南方科技大学
SOUTHERN UNIVERSITY OF SCIENCE AND TECHNOLOGY

本科生毕业设计（论文）

题　　目：	基于 GCNs 和 Deep RL 的人群导航策略的设计与实现
姓　　名：	周剑翔
学　　号：	11912923
系　　别：	机械与能源工程系
专　　业：	机器人工程
指导教师：	丁克蜜

2023 年 05 月 25 日

诚信承诺书

1. 本人郑重承诺所呈交的毕业设计（论文），是在导师的指导下，独立进行研究工作所取得的成果，所有数据、图片资料均真实可靠。

2. 除文中已经注明引用的内容外，本论文不包含任何其他人或集体已经发表或撰写过的作品或成果。对本论文的研究作出重要贡献的个人和集体，均已在文中以明确的方式标明。

3. 本人承诺在毕业论文（设计）选题和研究内容过程中没有抄袭他人研究成果和伪造相关数据等行为。

4. 在毕业论文（设计）中对侵犯任何方面知识产权的行为，由本人承担相应的法律责任。

作者签名:

_____ 年____ 月____ 日

基于 GCNs 和 Deep RL 的人群导航策略的设计与实现

周剑翔

（机械与能源工程系　指导教师：丁克蜜）

[摘要]：在一些生活场景中，机器人需要在人群中穿梭完成特定任务。由于这样的环境需要机器人具备避免与周围人群碰撞的能力，因此对机器人造成了很大的挑战。为了应对这个挑战，本毕业设计主要设计并实现一套用于机器人在人群中移动避障的算法。该算法利用图结构建模问题场景，并利用图卷积神经网络计算获取个体间更深层的交互特征，同时根据周围人群对机器人移动策略选择的影响对交互特征合并用于移动规划。此外该模型对人的动力学进行了建模，以涵盖对环境的全面理解，从而提高机器人的运动策略准确性。该算法模型采用深度强化学习算法进行训练，并将训练后的模型在仿真环境下与其他经典方法进行多方面的对比，结果证明此模型可以生成更高效和更安全的移动路径，具有更短的导航时间和更高的成功率。该算法将有助于提高机器人在拥挤场景下的应用效率和安全性。

[关键词]：图神经网络；深度强化学习；人群避障导航

[**ABSTRACT**]: In certain scenarios, robots need navigate through crowds to complete specific tasks. This environment poses great challenges as it requires robots to avoid collisions with surrounding individuals. To address this challenge, this graduation project mainly designs and implements algorithm for robot navigation in crowds. The algorithm utilizes graph structure to model the scenario, and uses graph convolutional networks(GCNs) to calculate deeper interactive features between agents. Based on the importance of human for robot, the interaction features are merged for movement planning. Additionally, the model also incorporates human dynamics, to provide more comprehensive considerations and improve the navigation strategy. The algorithm model is trained by deep reinforcement learning algorithm, and compared with other classical methods in a simulation environment. Through simulation experiments, it is demonstrated that the model can generate more efficient and safer movement paths, with shorter navigation time and higher success rate. The algorithm will improve the efficiency and safety of robots in crowded environments.

[**Key words**]: GCNs, Deep RL, robot crowd-navigation

目录

1 绪论 **1**
1.1 背景 1
1.2 国内外研究现状 2
1.2.1 基于规则的方法 2
1.2.2 基于轨迹预测的方法 2
1.2.3 基于机器学习的方法 3
1.3 研究内容和系统框架 4

2 算法设计 **5**
2.1 问题定义 5
2.2 算法框架 6
2.2.1 交互建模 7
2.2.2 关系汇聚 9
2.2.3 轨迹预测 10
2.2.4 路径规划 11
2.3 模型训练 12
2.3.1 深度强化学习模型 12
2.3.2 实现细节 14

3 仿真验证 **15**
3.1 仿真环境介绍 15
3.2 仿真结果对比 16
3.2.1 定量比较 16
3.2.2 定性比较 18

4 结论 **21**

参考文献 **22**

致谢 **24**

1 绪论

1.1 背景

21 世纪以来，随着机器人技术的快速发展，生活场景如人行道、建筑物内和走廊等出现了多种自主导航的机器人，并应用于某些特定任务，比如：(1) 移动机器人送餐：在餐馆、医院、办公室等地，人群中的移动机器人可以自主地导航，将食物或药品送到指定地点，同时避免与其他人或物品碰撞。(2) 人类与机器人共同工作：在工厂、仓库等环境中，人群中的机器人可以与工作人员共同协作，共同完成任务，机器人可以自主导航，为人类提供支持和帮助。在这些场景下，机器人必须以合乎社会规范的方式在人群中移动穿梭，并完成特定任务。对于机器人的移动策略研究，目标是以实现让机器人在人群中高效、安全和自主地导航，同时避免与其他人或障碍物碰撞，提高机器人的交互性和适应性。

与机器人相比，人群中移动的人类具有极高的机动性，人类知道如何在机场或商场等人数众多的地方穿行，也懂得如何遵循社会规范与其他人交互移动而避免碰撞，这些交互移动方式包括当交错行走时该走哪个方向避障，保留多大的空间间隔，以及何时超过缓慢行走的行人等。而对于机器人来说，这些场景是极具挑战的，其主要面临着俩个方面的难题。第一，整个场景问题是分布式的，每一个个体都有其自己的移动策略，而且其他移动个体的意图路径和目标对于机器人来说一般是未知的，且这些不可被观测到的状态是很难在线推测出来的；第二，人群中既包含移动物体，也包含静止物体，移动个体会受到其他移动或静止的个体的影响而改变移动方式，而这种个体间的交互影响同样是难以建模的。

尽管面临这些挑战，机器人人群中的自主导航已被广泛研究且已经有许多成功的验证。传统方法比如 ORCA（Optimal Reciprocal Collision Avoidance）和 SF(Social Force) 根据环境当前已知状态来决定机器人的最优行为[1][2][3]。而另一种方法则是基于轨迹预测的方法，首先预测其他个体未来的轨迹其次再规划一条路径来躲避碰撞[4][5]。不过这俩类方法都很容易面临机器人冻结问题，即在稠密人群中，机器人根据策略判断当前所有的移动路径都是不安全的，因此机器人选择不执行任何动作而冻结在原地，但该选择往往只是一个子优解[6]。第三种基于机器学习的方法则展现出了更加优秀的结果。该方法将机器人在人群中的自主导航任务看作是一个 Markov Decision Process (MDP)，即马尔可夫决策过程，并利用 Deep V-Learning 来解决[7][8][9][10][11]。这一类方法中，往往先利用一些专家策略来初始化网络模型，之后再利用强化学习继续训练模型，而决策者基于训练过后的值网络选择一个具有更高值的动作。

综上所述，研究如何让机器人在人群中更高效、安全地自主移动是具有挑战且有

意义的，该研究不仅可以扩展到其他分布式场景下的应用问题，而且可以用于实际场景中让机器人给人类以提供更有效的帮助。

1.2 国内外研究现状

以下将从基于规则的方法、基于轨迹预测的方法、基于机器学习的方法这三个方面来梳理在机器人人群导航算法研究的发展。

1.2.1 基于规则的方法

基于规则的方法已经有近二十多年的研究。其中，RVO(Reciprocal Velocity Obstacle)[1] 和 ORCA(Optimal Reciprocal Collision Avoidance)[2]方法是最经典的底层避障算法，这俩类方法均采用速度障碍法，会将未来任意可能与其他个体碰撞的速度均排除在外，而在剩下的可行速度域中选择一个最接近机器人倾向速度（即朝向目标点的最大速度）的无碰撞速度，同时，这些方法假定其他移动物体会遵循和机器人相似的移动避障策略来避免碰撞。而其他方法比如 Social Force (SF) 是一种用于模拟行人运动的模型，称为社会力模型[3]。该模型基于人类行为学和社会力学的理论，将行人视为受到多个力的作用，包括个人偏好力、群体效应力和障碍物排斥力等。这些力相互作用产生合力，驱动行人运动。社会力模型模拟了行人在不同场景下的运动，如拥挤的空间、狭窄的通道和紧急情况下的逃生等。不过这些方法都很容易面临机器人冻结问题[6]，意味着在密集人群中，机器人无法找到可行的移动路径，同时这些方法只考虑到移动物体当前的状态，而不考虑其隐藏意图（如目标点和期望的移动轨迹），因此机器人生成的移动路径是缺乏远见且有时是不自然的。

1.2.2 基于轨迹预测的方法

基于轨迹预测的方法主要分为基于概率的方法和基于深度学习的方法。其中，基于概率的方法主要考虑运动学约束和概率分布[4]，通过高斯过程、卡尔曼滤波、粒子滤波等方法实现轨迹预测。而基于深度学习的方法则主要通过神经网络学习历史轨迹数据的规律，进而预测未来运动轨迹[10]。目前基于深度学习的方法已经取得了许多优秀的结果，如 LSTM、CNN、TCN 等模型。基于轨迹预测的方法会预测其他移动物体的未来轨迹来生成可行路径，这使得机器人做出的决定是具有远见的。同时利用历史轨迹数据进行预测，避免了因为环境复杂而导致的传感器数据不准确的问题。同时，该方法能够实现更为精准的预测，从而使机器人做出正确的决策。然而该方法会面临着计算量大或需要对其他移动个体的隐藏状态进行建模的难题[12]，并且在实

体运行中每一次获得最新的观察（传感器状态更新）时都需要重新在线计算。此外，在复杂环境下对模型的准确性也存在众多限制。

1.2.3 基于机器学习的方法

基于机器学习的方法在机器人自主导航避障中的应用越来越受到关注。目前，该方法的研究主要分为基于规则的机器学习方法和基于深度学习的机器学习方法。基于规则的机器学习方法主要依赖于人工设计的规则，包括模糊规则、逻辑规则、专家系统等。比如，其中模仿学习方法仿效专家策略来让机器人模仿获得期望的行为[13][14]。此方法的优点是可以对机器人行为进行有效控制，可以避免机器人因环境复杂而产生混乱的行为，也可以降低机器人的学习成本。然而规则设置需要经验，并且灵活性不够，适应新环境的能力较弱。

基于深度学习算法的机器学习方法主要是利用深度学习模型对环境进行建模。当前研究主要采用基于强化学习和基于监督学习的方法。强化学习方法主要通过奖励函数的引导，鼓励机器人学习出更为优秀的行为，比如 Deep V-learning 算法[10][9][7][8][15]。监督学习方法则主要依赖于大量的数据集作为输入，利用神经网络等深度学习算法训练机器人的行为，使机器人从数据中得出规律，如 CNN、RNN 等模型[16]。相较于基于规则的机器学习方法，该方法的优点在于可以利用机器学习算法进行无模型学习，提高机器人的适应性和灵活性，能够在不同的环境下进行自主导航避障。此外，该方法还可以学习新的规律，在学习到新环境时，可以进行快速适应。然而，该方法需要大量的训练样本，训练过程中需要消耗大量的时间和计算资源，而且需要针对具体场景进行设计。

而关于环境建模，对于类似于人群避障的情境，个体之间的关系和个体可以被表示为一个图 (Graph)，其中个体被表示为图的节点 (nodes)，个体间的关系被表示为图的边 (edges)。而 Graph Neural Networks(GNNs) 是一个强有力的工具来学习基于图的函数。其中，对于人群移动交互问题，个体的动力学等状态被编码在 GNNs 的节点更新函数中，而个体间的交互关系被编码在边更新函数中。近年来已有许多研究工作利用 GNNs 来建模人群中的移动关系，并结合深度强化学习 (Deep Reinforcement Learning) 实现移动导航策略。在陈等人[10]的研究工作中，GCNs 被用于人群模型建模并学习人群导航中每个个体间的关系，作为 GNNs 的一个变体，其节点间的关系被定义为邻接矩阵，之后结合 Deep RL 训练模型，并在以上基础上扩增了蒙特卡洛树搜索算法，最终实现预测人群在未来一段时间的轨迹并得到相应的机器人移动策略。

1.3 研究内容和系统框架

在本篇毕设中，受前人研究的工作启发[17][10]，同样采用图结构来建模机器人在人群中自主移动避障的场景。利用 GCNs 来对图结构进行信息传递，学习场景下每个个体间的交互关系，并用学习到的交互特征用来进行值估计和个体的轨迹预测。在模型训练方面，首先利用模仿学习来初始化模型，这里采用经典模型 ORCA 作为专家策略，之后利用强化学习来训练模型。之后将训练得到的模型用于仿真环境进行定性和定量俩个方面的实验验证，并与其他常见的经典模型进行对比来验证论文所提出的模型的可行性。

本毕业论文将从绪论、算法设计、仿真验证、总结四个方面展开阐述。第一章绪论主要介绍研究背景和国内外研究现状。第二章算法设计介绍用于自主导航避障的网络模型设计。第三章仿真验证介绍所用的仿真环境，并与其他常见的方法进行多方面的对比来验证算法模型的可行性。最后一章总结则主要对整篇论文进行工作总结和对未来工作进行展望。

2 算法设计

2.1 问题定义

机器人在人群中的自主导航问题可以定义为让机器人穿过具有 N 个人的人群中，并尽可能高效且安全地到达目标点。而这个任务可以被看作是马尔可夫决策过程，并利用深度强化学习来解决。机器人在人群中的自主移动导航任务是一个复杂的问题，它既需要考虑要高效地到达目标点，又需要考虑避免与人群中的其他人碰撞，确保安全性。而马尔可夫决策过程可以很容易描述和解决这个问题。在马尔可夫决策过程中，输入到机器人的状态包括它的位置、速度、方向以及周围的人群状态，如人的数量、人群中每个人的位置和速度等等。机器人可以通过选择下一步的移动方向，来改变状态。每个状态转移之后，机器人会收到一个对其所做决策（即动作）的奖励信号，来调整其策略。这样，机器人可以在与人群交互的过程中不断学习，以最小化与人群碰撞的风险，同时达成它的任务目标。为了更好地解决这个问题，在模型中还需要定义奖励函数、策略和值函数。奖励函数的设定可以鼓励机器人选择安全、高效的行动，同时惩罚机器人与人群发生碰撞或进入他人舒适区的距离。策略是指机器人为了获得最大的奖励而选择下一步行动的概率分布。而值函数则给出了在给定策略下当前状态的值，即从当前状态开始，机器人在执行给定策略的情况下未来奖励的折现和。在马尔可夫决策过程的框架下，机器人可以学习一组最优的策略和值函数，以保证尽可能迅速、安全地穿越人群，到达目的地。而本小节将会对以上马尔可夫决策过程所需要的部分展开描述。

对于马尔可夫决策过程，首先需要定义整个系统当前的状态。每一个个体（人或机器人）都可以观测到其他个体的位置 $\mathbf{p} = [p_x, p_y]$，速度 $\mathbf{v} = [v_x, v_y]$ 和半径 r(一个粗略的尺寸)，而这部分为可观测状态。除此之外，每一个个体有着无法被其他个体观测的不可观测状态，包括全局位置 $\mathbf{p_g}$ 和倾向速度 v_{pref}。而一个个体的全部状态则定义为它的可观测状态加不可观测状态，即 $[\mathbf{p}, \mathbf{v}, r, \mathbf{p_g}, v_{pref}]$。此处设定 s_0^t 指机器人在 t 时刻的全部状态，s_i^t 则指第 i 个人在 t 时刻的可观测状态，而对于机器人来说，在 t 时刻的输入即当前时刻机器人的全部状态，加其他所有个体的可观测状态，即环境输入状态，数学符号可定义为 $\mathbf{S^t} = \{s_0^t, s_1^t, s_2^t, ..., s_N^t\}$。而马尔可夫决策过程中的最优策略则为环境输入状态 $\mathbf{S^t}$ 到下一时刻行动 $\mathbf{a}_t$ 的概率分布，数学符号定义为 $\pi^* : \mathbf{S^t} \mapsto \mathbf{a^t}$，

而最优策略的目标就是最大化未来的奖励和，公式(2.1)定义如下：

$$\begin{aligned}\pi^*(\mathbf{S^t}) &= \arg\max_{\mathbf{a_t}\in A} R(\mathbf{S^t},\mathbf{a^t}) + \gamma^{\Delta t}\int_{\mathbf{S}^{t+\Delta t}} P(\mathbf{S}^t,\mathbf{a}^t,\mathbf{S}^{t+\Delta t})V^*(\mathbf{S}^{t+\Delta t})d\mathbf{S}^{t+\Delta t} \\ V^*(\mathbf{S}^t) &= \sum_{k=t}^{T}\gamma^k R(\mathbf{S}^k,\pi^*(\mathbf{S}^k))\end{aligned} \tag{2.1}$$

式中 $R(\mathbf{S}^t,\mathbf{a}^t)$ 是 t 时刻所接受到的奖励，$\gamma\in(0,1)$ 是折现因子 (discount factor), V^* 是最优值函数。$P(\mathbf{S}^t,\mathbf{a}^t,\mathbf{S}^{t+\Delta t})$ 是系统状态从时间 t 到 $t+\Delta t$ 的转移概率，代表着系统动力学，即在个体的行为下，状态的变化，但对于机器人来说，其他个体的动力学是未知且难以建模的。而部分论文[7][9]在模型训练时假定该状态转移函数是已知的且在测试时将其简化为直线运动。虽然这种假设可以极大地降低问题的复杂性，但却与系统的实际动力学变化仍有不小的差距。而在该毕设中，通过强化学习来训练一个模型以预测人的未来轨迹，而非简化为直线运动计算。

马尔可夫决策过程的奖励函数参照了论文[9]，奖励函数公式 (2.2) 如下，其中 d_t 是时间 $[t-\Delta t,t]$ 内机器人与其他人的最短距离，$\mathbf{p}_t$ 是机器人的当前位置，$\mathbf{p}_g$ 是机器人的目标位置，0.2 设定为人的舒适距离区，当机器人处于这个范围内则会接受一个负值奖励。该奖励函数会奖励机器人到达目标位置，惩罚机器人碰撞人类或者进入其他人的舒适距离区，而在其他状况下则不会有任何奖励。但奖励或惩罚的状态在机器人实际训练过程中是少数的，即奖励是稀疏的，后续为使模型在深度强化学习时训练收敛，在最初会通过模仿学习专家策略来预训练模型保证算法收敛。

$$R_t(\mathbf{S}_t,\mathbf{a}_t) = \begin{cases} -2, & if\ d_t < 0 \\ -0.1 + d_t/2, & else\ if\ d_t < 0.2 \\ 1, & else\ if\ p_t = p_g \\ 0, & otherwise \end{cases} \tag{2.2}$$

2.2 算法框架

当人们在人群中行走时，他们的移动轨迹往往会受到其他人的影响。因此，对于需要在人群中行走的个体而言，其行动往往不可避免地会考虑到其他人对其移动的影响。针对这一问题，本研究旨在设计和建立一个能够有效地建模并编码人群间交互影响的模型，以此预测其他人未来的移动轨迹，并帮助个体选取最优的行动路径，以更高效和更安全地到达目标点。因此研究如何利用现有的数据来进行交互影响建模，预测人们的移动轨迹并帮助机器人做出更好的决策，是本研究的重要目标。

而关于建模机器人在人群中自主导航的场景，陈等人[10]的工作已经证实了关系图学习的可行性，利用 GCNs 编码了场景下每个节点间的交互信息，同时利用交互关

系来计算新的特征。受关系图学习和 GCNs 模型的启发，该篇毕设利用图结构来建模机器人人群避障场景，利用 GCNs 来计算个体间的交互关系。后续利用这些交互关系来预测其他人的未来移动轨迹和规划一条安全且更快速的移动路径。图2.1则展示了整个算法框架，其主要包含四个模块：

1. 交互建模：利用图结构建模问题场景，并利用 GCNs 进行关系图学习，计算个体间的交互关系特征。
2. 关系汇聚：根据每个个体对机器人影响的重要性，来比例结合机器人和其他所有个体的节点特征并合并为一个固定长度的向量用于后续路径规划。
3. 轨迹预测：将关系图学习后得到的每一个个体（人）的节点特征输入到一个训练后的神经网络中，来预测人的未来轨迹。
4. 路径规划：根据预测的人的未来轨迹，并结合关系汇聚得到的特征，通过值网络选取最优行动来完成任务。

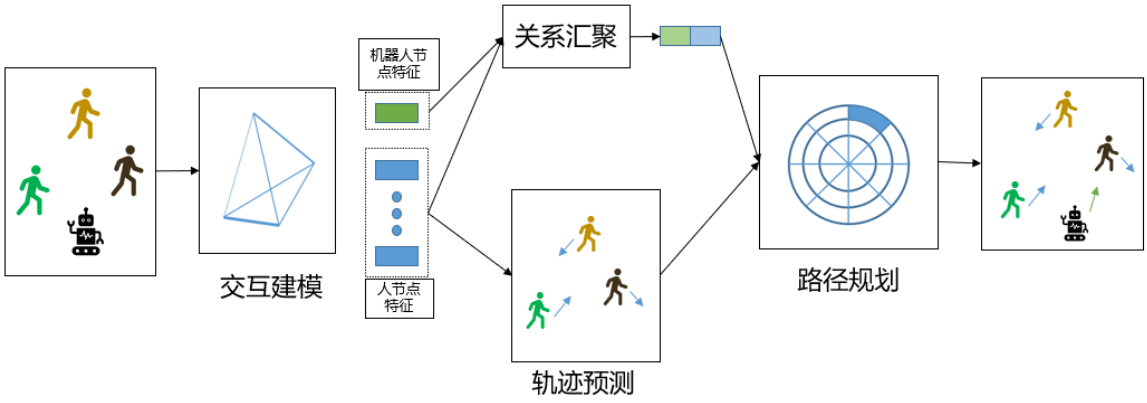


图 2.1 算法框架图，包含四个模块：交互建模，关系汇聚，轨迹预测，路径规划。其中，交互建模模块利用关系图学习建模场景下每个个体间的交互关系，关系汇聚模块将节点特征根据重要性汇聚成一个特征向量，而轨迹预测模块预测人下一时刻的轨迹，路径规划模块结合上述俩个模块的信息来规划一条安全的路径。

在后续的子章节中，将详细展开描述每个模块的算法框架和公式设计。

2.2.1 交互建模

对于机器人在人群中的自主导航场景，建模个体间的交互关系对于机器人导航策略和预测人的未来轨迹都是极其重要的。而利用图结构来建模该问题场景是很符合逻辑的，因为每个个体都有自己的状态和动力学，而这些信息可以被编码在图的节点函数中，而每个个体在移动时会考虑其他个体对他们移动策略的影响，而俩个个体

之间的相互影响程度也是不同的，利用有向图可以很好地表达该影响，其交互关系被编码在有向图的边函数中。如图2.2所示，因此在本篇论文中将该场景用一个有向图结构来建模，数学符号表示为 $G = (V, E)$，其中 $|V| = N + 1$，而边 $e_{i,j} \in E$ 则表示了个体 j 对个体 i 选择行动方式的影响程度。在最开始，机器人只能获取到整个系统下的环境输入状态，而个体间的交互关系是未知的，因此首先需要通过关系推测来获得交互关系。在完成关系推测后，利用图卷积神经网络进行信息传递，计算节点更深层的交互特征。

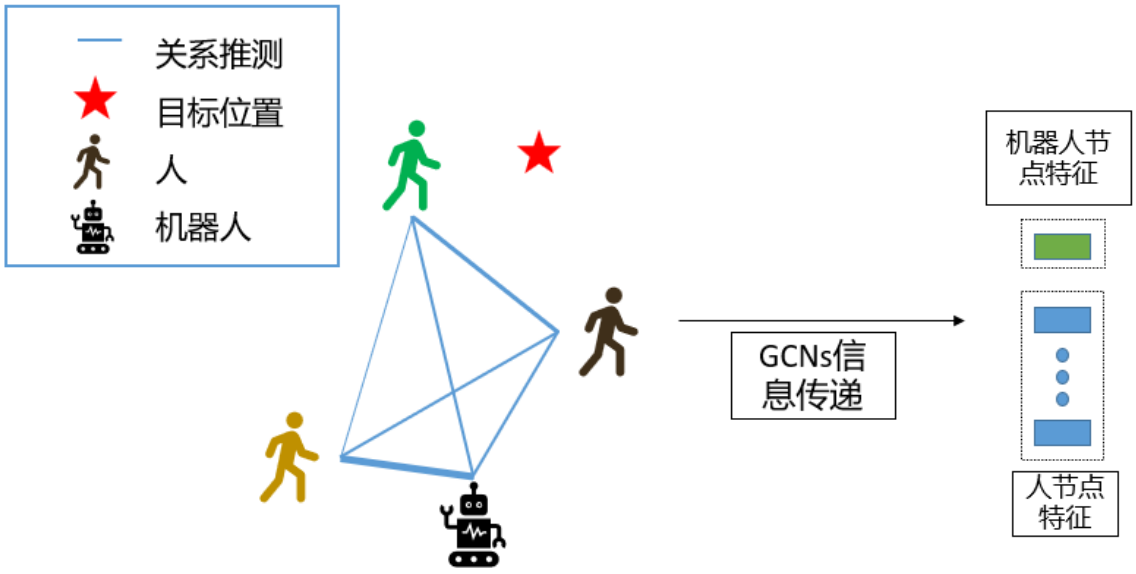


图 2.2 交互建模模块。利用图结构建模机器人在人群中自主导航的场景，通过关系推测初步推测个体间的交互关系矩阵，后续利用图卷积神经网络进行信息传递，计算个体间的交互特征。图中线的粗细程度表示关系推测得到的个体间的交互关系。

当机器人获取到环境输入状态时，图结构的节点 V 的初始值是每个个体的当前状态，即机器人的全部状态和其他人的可观测状态。因此最初机器人和人的节点具有不同的维度，因此在构建图结构对应的矩阵前，需要将初始状态传入到俩个不同的多层感知机 (MLPs) 中，分别为 $f_r()$and$f_h()$，来将它们映射到相同维度的空间中，从而构建特征矩阵 X，特征矩阵第一行为机器人的潜在状态，其余 N 行为人的潜在状态。给定特征矩阵 X，利用成对相似度函数可以计算关系矩阵。在机器学习中，一对数据点的“相似度”通常有多种度量方式，例如欧式距离、余弦相似度、相关系数等。这些度量方式可以通过成对相似度函数的形式来表示，从而用于在空间中描绘和理解数据。本篇毕设根据论文[18]，采用高斯函数作为相似度函数，数学符号表示为 $f(x_i, x_j) = e^{\theta(x_i)^T \phi(x_j)}$，矩阵形式表示为 $A = softmax(X^T W_A X)$，其中 $x_i = X[i,:], x_j = X[j,:], \theta(x_i) = W_\theta x_i, \phi(x_j) = W_\phi x_j, W_A = W_\theta^T W_\phi$。计算得到的关系矩阵在图2.2中表示为交互关系的推测，即个体间线的粗细程度。

基于特征矩阵 X 和关系矩阵 A，利用图卷积神经网络进行信息传递计算更深层的交互特征。信息传递规则参考论文[10]，公式 (2.3) 定义如下：

$$H^{(l+1)} = \sigma(AH^{(l)}W^{(l)}) + H^{(l)} \tag{2.3}$$

式中 $W^{(l)}$ 是第 l 层图卷积神经网络的权重矩阵，$H^{(l)}$ 是第 l 层的节点表征矩阵，σ 则表示激活函数。对该信息传递公式的选择则综合考虑了图神经网络的俩个元素，层间图和跳跃连接。层间图 (layerwise graph) 是指通过在网络中连接不同的层次来构建的图形结构，用于提取节点特征和学习图的结构信息，不同层之间的连接可以帮助学习不同尺度的信息，从而更好地理解图的结构，提高特征的重用性和可解释性，从而增强神经网络的性能。而跳跃连接 (skip connection) 则是指在神经网络中引入额外的连接，使得网络可以直接从输入层到达输出层，从而减轻信息在网络中传递时产生的模糊效应，其作用主要是加深网络的深度，从而增加网络的非线性能力，帮助网络学习到更复杂的问题空间，并加速梯度的反向传递。在实际模型搭建中，考虑到层间图会增加模型的复杂程度和计算复杂度，因此层间图设定为 False 加快计算而跳跃连接设定为 True 来加深网络深度。

通过信息传递，$l+1$ 层的节点特征编码了第 l 层该节点和周围相连节点的特征。在实际代码训练过程中，令 $H^{(0)} = X$，$L = 2$。经过俩层信息传递后，得到状态表征矩阵 $Z^t = H^{(2)}$，其中 $Z^t[i,:]$ 编码了个体 i 的特征和它的交互关系。之后将状态表征矩阵用于预测人的未来轨迹和选取最优移动路径。

2.2.2 关系汇聚

在实际场景下，机器人周围的人的数量会随着时间发生变化，因此所设计的模型需要能够处理任意数量的输入并得到固定维度的输出。一些研究提出将所有人的状态按照到机器人的距离由近到远排序并依次输送到 RNN 结构中进行处理[15]。然而，根据距离的远近并不能够完全对标个体间交互影响的程度，其他因素比如其他个体的速度和方向，对于个体的移动方式选择往往占据更大的影响程度。而在一些研究中，考虑到了多种因素对交互程度的影响，陈等人的工作通过构建多层感知机来学习交互关系的程度权重[8]，用于计算周围个体对机器人的总影响特征，但这却加重了整个模型的复杂程度。

而在该篇毕设论文中，提出了一种不需要多增加网络架构的模块来将不同人数的特征映射到同一维度上，既降低了模型架构的复杂度，也提升了最终的模型效果。整体框架在图2.3中表示。模块首先采用交互建模章节中的关系推测方法，来对经过

信息传递后得到的状态表征矩阵 Z 进行处理，计算新的交互影响程度表示，关系推测依旧采用高斯函数作为成对相似度函数，来得到关系矩阵 A^Z, 数学符号表示为 $f(z_i, z_j) = e^{\theta(z_i)^T \phi(z_j)}$，矩阵形式表示为 $A^Z = softmax(Z^T W_A Z)$，其中 $z_i = Z[i,:], z_j = Z[j,:], \theta(z_i) = W_\theta z_i, \phi(z_j) = W_\phi z_j, W_A = W_\theta^T W_\phi$，且 W_θ, W_ϕ 与交互建模章节中的关系推测共用同一套权重参数。其中，$A^Z_{0,j}, j \in [1, N]$ 则表示为第 j 个人对机器人做出行动方式的影响程度。之后利用 softmax 函数对向量 $[a_j | a_j \in A^Z_{0,j}]$ 进行处理映射到新的实数向量，其中每个元素的值在 0 和 1 之间，并且所有元素的和为 1，从而得到每个个体对机器人影响的权重系数向量 $\alpha = [\alpha_j | j \in [1, N]]$，之后计算得到总的人节点特征向量，计算公式 (2.4) 定义如下

$$z_r = \sum_{j=1}^{N} \alpha_j * z_j \tag{2.4}$$

其次合并机器人节点特征和总的人节点特征得到 $c = [z_0, z_r]$，用于后续输入到值网络进行移动路径规划。

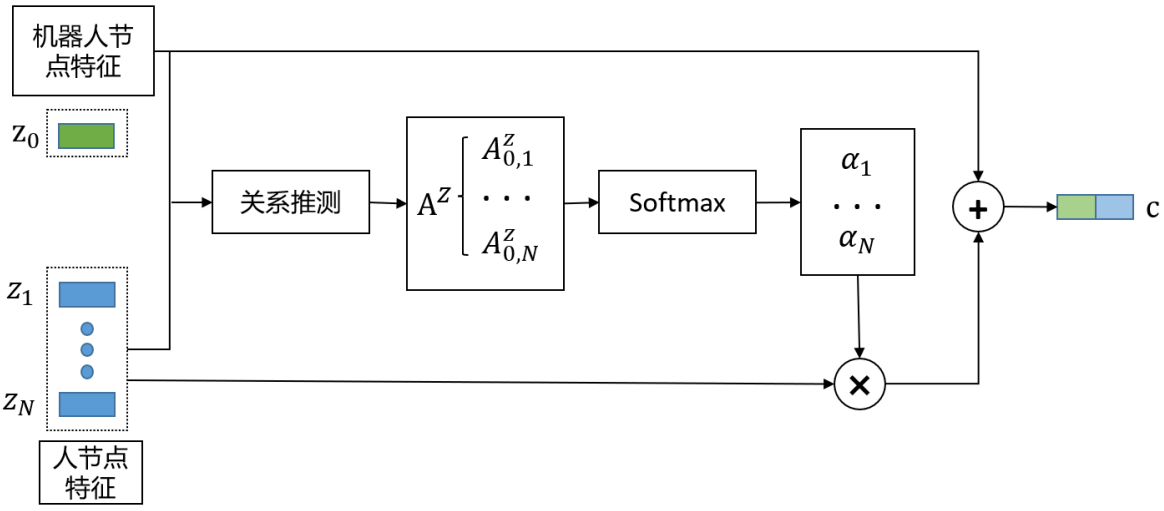


图 2.3 关系汇聚模块。利用关系推测再次计算特征表征矩阵中每个个体的交互关系，利用 Softmax 函数计算每个个体对机器人影响的权重比例，从而将不同人数的节点特征映射到同一个维度向量特征上，与机器人节点特征拼接用于后续路径规划。

2.2.3 轨迹预测

对于机器人的动力学，该篇论文假定机器人可以完美执行它的动作并到达下一状态，因此机器人下一时刻的状态估计 $\hat{s}_0^{t+1}$ 可以完全由当前状态 s_0^t 和动作 a_t 确定，计算公式 (2.5) 表示如下：

$$\hat{s}_0^{t+1} = s_0^t + a^t * \Delta t \tag{2.5}$$

在实际环境中，系统知道机器人所采取的行动并且可以用来计算它的下一时刻的状态，而人的移动方式往往是复杂且难以建模的。部分研究中将人的动力学假定为直线运动[1]，这虽然大大降低了问题的复杂性，但也同时降低了模型的表现性能。而在该篇论文中，采用监督学习训练了一个神经网络模型用来预测人的下一时刻的状态，整个模型表示在图2.4中。该模型计算公式 (2.6) 如下：

$$\hat{s}_i^{t+1} = f_p(Z_t[i,:]) \tag{2.6}$$

式中 $f_p(\cdot)$ 为一个多层感知机，是人的状态预测模块。模型的输入为交互建模模块得到的人的节点特征，即 $Z_t[i,:], 1 \leq i \leq N$，输出的 $\hat{s}_i^{t+1}$ 为第 i 个人下一时刻的状态估计。而总的下一时刻预测的系统状态则为 $\hat{S}^{t+1} = [\hat{s}_0^{t+1}, \hat{s}_i^{t+1}]$，基于预测的环境状态，可用于在奖励函数中计算当前时刻所采取的动作会获得的奖励。而预测人的下一时刻的状态，更好地建模了场景问题，并缓解了在人数多的情况下的机器人冻结问题。

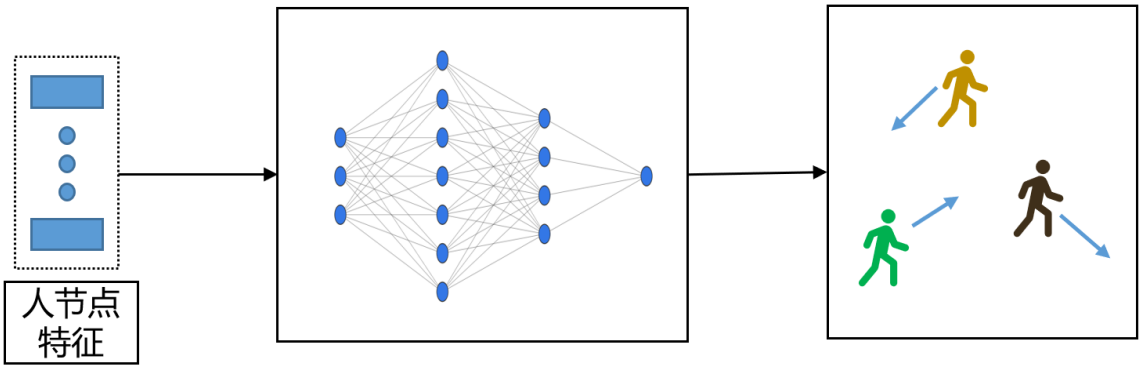


图 2.4 轨迹预测模块。将关系建模得到的人节点特征输入到一个监督学习训练后的轨迹预测网络中，来计算人的下一时刻的状态估计。

2.2.4 路径规划

图2.5展示了算法中的路径规划模块，该模块采用陈等人研究工作[10]中所使用的奖励函数，并结合轨迹预测模块得到的结果，来计算机器人采取行动后下一时刻会获得的奖励。同时，将关系汇聚模块得到的固定状态向量输入到一个值网络中进行状态值估计，从而进行路径规划。该值网络估计模块定义为 f_V，是利用多层感知机来构建，其计算公式 (2.7) 如下：

$$v = f_v(c) \tag{2.7}$$

式中 c 为关系汇聚得到的总特征向量，v 为当前状态的估计值。根据经过训练后得到的最优值网络，策略模型会选取具有最高值的状态，并计算要到达状态所需采取的动

作。路径规划所考虑的动作空间是离散的，根据实际运行情况可分为俩类。在仿真运行时，机器人的移动方式为全向运动 (holonomic)，即完整动力学模型机器人，数学符号表示为 $\mathbf{a} = (v_x, v_y)$。全向运动机器人具有完整的自由度，可以在任意位置和任意方向上移动和旋转，而在相关研究中，均采用全向运动学的情况来测试模型。而在实际环境运行中，所使用到的实体机器人一般是差速地盘，其运动学为差速驱动机器人 (unicycle)，即非完整运动学模型机器人。差速驱动机器人一般只有两个自由度，一个用于沿着机器人轴向移动，另一个用于绕 Z 轴旋转。

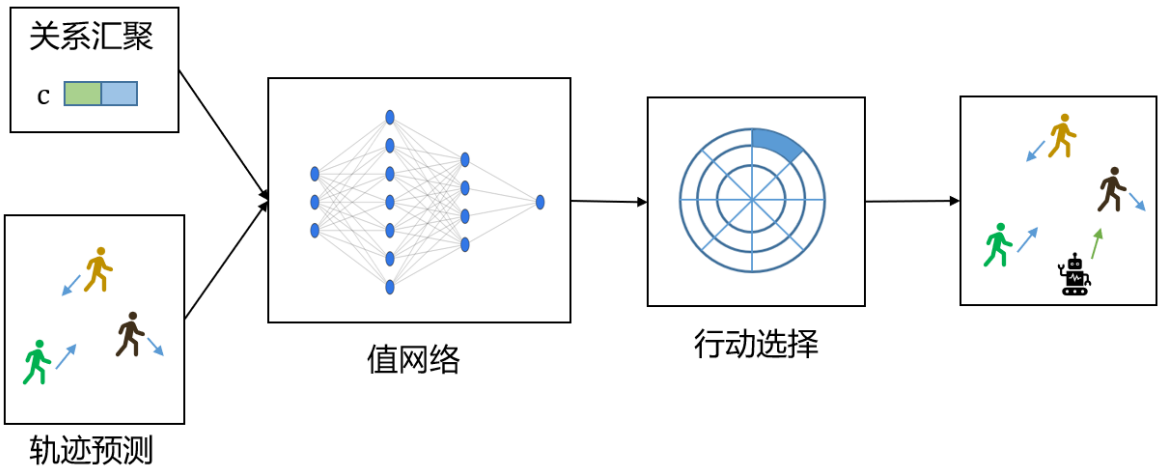


图 2.5 路径规划模块。结合人的轨迹预测，将关系汇聚模块得到的固定状态向量输入到一个值网络中进行状态值估计，从而在一个离散行动空间中选取最优行动。

2.3 模型训练

2.3.1 深度强化学习模型

关于轨迹预测网络 f_p 和值估计网络 f_V 的训练过程展示在算法1中，该算法架构采用了陈等人研究中的训练模型[10]。训练时，首先利用模仿学习初始化轨迹预测网络 f_p 和值估计网络 f_V（1-3 行）。这里采用经典算法模型 ORCA[1]作为专家策略，利用 ORCA 算法收集演示数据初始化模型，并将演示数据存放至经验回放缓存。经验回放缓存是用于存储智能体与环境交互历史记录的数据结构，在训练阶段中，智能体不会直接使用当前时刻生成的经验，而是从经验回放缓存中随机选择一批经验元组进行训练，从而平衡新旧经验的使用，使得智能体可以有效地利用以前的经验，减少数据的相关性，并更快地收敛到最优策略。同时对于后续强化学习训练值估计网络 f_V，需要拷贝 f_V 得到目标值网络 $\hat{f}_V$ 以保证强化学习收敛。而采用模仿学习初始化模型是因为该篇毕设中所选用的奖励函数是稀疏的，因此采用专家策略可以保证后续强化学习训练时模型可以收敛且训练更快。

Algorithm 1 f_P 和 f_V 模型训练

1: 使用演示数据 D 初始化 f_P 和 f_V
2: 初始化目标值网络 $\hat{f_V} \leftarrow f_V$
3: 初始化经验回放缓存 $E \leftarrow D$
4: **for** 对于每一个轮次 (episode) **do**
5: 初始化随机序列 S^0
6: **repeat**
7: $a_t \leftarrow \arg\max_{a_t \in A} \hat{R}(S^t, a^t, \hat{S}^{t+1}) + \gamma^{\Delta t} f_V(\hat{S}^{t+1})$, 其中 $\hat{S}^{t+1} = f_P(S^t, a^t)$
8: 执行动作 a_t，获得奖励 r^t 并到达下一状态 $S^{t+\Delta t}$
9: 存储元组 $(S^t, a^t, r^t, S^{t+\Delta t})$ 至 E
10: 从 E 中随机采样 mini-batch 元组
11: 设定值网络的目标: $y_i = r_i + \gamma^{\Delta t} \hat{f}_V(S_i)$
12: 通过最小化 $L_1 = ||f_V(S_i) - y_i||$ 更新 f_V
13: 设定轨迹预测网络的目标: S_{i+1}
14: 通过最小化 $L_2 = ||f_P(S_i, a_i) - S_{i+1}||$ 更新 f_P
15: **until** 达到终结状态 s_t 或者 $t \geq t_{max}$
16: 更新目标值网络 $\hat{f_V} \leftarrow f_V$
17: **end for**
18: **return** f_P, f_V

之后将执行一系列的轮次 (episode)，在每个 episode 中，首先初始化一个随机状态，记为 S^0，并初始化机器人和人的目标点位置（4-5 行）。其次在每个 episode 中判别当前策略模型和当前状态 S^t 以选择一个最优的动作 a^t，执行该动作并移动到下一个状态 $S^{t+\Delta t}$，之后随机采样 mini-batch 元组并设定值网络和轨迹预测网络的目标，其次反向传播更新神经网络 f_V 和 f_P 的权重，具体细节见算法1中的子步骤 (6-15 行)。在每一个步骤中，首先根据 $\epsilon - greedy$ 策略来选取动作。其结合了两种策略：一种是贪婪策略，即在目前已知的最佳策略上做出最佳动作；另一种是随机策略，即在所有动作中随机选择一个动作。具体地说，当执行 $\epsilon - greedy$ 策略时，智能体将以 ε 的概率选择一个随机动作（探索），以 $1 - \epsilon$ 的概率执行贪心动作（利用），其中贪心动作是基于当前已知策略获得的最佳动作，公式如下：

$$a_t \leftarrow \arg\max_{a_t \in A} \hat{R}(S^t, a^t, \hat{S}^{t+1}) + \gamma^{\Delta t} f_V(\hat{S}^{t+1}), where\ \hat{S}^{t+1} = f_P(S^t, a^t) \tag{2.8}$$

该公式会计算每个动作的 Q 值，其中 $\hat{S}^{t+1}$ 是使用轨迹预测网络 f_P 和当前动作 a^t 预测得到的下一个状态，其次选择 Q 值最大的动作作为当前状态下的执行动作 a_t。将选定的动作 a_t 应用到当前状态 S^t 上，其次获得新的状态 $S^{t+\Delta t}$ 和相应的奖励值 r^t。并将当前时刻 t 产生的元组 $(S^t, a^t, r^t, S^{t+\Delta t})$ 存储在经验回放缓存 E 中，以便后面的 minibatch 梯度下降训练过程中随机选择。其次每次从经验回放缓存 E 中随机选择一批元组，用于更新值估计网络 f_V 和轨迹预测网络 f_P 的参数。计算值网络目标值 y_i

的公式为：

$$y_i = r_i + \gamma^{\Delta t} \hat{f}_V(S_{i+1}) \tag{2.9}$$

式中 $\hat{f}_V(S_{i+1})$ 是目标值网络对下一个状态 S_{i+1} 的估计值。目标值网络与当前计算 Q 值的网络的不同点是它的参数不是实时更新的，而是以某个固定频率更新。这样，就可以避免更新目标值的神经网络和估算 Q 值的神经网络发生协同更新，从而降低了更新目标值过程中出现的误差和不稳定性，用于提高算法的效率和稳定性。其次用目标值 y_i 和当前值估计网络 f_V 对当前状态 S_i 输出的值进行比较，计算误差函数为 $L_1 = ||f_V(S_i) - y_i||$，然后采用梯度下降方法更新值估计网络的权重。而对于训练轨迹预测网络，则利用 minibatch 元组中下一状态和对应的动作作为预测网络的输入，计算预测网络对下一状态的预测值，计算误差函数为 $L_2 = ||f_P(S_i, a_i) - S_{i+1}||$，并通过梯度下降方法更新轨迹预测网络 f_P 对应的参数。

以上整个步骤重复执行，直到到达终止状态或者超过最大时间步长 t_{max}。而每个 episode 结束后 (16 行)，将当前的值估计网络 f_V 的权重拷贝到目标值网络 $\hat{f}_V$ 中。这样可以保证值网络在更新时的目标值相对稳定，有助于提高训练的稳定性和模型的收敛速度。当所有 episode 训练结束后即结束循环，返回训练好的轨迹预测网络 f_P 和值估计网络 f_V，以供测试阶段使用。

2.3.2 实现细节

在实际代码训练过程中，关系推测前的状态映射函数 $f_r(\cdot)$，$f_h(\cdot)$，轨迹预测网络 $f_p(\cdot)$，值估计网络 $f_v(\cdot)$ 的隐藏单元维度分别为 $(64, 32), (64, 32), (64, 5), (64, 64, 64, 1)$。网络模型参数训练上采用 Adam 算法[19]。对于模仿学习，首先利用专家策略 ORCA 收集 3000 个演示数据，并经过 50 个 epochs 的迭代训练，学习率设定为 0.001。对于强化学习训练，学习率设定为 0.001，折现因子 γ 设置为 0.9，而 $\epsilon - greedy$ 策略的探索率值最初设定为 0.5，在训练过程的前 5000 个轮次中，从 0.5 线性衰减到 0.1，在后续的轮次中维持 0.1 数值不变。而整个模型训练是在 AMD R7-6800H CPU 上训练的，大概需要训练 13 个小时。

在机器人动作的选择上，机器人的动力学模型设定为全向驱动模型，即机器人可以沿任意方向运动。动作空间包含 80 个离散动作，其中方向在 $[0, 2\pi)$ 内均匀选取 16 个朝向，而速度则在 $(0, v_{pref}]$ 范围内指数分布选取 5 个速度大小，从而构建 80 个离散动作。

3 仿真验证

3.1 仿真环境介绍

仿真环境采用陈等人论文[8]中所开源的仿真代码 CrowdNav[1]。整个仿真环境主要包含以下元素：

环境：环境代表着机器人在人群中自主导航的场景，其中一共包含 $N+1$ 个个体，即机器人穿梭具有 N 个人的人群到达指定目标点。场景中人的移动策略模型为 ORCA[1]，而每个人的模型参数采样自高斯分布以引入行为的多样性，来增加场景的复杂性。CrowdNav 仿真环境则基于 OpenAi gym 官方库搭建，主要实现以下俩个抽象类方法。

- reset()：环境将重置所有个体的位置，并返回机器人的观察值。
- step(action)：接受机器人的动作作为输入，环境会计算出每个个体的观察值，并调用 agent.act（observation）获取个体的动作。其次环境会检测个体之间是否发生碰撞。如果没有，个体的状态将被更新，然后返回观察值、奖励和完成情况。

个体：Agent（个体）是一个基类，其有两个派生类：human（人类）和 robot（机器人）。Agent 类持有一个智能体的所有物理属性，包括位置、速度、方向、策略等等。其中还包含以下几个重要部分：

- 可见性：人类的属性一定是可见的，但机器人被设置为不可见，这是为了防止在强化学习训练过程中，使机器人学习到偏侵略性的移动策略。
- 动力学：可以是完整动力学（沿任意方向移动）或单轮动力学（具有旋转约束），在仿真环境测试时选用完整动力学。
- act(observation) 函数：将其他个体的可观测状态结合自身全部状态得到环境输入状态，并将其传递给策略得到应采取的动作。

状态：在不同情况下，状态有多种定义，分类如下

- ObservableState：一个个体的位置、速度和半径
- FullState：一个个体的位置、速度、半径、目标位置、首选速度
- JoinState：一个个体的完整状态和所有其他个体的可观察状态的拼接

[1]https://github.com/vita-epfl/CrowdNav

动作：机器人所采取的动作类型根据动力学模型确定，分为以下俩类：

- ActionXY: (vx, vy)，完整动力学模型，即机器人可以沿任意方向移动
- ActionRot: (speed, rotation)，单轮动力学，即机器人具有旋转约束

3.2 仿真结果对比

仿真测试场景为五个人随机分布在一个半径等于 4m 的圆上，且这五个人的目标点位于圆上与起始位置圆心对称的地方。而所有个体的首选速度 $v_{pref} = 1m/s$。为更好地评估模型性能，机器人可见性设置为不可见，即人群只需要考虑其他人的影响并作出自己的路径规划，该设置一方面可以检验模型对场景的建模能力，另一方面则是为了防止模型训练生成偏侵略性的策略。最后将训练好的模型在 500 个随机的案例上进行评估测试。

以下将从定量定性俩个方面对模型性能进行对比。为了更好地评估该篇毕设所提出的模型性能，将与其他经典模型进行对比，来验证模型的可行性，其中包括 ORCA[1]，SARL[8]，RGL_OneStep[10]。而本篇毕设中所提出模型简称为 Gattn，同时为了证明轨迹预测模块带来的优异性能，增添了 Gattn_liner 模型，其轨迹预测部分则简化为直线运动。以下是这些模型的简要介绍：

- ORCA：基于规则的传统方法，在互惠假设下根据当前环境状态计算无碰撞速度。
- SARL：使用交互模块来建模人和机器人之间的交互关系，并利用自注意力机制汇聚人对机器人的交互信息。
- RGL_OneStep：RGL 模型包含 (MCTS) 蒙特卡洛树搜索算法，根据未来多个时间步长的轨迹预测，并计算相应的奖励和来作出动作选择。其完整模型考虑了后续多个时间步长，而为了更好地对比模型建构框架的性能，舍弃了 RGL 的 MCTS 部分，即 RGL_OneStep。
- Gattn：本篇毕设的算法框架，详细架构见章节2。
- Gattn_liner：在本篇毕设的算法框架上，将轨迹预测网络替换为直线运动建模，来对比验证轨迹预测网络的优越性。

3.2.1 定量比较

定量比较所采用的度量标准如下：

- 成功率：机器人成功达到目标点的比率。
- 碰撞率：机器人与人类碰撞的比率。
- 超时率：机器人未在限定时间到达目标点且未发生碰撞的比率。
- 导航时间：机器人到达目标点的平均时间。
- 总奖励：所有 episodes 下奖励和的平均值。

仿真实验结果汇总在表3.1中。

策略	成功率	碰撞率	超时率	导航时间 (s)	总奖励
ORCA	0.94	0.05	0.01	12.52	0.2531
SARL	0.97	0.02	**0.01**	10.73	0.3160
RGL_OneStep	0.96	**0.01**	0.03	**10.28**	0.3170
Gattn	**0.98**	**0.01**	**0.01**	10.40	**0.3211**
Gattn_linear	0.85	0.01	0.14	10.31	0.2721

表 3.1 定量对比结果。测试场景为五个人随机分布在一个圆形场景上，目标点在对称位置，机器人需穿过人群到达指定位置。

对表3.1的数据进行分析，可以发现基于规则的传统方法 ORCA 仍有不错的表现结果，但在五个模型中有着最高的碰撞率，这是因为在训练过程中，机器人的可见性设置为不可见，这与 ORCA 模型中互惠避障的假设恰恰相反。关于 Gattn 和 Gattn_linear 模型的对比，可以发现 Gattn_linear 模型的表现性能是最差的，具有最低的成功率和最高的超时率，这说明将人的动力学简化为直线运动并不能够很好地表征其运动方式，而 Gattn 所采用的轨迹预测模块可以更好地推测出人的未来轨迹，从而生成更好的移动策略。而通过 Gattn 与其他经典的基于机器学习的模型 RGL 和 SARL 对比，可以发现，三个模型的表现性能均很好，这说明 Gattn 利用图结构建模场景，并构建值估计网络和轨迹预测网络的可行性。综合比较，可以发现 Gattn 的表现性能是最好的，尽管该模型仍旧不是零碰撞，这是因为机器人的属性设置为不可见，从而导致一些碰撞是难以避免的。

为了对模型进行更全面的对比，将仿真场景下的人数设置为 5、10、15、20 四种情形，并将以上五种算法模型在四个不同人数的情况下进行测试，实验结果整理在图3.1中。整体来看，五种算法模型均在人数少的情况下有着好的表现性能，并随着人数的增加，超时率和碰撞率也不断增加。但该性能下降情况在 Gattn 模型上表现最为明显，这里的一个合理猜测是 Gattn 模型训练时并未对不同网络架构的参数进行结

果对比，且整个模型是在人数为 5 的场景下进行训练的，所以猜测网络架构参数并不合理，导致模型过拟合，因此随着人数增加，模型表现性能下降最大。后续需要尝试其他不同的架构参数进行对比，来优化 Gattn 模型在不同人数下的表现性能。

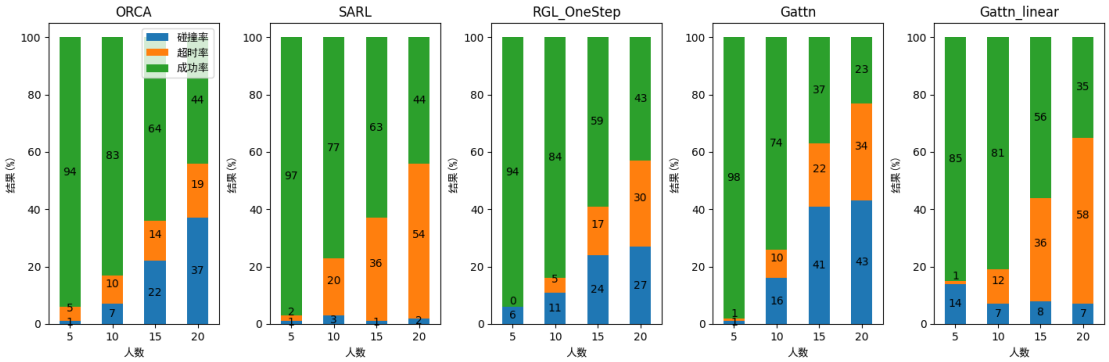


图 3.1 不同人数（5、10、15、20）下机器人自主避障导航任务的成功、超时以及碰撞情况。

3.2.2 定性比较

图3.2比较了 ORCA[1]，SARL[8]，RGL_OneStep[10]，Gattn，Gattn_linear 五个模型在同一场景下的运动轨迹。其中，机器人对于人是不可见的，从而保证所有场景下的人的移动路线是相同的，便于比较模型的性能。综合对比可以发现，ORCA 为避免与人群发生碰撞，采取了一条路径更长的路线，并以最长的时间到达目标点，动作选择上比较僵硬，基本只改变了俩次移动方向。而 SARL 模型通过对人群和机器人之间交互关系影响的建模，选取了一条最短的移动路径，但该路径的选择是偏侵略性的，与其他模型相比，SARL 生成的轨迹有着更多进入人的舒适范围的次数。RGL_OneStep 利用图结构建模场景，更深层地建模了场景下的交互关系，并选取了一条更安全的路径到达目标点。Gattn 相比 RGL_OneStep，增加了关系汇聚模块，编码了更高层的交互特征，用于移动策略选择。同时 Gattn 与其他模型相比，选取的路径移动时间最短，并与其他行人保持了安全距离。Gattn_linear 模型采用直线运动代替轨迹预测网络，虽然大大减少了模型的复杂性，但由于并未很好地考虑到人的移动方式，所生成的轨迹具有极高的震荡。

除了对整体路线轨迹进行分析，图3.3展示了俩个特殊场景下的动作策略选择，来定性分析 Gattn 模型的表现性能。在场景一中，机器人和五个人的距离都很接近，且数字标为 3 的人处于机器人和目标位置的中间。而 Gattn 模型预测沿 80° 方向以最高速度移动会有最高的值，即最高的折现奖励和。这是因为数字标为 3 的人在沿左上方移动，其他人的移动则在远离机器人，由此可知机器人沿 80° 方向全速移动并不会与人群发生碰撞，同时能更早地到达目标点。而沿着略高于 90° 的移动方向则可能会在

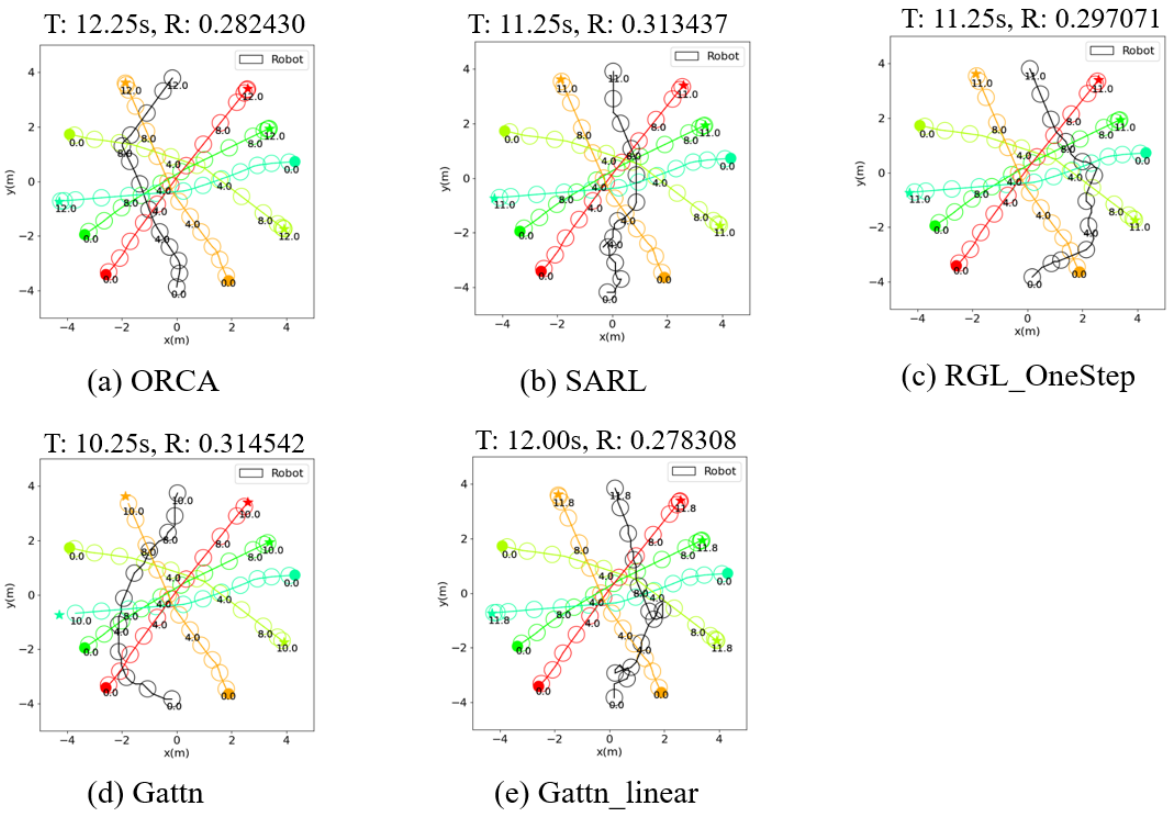
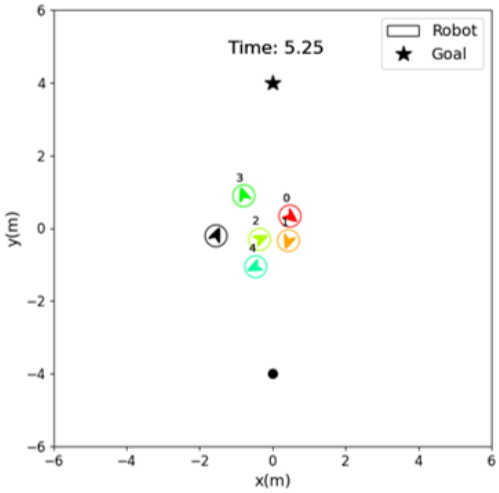
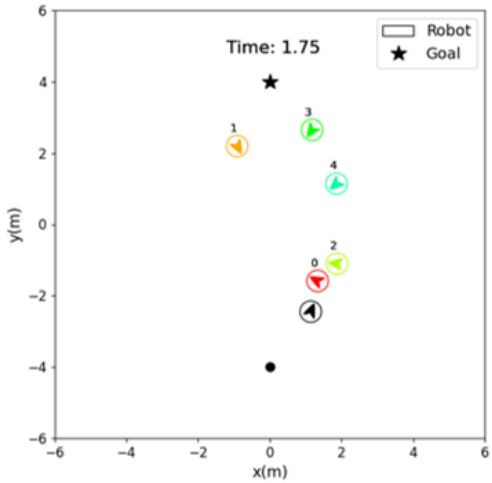


图 3.2 同一场景下的路径比较。其中，机器人的属性为不可见，彩色圆圈为人的移动路径，并标记有对应的时间，黑色圆圈为机器人的运动轨迹。

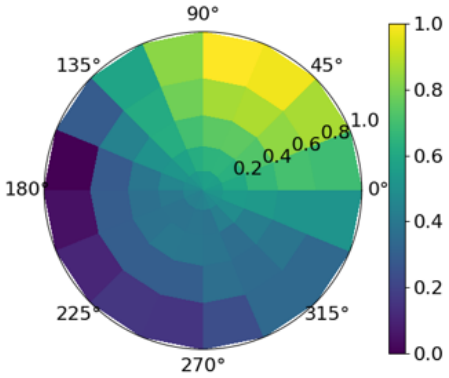
未来与数字标为 3 的人发生碰撞，因此在该方向上具有较低的值估计。而在场景二中，数字标为 0 和 2 的俩个人位于机器人的右上方，而目标位置则位于机器人的左上方，Gattn 模型计算认为沿着 80° 方向全速移动具有更高的值，这是因为数字标为 0 和 2 的俩个人均在向左上方移动。而机器人同样沿这个方向移动的话会进入人的舒适范围，由此可知机器人在该方向上计算得到一个略低于最优行动的值。以上俩个特殊场景均表明 Gattn 模型可以很好地建模该场景下机器人和人群的交互关系，并综合考量到人的轨迹预测，从而做出具有远见的最优行动选择。



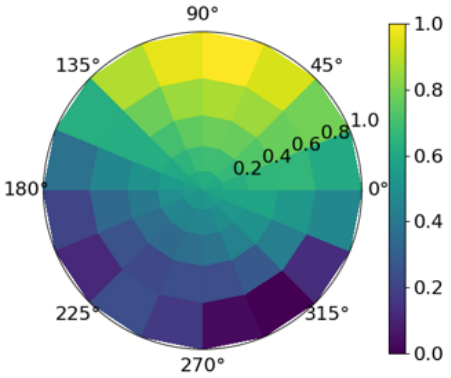
(a) 场景一



(b) 场景二



(c) 场景一的值估计



(d) 场景二的值估计

图 3.3 动作值估计。利用 Gattn 模型，对俩个特定场景进行值估计计算，(a)(b) 展示了俩个特殊的场景，(c)(d) 则是在这些场景下对机器人的动作空间进行值估计，其中黄色区域代表着该动作具有更高的值。

4 结论

本毕业设计主要设计并实现了一个用于机器人在人群中避障移动的算法模型。该算法模型采用图结构建模问题场景，利用图神经网络信息传递得到个体间更深层的交互特征，之后将这些信息用于人的轨迹预测和移动规划。通过人群中的交互关系建模，该模型可以综合考量到其他人的移动轨迹并作出行动规划。该算法模型采用 PyTorch 实现，并利用开源的强化学习算法进行训练，且在已有的仿真环境下进行了定性和定量俩方面的实验验证。通过与多个经典模型的对比，结果验证了该算法框架的可行性和其对问题场景的建模能力。而该模型所表现的优良性能不仅可用于机器人人群导航场景，也可以扩展到其他分布式场景中使用，用于建模分布式个体间更深层的交互特征。

然而，该算法框架只局限于仿真测试，并未部署到实物机器人上进行完整的测试。在实物测试上可能会遇到多种潜在问题，需要后续进行实体测试以发现问题。同时，该算法的构建包含了多个模型框架，虽然有着良好的表现性能，但导致训练时间更长，有进一步的改良空间。

参考文献

[1] VAN DEN BERG J, LIN M, MANOCHA D. Reciprocal velocity obstacles for real-time multi-agent navigation[C]//2008 IEEE international conference on robotics and automation. 2008: 1928-1935.

[2] VAN DEN BERG J, GUY S J, LIN M, et al. Reciprocal n-body collision avoidance[C]//Robotics Research: The 14th International Symposium ISRR. 2011: 3-19.

[3] HELBING D, MOLNAR P. Social force model for pedestrian dynamics[J]. Physical review E, 1995, 51(5): 4282.

[4] KUDERER M, KRETZSCHMAR H, SPRUNK C, et al. Feature-based prediction of trajectories for socially compliant navigation.[C]//Robotics: science and systems. 2012.

[5] AOUDE G S, LUDERS B D, JOSEPH J M, et al. Probabilistically safe motion planning to avoid dynamic obstacles with uncertain motion patterns[J]. Autonomous Robots, 2013, 35: 51-76.

[6] TRAUTMAN P, KRAUSE A. Unfreezing the robot: Navigation in dense, interacting crowds[C]//2010 IEEE/RSJ International Conference on Intelligent Robots and Systems. 2010: 797-803.

[7] CHEN Y F, LIU M, EVERETT M, et al. Decentralized non-communicating multiagent collision avoidance with deep reinforcement learning[C]//2017 IEEE international conference on robotics and automation (ICRA). 2017: 285-292.

[8] CHEN Y F, EVERETT M, LIU M, et al. Socially aware motion planning with deep reinforcement learning[C]//2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). 2017: 1343-1350.

[9] CHEN C, LIU Y, KREISS S, et al. Crowd-robot interaction: Crowd-aware robot navigation with attention-based deep reinforcement learning[C]//2019 international conference on robotics and automation (ICRA). 2019: 6015-6022.

[10] CHEN C, HU S, NIKDEL P, et al. Relational graph learning for crowd navigation[C]//2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). 2020: 10007-10013.

[11] CHEN Y, LIU C, SHI B E, et al. Robot navigation in crowds by graph convolutional networks with attention learned from human gaze[J]. IEEE Robotics and Automation Letters, 2020, 5(2): 2754-2761.

[12] DRIGGS-CAMPBELL K, GOVINDARAJAN V, BAJCSY R. Integrating intuitive driver models in autonomous planning for interactive maneuvers[J]. IEEE Transactions on Intelligent Transportation Systems, 2017, 18(12): 3461-3472.

[13] TAI L, ZHANG J, LIU M, et al. Socially compliant navigation through raw depth inputs with generative adversarial imitation learning[C]//2018 IEEE international conference on robotics and automation (ICRA). 2018: 1111-1117.

[14] LONG P, LIU W, PAN J. Deep-learned collision avoidance policy for distributed multiagent navigation[J]. IEEE Robotics and Automation Letters, 2017, 2(2): 656-663.

[15] EVERETT M, CHEN Y F, HOW J P. Motion planning among dynamic, decision-making agents with deep reinforcement learning[C]//2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). 2018: 3052-3059.

[16] ALAHI A, GOEL K, RAMANATHAN V, et al. Social lstm: Human trajectory prediction in crowded spaces[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2016: 961-971.

[17] GUPTA S, DAVIDSON J, LEVINE S, et al. Cognitive mapping and planning for visual navigation [C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2017: 2616-2625.

[18] WANG X, GIRSHICK R, GUPTA A, et al. Non-local neural networks[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2018: 7794-7803.

[19] KINGMA D P, BA J. Adam: A method for stochastic optimization[J]. arXiv preprint arXiv:1412.6980, 2014.

致谢

在此向毕业设计的指导老师丁克蜜副教授致以最诚挚的谢意。感谢您在整个毕业设计过程中的指导和支持，让我能够在本科生涯的最后阶段完成这个充满挑战和创新的项目。同时，我也要感谢柯文德副教授在毕业设计上提供的硬件支持和指导以及本科科研阶段贾振中助理教授的精心指导，这些都为我在毕业设计中的理论和实践工作提供了帮助。

在此之外，我还要感谢本科阶段一路走来的学长和同学们。感谢你们在我进行科研和竞赛的过程中提供的帮助和支持，让我能够更好地理解一些机器人技术的应用，并提高了我的团队合作能力。

最后，我要特别感谢我的家人。感谢你们一直以来的关心和支持，让我能够专注于学业和研究，面对挑战和困难时始终坚定前行。你们的支持是我不断前进的动力和信心来源。

尽管毕业设计仍有些许遗憾，但我会秉持着不放弃的精神，在后续的人生旅途中继续努力。