入状态提取器。请注意，图编码器中有 l_g 个模块。

目标编码器和状态提取器： 为了捕捉智能体 k 与其分配目标之间的关联性，目标编码器由一个 MLP 层 f_{goal}组成，并接收目标的位置P_g^k和智能体的观测 o^k作为输入。这可以表示为：

$$E_{goal}^k = f_{goal}(P_g^k, o^k). \tag{9}$$

在接收智能体与其目标之间的关系以及智能体 k在 G_s中的嵌入特征 E_a^k后，我们利用状态提取器在目标引导下增强团队表示，并输出 E_{all}^k。状态提取器由一个 MLP 层 f_{state}和循环神经网络（RNN）组成，以考虑当前状态与历史状态之间的高度关联性：

$$E_{all}^k = RNN\left(f_{state}\left(concat\left(E_{goal}^k, E_a^k\right)\right)\right). \tag{10}$$

4.3 多智能体指挥官-执行器训练

MAGE-X 遵循目标条件化多智能体强化学习框架，分别为目标指挥器和动作执行器训练两个策略网络，并通过强化学习最大化整个回合中累积的奖励来对其进行优化。我们使用多智能体近端策略优化（MAPPO） [44],作为

算法 1 MAGE-X 训练流程 输入：所有智能体的位置 P_a、所有智能体的联合观测 o，以及目标点的位置 P_g。输出：目标指挥器的最终策略 π_θ，以及动作执行器的最终策略 $\pi_{\theta'}$。1：初始化：智能体数量 N、每个回合的最大步数 T_e、目标指挥器的参数 θ、 ϕ、动作执行器的参数 θ'、 ϕ'、目标指挥器缓冲区 D、动作执行器缓冲区 D'。2：当 θ 和 θ' 尚未收敛时，执行循环。3：重置环境并获取 P_a、 P_g。4：初始化： D中的步数 $t \leftarrow 1$，以及 D'中的步数 $t' \leftarrow 1$。5：$P_{goal} \leftarrow$ 目标_指挥器(P_a, P_g)。6：计算 $\pi_\theta(a_t|\mathbf{o}_t)$ 和 $V_\phi(\mathbf{o}_t)$。7：执行 $a_t \sim \pi_\theta(a_t|\mathbf{o}_t)$。8：当 $t' < T_e$ 且尚未终止时，执行循环。9：对每个智能体 k，执行$E^k_{all} \leftarrow$ 动作执行器 $(\mathbf{o}, P^k_g)$。10：对每个智能体 k，计算 $\pi_{\theta'}(a'^k_{t'}|o'^k_{t'})$ 和 $V_{\phi'}(o'^k_{t'})$。11：对每个智能体 k，执行 $a'^k_{t'} \sim \pi_{\theta'}(a'^k_{t'}|o'^k_{t'})$。12：对每个智能体 k，接收 $r't'$ 和 $o'^k_{t'+1}$。13：将 $(o'^k_{t'}, a'^k_{t'}, \pi_\theta, (a'^k_{t'}|o'_{t'}) V_{\phi'}(o'^k_{t'}) r't', o'^k_{t'+1})$ 存入 D'。14：$t' \leftarrow t' + 1$15：结束循环。16：接收 r_t 和 o_{t+1}。17：$t \leftarrow t + 1$18：将 $(o_t, a_t, \pi_\theta, (\mathbf{a}_t|\mathbf{o}_t) V_\phi(o_t) r_t, o_{t+1})$ 存入 D。19：更新 θ、 ϕ、 θ' 和 ϕ'。20：结束循环

近端策略优化（PPO）的多智能体变体 [38], 作为策略优化器。

如算法 1 所示，MAGE-X 接收智能体的位置、智能体的观测以及地标的位置，然后为目标指挥器 π_θ 和动作执行器 $\pi_{\theta'}$ 生成最终策略。首先，我们初始化智能体数量以及若干训练参数，包括最大步数、目标指挥器的参数、动作执行器的参数、目标指挥器回放缓冲区 D以及动作执行器回放缓冲区 D'（第 1 行）。目标指挥器每个回合仅执行一个动作，在开始时为智能体分配目标（第 5 ~ 7 行），并在每个回合结束时接收奖励 r_t 和后续观测 $o_{(t+1)}$，其中 t 表示 D 中的步数（第 15 行和第 16 行）。之后，我们更新目标指挥器中的策略网络 π_θ对应的 θ，以及价值网络 V_ϕ对应的 ϕ。对于动作执行器，它为每个智能体 k输出动作 $a'^k_{t'}$，并在每个时间步 t'将一组数据存入 D'（第 9 ~ 13 行）。同样，我们更新动作执行器中的策略网络 $\pi_{\theta'}$对应的 θ'，以及价值网络 $V_{\phi'}$对应的 ϕ'。

5 实验

5.1 任务设置

为了评估我们的算法在大规模搜索空间中的有效性，我们选用拥有大量智能体的 MPE [24] 以及在 $3D$ 空间中的 pybullet-gym-drones [31]作为实验环境，如图 3 和图 4 所示。我们从这两个环境中选取了三个典型任务：简单扩散、推球和无人机。随后，我们在 MPE 的简单扩散任务中使用 $N \in \{5, 20, 50\}$ 个智能体进行实验，并在推球任务中使用 $N \in \{5, 20\}$ 个智能体进行实验。无人机是 gym-pybullet-drones 中一项更复杂的四旋翼导航任务，它采用四旋翼无人机的空气动力学模型，以缩小仿真与现实世界之间的差距。因此，无人机任务中的四旋翼无人机需要更强的协作能力以避免坠毁。我们在该 $3D$ 模拟器中使用 2 架和 4 架四旋翼无人机进行实验。

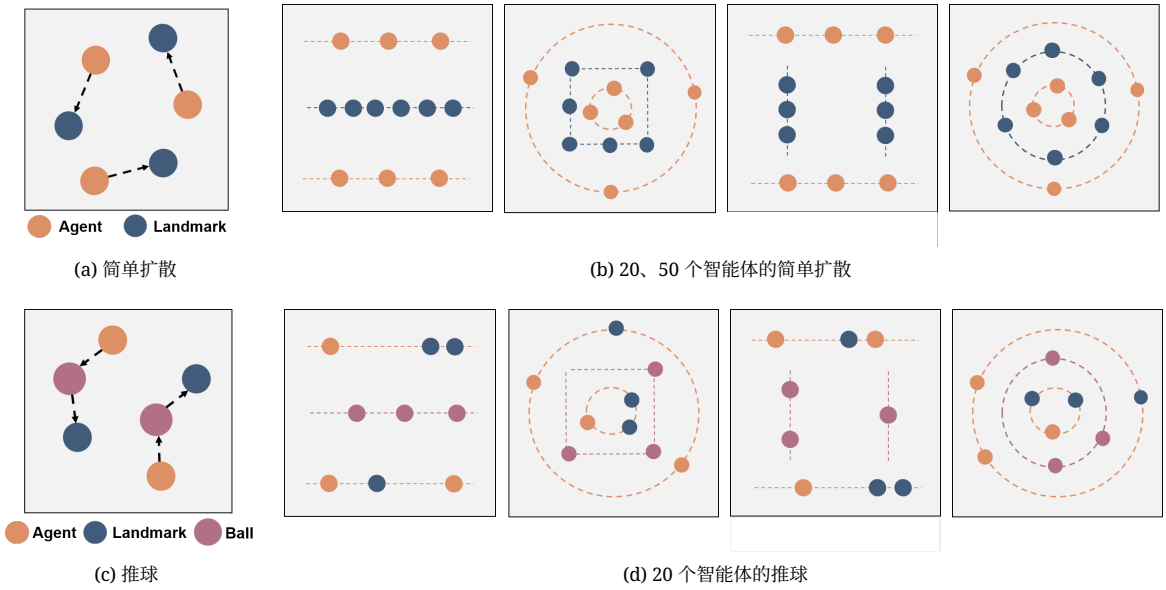
5.1.1 简单扩散 我们使用 MPE 中的简单扩散环境 [24], , 这是一种经典的二维导航任务。回合从初始化 N 个智能体和 N 个地标开始。当所有智能体到达地标时，该任务即为 100% 成功。智能体之间碰撞的结果是它们会彼此弹开，这不利于导航效率。智能体可用的离散动作包括 上、下、左和右。随着智能体数量和地图空间增大，任务难度也随之提高。实验分别在 5、20 和 50 个智能体的设置下进行。在 5 个智能体的设置中，智能体和地标的生成位置在大小为 4 的地图上随机分布。在 $N \in \{20,50\}$ 个智能体的设置中，智能体和地标的生成位置在四张具有挑战性的地图上随机初始化，地图大小分别为 36 和 100，如图

3 $N = 5$ 20 50 60, 100, 120,

(b) 所示。、、中的步数上限分别为和。

5.1.2 推球推球是 MPE [24], 中一项更复杂的两阶段导航任务，要求每个智能体先找到球，然后将球推到地标处。每个回合开始时，环境中会初始化 N 个智能体、 N 个球和 N 个地标。任务成功率是携球抵达地标的智能体数量占所有智能体总数的比例。智能体可执行的离散动作包括上、下、左和右。我们在智能体数量为 5 和 20 的设置下进行实验。在 $N = 5$ 场景中，智能体、球和地标的生成位置在大小为 16 的地图上随机分布。对于 $N = 20$，它们的初始位置按照图 3(d) 所示的四种具有挑战性的空间布局，分布在大小为 144 的地图上。 $N = 5$ 和 20 对应的步数上限分别为 100 和 200。

5.1.3 无人机 我们进一步采用 pybullet-gym-drones 中的 Drone [31], , 这是一款基于 pybullet、用于飞行四旋翼无人机的三维模拟器。Drone 对四旋翼无人机的动力学进行建模，并通过调节扭矩来控制它们。与 MPE 不同，四旋翼无人机之间发生碰撞会导致坠毁，进而使任务失败。在我们的导航任务中， N 四旋翼无人机和 N 地标的生成位置在坐标 x和 y中是随机的，但在坐标 z中固定，分别为 $x \in (-1.5, 1.5)$ $y \in (-1.5, 1.5)$ 和 $z = 1.5$。任务目标是在有限的时间预算内要求四旋翼无人机到达所有地标。每架四旋翼无人机的动作空间是在 x、 y、 z坐标方向上的加速度方向，每个坐标方向均包含前进、后退和停止 。我们考虑使用 2 架和 4 架四旋翼无人机进行实验。在我们的设置中，物理引擎步数的频率



(a) 简单扩散 (b) 20、50 个智能体的简单扩散

(c) 推球 (d) 20 个智能体的推球

图 3：MPE 中简单扩散和推球的实验环境。需要说明的是，(b) 展示了简单扩散中 4 种具有挑战性的任务模式的简化示意，其中智能体和地标的生成位置分别随机分布在橙色线和蓝色线上。(d) 展示了推球中 4 种典型任务模式的简化示意，其中智能体和地标的生成位置随机分布在橙色线上，球的生成位置随机分布在粉色线上。

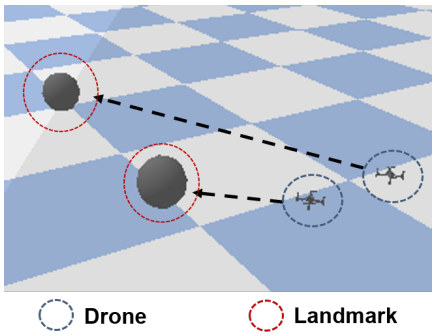


图 4：无人机的实验环境。

为 120 HZ；最大速度为 2 m/s，加速度为 5 m/s^2。在每个低层级步中，我们利用预测的加速度方向计算每架四旋翼无人机的目标速度，随后四旋翼无人机每经过四个物理引擎步执行一次动作。在 $N = 2$ 和 4 中，步数的时域为 120。

5.2 实现细节

为公平比较，每次强化学习训练均使用 3 个随机种子进行。评估得分以“均值（标准差）”的格式表示，并对总计 300 个测试回合取平均值，即每个随机种子对应 100 个回合。此外，我们使用成功率（智能体到达的地标数量占地标总数的比例）来表示各算法的性能。在无人机任务中，我们还考虑碰撞率，即坠毁的四旋翼无人机数量占四旋翼无人机总数的比例，以表示协作能力。我们的实验平台配备了 128 核中央处理器、256GB 随机存取存储器，以及一块拥有 24GB 显存的 NVIDIA GeForce RTX 3090Ti。

5.3 基线方法

为了展示我们方法的有效性，我们将其与四种多智能体强化学习基线方法进行比较。

- MAPPO [44]：这是将近端策略优化应用于多智能体强化学习的首个也是最直接的方法。该方法为所有智能体配置同一组共享参数，并通过智能体的聚合轨迹更新共享策略。
- 多智能体变换器（MAT） [41]：这是一种编码器-解码器架构，利用多智能体优势分解定理，将联合策略搜索问题转化为序列决策过程。
- DICG [22]：这种基于图的方法包含一个用于推断动态协调图结构的模块，以及一个图神经网络模块。后者根据前一个模块的输出，对联合动作或价值进行隐式推理。
- Maser [11]：这是一种目标条件化的多智能体强化学习方法，通过同时考虑个体Q值和总Q值，从经验回放缓冲区中自动为多个智能体生成子目标。

5.4 主要结果

我们展示并分析了 MAGE-X 和多智能体强化学习基线在三种竞争环境中的实验结果。

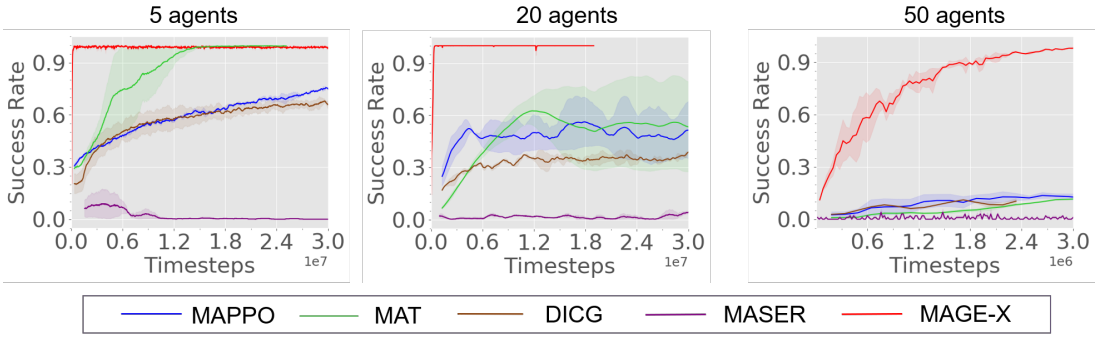


图 5：在简单扩散任务中，MAGE-X 与其他基线方法的比较， $N = 5$、20、50。

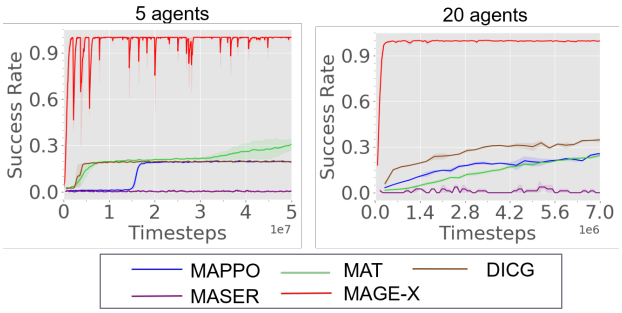


图 6：在推球任务中，MAGE-X 与其他基线方法的比较， $N = 5$、20。

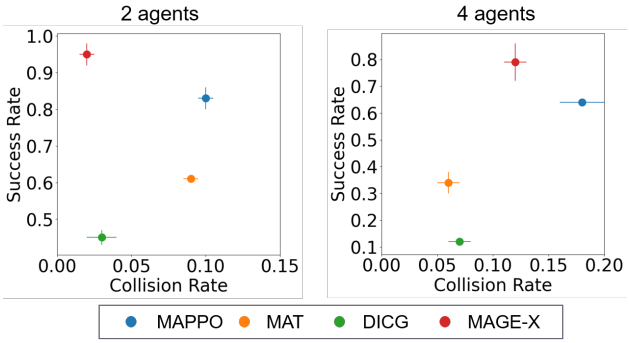


图 7：无人机任务中 MAGE-X 与基线方法的成功率和碰撞率评估性能。图中左上角的节点表示更高的性能。

5.4.1 简单扩散 我们在图 5 中展示训练曲线，并在表 1 中展示评估性能。结果表明，MAGE-X 表现最佳，尤其是在包含 50 个智能体的场景中，它能快速收敛至近似最优解，成功率达到 100%，训练效率比其他竞争者高出 10× 以上。结果显示，我们的算法表现最佳，尤其是在包含 50 个智能体的场景中。该图增强分层框架有助于智能体快速解决需要高度协作才能到达目标的问题。MAT

	MAPPO	MAT	DICG	MASER	MAGE-X
5 个智能体	0.82(0.03)	1.00(0.01)	0.71(0.05)	0.06(0.12)	**1.00**(0.01)
20 个智能体	0.50(0.31)	0.56(0.26)	0.41(0.04)	0.02(0.23)	**1.00**(0.01)
50 个智能体	0.13(0.02)	0.11(0.01)	0.12(0.02)	0.01(0.05)	**1.00**(0.01)

表 1：Simple Spread 中 MAGE-X 与基线方法的成功率评估性能。

	MAPPO	MAT	DICG	MASER	MAGE-X
5 个智能体	0.20(0.01)	0.32(0.04)	0.20(0.01)	0.01(0.01)	**1.00**(0.01)
20 个智能体	0.26(0.01)	0.25(0.02)	0.36(0.01)	0.01(0.01)	**1.00**(0.01)

表2：MAGE-X 与基线方法在推球任务中的成功率评估性能。

	2 个智能体		4 个智能体	
	成功率	碰撞率	成功率	碰撞率
MAPPO	0.83(0.03)	0.10(0.01)	0.64 (0.01)	0.18 (0.02)
MAT	0.61(0.01)	0.09 (0.01)	0.34(0.04)	**0.06**(0.01)
DICG	0.45(0.02)	0.03(0.02)	0.12(0.01)	0.07(0.01)
MAGE-X	**0.95**(0.03)	**0.02**(0.01)	**0.79**(0.07)	0.12 (0.01)

表 3：无人机中 MAGE-X 与基线方法的成功率和碰撞率评估性能。

表现优于其他多智能体强化学习基线，并且在 5 个智能体的设置中实现了 100% 的成功率，表明 MAT 智能体能够从队友的动作信息中获益。然而，随着智能体数量增加，MAT 的性能显著下降。DICG 与 MAPPO 表现相当，这表明直接采用基于图的方法并不能提升协作水平和效率。尽管 MASER 也是一种目标条件化分层强化学习方法，但它在所有竞争方法中表现最差。这说明，基于价值的方法 MASER 未能在有限的训练步数内发现有效的协作策略，因为它从经验回放缓冲区中推断子目标，且智能体与目标之间缺乏紧密联系。

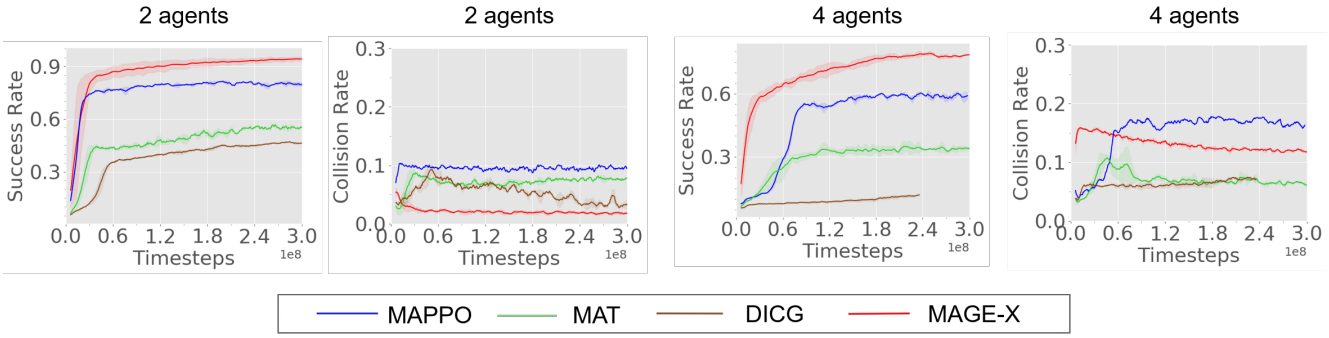


图 8：MAGE-X 与其他基线方法的比较在 Drone 中。

5.4.2 推球 如图 6 和表 2 所示，我们在推球任务中使用 $N \in \{20,50\}$ 个智能体进行实验。推球任务的难点在于需要进行两阶段目标分配：智能体首先获取指定球，然后携球到达目标地标。尽管如此，MAGE-X 仍只需较少的训练步数便实现了 100% 的成功率。相比之下，除 MAT 的成功率略有提高外，其他基线方法在 5 个智能体的设置中仅获得次优策略，成功率为 20%。在有 20 个智能体的场景中，所有基线方法均未能到达目标。结果表明，MAGE-X 中的高层目标指挥器有望同时处理多个目标分配问题，因为它将这一 M阶段目标分配拆分为 M 相互独立的任务，只需获取已分配目标的信息。

5.4.3 无人机 我们在图 8 中报告训练性能，在图 7 和表 3 中报告评估结果。实验表明，在物理真实的 Drone 三维环境中，MAGE-X 优于其他竞争方法，在 $N = 2$ 中成功率达到 95%，在 $N = 4$ 中达到 79%。与 MPE 不同，Drone 中基线方法里表现最佳的竞争方法是 MAPPO，而非 MAT，这表明在智能体数量较少的复杂任务中，MAPPO 的能力更强。具体而言，MAGE-X 优于 MAPPO，在 $N = 2$ 和 4 中的成功率分别高出 12% 和 15%。此外，MAGE-X 展现出很强的协同性，在 $N = 2$ 和 $N = 4$ 中的碰撞率分别低至 0.02% 和 0.12%，这表明 MAGE-X 能够成功地为智能体分配目标，并协调智能体之间的交互。尽管 MAPPO 在基线方法中表现最佳，但其碰撞率却明显偏高。我们推测，MAPPO 智能体追求的是较强的个体能力，而非协作。此外，在 4 个智能体的设置下，MAGE-X 的样本效率比 MAPPO 高 2× 。

5.5 消融研究

为说明 MAGE-X 各组件的有效性，我们在简单扩散任务中使用 50 个智能体，对方法的 3 个变体进行了考察：

- **带 RG 的 MAGE-X**：我们将目标指挥器中的目标调度器替换为不放回随机采样，为每个智能体分配一个随机目标。
- **MAGE-X-Atten**：我们移除观测编码器中的图卷积网络，并将动作执行器中的图编码器替换为注意力模块 [40]，以提取智能体之间的关系。将 $(f_o(o_1)\ f_o(o_2)\ ..f_o(o_N))$ 拼接后输入注意力模块。

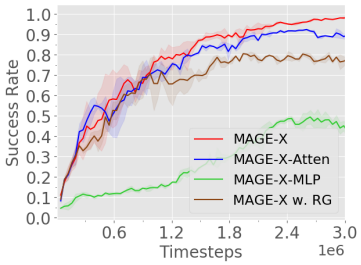


图 9：在包含50 个智能体的简单扩散任务中对 MAGE-X 进行的消融研究。

	带 RG 的 MAGE-X	MAGE-X-Atten	MAGE-X-MLP	MAGE-X
成功率	0.83(0.01)	0.93(0.01)	0.50(0.08)	**1.00**(0.01)

表 4：MAGE-X 及其变体在简单扩散任务中的成功率评估性能。

- **MAGE-X-MLP**：我们将 MLP 层作为动作执行器的替代方案，以捕捉智能体与目标之间的关联性。MLP 层以 $(f_o(o_1)\ f_o(o_2)\ ..f_o(o_N)\ P_g)$ 的拼接结果作为输入。

图 9 和表 4 分别总结了 MAGE-X 及其变体在训练和评估中的性能。MAGE-X 在数据效率和最终性能方面表现出色，成功率达到 100%。*MAGE-X-MLP* 的性能下降最为明显，这表明 MLP 层无法根据给定信息区分智能体与目标点之间的关联性。带 *RG* 的 *MAGE-X* 在高层目标指挥器中缺少合适的目标调度器，这可能导致智能体被分配到距离较远的目标。因此，带 RG 的 MAGE-X 表现较差，成功率为 83%。*MAGE- X-Atten* 略逊于 MAGE-X，成功率低了 7%。我们推测，*MAGE-X-Atten* 向每个智能体提供所有邻居的注意力权重，而不必要的队友可能会对智能体造成影响。相反，MAGE-X 智能体只关注子图中携带有用信息的关键邻居。

6 结论与未来工作

在本文中，我们提出了一种目标条件化多智能体强化学习框架，多智能体指挥器-执行器 （MAGE-X），旨在提高数据效率和多智能体导航任务中的协作能力，尤其是

在大规模空间场景中（例如，智能体数量众多或 3*D* 模拟器复杂的场景）。MAGE-X 由高层目标指挥器和低层动作执行器组成，其中，指挥器通过目标的概率分布为智能体分配目标点，而执行器则利用图神经网络和目标编码器，在强大的目标引导下捕获团队表示。全面的实验表明，在具有大规模智能体的多智能体粒子环境（MPE）以及更加复杂的四旋翼导航任务中，MAGE-X 的样本效率更高、性能优于所有最先进的多智能体强化学习基线。目前，MAGE-X 主要聚焦于多智能体导航任务，未来我们将尝试把 MAGE-X 应用于导航之外的其他多智能体任务。

致谢

本研究得到国家自然科学基金（编号：62203257、U19B2019、M-0248）、清华大学自主科研计划、清华大学-美团数字生活联合研究院、北京信息科学与技术国家研究中心（BNRist）和北京未来芯片技术高精尖创新中心的资助。

如需了解更多信息，建议访问
https://sites.google.com/view/mage- x23。

MAGE-X 详情

A.1 匈牙利算法

匈牙利算法 [14] 是一种用于解决指派问题的组合优化算法。具体而言，指派问题可以用二分图进行形式化表示。假设我们有一个完全二分图 $G = (S, T; E)$，其中包含n工人顶点 S和 n任务顶点 T，且每条边都有一个非负成本 $c(i, j)$。我们的目标是找到总成本最小的完美匹配。此外，图及其成本也可以转换为矩阵，其中第 i_{th}行第 j_{th}列的元素表示将第 j_{th}个任务分配给第 i_{th}个工人的成本。然而，匈牙利算法并不能得到精确的最优解，并且会以难以承受的时间复杂度 $O(n^3)$ 产生近似最优解。因此，我们提出了一个基于学习的模块——目标指挥器，用于以更少的推理计算求得（近似）最优解。此外，我们将匈牙利算法实现为专家，通过计算如下形式的奖励来评估目标指挥器的性能：

$$R_c = 1 - \frac{C_c}{C_h}.$$

R_c表示我们的分配策略 C_c相对于匈牙利算法 C_h的距离代价。

A.2 观测空间和动作空间

对于目标指挥器而言，观测空间是所有智能体的位置（x, y） $\in \mathrm{R}^2, x, y \in [0, mapsize]$，以及目标点的位置（$g_x, g_y$） $\in \mathrm{R}^2, x, y \in [0, mapsize]$。目标指挥器的动作空间是智能体的已分配目标的索引，$A_{com} = \{i, i \in N\}$，其中 i是已分配目标的索引，N表示目标的数量。对于每个动作执行器 i，观测空间包含智能体 i的位置（x_i, y_i） $\in \mathrm{R}^2, x_i, y_i \in [0, mapsize]$、指挥器为其指定的目标（$g_i^x, g_i^y$） $\in \mathrm{R}^2, g_i^x, g_i^y \in [0, mapsize]$、其他智能体的位置（$x_{-i}, y_{-i}$） $\in \mathrm{R}^2, x_{-i}, y_{-i} \in [0, mapsize]$，以及用于表示其是否已到达目标的独热标签$L_i$。每个动作执行器的动作空间是每个智能体的原始动作，$A_{exe}^i = \{a^i | a^i \in action\}$。我们将 $action$ 记为可用的原始动作，它是从分类分布中采样的离散变量。

A.3 动作生成器

动作生成器接收状态提取器提供的具有强目标引导的团队特征，然后为每个智能体推断一个原始动作，以完成导航任务。

B 实验

B.1 训练细节

我们使用 MAPPO [44], 训练策略，MAPPO 是近端策略优化（PPO） [38]的多智能体扩展版本。训练超参数见表 1，MAGE-X 的详细架构见表 2。

在 MAGE-X 中，我们采用特征归一化和奖励归一化来增强鲁棒性。我们使用 Adam优化器

常见超参数	价值
梯度裁剪范数	10.0
广义优势估计参数	0.95
gamma	0.99
价值损失	Huber损失
Huber阈值	10.0
小批量大小	批量大小 / 小批量
优化器	Adam
优化器 epsilon	1e-5
权重衰减	0
网络初始化	正交初始化
使用奖励归一化	True
使用特征归一化	True
学习率	2.5e-5
并行环境线程	10
本地步数	15

表 1：MAPPO 超参数

架构超参数	价值
f_{sche}	$[4 \times N, 1 \times N, 1]$
f_o	$[2 \times N, 32, 1]$
f_g	$[32 \times N, 32 \times N, 1]$
f_{goal}	$[2 \times (N + 1), 32, 1]$
f_{state}	$[32 \times 2, 32, 1]$
特征维度	32
l_g	2

表 2：架构超参数

，优化器 epsilon 为 $1e^{-5}$，学习率为 $2.5e^{-5}$。价值损失采用 Huber损失，Huber阈值为 10。0。我们将梯度裁剪范数设为 10。0，广义优势估计参数设为 0。95。期望奖励的折扣因子 $gamma$为 0。99，且未使用权重衰减。环境的并行线程数为 10，本地步数为 15。神经网络架构采用正交初始化，特征维度为 32。

B.2 基线

在 MAGE-X 中，我们将其与一种基于规划的竞争方法 Multi-Agent $A*$，以及四种多智能体强化学习基线算法 MAPPO、MAT、DICG 和 Maser 进行比较。

B.2.1 多智能体 A*（MA A*）[6] MA A* 是 $A*$ 算法的多智能体版本。它首先将连续空间转换为网格空间，并计算每个网格的成本。成本包括网格与智能体之间的距离，以及网格与目标之间的距离。此外，它还会考虑智能体之间的碰撞。目标是以最小总成本找到一条从智能体到目标的路径。

B.2.2 MAPPO [44] MAPPO 是 PPO 的多智能体版本，在《星际争霸 II》、Google

智能体	5	20	50
MA A*	**1.00**(0.01)	0.93(0.03)	0.77(0.03)
MAGE-X	**1.00**(0.01)	**1.00**(0.01)	**1.00**(0.01)

表 3：MAGE-X 与 MA A* 在简单扩散中的成功率评估性能。

Research Football、MPE 和花火中均取得了出人意料的优异表现。与常见的基于价值的多智能体算法相比， [25, 34, 39], Mappo 是在多智能体强化学习中应用 PPO 的首个也是最直接的方法。它为所有智能体配备一套共享参数，并通过智能体的聚合轨迹更新共享策略，最终奖励表现具有竞争力甚至更优，同时不损失样本效率。

B.2.3 MAT [41] MAT 是一种编码器-解码器架构，它利用多智能体优势分解定理，将联合策略搜索问题转化为序贯决策过程。不同于 Decision Transformer 等仅适用于预先收集的离线数据的先前方法，MAT 通过在环境中进行在线试错，以同策略方式训练。

B.2.4 DICG [22] DICG 是一种基于图的方法，由一个用于推断动态协调图结构的模块，以及一个根据前者的输出对联合动作或价值进行隐式推理的图神经网络模块组成。它使智能体能够通过标准的演员-评论家方法学习在完全集中化与去中心化之间进行权衡，从而显著提升智能体数量较多的领域中的协同。

B.2.5 Maser [11] Maser 是一种目标条件化多智能体强化学习方法，它通过同时考虑个体Q值和总Q值，从经验回放缓冲区中为多个智能体自动生成子目标。该方法为每个智能体采用个体内在奖励，使其实现分配的子目标并最大化联合动作价值。

B.3 补充结果

B.3.1 与 MA A* 的比较表1展示了 MAGE-X 与基于规划的基线方法 MA A* 在简单扩散中的性能。结果表明，MA A* 在 $N = 5$ 20 $N =$ 个智能体时的性能与 MAGE-X 相当，但在 时表现较差。这意味着 MA A* 有潜力处理多智能体多目标问题，但当智能体数量增加时，可能难以找到最优路径。

参考文献

[1] Guillaume Bresson、Zayed Alsayed、Li Yu 和 Sébastien Glaser. 2017. 同步定位与建图：自动驾驶当前趋势综述。IEEE智能车辆学报 2，3（2017），194–220。[2] Daniele Calisi、Alessandro Farinelli、Luca Iocchi 和 Daniele Nardi. 2005. 救援环境中的自主导航与探索。载于 2005 年 IEEE 国际安全、安保与救援机器人研讨会。IEEE，54–59。[3] Elliot Chane-Sane、Cordelia Schmid 和 Ivan Laptev. 2021. 具有想象子目标的目标条件强化学习。载于国际机器学习会议。PMLR，1430–1440。[4] Rui Chen、Peide Huang 和 Laixi Shi. 2021. 面向多智能体目标条件自监督模仿学习的潜在目标分配。载于神经信息处理系统进展。[5] Peter Dayan 和 Geoffrey E Hinton. 1992. 封建强化学习。神经信息处理系统进展 5（1992）。[6] Abdul Qadir Faridi、Sanjeev Sharma、Anupam Shukla、Ritu Tiwari 和 Joydip Dhar. 2018. 利用人工蜂群和进化规划在未知环境中进行多机器人多目标动态路径规划。智能服务机器人 11（2018），171–186。[7] Kaivuan Gao、Jing Xin、Han Cheng、Ding Liu 和 Jiang Li. 2018. 面向智能物流的多移动机器人自主导航系统。载于 2018 年中国自动化大会（CAC）。IEEE，2603–2609。[8] Sorin Grigorescu、Bogdan Trasnea、Tiberiu Cocias 和 Gigel Macesanu. 2020. 用于自动驾驶的深度学习技术综述。野外机器人学杂志 37，3（2020），362–386。[9] Christopher Hoang、Sungryull Sohn、Jongwook Choi、Wilka Carvalho 和 Honglak Lee. 2021. 面向长时域目标条件强化学习的后继特征地标。神经信息处理系统进展 34（2021），26963–26975。[10] Eric Jang、Shixiang Gu 和 Ben Poole. 2016. 使用 Gumbel-Softmax 的分类重参数化。arXiv 预印本 arXiv:1611.01144（2016）。[11] Jeewon Jeon、Woojun Kim、Whiyoung Jung 和 Youngchul Sung. 2022. MASER：利用从经验回放缓冲区生成的子目标进行多智能体强化学习。载于国际机器学习会议。PMLR，10041–10052。[12] Jiechuan Jiang、Chen Dun、Tiejun Huang 和 Zongqing Lu. 2018. 图卷积强化学习。arXiv 预印本 arXiv:1810.09202（2018）。[13] Yue Jin、Yaodong Zhang、Jian Yuan 和 Xudong Zhang. 2019. 利用交错深度强化学习在未知环境中实现高效的多智能体协作导航。载于 ICASSP 2019-2019 IEEE 声学、语音与信号处理国际会议（ ICASSP ）。IEEE，2897–2901。[14] Roy Jonker 和 Ton Volgenant. 1986. 改进匈牙利分配算法。运筹学快报 5，4（1986），171–175。[15] Junsu Kim、Younggyo Seo 和 Jinwoo Shin. 2021. 分层强化学习中的地标引导的子目标生成。神经信息处理系统进展 34（2021），28336–28349。[16] Thomas N Kipf 和 Max Welling. 2016. 利用图卷积网络进行半监督分类。arXiv 预印本 arXiv:1609.02907（2016）。[17] Alexander Kleiner、Johann Prediger 和 Bernhard Nebel. 2006. 基于 RFID技术的搜救探索与 SLAM。载于 2006 IEEE/RSJ智能机器人与系统国际会议。IEEE，4054–4059。[18] Songsang Koh、Bo Zhou、Hui Fang、Po Yang、Zaili Yang、Qiang Yang、Lin Guan 和 Zhigang Ji. 2020. 基于实时深度强化学习的车辆导航。应用软计算 96（2020），106694。[19] Abdul Rahman Kreidieh、Glen Berseth、Brandon Trabucco、Samyak Parajuli、Sergey Levine 和 Alexandre M Bayen. 2019. 分层强化学习中的层间协作。arXiv 预印本 arXiv:1912.02368（2019）。[20] Tejas D Kulkarni、Karthik Narasimhan、Ardavan Saeedi 和 Josh Tenenbaum. 2016. 分层深度强化学习：整合时间抽象与内在动机。神经信息处理系统进展 29（2016）。[21] Andrew Levy、George Konidaris、Robert Platt 和 Kate Saenko. 2017. 利用事后经验学习多层级层次结构。arXiv 预印本 arXiv:1712.00948（2017）。[22] Sheng Li、Jayesh K Gupta、Peter Morales、Ross Allen 和 Mykel J Kochenderfer. 2020. 用于多智能体强化学习的深度隐式协调图。arXiv 预印本 arXiv:2006.11438（2020）。[23] Shaotang Liu 和 Lixin Hu. 2009. 北斗导航卫星系统在物流与运输中的应用。载于《物流：中国交通与发展新前沿》。1789–1794。[24] Ryan Lowe、Yi I Wu、Aviv Tamar、Jean Harb、OpenAI Pieter Abbeel 和 Igor Mordatch. 2017. 用于混合协作竞争环境的多智能体演员 - 评论家。神经信息处理系统进展 30（2017）。[25] Ryan Lowe、Yi I Wu、Aviv Tamar、Jean Harb、OpenAI Pieter Abbeel 和 Igor Mordatch. 2017. 用于混合协作竞争环境的多智能体演员-评论家。神经信息处理系统进展 30（2017）。[26] Enrico Marchesini 和 Alessandro Farinelli. 2020. 用于无地图导航的离散深度强化学习。载于 2020 IEEE机器人国际会议与自动化（ICRA）。IEEE，10688–10694。[27] Enrico Marchesini 和 Alessandro Farinelli。2022。通过单机器人进化策略搜索增强多机器人导航的深度强化学习方法。载于：2022年国际机器人与自动化会议（ICRA）。IEEE，5525–5531。[28] Ofir Nachum、Shixiang Shane Gu、Honglak Lee 和 Sergey Levine。2018。数据高效的分层强化学习。神经信息处理系统进展 31（2018）。[29] Suraj Nair 和 Chelsea Finn。2019。分层前瞻：通过视觉子目标生成自监督学习长时程任务。arXiv预印本 arXiv:1909.05829（2019）。[30] Yaru Niu、Rohan R Paleja 和 Matthew C Gombolay。2021。多智能体图注意力通信与协作。在 AAMAS。964–973。[31] Jacopo Panerati、Hehui Zheng、SiQi Zhou、James Xu、Amanda Prorok 和 Angela P Schoellig。2021。学会飞行——一种采用 pybullet 物理引擎、用于多智能体四旋翼控制强化学习的 Gym 环境。载于：2021年 IEEE/RSJ智能机器人与系统国际会议（IROS）。IEEE，7512–7519。[32] Karl Pertsch、Oleh Rybkin、Frederik Ebert、Shenghao Zhou、Dinesh Jayaraman、Chelsea Finn 和 Sergey Levine。2020。利用目标条件分层预测器进行长时程视觉规划。神经信息处理系统进展 33（2020），17321–17333。[33] Zhenhuan Rao、Yuechen Wu、Zifei Yang、Wei Zhang、Shijian Lu、Weizhi Lu 和 ZhengJun Zha。2021。基于深度强化学习的多目标视觉导航。IEEE神经网络与学习系统汇刊 32，12（2021），5445–5455。[34] Tabish Rashid、Mikayel Samvelyan、Christian Schroeder、Gregory Farquhar、Jakob Foerster 和 Shimon Whiteson。2018。QMIX：用于深度多智能体强化学习的单调价值函数分解。载于：国际机器学习会议。PMLR，4295–4304。[35] Jingqing Ruan、Yali Du、Xuantang Xiong、Dengpeng Xing、Xiyun Li、Linghui Meng、Haifeng Zhang、Jun Wang 和 Bo Xu。2022。GCS：多智能体强化学习的基于图的协调策略。arXiv预印本 arXiv:2201.06257（2022）。[36] Heechang Ryu、Hayong Shin 和 Jinkyoo Park。2020。采用分层图注意力网络的多智能体演员-评论家。载于：AAAI人工智能会议论文集，第34卷。7236–7243。[37] Franco Scarselli、Marco Gori、Ah Chung Tsoi、Markus Hagenbuchner 和 Gabriele Monfardini。2008。图神经网络模型。IEEE神经网络汇刊 20，1（2008），61–80。[38] John Schulman、Filip Wolski、Prafulla Dhariwal、Alec Radford 和 Oleg Klimov。2017。近端策略优化算法。arXiv预印本 arXiv:1707.06347（2017）。[39] Kyunghwan Son、Daewoo Kim、Wan Ju Kang、David Earl Hostallero 和 Yung Yi。2019。QTRAN：通过变换学习分解协作式多智能体强化学习。载于：国际机器学习会议。PMLR，5887–5896。[40] Ashish Vaswani、Noam Shazeer、Niki Parmar、Jakob Uszkoreit、Llion Jones、Aidan N Gomez、Łukasz Kaiser 和 Illia Polosukhin。2017。注意力就是一切。神经信息处理系统进展 30（2017）。[41] Muning Wen、Jakub Grudzien Kuba、Runji Lin、Weinan Zhang、Ying Wen、Jun Wang 和 Yaodong Yang。2022。多智能体强化学习是一个序列建模问题。arXiv预印本 arXiv:2205.14953（2022）。[42] Lijun Xia、Chao Yu 和 Zifan Wu。2021。用于协作式多智能体导航的基于推理的分层强化学习。载于：2021年 IEEE第33届人工智能工具国际会议（ICTAI）。IEEE，57–64。[43] Xinyi Yang、Chao Yu、Jiaxuan Gao、Yu Wang 和 Huazhong Yang。2022。SAVE：空间注意力视觉探索。载于：2022年 IEEE图像处理国际会议（ICIP）。IEEE，1356–1360。[44] Chao Yu、Akash Velu、Eugene Vinitsky、Yu Wang、Alexandre Bayen 和 Yi Wu。2021。近端策略优化在协作式多智能体游戏中的惊人效果。arXiv预印本 arXiv:2103.01955（2021）。[45] Chao Yu、Xinyi Yang、Jiaxuan Gao、Huazhong Yang、Yu Wang 和 Yi Wu。2022。学习高效的多智能体协作视觉探索。载于：欧洲计算机视觉会议。Springer，497–515。[46] Tianren Zhang、Shangqi Guo、Tian Tan、Xiaolin Hu 和 Feng Chen。2020。在分层强化学习中生成邻接约束子目标。神经信息处理系统进展 33（2020），21579–21590。[47] Yuke Zhu、Roozbeh Mottaghi、Eric Kolve、Joseph J Lim、Abhinav Gupta、Li Fei-Fei 和 Ali Farhadi。2017。在室内场景中使用深度强化学习进行目标驱动视觉导航。载于：2017年 IEEE国际机器人与自动化会议（ICRA）。IEEE，3357–3364。