

基于深度强化学习的多智能体策略优化研究

Research on Multiagent Policy Optimization Based on Deep Reinforcement Learning

一级学科: 软件工程
学科专业: 软件工程
作者姓名: 郑岩
指导教师: 孟昭鹏

天津大学智能与计算学部软件学院
二〇一九年五月二十二日

独创性声明

本人声明所呈交的学位论文是本人在导师指导下进行的研究工作和取得的研究成果，除了文中特别加以标注和致谢之处外，论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得 天津大学 或其他教育机构的学位或证书而使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示了谢意。

学位论文作者签名: 签字日期: 年 月 日

学位论文版权使用授权书

本学位论文作者完全了解 天津大学 有关保留、使用学位论文的规定。特授权 天津大学 可以将学位论文的全部或部分内容编入有关数据库进行检索，并采用影印、缩印或扫描等复制手段保存、汇编以供查阅和借阅。同意学校向国家有关部门或机构递交论文的复印件和磁盘。

(保密的学位论文在解密后适用本授权说明)

学位论文作者签名: 导师签名:
签字日期: 年 月 日 签字日期: 年 月 日

摘要

近年来，关于深度强化学习的研究受到了广泛的关注并取得了大量的研究成果。如何有效促进智能体进行策略优化是深度多智能体强化学习领域的重要研究问题，然而，在有效地解决多智能体环境下的策略优化问题方面，仍存在一定的局限性与挑战。首先，从环境的角度，既有的深度强化学习算法对于环境中多元感知信息的处理存在一定的局限性；其次，从强化学习算法的角度，既有算法存在估值偏差的局限性以及对奖赏值中噪声处理的局限性；最后，从多智能体系统的角度，既有算法在面对独立学习智能体时，存在难以实现策略协同优化的局限性，以及面对非静态对手时，存在对手判别不准确的局限性。针对上述挑战，本文聚焦于基于深度强化学习的多智能体策略优化研究，从环境、强化学习算法以及多智能体三个角度展开，对既有算法存在的局限性进行分析，并提出相应的解决方法。论文的主要工作内容如下：

首先，本文考虑多模态信息输入的智能体策略优化问题，针对既有算针对多源感知信息输入的局限性，提出了基于分离式多模态输入的强化学习框架，拓展了强化学习算法处理多模态输入的能力。进一步，针对一般注意力机制对多模态信息输入权重分配的局限性，提出了层次注意力机制，实现了多模态间以及模态内的注意力权重分配，增强了多尺度的特征提取能力。最后，针对LSTM网络处理多模态输入的局限性，对LSTM进行了拓展，提出了基于多信息流的LSTM网络结构，实现了对多源信息输入的有效处理。本研究增强了既有算法处理多模态信息输入的能力，有效地利用多模态信息实现智能体的策略优化。

其次，本文考虑了噪声环境下独立学习智能体的估值纠偏与策略优化问题，针对强化学习算法存在估值偏差的局限性，提出了基于双权估计器的WDDQN算法，实现了有效的估值纠偏。接着，针对既有算法对奖赏值中噪声处理的局限性，提出了奖赏值网络RN实现了有效的降噪。同时，针对多智能体环境中，既有算法难以促进独立学习智能体实现协同收敛优化的局限性，提出了宽容的奖赏值网络LRN，有效的促进智能体之间的协同策略优化。最后，针对多智能体系统中策略优化效率偏低的局限性，提出了调度经验回放策略SRS，有效地提升了策略优

化的效率。综上，本研究有效地实现了深度强化学习算法的估值纠偏，促进了多智能体间策略协同优化，以及帕累托最优纳什均衡策略的求解。

最后，本文考虑了面向非静态对手环境下的多智能体策略优化问题，针对多智能体环境下非静态对手的挑战以及既有算法使用单一策略来应对非静态对手的局限性，提出了基于贝叶斯策略重用的deep BPR+算法，有效应对非静态对手的复杂行为。提出使用对手模型来构建修正的置信模型RBM，从奖赏值信号和对手行为型号双重角度实现准确的对手策略检测。此外，提出使用蒸馏策略网络DPN作为应对策略库，实现了对未知策略的快速学习、高效的策略存储与重用。综上，本研究实现了准确的对手类型检测以及高效的策略重用，对于应对多智能体环境下非静态对手的多智能体策略优化问题具有一定的指导意义。

综上，本文以基于深度强化学习的多智能体策略优化为研究目标，从环境、强化学习算法以及多智能体系统三个角度展开研究，深入探讨了面向多模态感知信息输入的智能体策略优化问题、面向噪声环境下独立学习智能体策略优化问题，以及面向非静态对手环境下的策略优化问题，并通过实验论证了本文研究成果的有效性。本研究兼顾工程实践与科学研究，对使用强化学习算法解决实际问题起到了一定的指导作用。同时，为多模态强化学习、独立学习智能体的策略优化、帕累托最优纳什均衡策略的求解以及应对非静态对手等领域提供了一定的参考价值。

关键词： 深度强化学习；多智能体系统；非静态对手；贝叶斯策略重用；策略优化；多模态学习

ABSTRACT

Recently, DRL has received extensive attention and achieved many research results. In the domain of deep reinforcement learning (DRL) and multiagent system (MAS), achieve efficient policy optimization is a key problem and has some limitations and challenges. Firstly, from the perspective of the environment, the existing DRL algorithms have limitations on handling the multimodal inputs. Secondly, from the perspective of DRL algorithms, there exists biases in the estimation of Q-value and existing DRL algorithms are unable to handle noise in the received rewards. Lastly, from the perspective of MAS, existing algorithms are unable to achieve efficient cooperation between independent learners, as well as policy optimization against non-stationary opponents. To address these, this paper focuses on research of multiagent policy optimization based on DRL algorithms and tries to develop effective policy optimization algorithm by overcoming these limitations from perspectives of environment, DRL algorithms, and MAS. The main contents of this paper are as follows:

First, this paper studies the problem of policy optimization with multimodal inputs. The separated multimodal network (SMMN) is proposed to overcome the shortcoming of handling multimodal inputs. SMMN can be easily combined with vanilla DRL algorithms to handling multimodal inputs. Besides, hierarchical attention (HA) mechanism is proposed to achieve weight allocation between and within multimodal inputs, resulting in better feature extraction results. At last, a modified LSTM network is proposed to effectively handle multiple inputs. This study enhances the ability of existing algorithms in handling multimodal inputs and effectively achieving policy optimization.

Second, this paper studies the problems of estimation correction and policy optimization of independent learners under noisy environments. To reduces the estimation bias in DRL algorithms, WDDQN is proposed based on weighted double estimators. The reward network (RN) is proposed to handle the noise in rewards. Meantimes, to encour-

age agents to achieve cooperation, the lenient reward network (LRN) is proposed based on the notion of leniency. At last, the scheduled replay strategy is proposed to achieve efficient policy optimization. In summary, this study achieves effective estimation correction, cooperative policy optimization between independent learners and improves the probability of converging to Pardons-optimal Nash equilibrium.

Lastly, this paper studies the policy optimization against the non-stationary agent. Existing multiagent reinforcement learning algorithms do not explicitly classify the non-stationary opponents, but try to deal with them with one general policy. To overcome the challenge of non-stationary agents in MAS and the limitation of existing algorithms that using single response policy against the non-stationary opponent. Based on the opponent model, the rectified belied model achieves accurate opponent detection from the perspectives of reward signals and opponent behaviors. In addition, the distillation policy network(DPN) is proposed as a policy library to achieve fast policy switching, convenient policy reuse, and efficient policy storage. In summary, this study achieves accurate opponents classification and efficient strategy reusing, which shed a light on researches of playing against non-stationary opponents.

In summary, this paper takes multiagent policy optimization based on DRL algorithms as the research goal and studies it from the perspectives of the environment, DRL algorithms, and MAS. Specifically, this paper studies the problems of policy optimization with multimodal inputs, policy optimization of independent learners and policy optimization against non-stationary agents. The empirical experiments confirm the effectiveness of the proposed methods. This paper focuses on both engineering practice and plays a guiding role in applying DRL algorithms in solving practical problems. Meantimes, it sheds a light on further research on multimodal reinforcement learning, policy optimization of independent learners, finding Pareto optimal Nash equilibrium and dealing with non-stationary opponents.

KEY WORDS: Deep reinforcement learning; Multiagent system; Non-stationary agent; Bayesian policy reuse; Policy optimization; Multimodal learning.

目 录

摘 要	I
ABSTRACT	III
目 录	V
第一章 绪论	1
1.1 研究背景与问题的提出	1
1.1.1 研究背景	1
1.1.2 研究意义	5
1.2 研究内容与技术路线	7
1.3 研究创新点	9
1.4 本文的组织结构	11
第二章 研究理论基础与文献综述	13
2.1 研究理论基础	13
2.1.1 强化学习	13
2.1.2 多智能体系统	16
2.2 主要相关算法	17
2.2.1 基于值函数估计的强化学习算法	18
2.2.2 基于策略的强化学习算法	21
2.2.3 基于行动者评论家的强化学习算法	22
2.3 多模态信息输入的智能体策略优化分析	24
2.4 噪声环境下的智能体策略优化分析	26

2.5	非静态对手环境下的智能体策略优化分析	28
2.6	本章小结	30
第三章	面向多模态信息输入的智能体策略优化研究	31
3.1	引言	31
3.2	基于多模态信息输入的策略优化算法框架	33
3.3	基于多模态信息输入的网络架构	34
3.4	层次注意力机制	36
3.4.1	注意力机制	37
3.4.2	层次注意力机制	38
3.5	多模态信息融合	40
3.5.1	基于单信息流的LSTM网络	40
3.5.2	基于多信息流的LSTM网络	42
3.6	实验验证	43
3.6.1	充分信息的多模态问题	44
3.6.2	不充分信息的多模态问题	46
3.6.3	多模态自动驾驶问题	48
3.7	本章小结	51
第四章	面向噪声环境下独立学习智能体的策略优化研究	53
4.1	引言	53
4.2	基于双权估计的多智能体策略优化算法框架	54
4.3	双权深度Q网络	57
4.3.1	估值偏差的机理分析	57
4.3.2	基于双权估计器的估值纠偏	58
4.4	奖赏值网络与宽容机制	59
4.4.1	奖赏值网络	60

4.4.2	宽容的奖赏值网络	61
4.5	调度经验重放策略	64
4.5.1	优先级经验重放	65
4.5.2	调度经验重放策略	66
4.5.3	混合优先级经验重放池	68
4.6	实验论证	69
4.6.1	估值纠偏实验	70
4.6.2	合作式多智能体实验（离散动作空间）	71
4.6.3	合作式多智能体实验（连续动作空间）	73
4.6.4	基于层次任务的合作式多智能体实验	76
4.7	本章小结	78
第五章	面向非静态对手环境下的多智能体策略优化研究	81
5.1	引言	81
5.2	基于深度贝叶斯策略重用的多智能体策略优化算法框架	82
5.3	贝叶斯策略重用理论	85
5.4	基于置信模型的对手策略检测	86
5.4.1	置信模型	86
5.4.2	对手建模	87
5.4.3	修正置信模型	89
5.5	基于策略蒸馏的策略优化	89
5.5.1	策略蒸馏	90
5.5.2	蒸馏策略网络	90
5.6	实验论证	92
5.6.1	环境描述	92
5.6.2	非静态对手的策略检测	94

5.6.3 针对未知策略的有效学习	95
5.7 本章小结	97
第六章 总结与展望	99
6.1 研究内容总结	99
6.2 展望	100
参考文献	102
发表论文和参加科研情况说明	112
致 谢	115

第1章 绪论

1.1 研究背景与问题的提出

1.1.1 研究背景

近年来，深度学习（Deep Learning, DL）的相关研究工作取得了飞速的发展，一个主要的原因是由于硬件计算能力的大幅提升，为大量深度学习算法提供了充足的计算能力。深度强化学习（Deep Reinforcement Learning, DRL）作为其中受益的一个研究领域，也取得了大量突破性的研究成果。例如，AlphaGo^[1], AlphaZero^[2]等基于深度强化学习的人工智能算法，在以往人类擅长的围棋比赛中首次战胜了人类，并在与世界围棋冠军的比赛中获得了远超人类智慧的表现^[3]。又比如，中国最大的电子商城淘宝也使用深度强化学习技术来解决其电子商城中商品推荐的问题，实现了买家、卖家以及平台的三方利益最大化^[4]。此外，在计算机视觉^[5,6]、自然语言处理^[7,8]、机器人控制^[9-11]、无人驾驶^[12,13]、智能电网^[14-16]、智能交通^[17]等研究领域，深度强化学习算法都取得了一定得研究成果。然而，纵使越来越多的研究者开始尝试使用深度强化学习来解决人类社会中的复杂问题，但仍然存在一定的难题与挑战。这不仅是因为深度强化学习算法本身存在一定的局限性，还因为许多真实的人类社会问题中通常存在多个拥有自主决策能力的个体（智能体），个体之间被要求以某种特定的行为来完成特定的任务。但由于每个智能体不断地改变行为，彼此之间相互影响，这无疑加剧了问题本身的复杂度与难度。

古往今来，大多人类社会的现实问题中都存在着明确的合作和分工，生活中我们可以看到大量合作共赢的案例。从石器时代的合作捕猎到工业时代的流水线制造工厂，再到现在科技公司之间的优势互补，无一不体现了合作共赢的重要性。此外在自然界有一定智能的个体之间，比如蚁群、鱼群、鸟群，也都存在着各式各样的合作。这一定程度上也说明合作广泛存在于群体之前，并且能够有效地实现群体的利益最大化。因此，大多人类社会问题都要求能够高效的实现智能

体之间的合作，共同实现群体利益最大化。

与此同时，当下社会中存在诸多拥有巨大经济效益的科学问题急需解决，如城市公交智能调度^[18]、机场航班智能调度、智能电网的电力分配^[15,16]、水权市场的定价机制^[19]、电商系统中的商品推荐机制等^[20]。这些问题通性是都包含了由多个智能体组成的复杂群体，并在此复杂环境下进行优化求解。针对此类问题，一种行之有效的方法是将其建模成多智能体系统，并使用相应的优化算法进行求解。由此，研究在多智能体系统下，如何促使智能体协作，以实现群体利益最大化就显得尤为重要。尽管个体间的合作一直广泛存在，但是针对智能体之间合作的研究并不是非常成熟，因此，如何有效且高效地促进多智能体的协同合作一直是当前的一个热点研究问题。这也侧面印证了多智能体系统研究的重要性和意义。

深度强化学习与多智能体系统的研究受到国内外研究者的广泛重视，均属于人工智能的前沿研究。其中强化学习算法作为一种面向单智能体的策略优化算法，侧重于针对序列决策问题进行策略优化求解^[21]。而多智能体系统常用于大型、复杂的现实问题建模^[22]，侧重于关注多个智能体相互影响，并实现问题求解。因此，有效的结合深度强化学习与多智能体系统，不仅能够进行最优序列决策的求解，还能实现多智能体之间的相互配合，以获得群里利益最大化，解决复杂的现实问题。

以往的研究工作表明，强化学习能够用于解决多智能体系统问题，但在深度神经网络复兴之前，由于计算资源的限制，针对问题的建模往往比较简单，限制也比较严格。比如问题的状态表征空间一般用低维特征，否则会极大增加计算资源；又比如，针对问题的建模也相对简单，一般不考虑可能潜在的噪声或者多模态信息输入等复杂的场景。大部分的约束都是由于计算资源有限所致。这些约束一方面限制了强化学习算法本身处理真实复杂问题的能力，另一方面也阻碍了强化学习与多智能体系统研究的应用推广。现如今，随着硬件计算资源的进步以及深度学习的发展，计算资源已经不再是难以突破的约束。深度强化学习算法已经能够赋予智能体处理含有高维状态特征的能力^[3]，并根据得到的信息进行策略优化，做出最优决策。同时由于深度强化学习的通用性及其端到端的易训练性，使得深度强化学习成为了解决策略优化问题的不二选择。学术界也涌现出了越来越多的结合深度强化学习与多智能体系统的研究工作^[23,24]，这使得利用深度强化学习算法来解决基于多智能体系统建模的真实问题成为了可能。

尽管如此，在试图结合深度强化学习与多智能体系统，以实现高效的智能体策略优化的研究中，仍然存在许多难题与挑战。这些难题与挑战可能来源于诸

多方面，例如问题本身可能存在多源感知信息的挑战^[25]；智能体接收到的反馈信息中存在噪声的挑战^[26]；深度强化学习算法本身存在估值偏差的局限性的挑战^[27,28]；以及多智能体系统中如何实现独立学习智能体间高效配合的挑战^[29]；以及如何应对多智能体系统中不断变化策略的其余智能体的挑战^[30]。上述五大挑战，是使用深度强化学习算法解决多智能体策略优化研究中亟需解决挑战，后文针对这些挑战对策略优化的影响以及解决必要性进行详细论述。

图1-1描述了本文的主要研究内容，本文针对基于深度强化学习的多智能体策略优化问题展开研究，从环境层面，强化学习算层面以及多智能体层面对研究现状以及既有算法的局限性进行了详细分析，并给出了对应的解决方法。接下来针对策略优化中存在的局限性与挑战进行分析。

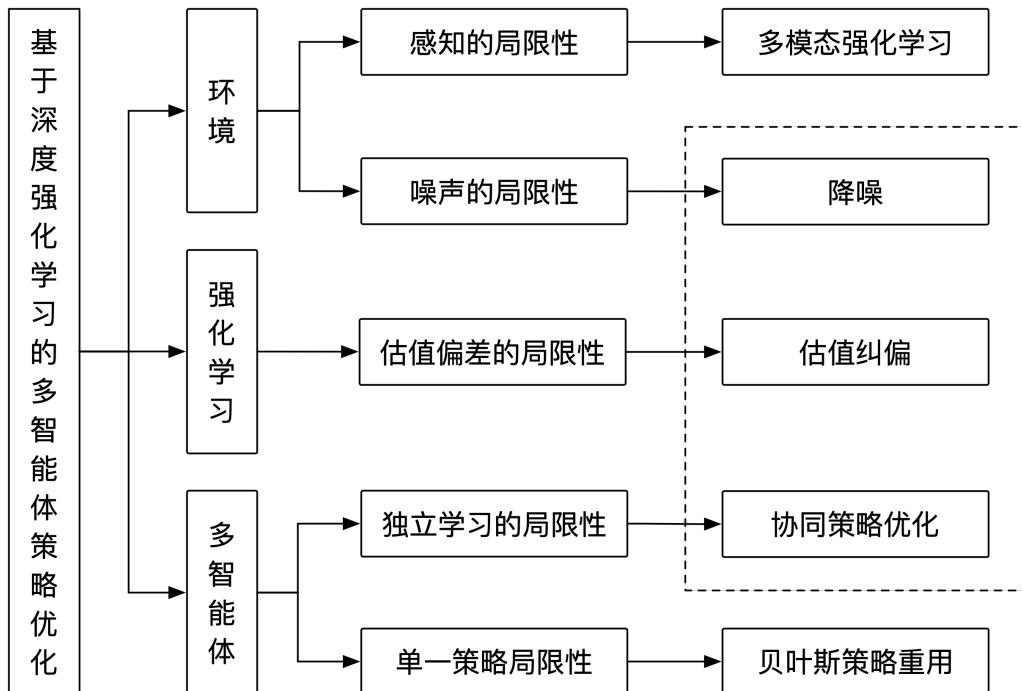


图 1-1 基于策略优化中存在的局限性分析及具体研究内容

第一个挑战来自于问题本身可能存在多源感知信息的挑战。通常来说，智能体通过观察来获取信息，并基于观察到的信息进行下一步的行为决策。这些智能体观察到的信息被称之为感知信息。感知信息是智能体在决策时依赖的必要信息，有时甚至是唯一信息。不同的感知信息直接影响智能的行为决策。形式化的，不同来源的感知信息又被称之为模态。单模态信息提供的信息有限，造成智能体在策略优化时遇到信息不足的难题，致使策略优化效率低下，乃至难以学习到最优

策略等问题。直觉上，由于不同模态信息承载的信息不同，利用多模态信息可以有效地缓解信息不足的问题，有助于智能体进行提升策略优化效率，寻找最优策略。例如，若能够有效地融合多模态感知信息，产生更完整的表征信息，并基于该表征信息，实现更智能的策略决策，使智能体拥有向人类一样处理多模态数据的能力。这将赋予智能体更高的智能，使其能够更全面的分析观测到的数据并进行决策，达到接近人类智能的能力。就深度强化学习算法的研究现况而言，基于单模态信息的研究工作尚未研究成熟，基于多模态的观测信息的研究工作更是困难重重。尽管如此，基于多模态感知信息的强化学习算法研究，不仅对深度强化学习的研究本身有着重要的意义，对于后续使用深度强化学习解决多智能体策略优化问题，也有着一定的借鉴意义。

第二个挑战来自于智能体接收到的反馈信息中存在噪声的挑战。一般来说，智能体决策后，会收到环境的反馈信息（奖赏值或收益）。智能体通过奖赏值来判别当前行为的优劣，并基于奖赏值调进行策略优化，实现长期收益最大化^[31]。这说明反馈信息是智能体进行决策的关键因素之一，对智能体策略优化有着重要的影响。现实问题中充满了各式各样的扰动和噪声，因此智能体接收到的奖赏值一般也都含有噪声。然而一般的深度强化学习算法并没有考虑噪声的存在，这增加了在带噪声的环境中进行策略优化难度。为了有效解决真实问题，需要考虑在带噪声的环境中，如何有效地进行策略优化。这对强化学习算法研究进行了补充，并且对多智能体策略优化研究奠定了基础。

第三个挑战来自于深度强化学习算法本身存在估值偏差问题的挑战。深度强化学习算法要求智能体在决策时，对所有行为可能带来的未来奖赏值收益进行估计，并选择拥有最高估值的行为执行。因此，行为估值直接决定了智能体的策略，而估值准确性直接影响了智能体的策略优化。现阶段研究表明，既有的强化学习算法对行为的估值存在着估值偏差^[27,28]，这可能直接导致策略可能朝着错误的方向进行优化，严重影响智能体的策略优化。因此，进行估值纠偏对于深度强化学习与多智能体策略优化本身有着重要意义。

第四个挑战来自于多智能体系统中如何实现独立学习智能体间高效配合的挑战。多智能体系统存在着天然的不稳定性。这是由于多个智能体共同存在，相互影响，且每个智能体的行为也是不断变化的，这也对最优策略优化有着关键的影响。这种影响产生的主要原因是因为智能体把其余智能体当作环境的一部分，这种智能体被称为独立学习智能体。因此，独立学习的智能体会把其余智能体的行为变化当做是环境中的不稳定噪声，这种不稳定性会影响独立学习智能体的策略

优化。同时，由于群体里利益最大化需要有多个智能体配合完。如何有效促进多个智能体朝着群体利益最大化的方向，通过优化各自的策略，尽可能地收敛到帕累托纳什均衡策略，实现整体收益的最大化，是多智能体策略优化研究中心急需解决的挑战。

第五个挑战来自于多智能体系统中不断变化策略的其余智能体的挑战。独立学习的智能体把其余智能体当做环境的一部分，这种做法因为完全无视对手的存在，因此存在一定的局限性。而在多智能体系统中，智能体之间需要相互配合才能实现群体利益最大化。如果智能体完全不考虑其余智能体的行为单独做出决策，智能体之间的配合就会出现困难，最终导致无法收敛到最优策略，损害群里利益最大化。因此对每一个智能体而言，根据有限的观察信息，准确的识别其他智能体的意图，并采用适合的策略与其配合，是实现群体收益的最大化有效方法。因此，如何高效而准确的判别对手行为，并使用相应的策略快速应对，最大化长期收益，是多智能体策略优化所面临的另一大挑战。

综上所述，基于深度强化学习的多智能体系统研究，对于解决实际问题有种重要的意义，同时也存在诸多尚未解决的难题与挑战。本文以策略优化为研究目标，主要针对前述的五大挑战展开研究：从深度强化学习算法与多智能体系统角度出发，首先，研究当拥有有多种模态感知信息输入时，如何有效的利用多模态信息进行智能体策略优化。其次，研究噪声环境下，如何设计拥有抗噪能力的深度强化学习算法。然后，针对深度强化学习算法本身存在的估值偏差问题进行研究，提出有效的估值纠偏方法。接着，研究在多智能体环境下，如何有效促进独立学习智能体之间高效协作，实现协同策略优化，最大化群体收益。最后，研究当面对非静态智能体时，如何使用策略重用算法来高效地应对不断变化行为的对手，实现长期利益最大化。本研究对于深度强化学习理论与多智能体系统理论结合方面具有一定的实践意义。同时，对于使用多智能体系统来建模并解决实际问题，具有重要的理论和现实指导意义。

1.1.2 研究意义

使用多智能体系统建模能够最大程度的还原真实问题，在此基础上使用深度强化学习算法进行问题求解也是行之有效的解决方案。但在智能体进行策略优化时，以往的方法存在一定的局限性。

例如，从环境层面，既有方法进行策略优化时候，并不考虑多模态感知输入

的情况，也不考虑环境返回的奖赏值中存在噪声的情况；其次，从强化算法本身考虑，既有算法不太考虑算法本身存在的估值偏差可能带来的影响；最后，从多智能体系统层面，既有算法并没有考虑如何在噪声环境下实现独立智能体之间的有效合作，也没有考虑如何通过策略重用来应当不断变化行为的其余智能体问题。

本文主要针对在使用深度强化学习算法进行多智能体策略优化中潜在的局限性展开研究，并提出相应的解决方法。主要涵盖了环境层面的局限性研究、强学习算法层面的局限性研究以及多智能体系统层面的局限性研究。本文针对策略优化过程中存在的局限性提出的解决方法，对于深度强化学习研究以及多智能体研究来说，都具有较大的理论意义与实践意义。

(1) 理论层面

首先，环境层面，本文针对多模态信息输入的研究填补了强化学习算法针对多模态输入的研究空白。其次，本文研究的在带噪声环境下的降噪研究也弥补了一般强化学习算法不考虑带噪声问题的缺点。第三，从深度强化学习算法层面，本文针对算法本身存在的估值偏差进行估值修复研究，补充并完善了既有算法针对估值偏差的研究。第四，从多智能体系统层面，本文研究了在噪声环境下高效的促进独立学习智能体实现协同策略优化。最后，本文研究了如何使用策略重用来应对不断变化行为的智能体，最大化长期收益。该工作弥补了以往工作的不足，充实了基于深度强化学习的多智能体策略优化研究，也丰富和完善了深度强化学习以及多智能体系统的相关理论。

(2) 实践层面

本文的研究可以为深度强化学习与多智能体系统中的策略优化问题提供参考价值。首先，本文针对多模态信息输入下的策略优化研究工作，为现实中需要通过融合多种输入信息来实现智能决策的问题使用提供一定的参考意义。其次，本文针对带噪声环境的研究以及针对强化学习算法估值修复的研究，弱化了以往对环境的假设，增强了抗噪能力，增加了估值准确度。为解决现实中带噪声的真实问题，提供了一定的指导意义。最后，本文针对独立学习智能体之间的协同策略优化研究，对现实中当无法观测到其余智能体信息时，如何实现智能体的策略协同优化提供了指导意义。此外，本文针对不断变化行为智能体的研究，为现实中需要准确地判断对手意图，并快速做出响应的问题，提供了一定的借鉴意义。

综上，本文的研究对基于深度强化学习与多智能体系统的策略优化问题，提供了借鉴意义和现实指导意义。

1.2 研究内容与技术路线

由于多智能体系统能够有效地对现实问题进行建模，而深度强化学习算法也常用于解决多智能体问题。既有研究工作也侧面证明使用深度强化学习算法来解决多智能体问题是目前主流的研究方向之一。因此，本文以智能体策略优化为目标，主要针对在使用深度强化学习算法进行多智能体策略优化（后文统一简称为策略优化）中潜在的局限性与缺陷进行研究，论述了分析缺陷产生的原因，可能造成的影响，并提出相应的解决方法。

具体来说，本文主要关注在智能体策略优化的过程，考虑环境层面潜在的缺陷、深度强化学习算法层面潜在的缺陷以及多智能体层面潜在的缺陷对智能体策略优化带来的影响。本文从三个层面探讨了策略优化过程中存在的五大局限性以及对应的解决方法。并分别在文中的第三章、第四章、第五章进行详细的论述。

综上，本文针对使用深度强化学习算法进行多智能体策略优化中存在的缺陷进行研究，并提出对应的解决方法。图1-2展示本文的研究技术路线图，具体来说，全文的组织结构如下：

第一章绪论提出了本文的研究背景。首先介绍了近年来深度强化学习取得的突破性研究成果，然后阐述了使用多智能体系统对当下社会中的现实问题进行建模的必要性，论述了使用深度强化学习算法进行多智能体策略优化研究中存在的问题以及挑战。强调了本文研究工作的理论以及实践意义，介绍了本文的研究内容、技术路线与研究创新点。

第二章相关工作研究，主要介绍了本文涉及到的基础理论，包括深度强化学习、多智能体系统等基础概念。此外还介绍了部分重要基础算法，包括Q-learning算法、REINFORCE算法、DDPG算法、Actor-Critic算法以及A3C等强化学习算法。最后对存在的相关工作进行了详细的论述，并分析总结了现有工作的优缺点，总结待解决的核心难点问题。

第三章基于多模态信息输入的智能体策略优化研究，主要论述了如何将多模态学习与深度强化相结合，以改进传统强化学习算法使用单模态输入可能潜在的信息不足的问题。首先，提出了分离式多模态输入的强化学习框架，拓展了一般强化学习算法处理信息输入的能力，实现了针对多种模态信息的有效处理。针对多模态强化学习中存在的注意力分配问题，其次，提出了层次注意力机制，实现了在多模态信息间以及各个模态信心内部的注意力权重分配方法，弥补了一般注

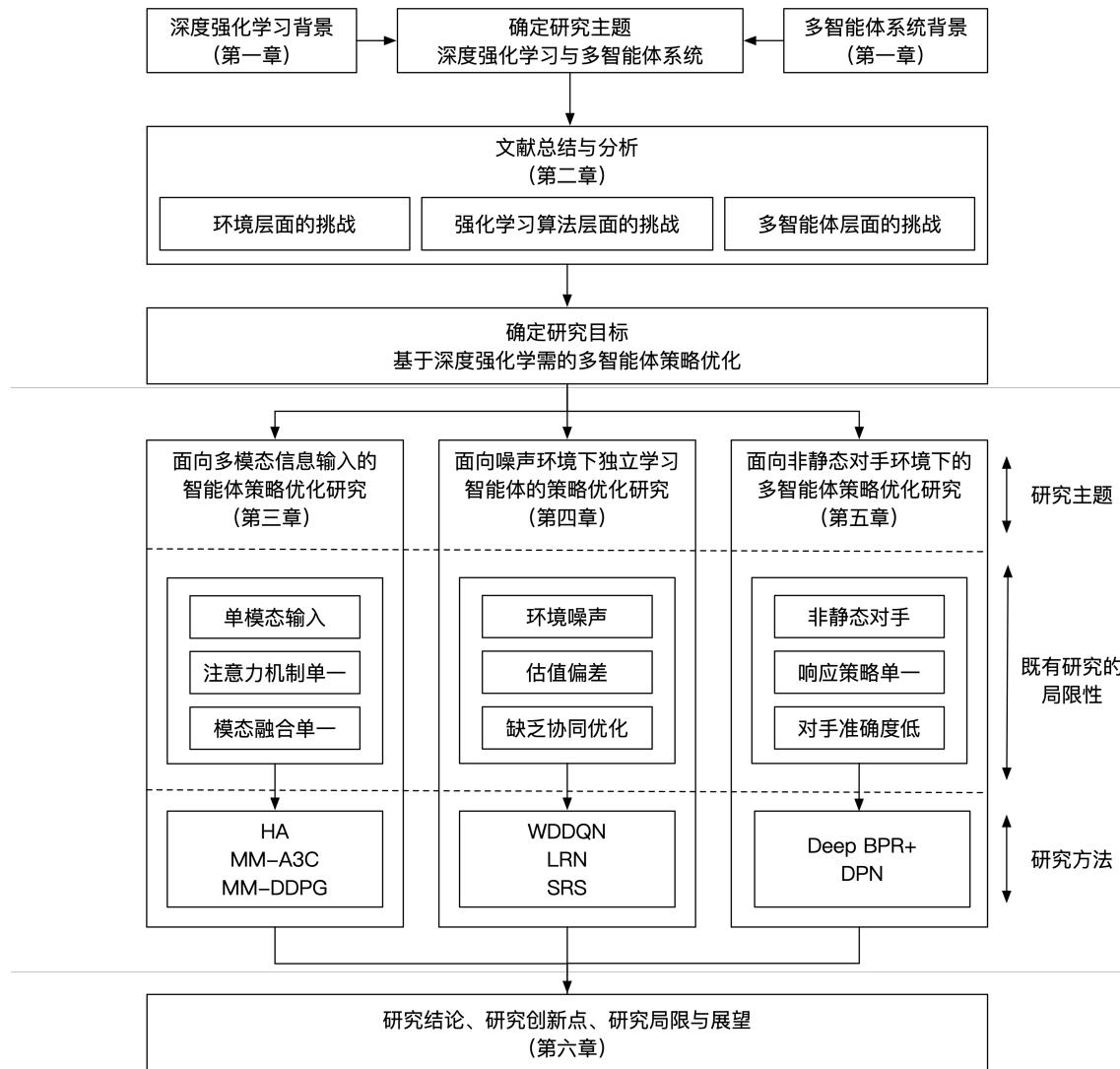


图 1-2 技术路线图

意力机制只处理单模态信息的缺陷，实现了对多模态信息输入的层次化注意力分配，有效提高策略优化效率。最后，针对多模态输入信息融合的问题，提出使用基于LSTM网络的多模态融合机制进行信息融合，有效提高了算法处理多模态信息的能力。

第四章带噪声环境下的独立学习智能体协同策略优化研究，主要论述了在带噪声的多智能体环境中，如何有效的降低环境中的噪声并促进独立学习智能体之间有效地进行策略协同优化，并提高收敛到帕累托最优纳什均衡策略的概率。本章解决了策略优化过程中存在的噪声挑战、估值偏差挑战以及独立学习智能体协同优化的挑战。首先，针对环境中噪声的挑战，提出了奖赏值网络对环境信息进

行拟合，有效解决环境中的噪声问题。其次，针对估值偏差的挑战，提出了使用基于双权Q网络的强化学习算法来降低估值偏差，有效地进行估值纠偏。再次，关于独立学习智能体之间协同策略优化的挑战，本章将宽容的思想与奖赏值网络想结合，提出宽容奖赏值网络，有效地促使独立学习的多智能体之间有效的进行协同策略优化，提升了学习到帕累托最优纳什均衡的概率，实现群体利益最大化。最后，针对策略优化的效率问题，提出了调度经验回放策略，有效地提高了多智能体策略优化的效率。

第五章面向非静态智能体的策略优化研究，主要论述了在多智能体环境下，如何应对不断变化行为智能体的挑战，以实现长期收益最大化。首先，基于贝叶斯理论提出了深度贝叶斯策略重用算法，实现了有效的对手策略变化检测以及准确的对手类型判别，快速切换并执行针对该对手的最佳响应策略。其次，为了提高对手策略检测度的准确性，提出的使用对手建模机制，用于修正贝叶斯策略重用算法中的原始策略检测机制，进一步提高了检测准确度。最后，本章提出了使用策略蒸馏算法，对多个相应策略蒸馏得到单一的策略整流网络。策略整流网络不仅能够实现快速的策略切换，有效地降低了空间使用率，还显著提升了针对未知对手进行策略优化时的优化效率。

论文的最后一章对全文的研究内容和结论进行了分析与总结，同时阐述了本研究潜在的局限性，并展望了未来可进一步深入研究的方向。

1.3 研究创新点

通过上述论述可知，本文针对深度强化学习算法在多智能体环境下的策略优化问题展开研究；分析了来自包括环境、强化学习算法以及多智能体三个层面的挑以及面临这些挑战时既有算法中存在的缺陷，论述了缺陷对策略优化的影响；并提出了行之有效的解决方法。具体来说，本文的主要创新点包括以下几个部分：

(1) 基于多模态信息输入的强化学习框架

本研究主要从环境层面出发，论述了当决策所需的感知信息有限时，对智能体策略优化的影响。进而研究了如何有效地利用多模态感知信息时进行策略优化。由于目前基于多模态的强化学习的研究相对较少，而很多实际问题又需要使用使用多模态的学习算法来解决的，例如自动驾驶问题，驾驶员需要结合感知到视觉

信息、声音信息、传感器信息等信息协同完成安全的行车驾驶，导致多模态的强化学习算法的研究具有十分迫切的实际需求以及研究意义。本研究提出了一种基于多模态信息输入的强化学习算法框架，不同于一般强化学习算法只使用单模态信息输入作为智能体决策的唯一依据，本算法通过融合多模态的感知数据，有效地进行策略优化，实现长期受益最大化。此外，本文提出了层次注意力机制实现了注意力机制的层次化分配，有效提高策略优化效率。最后，本文提出使用基于LSTM网络的多模态融合机制针对多模态输入信息进行有效融合，提高了算法处理多模态信息的能力。本研究工作一定程度上弥补了在强化学习研究领域关于多模态工作的缺陷，同时本研究成果可直接拓展到多智能体环境下，作为前序研究，为后续多智能策略优化奠定了基础。

(2) 基于双权深度Q网络的强化学习算法

本研究考虑了多种策略优化过程中的挑战，涵盖了环境中噪声的挑战、强化学习算法中估值偏差的挑战以及多智能体之间协同策略优化的挑战。本研究的创新点包括四个方面。首先，由于一般的强化学习研究重点都是关注如何提高算法效率，因而不会假设环境中存在噪声。而本文解决带噪声环境下的策略优化问题，并提出了奖赏值网络来降低环境中的噪声，该方法提高了一般强化学习算法面对噪声环境的鲁棒性。其次，一般强化学习算法中存在估值偏差的问题，本文提出使用深度双权估计器来降低估值中的偏差，有效地进行估值纠偏，促进了策略优化效率。此外，针对多智能体环境下，独立学习智能体难以进行协同策略优化的问题，本研究提出了使用宽容奖赏值网络，促进独立学习智能体之间的协同策略优化，增加了学习到帕累托最优纳什均衡的概率。最后，本研究设计了调度经验回访策略，相比传统优先级经验回放策略，有效地提升独立学习智能体之间的策略协同优化效率。综上，本文考虑多智能体环境下，提出了有效促进独立智能体之间进行高效的策略协同优化的方法。

(3) 基于深度贝叶斯策略重用机制的强化学习算法

本研究考虑了在多智能体环境下，由于智能体不断变化行为而带来的挑战。一般在合作式多智能体问题中，智能体之间需要相互配合实现群体收益最大化。可当环境中部分智能体不停切换其行为时，实现智能体间相互配合的难度会极具增加。针对此问题，既有的研究工作分为两大类：一种是不对智能体类型进行显示分类，而是学习单一通用策略来应对不断变化行为的智能体；另一种做法是显示的对智能体进行分类，并执行相应的应对策略，实现智能体合作。本文提出了一种基于贝叶斯策略重用的多智能体强化学习算法，以有效的应对不断变化策略

的智能体。该算法属于显示分类的范畴，相比不显示分类，能够快速地响应其余智能体的行为变化，并使用合适的策略进行响应，以实现长期效益最大化。而相比既有的显示分类算法，本算法实现了更高的检测准确度以及更好的算法表现。此外，考虑到显示对手分类过程中，关于对智能体分类类型的先验知识是有限的，需要通过在线学习来应对新的对手类型。而在线学习一般来说效率低，需要耗费大量的学习时间。为了解决这一难题，本文提出了一种基于策略蒸馏的算法，有效地蒸馏出已经学习到的知识，用于在线学习，显著地提升了在线学习的速度以及策略优化效率。

1.4 本文的组织结构

本文围绕基于深度强化学习的多智能体策略优化研究展开讨论，具体的组织结构如下：

第一章介绍了本文的研究背景与研究意义，研究内容与技术路线，本文的创新点和全文的组织结构。

第二章主要介绍了本文涉及到的基础理论与概念，部分重要算法，并总结了相关工作。最后针对文本拟展开的研究问题进行了详细的论述，总结待解决的核心难点问题。

第三章主要对面向多模态输入的智能体策略优化问题展开研究，提出了基于分离式多模态输入的强化学习框架，有效地实现了智能体的策略优化。

第四章主要对面向带噪声环境下的独立学习智能体策略优化问题展开研究，提出了基于双权估计器的WDDQN算法，有效地实现了估值纠偏，促进了多智能体间策略协同优化，以及帕累托最优纳什均衡策略的求解。

第五章主要对面向非静态对手环境下的多智能体策略优化问题展开研究，提出了基于贝叶斯策略重用的deep BPR+算法，实现了准确的对手类型检测以及高效的策略重用，能够有效应对多智能体环境下的非静态对手。

第六章对本文的研究内容进行总结，并展望了可以深入研究的方向。

第2章 研究理论基础与文献综述

2.1 研究理论基础

2.1.1 强化学习

强化学习(Reinforcement Learning, RL)是机器学习的一个重要分支，其核心思想起源于行为学，主要借鉴了仿生学中的“尝试与试错”的机制，强调在与环境的不断交互过程中学习，通过交互后的获得的反馈信号进行策略的修正与优化^[32]。由于强化学习是具有很强的普适性，在诸多领域都有着广泛的运用，例如博弈论、控制理论、运筹学、信息论、多智能体系统、遗传算法等研究领域都能看到与强化学习结合的成功案例。

(1) 强化学习的组成要素

强化学习一般包括如下四个要素：

- 智能体 (Agent): 拥有智能的行为主体，一般来说可以被智能体类比成为人，通过优化自身行为实现长期利益最大化。
- 状态 (State): 是对环境的在某一特定时刻的描述，一般来说是以高维向量的形式，同时也是智能体决策时的考虑的重要依据。
- 动作 (Action): 是智能体的根据观察到的状态所作出的行为反应，一般来说智能体的动作会返回用于环境并改变环境状态。
- 奖赏值 (Reward): 当智能体作出的行为作用于环境后，环境给智能体返回的信号，一般是实数的形式，用于衡量智能体行为的短期收益。

在强化学习中，智能体通过观察到的环境状态，作出决策动作，收到环境反馈的奖赏值，并基于收到的奖赏值，进行策略优化。与其他机器学习方法的一个关键不同在于强化学习强调解决序列决策优化问题，意味着强化学习的目标是使得智能体获得累计收益（长期收益）最大化。

强化学习是一个通过试错来修正并优化策略的算法框架，其形式化的定义如图2-1所示^[21]：智能体Agent在 t 时刻观察到环境的状态表征 s_t 后给出当前时刻的

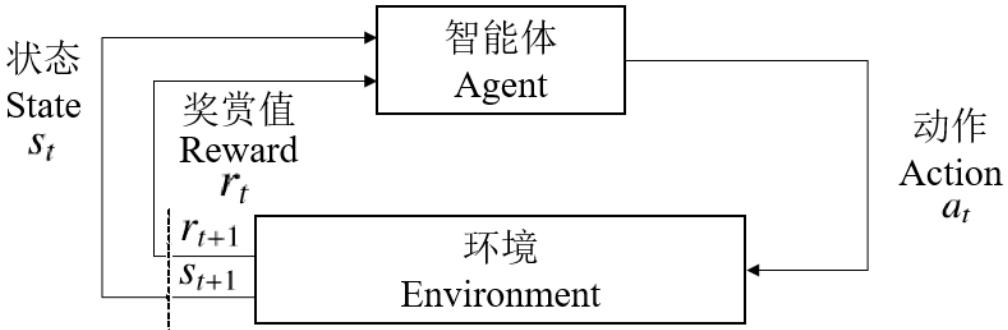


图 2-1 强化学习算法框架图

行为策略 a_t ; 环境受到智能体的决策 a_t 的影响后, 演化到新的状态 s_{t+1} , 并返回给Agent一个反馈信号 r_t , Agent接受环境返回的 $t + 1$ 时刻的状态表征 s_{t+1} 与奖赏信号 r_t , 并在接下来的 $t + 1$ 时刻继续作出决策, 如此循环反复直到环境给出终止的信号, 表示任务完成。此外, 强化学习策略优化的优化目标如下公式:

$$\text{maximize} \sum_t r_t \quad (2-1)$$

综上, 强化学的目标在于针对序列解决问题进行最优策略求解, 其更侧重于最大化长期收益, 而不在意一两次决策是否会带来较低的收益。

(2) 马尔科夫决策过程

马尔科夫决策过程 (Markov Decision Process, MDP) 主要基于马尔科夫过程与动态规划理论^[33], 是序列决策问题研究的基础。单智能体强化学习主要就是使用的MDP作为基础研究理论进行智能体策略优化的研究。马尔科夫决策过程继承了马尔科夫性质, 意味着系统下一时刻的状态值只依赖系统当前时刻的状态, 而和之前的历史状态并无关系。一般来说, 马尔科夫决策过程由以下五元组组成:

- S : 环境所有可能状态的集合, 又称为状态空间。
- A : 智能体所有可能动作的集合, 又称为动作空间。
- $R: S \times A \rightarrow \mathbb{R}$: 环境中的奖赏值函数。
- $T: S \times A \times S \rightarrow [0, 1]$: 环境的状态转移概率函数。
- γ : 折扣因子

马尔科夫决策过程中的 S 、 A 分别表示了状态集合、行为集合。此外, R 是奖赏值函数, 用于计算智能体在状态 s 时, 选择动作 a 后, 环境返回的奖赏值 $R(s, a)$ 。 T 是环境的状态转移概率函数, 代表了智能体在状态 s 时, 选择动作 a 后, 环境切换

到 s' 的概率 $P(s, a, s')$ 。折扣因子 γ 是用于计算智能体获得的长期收益，例如智能体从 t 时刻开始的长期收益 R_t 计算公式如下：

$$R_t = \sum_{i=0}^{\infty} \gamma^i r_{t+i} = r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \dots \quad (2-2)$$

强化学习算法旨在对智能体策略进行优化以最大化 R_0 。

智能体的策略（Policy）是一个概率分布，一般用 $\pi(s, a) \rightarrow [0, 1]$ 表示，代表了智能体在状态 s 时，执行动作 a 的概率。强化学习的目标是要最大化 R_0 ，但当智能体使用不同的策略 π 时，会产生不同的奖赏值序列（由于不同策略会产生不同的状态转移结果）。

状态值函数 $V_\pi(s)$ 表示在 s 状态下，智能体执行策略 π 而产生的长期奖赏值，具体计算公式如下：

$$\begin{aligned} V_\pi(s) &= \mathbb{E}_\pi \left[\sum_{i=0}^{\infty} \gamma^i r_{t+1+i} | S_t = s \right] \\ &= \mathbb{E}_\pi [r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \dots | S_t = s] \\ &= \mathbb{E}_\pi [r_{t+1} + \gamma V_\pi(s_{t+1}) | S_t = s] \end{aligned} \quad (2-3)$$

上述公式中的 $V_\pi(s)$ 表示在状态 s 下，使用策略 π 所获得长期收益的期望。因此，强化学习的目标在于通过策略优化算法求解最优策略 π^* ，求解公式如下：

$$\pi^* = \arg \max_{\pi} V_\pi(s) \quad (2-4)$$

行动值函数 $Q_\pi(s, a)$ 也可用于衡量智能体执行策略 π 而产生的长期奖赏值，是和状态值函数等价的一种度量方式，定义如下：

$$\begin{aligned} Q_\pi(s, a) &= r_{t+1} + \gamma \mathbb{E}_\pi \left[\sum_{i=0}^{\infty} \gamma^i r_{t+1+i} | S_t = s, A_t = a \right] \\ &= r_{t+1} + \gamma \mathbb{E}_\pi [r_{t+2} + \gamma r_{t+3} + \dots | S_t = s, A_t = a] \\ &= r_{t+1} + \gamma Q_\pi(s_{t+1}, a_{t+1}) \end{aligned} \quad (2-5)$$

$Q_\pi(s, a)$ 表示的是在状态 s 下，智能体通过策略 π 选了并执行动作 a 后获得的长期收益。此时，最优策略 π^* ，求解公式如下：

$$\pi^* = \arg \max_{\pi} Q_\pi(s, a) \quad (2-6)$$

无论使用状态值函数或者行动之函数求解出来的最优策略 π^* 是等同的。在任意给定的状态 s 下，智能体通过执行最优策略 π^* ，都能够实现长期收益的最大化。

2.1.2 多智能体系统

多智能体系统（Multi-agent System, MAS）一般由多个拥有智能的个体组成^[34]，智能体之间通过相互配合或者竞争，在群体层面呈现出有规律的协同运动或者竞争行为^[22,29,35]。每个智能体都有其各自的行为，智能体之间既有配合也有竞争关系，因此多智能体系统有着天然的复杂性。在一个多智能体系统中，单个智能体往往不具备独立解决问题的能力，需要多个智能体之间相互配合协同完成任务。此外，如果智能体数量较多，会造成算法的求解空间过于庞大，导致最优策略无法求解。

(1) 马尔科夫博弈游戏

本文着重研究的是马尔科夫游戏 (Markov Game)，这是一种常见的多智能体系统，通常可用于多个智能体相互交互的场景，其形式化定义如下：

一个智能体系统通常由一个六元组 $\langle N, S, \mathbf{A}, Tr, R_1, \dots R_N, \gamma \rangle$ 组成：

- N : 智能体的个数；
- S : 状态空间；
- $\mathbf{A} = A_1 \times \dots \times A_N$: 所有智能体组成的策略空间, A_i 表示第*i*个智能体的策略；
- $T : S \times A \times S \rightarrow [0, 1]$: 代表环境中的转移函数；
- R_i : 第*i*个智能体的奖赏值函数；
- $\gamma \in [0, 1]$: 代表折扣因子

其中 $T(s, a, s')$ 表示在状态 s 下，所有智能体采取决策 \mathbf{A} 后，环境转换到 s' 的概率。一般来说，状态 S 是所有智能体都可观测到的，每一个智能体*i*拥有自己的策略 $\pi_i : S \times A_i \rightarrow [0, 1]$ ，描述了智能体选择不同行为的概率。

一个完全合作式的马尔科夫博弈游戏中，所有的智能体拥有相同的奖赏值函数。也就是说任何一个时候，所有智能体收到的奖赏值是相同的。如此，所有智能体在优化各自的策略时，都有一个相同的目标，即使得群体利益最大化。

(2) 独立学习智能体

Claus^[36]从智能体的角度将多智能体系统划分为了独立学习智能体 (Independent Learners) 与联合动作智能体 (Joint Action Learners) 两种类型。两者的主要区别在于智能体进行策略优化时，智能体观察的信息不同。具体来说，独立学习智能体无法观测到其余智能体的行为与获得的奖赏值，在策略优化时，把其余智能体当做是环境中的一部分；而联合动作学习智能体会单独考虑其余所有智能体的行为与奖赏值，并在策略优化过程中使用其余智能体的行为与奖

赏值。

在与环境的交互中，联合动作智能体要求其余智能体的行为以及获得的奖赏值是可观测的。然而，很多现实问题中，智能体无法观测到其余智能体的行为与奖赏值^[37]。此时，智能体仅能使用自身感知到的环境的反馈信息进行有效地策略优化。虽然独立学习智能体无法有效地观察其余智能体的信息，可能一定程度地降低学习效率，但是独立学习智能体的设定更具有一般性，适用于解决大部分问题。除此之外，基于独立学习智能体的强化学习算法实现起来比较简单，从算法的角度看，不会因为对手的变化而引入过多的复杂度。在合作式马尔科夫博弈游戏中，独立学习智能体一般能够有效地进行策略优化，实现群体利益最大化。综上，后文主要考虑基于独立学习智能体的强化学习算法进行研究。

2.2 主要相关算法

强化学习算法广义上可分为基于模型（Model-Based）的强化学习算法^[38–40]以及无模型（Model-Free）的强化学习算法^[41–43]。前者显示地对环境中的状态转移概率函数 T 进行学习，而后者不对 T 进行学习，而是直接通过与环境的迭代式交互实现策略优化。Model-Based的强化学习算法由于需要对环境中的状态转移概率函数进行拟合，因此需要一定的计算资源，并且当面对不同的环境都需要重新学习状态转移概率函数。而Model-Free的算法并没有这方面的约束，实现较为简单，并且拥有较强的泛化性，因此本文的研究主要都是基于Model-Free的强化学习算法。

Model-Free的强化学习算法大致上可分为三类：基于值函数估计（Value-Based）的强化学习算法，基于策略（Policy-Based）的强化学习算法、以及基于行动者评论家（Actor-Critic）的强化学习算法。

（1）经典强化学习

基于Value-Based的强化学习算法中，比较典型的代表是Q学习算法^[44,45]，双Q学习算法^[46]， $TD(\lambda)$ 算法^[31]等。此类算法使用值函数对 $Q(s, a)$ 值进行估计，通过公式2-6求解最优策略 π^* 。基于策略（Policy-Based）的强化学习算法中比较有代表性的是REINFORCE^[47]算法，确定性策略梯度下降（Deterministic Policy Gradient, DPG）^[48]算法等，此类算法的特点在于其不对动作值函数 $Q(s, a)$ 进行估计，而是以优化期望值回报为目标，直接对智能体策略 π 进行优化。使用Policy-Based强化学习算法的智能体，由于直接对其策略进行优化，因此可以解决连续动作空

间的问题，这是基于值估计的强化学习所不具备的优势。最后一类是基于行动者评论家（Actor-Critic）的强化学习算法，此类算法结合了Value-Based与Policy-Based强化学习算法的思想进行智能体策略优化。具体来说，此类算法中使用行动者（Actor）和评论家（Critic）的两个角色，Critic针对具体的智能体行动值 $Q(s, a)$ 进行估值，而Actor则使用该值进行其策略优化。其中经典的代表算法是AC算法^[49]等，此类算法一般来说拥有较高的训练稳定性，并且能够取得较高的性能表现，是目前强化学习领域的热门研究方向。针对这三类强化学习方法，本文的研究都有所设计，后文主要针对这三种类型，选择本文主题相关的基础强化学习算法进行详细介绍与分析。

（2）深度强化学习

深度强化学习作为一种崭新的机器学习算法，借助深度神经网络的优势，增强了传统强化学习算法对状态表征空间的感知能力。深度强化学习算法的缘起可追溯到2013年，Minh等^[50]第一次将深度神经网络与强化学习算法结合来训练智能体像人一样玩Atari游戏，并取得了初步的成功。接着在2015年，Minh等^[3]提出了DQN算法，通过借助经验回放机制与目标网络机制，有效缓解了神经网络训练过程中的不稳定问题，并取得了突破性的进展。由此拉开了深度强化学习兴起的序幕，标志着深度强化学习进入了一个崭新的阶段，也涌现了大量深度强化学习的算法。

其中Value-Based的深度强化学习算法有深度Q网络（Deep Q-Network, DQN）算法^[3]、深度双Q网络（Double Deep Q-Network, DDQN）算法^[28]。Policy-Based的深度强化学习算法中具有代表性的则是深度确定性策略优化（Deep Deterministic Policy Gradient, DDPG）算法^[51]。基于Actor-Critic的深度强化学习算法则是异步行动者评论家（Asynchronous Advantage Actor Critic, A3C）算法^[52]等。后文主要针对这三种类型，选取和本文主题相关的深度强化学习算法进行详细介绍与分析。

2.2.1 基于值函数估计的强化学习算法

（1）Q学习算法

Q学习（Q-learning）算法使用 $Q(s, a)$ 值的大小来表示智能体在状态 s 时，选择动作 a 后获得的长期受益期望，是一个经典的基于值函数估计的强化学习算法，也是目前最广泛使用的无模型的强化学习算法。

算法 1: Q-learning 算法

```

1 初始化 $Q(s, a)$ 值函数估计器。
2 // 智能体与环境迭代交互，直到环境终止。
3 for  $t = 0 \dots \text{do}$ 
4   根据 $Q(s_t, a)$ 的值，选择拥有最大Q值的动作 $a_t^*$ 。
5   使用 $\epsilon$ -greedy方法在随机动作与 $a^*$ 动作之间选择，执行后获取环境返回
     的奖赏值 $r_t$ 以及下一个状态 $s_{t+1}$ 。
6   更新值函数： $Q(s_t, a_t) = (1 - \alpha) Q(s_t, a_t) + \alpha \left( r_t + \gamma \max_a Q(s_{t+1}, a) \right)$ 

```

Q-learning 算法通过不断的与环境交互，并使用每一次交互环境返回的 r_t 来更新值函数 $Q(s, a)$ ，不断循环跌倒到收敛后， $Q(s, a)$ 表示的就是最优策略，即当给定状态 s 时，含有最大 Q 值的动作 $a^* = \arg \max_a Q(s, a)$ 就是最优动作。因此，智能体能够使用学习到的值函数，进行最有动作的选择。需要注意的是，对于状态与动作空间较小的问题，一般采用表格化的存储方式值函数 $Q(s, a)$ ，例如将所有的状态 s 对应的 $Q(s, a)$ 存储下来。

(2) 双Q学习算法

双Q学习算法（Double Q-learning）是基于Q-learning的一种改进算法，其核心思想是使用两个值函数估计器进行Q值的学习，这么做的好处是能够使得学习过程更加稳定，取得更高的算法性能，算法2展示了具体流程。

算法2中第4行展示了Double Q-learning中使用了两个值函数估计器，在每一轮与环境交互的时候，交替选择其中的一个值函数估计器用于选择动作并选择，而另一个则用于动作值估计。具体来说，如算法2中第5行所示，随机选择两个估计器中的一个用于动作选择（第6、13行）。此外，在进行值函数估计的时候，使用另一个估计器进行值函数的估计（第11、18行）。既有的研究成果证明，双Double Q-learning方法取得了更高的稳定性以及更好的算法表现（更高的长期收益）。

(3) 深度Q网络

深度Q网络（Deep Q-Network, DQN）算法^[3]是一种基于Q-learning的深度强化学习算法，其使用深度神经网络作为值函数估计器的具体实现，以实现端到端的训练。主要动机是因为，状态空间的大小是强化学习要考虑的一个重要因素，而传统基于表格存储的强化学习算法无法有效地解决状态空间较大甚至无穷的复杂问题。此外，与Q-learning方法相同，DQN基于时间差分的动态规划思想^[31]，通过

算法 2: Double Q-learning 算法

```

1 初始化  $Q^U(s, a)$  与  $Q^V(s, a)$  两个值函数估计器。
2 // 智能体与环境迭代交互，直到环境终止
3 for  $t = 0 \dots \text{do}$ 
4   // 随机选择  $Q^U$  与  $Q^V$  中的一个值函数估计器，来选择动作执行。
5   if 如果使用  $Q^U$  then
6      $a_t = \arg \max_a Q^U(s_t, a)$ 
7     执行  $a_t$ , 观察奖赏值  $r_t$  以及最新状态值  $s_{t+1}$ 
8     // 使用  $Q^U$  选择  $s_{t+1}$  时刻的最优动作  $a_{t+1}^*$ 
9      $a_{t+1}^* = \arg \max_a Q^U(s_{t+1}, a)$ 
10    // 使用  $a_{t+1}^*$  更新值函数
11     $Q^U(s_t, a_t) = (1 - \alpha)Q^U(s_t, a_t) + \alpha(r_t + \gamma Q^V(s_{t+1}, a_{t+1}^*))$ 
12  else
13     $a_t = \arg \max_a Q^V(s_t, a)$ 
14    执行  $a_t$ , 观察奖赏值  $r_t$  以及最新状态值  $s_{t+1}$ 
15    // 使用  $Q^V$  选择  $s_{t+1}$  时刻的最优动作  $a_{t+1}^*$ 
16     $a_{t+1}^* = \arg \max_a Q^V(s_{t+1}, a)$ 
17    // 使用  $a_{t+1}^*$  更新值函数
18     $Q^V(s_t, a_t) = (1 - \alpha)Q^V(s_t, a_t) + \alpha(r_t + \gamma Q^U(s_{t+1}, a_{t+1}^*))$ 

```

与环境的迭代循环交互来更新Q值，进而收敛到全局最优策略。 Q 值的更新方式如公式2-7所示。

$$Q(s, a; \theta) = (1 - \alpha)Q(s, a; \theta) + \alpha[r + \gamma \max_{a'} Q(s', a'; \theta)] \quad (2-7)$$

上式中，DQN算法采用参数化的神经网络 $Q(s, a; \theta)$ 来表示Q值函数估计器，其中 θ 为神经网络的参数。智能体通过从环境中获得的激励信号 r 来更新 Q 网络， α 表示网络更新过程中的线性组合权重。

DQN算法的成功还取决于另外两个机制设计：经验回放（experience replay）与目标网络（target network）^[3]。经验回放机制打破了采样样本之间的高度依赖性，使得用于训练的样本尽可能的保留了独立同分布的特性，并且也极大的提高了样本的训练效率。目标网络的使用，缓解了神经网络在训练过程中的不稳定性，

极大的提高了算法的收敛效率与鲁棒性。这两种机制相互配合使得DQN算法在大部分Atari游戏上都取得了接近人类，甚至超越人类的表现^[3,50]。

(4) 深度双Q网络算法

深度双Q网络算法（Double Deep Q-Network, DDQN）是基于Double Q-learning的思想，使用神经网络作为Q值估计器的深度强化学习算法。大致上DDQN对Double Q-learning的拓展思路和DQN对Q-learning的拓展相似，算法流程基本和算法2相似。但需要注意算法2中的第10、18行，DDQN算法在更新Q网络的时候，更新公式如下：

$$Q(s_t, a_t; \theta) = (1 - \alpha)Q(s_t, a_t; \theta) + \alpha Q(s_{t+1}, \arg \max_a Q(s_{t+1}, a; \theta); \theta') \quad (2-8)$$

上述公式中的两个网络 $Q(s, a; \theta)$ 与 $Q(s, a; \theta')$ ，分别对应原始Double Q-learning算法中的两个值估计器。类似的，DDQN也取得了比DQN更好的稳定性以及更好的算法性能。

2.2.2 基于策略的强化学习算法

区别与Value-Based强化学习算法直接保存Q值的做法，Policy-Based强化学习算法一般直接保存智能体策略 $\pi(s, a) \in [0, 1]$ 。 $\pi(s, a)$ 表示在给定状态 s 后，智能体选择动作 a 的概率。另一个区别是基于策略的强化学习算法可用于解决决策空间是连续的问题，可应用于控制问题，比如机械臂控制，无人车驾驶等问题。

(1) REINFORCE算法

REINFORCE算法又被称为蒙特卡洛策略梯度强化算法^[21]，是一种Policy-Based强化学习算法。一般来说REINFORCE算法会将智能体策略 π 用 θ 参数化，表示为 π_θ 。如公式2-3所示，当给定策略 π_θ 时，强化学习的目标是最大化 $V_{\pi_\theta}(s)$ 。因此，REINFORCE算法的目标是通过调整参数 θ 实现智能体长期收益最大化，具体如下公式所示：

$$\text{maximize } J(\theta) = V_{\pi_\theta}(s_0) = \sum_s d_{\pi_\theta}(s) \sum_a \pi_\theta(s, a) R(s, a) \quad (2-9)$$

上述公式中， $J(\theta)$ 是REINFORCE算法的目标函数， $d_{\pi_\theta}(s)$ 概率分布函数，描述了的那当使用策略 π_θ 时，状态 s 出现的的概率， $R(s, a)$ 表示的是环境返回的奖赏值函数。REINFORCE算法通过梯度上升（Gradient Ascent）算法，调整参数 θ 实现目标函数 $J(\theta)$ 的最大化。Gradient Ascent算法的核心思想是在更新参数 θ 的时候沿着梯度

的方向更新参数，实现目标函数上升的目的。 θ 的具体更新公式如下：

$$\theta_{t+1} \leftarrow \theta_t + \alpha \nabla_\theta J(\theta) \quad (2-10)$$

公式中 $\nabla_\theta J(\theta)$ 表示目标函数 $J(\theta)$ 在 θ 处的梯度，计算公式如下：

$$\begin{aligned} \nabla_\theta J(\theta) &= \nabla_\theta \left[\sum_s d_{\pi_\theta}(s) \sum_a \pi_\theta(s, a) R(s, a) \right] \\ &= \sum_s d_{\pi_\theta}(s) \sum_a \pi_\theta(s, a) \nabla_\theta \log \pi_\theta(s, a) R(s, a) \\ &= \mathbb{E}[\nabla_\theta \log \pi_\theta(s, a) R(s, a)] \end{aligned} \quad (2-11)$$

上述过程描述了REINFORCE算法策略优化的具体细节。目标函数 $J(\theta)$ 的梯度由 $\nabla_\theta \log \pi_\theta(s, a)$ 和 $R(s, a)$ 两部分的乘积组成。其中 $\nabla_\theta \log \pi_\theta(s, a)$ 代表了策略 π_θ 关于 θ 的梯度，朝着该方向更新策略 π 会增加策略选择动作 a 的概率； $R(s, a)$ 代表了动作的奖赏值。二者的乘积用于智能体策略 π_θ 更新，能使更新后的策略更新倾向于选择能够带来更高收益的行为，而避免选择无法带来更高收益的行为。

算法3展示了REINFORCE的详细流程。REINFORCE使用环境返回的 r_{t+1} 以及折扣因子 λ 计算在状态 s_t 下行为 a_t 带来的长期收益 R （算法第5行）。接着联合与梯度策略 $\nabla_\theta \log \pi_\theta(s_t, a_t)$ 一起更新策略 π_θ （算法第6行）。

算法 3: REINFORCE 算法

- 1 初始化策略 π_θ 。
 - 2 使用 π_θ 与环境交互，直到回合终止，得到轨迹 $(s_0, a_0, r_1, \dots, s_{T-1}, a_{T-1}, r_T)$
 - 3 $R = 0$
 - 4 **for** $t = T - 1 \dots 0$ **do**
 - 5 $R = r_{t+1} + \lambda R$
 - 6 $\theta \leftarrow \nabla_\theta \log \pi_\theta(s_t, a_t) R$
-

REINFORCE算法需要在每个回合结束的时候才能开始计算用于策略更新的 R ，针对一些奖赏值十分稀疏的强化学习问题，Policy-Based强化学习算法有时候学习效果并不好。为解决此类难题，研究者们提出了Actor-Critic强化学习算法。

2.2.3 基于行动者评论家的强化学习算法

以REINFORCE为代表的基础Policy-Based强化学习算法用于策略更新的奖赏值只有等待回合结束才能获得，这导致策略更新得等到回合结束，属于十分低效

的做法。此外，由于使用整个回合轨迹进行策略更新会带来较大的更新方差，使得策略优化的过程产生震荡，致使优化效率低下，收敛到局部解甚至完全不收敛。为了解决上述切点，研究者们提出了Actor-Critic强化学习算法。

Actor-Critic强化学习算法同时学习策略和状态值函数，其中Actor和Policy-Based强化学习算法类似，一般直接使用参数化的策略 π_θ 与环境交互，并使用Gradient Ascent算法来更新策略参数 θ 。同时，Critic使用值函数估计器来估计长期奖赏。值函数的更新与前文介绍的Value-Based强化学习相似。Critic的估值可用于Actor的策略更新。这样做的一个好处是能够减少方差，缓解学习过程中网络训练的不稳定的问题^[31]。

Actor-Critic强化学习算法与Policy-Based强化学习算法大致流程是一致的，区别在于用于策略更新的奖赏值不同。如公式2-11所示，Policy-Based强化学习算法用语策略更新的长期奖赏值 $R(s, a)$ 需要等到回合结束时才能计算（算法3第5行）。而在Actor-Critic强化学习算法使用了额外的值函数估计器用于估计长期奖赏值，使得智能体能够在每一步进行策略更新，而不用等到回合结束。实验证明，Actor-Critic强化学习算法不仅拥有更高的稳定性，也取得了更高的算法性能。

(1) 异步优势行动者评论家算法

异步优势行动者评论家（Asynchronous Advantage Actor Critic, A3C）算法由Mnih等^[52]提出，是一种改进的Actor-Critic强化学习算法。其最大的优点是A3C算法使用异步多线程进行策略优化，极大地提升了策略优化的效率。试验表明A3C算法在离散及连续决策空间的问题上都取得了惊人的效果。

Schulman等^[53]提出的广义优势估计器（Generalized Advantage Estimation, GAE），主要对A3C算法过程中值函数部分进行了改进。GAE通过综合多步时差分的误差来改进值函数更新过程，提高了值函数估计的准确性，也提高了策略优化的效率，提升了策略效果。

(2) 基于GPU的异步优势行动者评论家算法

受到A3C算法成功归借助分布式计算带来的算法效率提升的启发，Babaeizadeh等^[54]提出了基于GPU的异步优势行动者评论家（GPU-Based Asynchronous Advantage Actor-Critic, GA3C）算法。GA3C算法是一种混合CPU和GPU的强化学习算法，通过消息队列的形式极大地提高了训练过程中GPU的使用率，提高了策略优化的效率。

(3) 确定性策略梯度下降算法

Silver等^[48]提出了确定性策略网络（Deterministic Policy Gradient, DPG）算

法。一般Policy-Based强化学习算法中策略 π 的输出是不同动作的概率，而DPG算法证明了策略 π 的输出可以是连续空间上具体的一个值（比如实数域上的一个值）。DPG的提出成功解决了决策空间是连续空间的问题，能够成功用语解决例如自动驾驶，机器人控制等拥有复杂状态空间的真实问题。

(4) 深度确定性策略梯度下降算法

深度确定性策略梯度下降（Deep Deterministic Policy Gradient, DDPG）算法是一种深度强化学习算法，由Lillicrap等^[51]提出。通过结合神经网络与DPG算法，DDPG能够处理如MuJoCo^[55]一类的复杂控制类问题。Hausknecht等^[56]在DDPG基础上考虑了参数化的决策空间，并在RoboCup Soccer的环境上取得了较好的实验效果。

2.3 多模态信息输入的智能体策略优化分析

模态一般指的是人接受信息的方式，比如视觉信息、听觉信息、感觉信息都是不同的模态。以往的研究都是大部分都是关注单模态的学习，比如图形分类问题只关心图片信息；语音识别只关注音频信息；机器翻译只关心文本信息等。但近年来涌现出了越来越多基于多模态融合学习的研究工作。比如Ngiam等^[57]就讲多模态学习运用在了深度学习领域，但目前尚且没有结合多模态学习与深度强化学习的相关工作，侧面证明了多模态学习与深度强化学习结合的重要研究意义。

(1) 多模态学习

多模态学习是一个热点研究问题，主要研究如何有效地利用多种模态输入信息进行学习。多模态学习在诸多研究领域都取得了研究成果，例如，在机器人控制领域，研究者们就如何将多个传感器信息数据进行融合展开了研究^[58-60]。此外，在融合声音与视觉数据方面也有相关的工作^[57,61-63]。从机器学习算法层面看，研究者们成功地使用图模型^[61]与条件随机场算法^[62,63]有效地进行多模态信息融合。值得注意的是，近期使用深度学习算法来学习跨模态的特征表达，进而实现多模态信息融合的工作也取得了一定的研究成果^[57,63-65]。既有研究工作证明，多模态学习与深度学习算法能够有效地结合的，为多模态信息融合与深度强化学习提供了有效地借鉴。

(2) 注意力机制

基于多模态输入的强化学习算法需要首先解决如何有效地利用多种模态信息融合的问题。因为不同时刻不同模态信息对于决策的影响程度是不同的，一种直观的做法是对多模态信息分配不同的权重，进行线性组合，并使用组合后的多模态信息作为决策的输入。该设计的优点是能够在不同时刻，通过动态调节不同模态信息的权重，实现关键模态信息的抽取，实现更高效的学习与更智能的决策。比如自动驾驶领域，视觉和声音数据是重要的模态信息，大部分行驶过程中，驾驶员都是通过视觉输入进行车辆的驾驶，此时视觉输入是重要的决策依据，应该赋予其较大的权重。但是当距离传感器探测到危险而发出警报时，此时对于智能体决策来说，声音模态信息相比其他模态信息更加重要，应该被赋予更高的权限。

注意力机制是一种权重分配算法，能够对不同模态信息赋予不同的权重，以实现重要信息的提取，近年来也有越来越多的研究人员在监督学习领域，对注意力机制进行了研究。例如针对机器翻译的研究问题，有使用注意力机制^[66,67]来解决多语言之间的翻译难题的研究工作。比如Caglayan等^[68]直接在将注意力机制用于融合文本和声音两种模态信息来解决翻译问题。

注意力机制也广泛用在了强化学习领域的研究中，例如Hausknecht等^[69,70]将注意力机制用在空间信息的处理上，并与深度循环Q网络，用以解决强化学习问题。此外还有使用注意机制用于解决图像分类问题^[71,72]。Nan等^[73]基于注意力机制提出了最新的稀疏注意力算法来解决长时间的记忆力衰退的难题。该方法主要是设计了一种新型的循环神经网络架构，使得其能够在长时间序列上保证梯度不会弥散。这样一来就能使得信息在时间维度上能够传递的更久远，以解决较复杂的强化学习问题。除了循环神经网络的结构，Junhyuk等^[74]提出了基于记忆力的强化学习算法，该算法使用了记忆单元进行策略优化。记忆单元能够存储时间和空间上的数据序列，在此基础上，注意力机制就能够在时间空间维度上分析智能体行为的前后以来关系，并修正智能体行为，实现策略优化。该算法只使用注意力机制与记忆单元，并没有使用循环神经网络。这说明注意力机制具有一定的鲁棒性，可以和多种底层网络结构想结合，为研究者进行网络结构设计提供了一定的灵活性。

上述研究工作的成功证明了注意力机制确实能够有效地对数据进行权重分配，在时间与空间上进行重要信息的抽取。然而目前尚没有相关工作研究如何将注意力机制与多模态强化学习进行结合。直观上看，注意力机制高效处理时空数据的能力是得到理论与实验论证的，因此将其与多模态强化学习结合，用于处理多模

态信息输入的全部分配，提取决策所需的关键信息，具有一定的可行性，是一项十分有意义的研究工作。

(3) 多模态强化学习

一般的强化学习算法通常仅使用单模态输入作为进行策略决策，例如智能体仅使用视觉输入进行自动驾驶任务。但现实问题中，智能体通常能够感知到多种模态信息，例如声音，视觉、文字以及传感器数据等。为高效的利用多模态数据，有效地解决强化学习问题，研究者将多模态学习与强化学习技术相结合，提出了多模态强化学习，旨在使用多模态信息输入，有效地进行智能体策略优化。

Omidshafiei^[25]于提出了多模态的强化学习算法（CASL），用以解决基于视觉和声音的强化学习问题。CASL成功地将多模态学习与强化学习进行了结合，能够解决多模态输入的强化学习问题。尽管CASL推动了多模态强化学习的研究，同时也存在一定的局限性。首先，CASL只在多模态间使用了注意力机制，并没有考虑模态内部的权重分配，这使得算法仍然存在一定的提升空间。其次，CASL直接将多模态信息简单拼接作为LSTM的输入，直接的拼接会导致一定的信息丢失，损害算法性能。最后，CASL虽然使用了A2OC层次强化学习算法^[75]，但其效果却不如同时层次强化学习的Feudal-Network算法^[76]，说明算法性能还有提升的空间。

综上所述，基于多模态信息输入的智能体策略优化研究并不成熟。如前文所述，该领域的研究仍存在一定的不足与挑战，其中包括既有强化学习算法无法有效处理多模态信息输入的问题、既有注意力机制无法同时在模态间于模态内进行权重分配的问题、直接拼接多模态信息会造成决策时一定的信息丢失的问题等。本文第三章将针对这些问题进行详细的分析讨论，并给出相应的解决方法。

2.4 噪声环境下的智能体策略优化分析

近年来，越来越多的研究工作尝试使用深度强化学习技术解决独立学习智能体的策略优化的研究问题^[29,77,78]。然而，既有的研究工作依然受到多智能体系统的两个内在难题的挑战，即环境中的噪音所产生的随机性以及环境中其余智能体的动态行为产生的随机性。

由于DQN与DDQN等基于值函数估计器的深度强化学习算法本身存在估值偏差的缺陷^[27,28]，会导致策略优化过程震荡甚至产生策略不收敛的情况，影响最优策略求解。此外，环境中固有的噪音以及智能体的动态行为产生的随机性，会导

致策略优化的过程中产生额外的估值偏差，进而极大的降低深度强化学习的算法性能。综上原因，估值偏差会导致独立学习智能体在进行策略优化时，无法实现智能体之间的协同优化，增加算法收敛到帕雷多纳什均衡，甚至是纳什均衡的难度。特别是在合作式强化学习环境中，目前的多智能体强化学习算法并没有理论保证能够收敛到最优策略。

本文主要研究在带噪声的合作式多智能体环境下，独立学习智能体的策略优化问题。虽然独立学习智能体之间共享相同的奖赏值函数，但由于独立学习智能体无法观测到其余智能体的行为以及奖赏值^[36]，既有的深度强化学习算法在实现独立学习智能体之间的协同策略优化，以最大化群体长期利益时，存在如下挑战：

(1) 估值偏差的挑战

Q-learning与Double Q-learning算法已经被证明在估计Q值时，存在高估与低估的问题^[27,28]。为了解决估值偏差的问题，尽可能降低估值偏差对算法的负面影响，Zhang等^[26]提出了一种基于加权双Q值学习算法。该算法能够有效地降低Q值估计过程中的估值偏差，但该算法只使用了较为简单的环境进行了实验论证，并没有使用包括基于原始图片作为输入的环境以及合作式马尔科夫博弈游戏此类较为复杂的环境进行算法验证。因此其估值修复的效果需要进一步验证。

(2) 环境中噪声的挑战

针对合作式马尔科夫博弈游戏，特别是当环境中存在较大噪音的问题，既有的基于独立学习的多智能体算法，在理论上无法保证一定能收敛到帕雷多最优纳什均衡策略^[29]，目前的既有算法也无法有效地解决环境中存在噪声的问题。

(3) 策略协同优化的挑战

为了提高多智能体环境下独立智能体学习到相互配合策略的可能性，Potter等^[79]提出了宽容的Q学习算法（Lenient Q-learning）。该算法的核心思想是使每智能体在学习初期都对Q值的估计保持乐观，这样有利于智能体之间一同朝着合作的策略方向更新。这种宽容的思想已经多次被验证能够有效的使得智能体在实现合作策略的同时，避免陷入局部最优^[80-83]。大部分基于宽容的Q学习算法并没有在复杂的问题中进行验证。虽然Palmer等^[81]提出的深度宽容Q学习算法将该思想进行拓展，但是其并没有考虑环境中的反馈中的噪声，这一缺陷有可能会降低算法学习效率，甚至收敛到次优解。

(4) 经验回访机制的挑战

经验回放机制（Experience replay）^[3]，特别是优先经验回放机制（Prioritized

experience replay) [84]，是用于解决单智能体场景下强化学习问题。但是针对基于独立学习的多智能体问题，经验回放机制反倒可能降低学习效率与效果。因此，需要针对独立学习智能体的环境设计高效地经验回放机制。

综上所属，面向噪声环境下的独立学习智能体策略优化研究并不成熟。如前文所述，该领域的研究仍存在一定的不足与挑战。其中包括强化学习算法本身存在值偏差问题、环境中存在不稳定噪声的问题、独立学习的智能体之间的协同策略优化问题以及经验回放机制导致策略优化效率下降的问题。后文的第四章节会针对这些问题进行详细的分析讨论，并给出相应的解决方法。

2.5 非静态对手环境下的智能体策略优化分析

在多智能体系统的研究中，一个关键的挑战是研究如何应对环境中不断变化行为的其他智能体（对手）^[30]。因为当环境中的对手改变行为时，从独立学习智能体的角度看，相当于环境发生了变化，这间接改变了智能体策略的优化方向。由此可见，在面向非静态对手环境下的多智能体策略优化过程中，是否进行对手行为建模，以及有效地进行对手行为建模，成为了面向非静态对手环境下的多智能体策略优化研究中的一个重要问题。注意到为了避免歧义，在后文中，无论是合作式或者是竞争式多智能体环境，都将环境中存在的其余智能体统称为“对手”。

(1) 非静态智能体建模

标准的独立学习智能体假设对手的行为以及奖赏值信息是无法观测的，但现实情况下，独立学习智能体是能够观测到有限的对手信息的，因此在具体算法设计时，可以考虑使用有限观测的信息辅助智能体策略优化，提升策略优化效率。从独立学习智能体使用有限的观测信息进行策略优化的角度，Hernandez^[30]针对非静态智能体建模的研究，提出了如下五种建模方法：

- 忽略模型 (Ignore): 这是最简单的建模方式，独立学习智能体完全忽略对手的策略变化，认为对手始终使用的是静态策略，甚至完全把对手当做是环境的一部分。Q-learning算法^[44]以及虚拟博弈^[85]都属于此类范畴。这类建模的优点是实现简单，但是无法应对较复杂的多智能体问题。
- 遗忘模型 (Forget): 独立学习智能体根据观察到的对手行为，针对对手的策略进行策略建模，并用实时观测的对手行我数据来更新策略模型。该模

型在解决非静态智能体问题时，具有一定的有效性。但其缺点是始终认为对手只使用一种策略，并且需要实时更新对手的策略模型。WoLF-PHC算法就属于此类范畴^[86]。

- 特定对手响应模型（Respond to target opponents）：该方法认为对手是某种特定的类型，拥有一定的目标，并根据此目标变化行为。例如Minimax-Q算法^[87]认为其余的智能体的目标是为了降低当前智能体的收益。
- 学习模型（Learn opponent models）：此类模型认为对手在一系列的静态策略之间不断切换，每次选取一个策略进行执行。因此，独立学习智能体通过检测对手使用的策略类型，并执行合适的应对策略来实现长期收益的最大化。Hernandez^[88]基于此思路，提出一种高效的策略检测方法，能够有效检测其余智能体使用的策略类型，但缺点是并没有考虑对手会有策略性地切换行为。学习模型的建模方式与实际问题较为接近，应用也较为广泛。
- 心智模型（Theory of mind）：该建模方法是最复杂的一类模型，主要考虑对手会有策略地切换行为。具体来说，所有智能体都认为其余智能体在策略优化时，会考虑自己的行为变化。因此这里引入了循环推理的心智模型。Piotr等^[89]提出了一种循环推理的方法来解决此类问题，但是缺点就是计算复杂度较高。因此，此类模型是面向非静态对手的研究中最复杂且难度最大的建模方式。

尽管独立学习智能体的严格定义要求智能体完全不使用额外关于对手信息进行策略优化，但显示问题中，使用有限的观察信息是有助于策略优化过程。因此，在解决多智能体问题时候，并不一定选择完全忽略其余智能体的行为。此外，考虑到Ignore、Forget以及Respond to target opponents模型对对手的假设过于简单，而Theory of mind模型又太耗费计算资源，因此，本文使用学习模型（Learn opponent models）对智能体策略进行建模。下一节主要针对基于学习模型的相关工作展开讨论。

（2）基于学习模型的相关工作

当面对拥有多个策略，并且在多个策略间不断切换的对手时。学习模型是一种有效的建模方法。简单来说，学习模型主要是根据观察到的对手信息，判断对手当前使用的策略，进而使用合适的应对策略执行。Silver等^[90]提出RL-CD算法与Hernandez^[91]提出的DriftER算法都使用了此类对手建模的方法，但不同的是，DriftER使用R-max算法^[92]进行最优策略求解，要求较高的计算资源。

学习模型中，对手所使用的策略及应对策略一般作为先验知识，在线交互

时，智能体只需要关注检测对手当前使用的策略并选择相应的应对侧策略执行即可。但对于一些特殊的场景，对手可能在线使用未知的策略，为了此类问题，Hernandez等^[93]拓展了BPR算法，并提出了BPR+算法^[88]。BPR+算法不仅能够高效的检测对手类型，并选择对应的应对策略执行，还能够检测对手是否使用未知策略。这使得BPR+能够不受先验知识的限制，拥有持续学习的能力。

另一方面，BPR+算法仍存在一定的不足，首先，BPR+只使用和对手交互过程中收到的奖赏值作为对手类型判断的依据，这在合作式多智能体环境下会造成策略检测度的下降，导致无法正确判断对手类型。其次，在针对未知对手的应对策略优化时，BPR+使用R-max用于求解未知策略的应对策略，这无法有应用在基于图片为输入的复杂问题中，再次，BPR+针对每一个对手的相应策略进行单独保存，这造成了一定的空间浪费。最后，BPR+并没有与深度学习结合，也未拓展到基于视觉输入的复杂问题上，其算法有效性需要使用相对复杂的多智能体问题进一步验证。

综上所属，面向非静态对手环境下的多智能体策略优化研究并不成熟。如前文所述，该领域的研究仍存在一定的不足与挑战。其中包括对手策略检测不准确的问题、针对未知对手的应对策略优化问题、相应策略保存的问题以及在复杂问题下算法有效性的问题。后文的第五章节会针对这些问题进行详细的分析讨论，并给出相应的解决方法。

2.6 本章小结

本章首先介绍了强化学习与多智能体系统的相关理论基础以及本研究所涉及的主要相关算法。接着，以基于深度强化学习的多智能体策略优化为研究目标，从环境、强化学习算法以及多智能体系统三个角度，详细分析了既有算法在解决智能体策略优化时存在的局限性。进而引出本文拟解决的研究问题。

具体来说，本文旨在从前述三个角度切入，系统性的解决智能体策略优化问题。首先，从环境角度，本文拟解决多模态感知信息输入下的智能体策略优化问题（第三章）；其次，从强化学习算角度，本文拟解决噪声环境下的智能体策略优化问题（本文第四章）；最后，从多智能体系统角度，本文拟解决非静态对手环境下的智能体策略优化问题（本文第五章）。

第3章 面向多模态信息输入的智能体策略优化研究

本章主要研究面向多模态信息输入的智能体策略优化问题。主要论述了多模态感知信息对智能体决策的重要性，针对既有算法中存在的缺陷，包括注意力机制的局限性以及LSTM对多模态信息融合的不足进行了详细的分析，并提出了相应的解决方法。最后本章节使用由潜及深的使用多模态输入信息的复杂问题来验证新算法的有效性。

3.1 引言

在强化学习算法框架中，智能体需要根据感知到的信息（state）进行决策，再根据环境返回的奖赏值进行策略优化，实现长期利益最大化。因此，环境返回的感知信息输入是影响智能体决策的关键因素，在某些情况下甚至是唯一因素，这说明了感知信息输入对智能体决策以及后续策略优化的重要影响。具体来说，不同的感知信息会影响智能体做出不同的决策，不同的决策会使得环境返回不同的奖赏值，并且使得在决策后的下一时刻，智能体收到不同的感知信息（下一时刻的state）。由于强化学习问题是一个序列决策优化问题，某一时刻的决策不同，都有可能在后续的序列决策中被放大，进而循环往复地不断影响后续的决策，导致产生完全不同，甚至差异极大的序列决策结果。因此，在强化学习策略优化过程中，关于有效利用感知输入的研究，是实现高效地智能体策略优化的基础。

近年来，涌现了大量关于深度强化学习算法的研究，但既有的研究工作大多是基于单模态信息数据进行策略优化的研究，比如基于单模态视觉输入信息进行Atari游戏的研究^[3]，又如基于单模态传感器输入信息进行机器人控制的研究^[4]等。这些研究工作的侧重点在于如何从算法优化本身出发，对策略优化问题展开研究，却忽略了单模态信息可能存在信息不充分的局限性。例如自动驾驶问题，驾驶员一般能够感知到多模态的信息源，包括视觉信息输入和听觉信息输入等；大多时候，驾驶员通过视觉感知信息进行驾驶，但在危险的时候，车上传感器会发出声音警报信息提醒驾驶员当前的路况信息。上述案例说明，单模态信

息可能存在信息不足的问题，在真实而复杂的现实问题中，基于单模态信息输入的强化学习可能不足以进行有效的策略优化。因此，如何设计能够有效处理多模态信息输入的强化学习算法，进行智能体策略优化是多模态强化学习研究所面临的一大挑战。

其次，注意力机制^[71]是一种有效的信息提取机制，通过对感知输入信息中的特征进行不同的权重分配，实现信息的提取，进而实现更加智能的决策。既有的研究工作表明，基于单模态输入的机器学习算法能够有效的与注意力机制结合，有效的解决翻译问题^[67,68,94]，图像识别问题^[72]以及图像分割^[95]等问题。此类算法主要研究如何对单模态输入内部的特征信息进行权重分配，实现算法优化。与此同时，关于模态间的权重分配的注意力机制研究也取得了一定的研究成果^[58-61]，此类算法主要进行模态间进行注意力权重的分配。综上，针对单模态信息的注意力分配机制只考虑模态内的信息提取，属于细粒度的权重分配；而针对模态间的注意力分配机制，又只考虑了不同模态间的注意力权重分配，属于粗粒度的权重分配。因此，如何设计面向多模态信息输入的注意力机制，平衡模态内与模态间的权重分配问题，也是目前面临的第二个挑战。

最后，既有的工作表明，LSTM确实能够有效地与强化学习算法相结合，实现有效的策略优化^[56,96]。由于强化学习算法是一种解决时序决策问题的优化算法，因此，智能体能在进行策略优化过程中，如果能够有效利用时序上的前后信息，一般来说能够取得更好的策略优化效果。然而，由于LSTM本身并不考虑多模态信息输入的问题，因此其无法有效处理多模态输入的强化学习问题。对此，如何从多模态信息输入的角度来考虑时序上的前后信息，以实现智能体策略优化，是目前面临的第三个挑战。

综上所述，本章节研究基于多模态信息输入的智能体系统策略优化问题。主要研究在含有多模态信息的环境下，智能体如何对多模态信息进行有效的知识融合，并在此基础上实现高效的策略优化。具体来说主要考虑多模态信息输入的环境下，针对前文所述强化学习算法针对多模态感知输入的局限性、多模态信息输入下注意力机制的局限性以及LSTM处理多模态信息输入的局限性进行了详细的分析，并提出了相应的解决方法。

总体来说，针对强化学习算法针对多模态感知输入的局限性问题，本文使用基于多模态信息输入的网络架构，以处理多模态输入信息，实现高效的智能体策略优化。其次，针对多模态信息输入下注意力机制的局限性问题，文提出了层次化注意力机制，同时考虑模态内与模态间的注意力权重分配，并在此基础上与强

化学习算法相结合，实现有效的智能体策略优化。最后，针对LSTM处理多模态信息输入的局限性问题，本文提出了一种LSTM的变种，能够有效地使用多模态信息输入进行智能体策略优化。章节的最后使用了包括自动驾驶等三种基于多模态信息输入的环境来验证新算法的有效性。

3.2 基于多模态信息输入的策略优化算法框架

强化学习算法要求智能体根据观察到的感知信息进行行为选择。因此，关于感知信息的研究是强化学习研究以及多智能体策略优化的研究基础。本文主要研究在多模态信息输入的环境下，如何有效的利用多种模态信息进行策略优化。具体来说，本文使用一种基于多模态信息输入的网络架构，并在此基础上提出了层次注意力机制设计以及基于多模态信息融合的LSTM网络。这三者的关系如下：

(1) 基于多模态信息输入的网络架构

为解决全文所述一般强化学习算法针对多模态感知输入的局限性问题，本文使用了基于多模态信息输入的网络架构，通过对多模态信息输入进行有效融合，实现高效的策略优化。概括来说，该框架针对不同的模态输入单独设置感知层神经网络，实现不同模态内的知识提取，为后续智能体决策以及策略优化提供多模态的感知特征。基于多模态信息输入的网络架构一种通用的神经网络结构，具有较高的泛化性，能够和大部分主流的Value-Based、Policy-Based、Actor-Critic强化学习算法相结合进行策略优化。本文将其与DQN算法、DDPG算法以及A3C算法相结合，分别提出了MM-DQN、MM-DDPG与MM-A3C算法，并在实验部分论证了新算法的有效性。详细的内容在3.3节中展开。

(2) 层次注意力机制

为解决前文所述的多模态信息输入下注意力机制的局限性，本文提出了层次化注意力机制，实现层次内与层次间的注意力权重分配。概括的说，层次注意力机制在每个模态内部都放置了权重分配模块，对每个模态的信息输入特征进行权重分配；同时，当经过权重分配的多个模态间信息准备信息融合之前，还增加了模态间的权重分配模块，实现模态间的权重分配。层次注意力分配机制实现了多层次的注意力权重分配，并且可使用端到端的训练方式进行网络训练，能够有效地促进智能体策略优化。详细的内容在3.4节中展开。

(3) 基于多模态信息融合的LSTM网络

为了解决前文所述的LSTM处理多模态信息输入的局限性，本文提出了一种LSTM网络的变种。经典的LSTM网络一般只处理单模态信息输入，这导致其无法处理多模态信息输入的问题。尽管可以将多种模态信息进行直接拼接，形成一种特殊的“单模态”信息输入，但这种做法可能到时LSTM在进行内部状态更新时，导致一定程度上的信息丢失。为此，本文提出了一种基于多模态信息融合的LSTM网络，通过针对不同模态信息设计单独的遗忘门，实现对多模态信息输入的有效处理，促进智能体策略优化。详细的内容在3.5节中展开。

3.3 基于多模态信息输入的网络架构

(1) 多模态信息与强化学习

多模态信息通常指多种形式的信息，例如声音信息和视觉信息就是两种模态信息。多模态学习旨在同时使用多模态信息进行学习，其优点是能够融合多个模态中的信息，更加全面的做出更智能的决策。多模态学习在多媒体领域有众多诸多的成功运用案例。但是在强化学习领域中基于多模态学习的相关工作较少，主要是因为基于多模态信息的强化学习存在一定的挑战，例如使用多模态信息进行学习的时候，多个模态信息之间可能存在相互干扰，影响强化学习效果；其次，由于强化学习本身是用于解决序列决策优化问题；而既有的基于多模态学习研究主要是和监督学习相结合，解决单次决策问题（比如图像分割等问题）。因此，关于将多模态学习与强化学习相结合的研究存在一定的不足。

一般来说，不同模态的数据通常包含不同的知识，例如自动驾驶问题，视觉感知能够提供车道、红绿灯、行人等信息；而听觉感知能够提供警报等相关信息，综合使用多模态信息往往能够获得更全面的信息。直觉上，更全面的信息有助于智能体做出最优决策。因此，基于多模态信息输入的强化学习研究有着重要的研究意义，是强化学习研究领域的一个重要研究方向。

本文从智能体收到的感知信息的角度，尝试设计能够有效处理多模态信息输入的强化学习算法，以解决含有多种模态信息的复杂问题。首先本文针对能够有效地处理多模态输入的网络架构展开了研究。

图3-1展示了著名的蒙特祖玛的复仇（Montezuma Revenge）游戏场景中的视觉和听觉两种模态感知信息，分别用分别是 s_v 和 s_a 表示。该场景中，智能体需要基于多模态信息实现有效的策略优化。

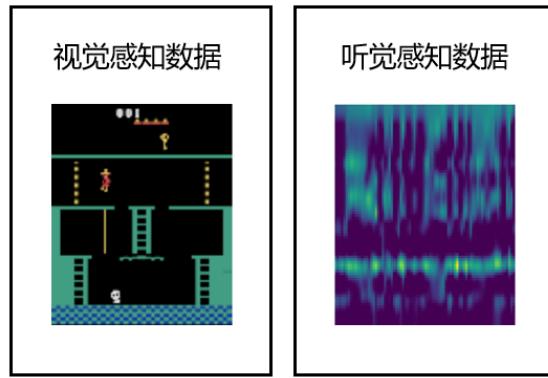


图 3-1 多模态感知信息输入

(2) 拼接式多模态信息输入的网络结构

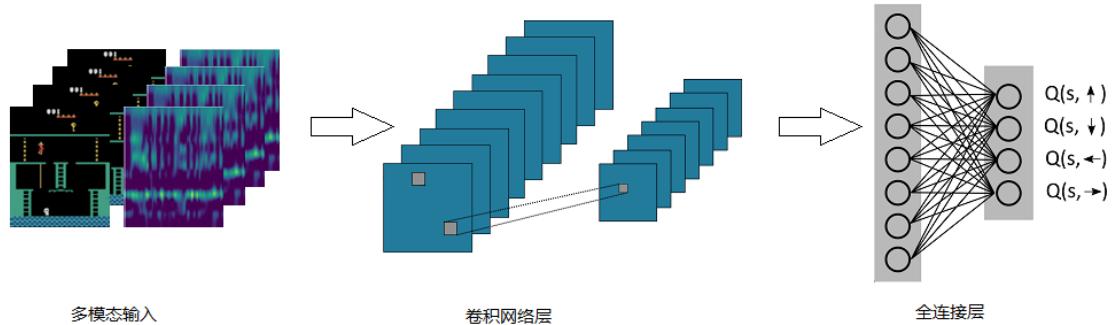


图 3-2 拼接式多模态信息输入的网络结构图

由于一般深度强化学习算法（例如DQN算法）只使用单模态信息作为输入进行策略优化，并取得了成功。因此，一种直接而简单的设计是直接将多模态的输入拼接起来作为网络的输入进行端到端的训练（图3-2）。本文称之为拼接式多模态输入（Concatenated Multimodal Network, CMMN）。CMMN的输入包含了完整的原始多模态信息，并没有信息丢失，因此在理论上能够进行有效的策略优化。

(3) 分离式多模态信息输入的网络结构

基于单模态信息输入的DQN算法效率较为低下，这主要是源于视觉信号内的复杂性以及环境本身存在稀疏奖赏值等原因。此外，考虑到多个模态本身拥有各自的复杂性，拼接式多模态信息的做法，很大程度上会进一步放大强化学习算本身训练效率低下的缺陷。与此同时，拼接式多模态信息会增加神经网络处理混合输入的复杂度，增加网络训练的难度，甚至导致策略优化不收敛。因此，为了降低多模态学习过程中的复杂度，提高学习效率，本文提出使用分离式多模态信

息输入的网络结构 (Seperated Multimodal Network, SMMN) (如图3-3)。

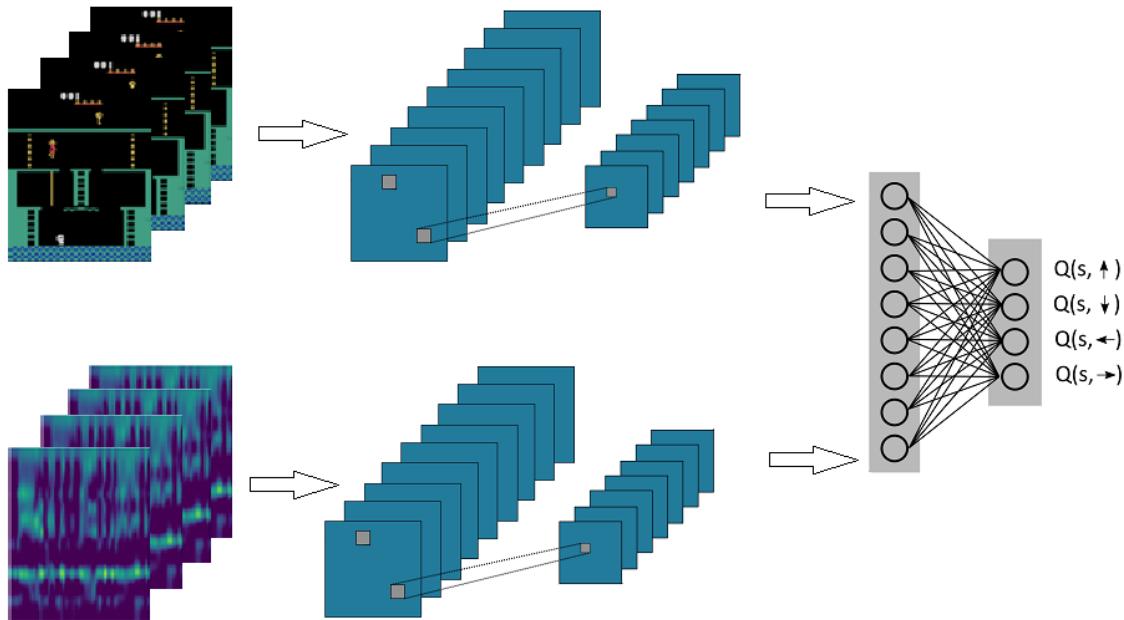


图 3-3 分离式多模态信息输入的网络结构图

SMMN针对每个模态输入都使用独立的神经经网络进行处理，确保每个神经网络都专注于抽取单个模态中的信息，且只在最后使用全连接层对各个模态的网络输出结果进行拼接，输出决策。该设计不仅有利于每个神经网络处理各自的单模态输入数据，而且从实现的角度也相对简单。SMMN是本章节后续研究所使用的基础网络结构，接下来的层次注意力机制设计以及多模态信息融合的研究，都基于该网络结构进行展开。

3.4 层次注意力机制

长短期记忆网络(Long Short Term Memory, LSTM)^[97]是一种特殊的循环神经网络(RNN)^[98]，能够有效地处理基于时序的序列输入。而注意力机制^[94]是一种基于LSTM实现的权重分配机制，能够与深度强化学习算法结合^[56,70]，有效地实现智能体的策略优化。鉴于此，本文提出一种基于层次注意力机制 (Hierarchical Attention, HA)，能够有效的与SMMN进行结合，实现注意力权重的层次化分配。

具体来说，HA的设计思路来源于两方面考虑：一方面，不同模态信息在不同时刻所携带的信息重要性程度不同，因此智能体在不同时刻需要有所侧重选择不

同的模态输入星星，进行策略选择。鉴于此，HA的设计考虑了不同模态间的注意力权重分配问题。另一方面，每个模态信息内的不同特征的重要性也有所不同，也需要有所侧重的选择。鉴于此，HA的设计还考虑了单一模态内不同特征的注意力权重分配问题。因此，HA是一种考虑了多尺度的注意力权重分配机制，能够有效地与多模态强化学习算法相结合，增加策略优化的收敛速度与算法效果。

3.4.1 注意力机制

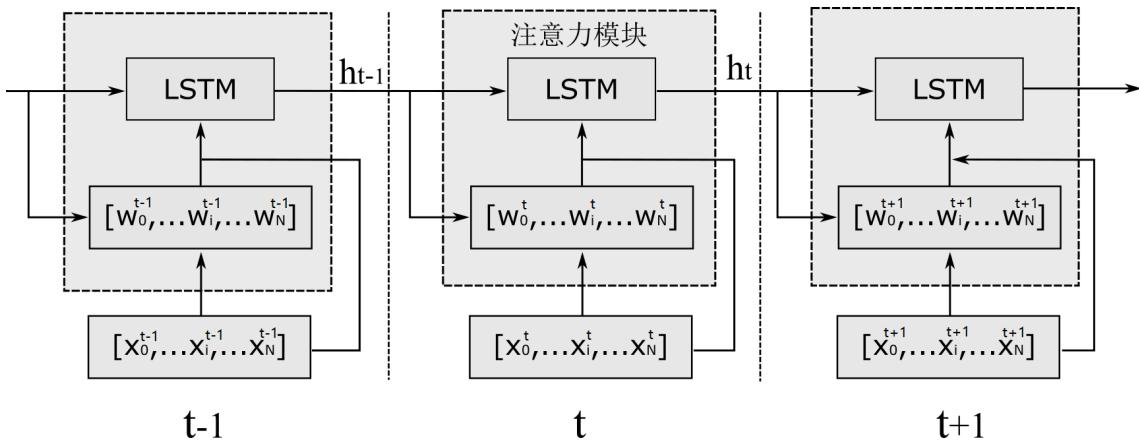


图 3-4 注意力机制权重分配流程图

注意力机制是一种基于LSTM实现的权重分配算法^[97]，图3-4展示了注意力权重分配的流程，具体来说，当给定 N 个多模态的感知特征时， x_i 表示第*i*个模态感知特征经过神经网络处理后提取出的特征。在*t*时刻，注意力机制模块的输入是经过*N*个神经网络处理过的多模态特征 x_i^t ，以及前一时刻LSTM的隐藏层状态值。注意力机制的输出是一个*N*维向量**w**，形式如下：

$$\mathbf{w} = [w_1, \dots, w_i, \dots, w_N] \quad (3-1)$$

其中每*i*个维度的值 w_i 代表分配给第*i*个模态感知数据的注意力权重。而权重**w**的计算公式如下：

$$w^t = Softmax(\text{Linear}(\text{Tanh}(\sum_{i \in N} \text{Linear}(x_i^t) + \text{Linear}(h^{t-1})))) \quad (3-2)$$

上述公式中 h^{t-1} 表示的是上一个时刻LSTM的隐藏层状态值。 Linear 、 Tanh 、 Softmax 分别表示线性函数、激活函数以及归一化处理函数。在*t*时刻，注意力机制模块的输出**w^t**是权重向量，代表对*N*个不同模态的感知特征的注意力程度。将权重

一一分配给多模态感知特征，并将权重化后的结果出入到LSTM网络，得到最终的策略。

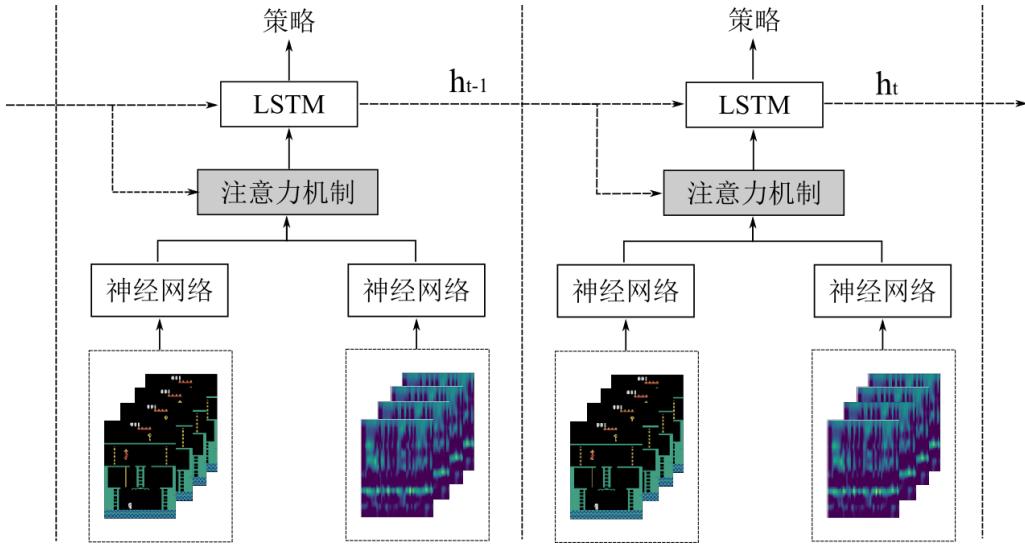


图3-5 基于注意力机制的强化学习框架图

图3-5展示了注意力机制的权重分配流程，首先多模态感知特征需要经过各自的神经网络进行特征提取，得到 x_i 后将其输入到注意力机制模块得到注意力权重 w ;然后进行注意力权重分配，将 w_i 与 x_i 相乘，得到权重化后的输出 $[w_1x_1, \dots, w_Nx_N]$ 。最后将权重化结果输入到LSTM模块中，得到最终决策。整个架构使用端到端的训练。

图3-5提出的架构考虑到了模态之间的权重分配，这使得智能体在不同时刻能够使用不同的模态感知特征，基于更全面的信息进行更智能的决策。但该方法存在一个潜在的缺陷，即注意力机制只有在 N 个多模态感知信息之间进行权重分配，并没有考虑针对每个模态内部进行注意力分配。考虑到注意力机制权重分配本身可用在多尺度上，本文提出了另一种层次注意力机制，可针对每个模态内部再次进行注意力的分配。

3.4.2 层次注意力机制

图3-6展示了使用层次注意力机制的强化学习算法框架图。不仅含有用于多模态间的注意力分配模块，还增加了用于各个模态内的注意力分配模块。

模态信息内部的注意力分配机制和模态间的注意力分配机制核心思想是类似的。举例来说，针对视觉输入， v 表示原始视觉输入中的特征信息。将 v 和前一时

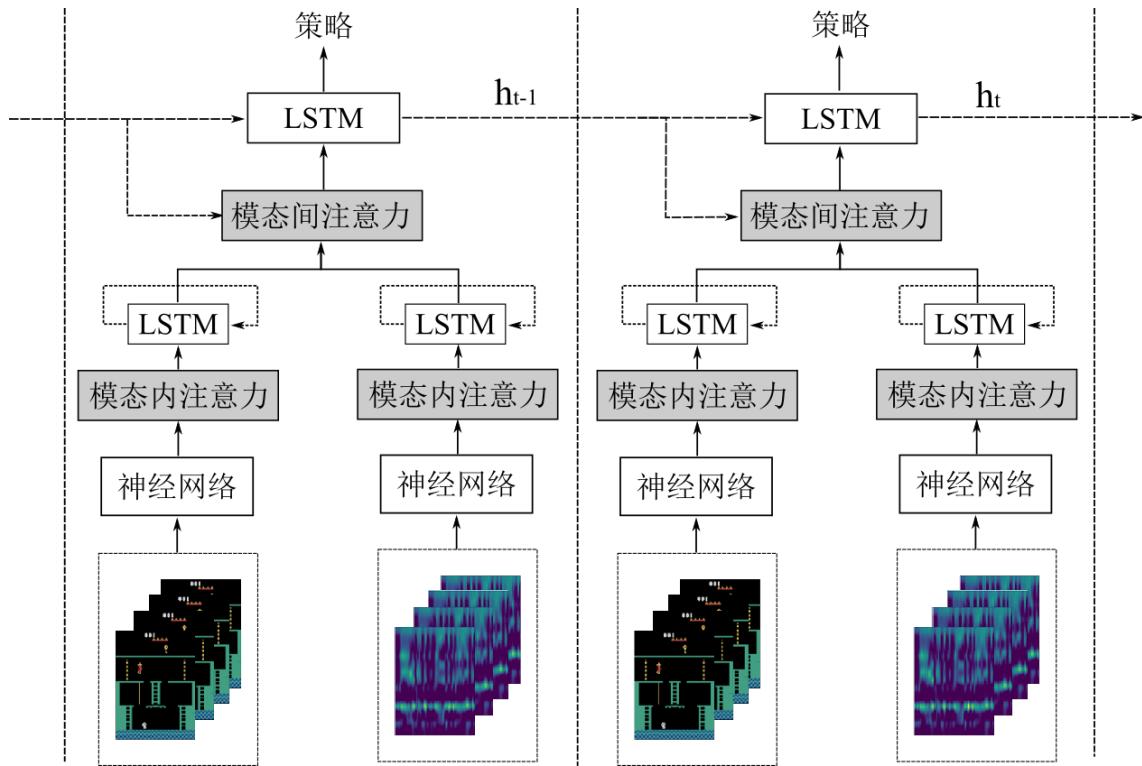


图 3-6 使用层次注意力机制的强化学习算法框架图

刻LSTM的隐藏层状态值输入到模态间注意力分配模块，得到模态内的注意力分配权重 α 。这里 α 是一个权重向量，形式如下：

$$\alpha = [\alpha_0, \dots, \alpha_i, \dots, \alpha_{size(v)}] \quad (3-3)$$

其中第*i*个维度的值 α_i 代表分配给 v 中第*i*维特征的注意力权重，而 $size(v)$ 表示 v 的特征维度。权重向量 α 的计算公式如下：

$$\alpha^t = \text{Softmax}(\text{Linear}(\text{Tanh}(\sum_{i \in N} \text{Linear}(v(i)^t) + \text{Linear}(h^{t-1})))) \quad (3-4)$$

注意这里的 $v(i)$ 表示的是视觉感知输入 v 中的第*i*维特征值。将注意力分配权重 α 与原始视觉模态信息 v 相乘，得到模态内注意力分配的结果 $X_0 = \alpha \cdot v$ ，并将其作为后续模态间注意力分配的输入。

本方法的目的在于优先在模态内对数据进行注意力分配，再在模态间进行注意力权重分配。这样做的优点在于层次注意力机制不仅能够重点关注每个模态输入中有用的特征信息；还能尽量避免学习过程中模态间的相互干扰，提升算法效率。

3.5 多模态信息融合

大量的研究工作表明，LSTM能够有效地解决时序上的序列输入问题，而使用了LSTM的深度强化学习算法能够高效地进行策略优化。然而，既有研究工作大多只针对单模态输入进行研究，并且LSTM网络结构本身也只处理单模态的信息输入，导致了LSTM无法直接用于解决多模态输入的问题。

从策略优化的角度，多模态信息的融合的目的在于有效地提取多源数据中的信息，并基于融合的信息实现有效的策略优化。这里不仅要考虑信息在时序决策上的传递，还要考虑信息融合过程中的完整性，有效地数据融合机制对策略优化效果有十分重要的影响。因此，从信息在序列决策上传递的角度考虑，结合既有的研究工作，本文选择使用LSTM网络结构，而从信息的完整性角度考虑，本文主要考察拼接式和分离式两种数据处理方法的有效性。

从强化学习网络结构的角度，多模态信息的融合的目的在于融合经过权重分配后的多模态输出，并将其输入进LSTM以实现策略决策。因此，本章节的研究重点介于层次注意力模块之后，LSTM之前，主要研究如何有效的将多模态信息进行融合，并将其与LSTM相结合，实现策略优化。

3.5.1 基于单信息流的LSTM网络

首先考察拼接式数据融合方法，简单来说，拼接式数据融合方法直接将多源信息进行拼接，将拼接后的单信息流作为LSTM网络的输入。这种方法在理论上能够保留信息完整性，同时能够直接和LSTM网络结合，以处理信息在时序上的传递问题，是一种直观而简单的数据处理方法。具体细节如下：

拼接式数据融合方法，对层次注意力机制分配权重后的多个数据流 $[w_0X_0, w_1X_1]$ 进行拼接，将拼接后的单信息流 X 作为LSTM网络的输入，具体形式如下：

$$X = [w_0X_0, \dots, w_iX_i, \dots, w_NX_N] \quad (3-5)$$

这里 w_iX_i 代表经过注意力权重加权后的第*i*个模态数据。直接将*N*个模态数据进行拼接处理的好处在于实现比较简单，并且也没有信息丢失。经过拼接处理后的 X 携带着多源信息输入到LSTM网络中进行处理。

图3-7展示了LSTM网络处理对输入 X 的算法处理细节，其中在LSTM中存在三

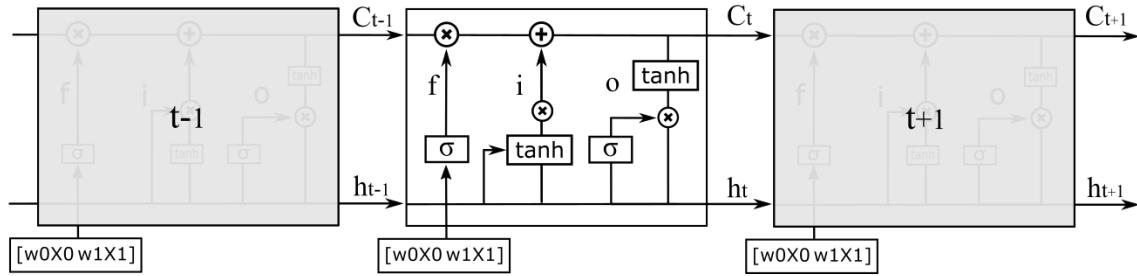


图 3-7 基于单信息流的LSTM网络结构

个逻辑门操作，分别是遗忘门 f 、输入门 i 以及输出门 o ，此外LSTM还会维护一个内部状态 C ，针对每个输入 X ，LSTM会输出 h 。概括地说，LSTM通过内部状态 C 存储信息；每一时刻，LSTM会使用遗忘门 f 选择性地丢弃旧信息，同时使用输入门 i 选择性的筛选新信息来更新内部状态 C ，实现信息在时序上的有效更新与传递；此外LSTM使用输出门 o 来计算当前时刻的网络输出，具体的形式化的定义如下：

$$f_t = \sigma(\text{Linear}(h_{t-1}, X_t)) \quad (3-6)$$

$$i_t = \sigma(\text{Linear}(h_{t-1}, X_t)) \quad (3-7)$$

$$o_t = \sigma(\text{Linear}(h_{t-1}, X_t)) \quad (3-8)$$

在 t 时刻，遗忘门 f 、输入门 i 以及输出门 o 分别使用当前时刻的信息输入 X_t 以及上一个时刻的隐藏层状态 h_{t-1} 作为输入，通过Linear线性变换以及Sigmod激活函数 σ 计算输出，用于更新网络内部状态 C 等后续操作，具体更新公式如下：

$$\tilde{C}_t = \tanh(\text{Linear}(h_{t-1}, X_t)) \quad (3-9)$$

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \quad (3-10)$$

上述公式表明，LSTM在每一时刻都会对内部状态 C 进行更新，确保信息在时序上有效的传递。具体来说，LSTM会使用遗忘门输出 f_t 选择性的忘记一些信息 $(f_t * C_{t-1})$ ；同时通过输入门的输出 i_t ，选择性的记住一些最新的信息 $(i_t * \tilde{C}_t)$ 。最后LSTM会根据输出门的输出 o_t ，选择合适的信息 $o_t * \tanh(C_t)$ 作为当前时候的网络输出，具体计算公式如下：

$$h_t = o_t * \tanh(C_t) \quad (3-11)$$

3.5.2 基于多信息流的LSTM网络

信息在时序上的有效传递是个十分复杂的研究问题，尽管LSTM一定程度上能够解决此问题，但仍然面临训练复杂度较大、学习速率较慢等难题。因此，尽管拼接式数据融合是一种简单而直观的数据融合方法，但将多信息流直接拼接成单信息流作为网络输入的做法，可能会导致LSTM网络训练效率过低的问题。此外，直接将所有信息流拼接成单信息流，可能会在LSTM网络在训练时，产生多信息流之间的相互干扰的潜在风险，影响训练效果以及学习速率。鉴于此，本文直接使用多源信息流作为输入，通过拓展LSTM的基础结构，实现针对多模态信息的单独处理。该设计使得LSTM能够专注于处理时序上的信息传递问题，而不用考虑多源信息间的相互干扰问题，更好地发挥LSTM的优势。

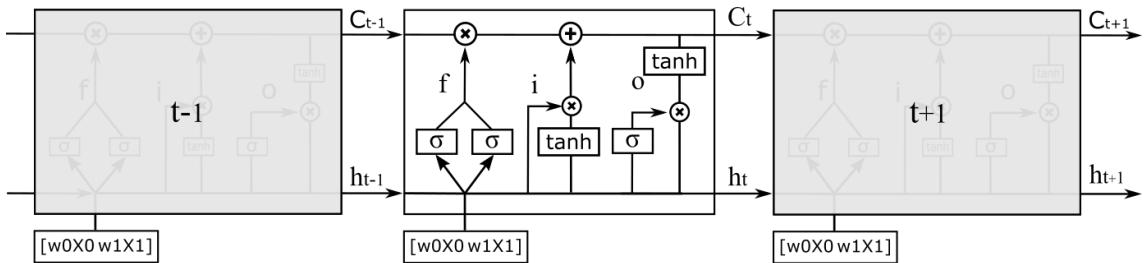


图 3-8 基于多信息流的LSTM网络结构

图3-8中描述了基于多信息流的LSTM网络架构，相较于基于单信息流的LSTM网络结构，针对LSTM网络的遗忘门进行了调整，拓展了LSTM网络处理多信息流输入的能力。具体来说，改进后的网络结构针对每一种模态信息输入，都单独添加了独立的遗忘门，用于处理该模态输入；并联合考虑所有信息流的处理结果，进行LSTM内部状态C的更新，实现信息在时序上的有效传递。遗忘门的形式化的定义如下：

$$f_{i,t} = \sigma(\text{Linear}(h_{t-1}, (w_i X_i)_t)) \quad (3-12)$$

在基于多信息流的LSTM网络结构中，多个遗忘门 f_i 与多个模态*i*之间形成一一对应的关系，且对应的输入也修改为 $(w_i X_i)$ ，其目的在于使得遗忘门 f_i 专门处理模态*i*的信息输入。

此外，关于LSTM内部状态C的更新也有所不同。基于多信息流的LSTM网络

结构联合多个遗忘门的处理结果，对内部状态 C 进行更新，更新公式如下：

$$C_t = \sum_{i=1 \dots N} f_{i,t} * C_{t-1} + i_t * \tilde{C}_t \quad (3-13)$$

上述公式表明，内部状态 C 的更新受到多个遗忘门 f_i 共同影响的结果，该设计尽可能保证模态之间对于内部状态 C 的影响是独立的，最大程度降低模态间的相互干扰，实现信息的有效传递。此外，内部状态 C 的更新受到遗忘门 f 与输入门 i 的共同影响，但本文选择只针对遗忘门 f 进行拓展的原因有如下几点：首先相比输入门 i 遗忘门 f 是直接作用在 C 上的，有着最直接的影响，而输入门 i 是通过 \tilde{C} 对 C 产生间接的影响；其次，考虑到LSTM本身训练效率的问题，对遗忘门和输入门同时进行拓展会极大的增加算法复杂度以及实现难度。因此，本文最终选择在遗忘门进行拓展。

综上所属，本节提出了基于多信息流的LSTM网络结构，通过对LSTM网络中的遗忘门进行拓展，实现对多模态信息输入的有效处理，增强了LSTM网络处理复杂信息输入的能力。同时，基于多信息流的LSTM网络结构能够与层次注意力机制相结合，进行端到端的网络训练，实现多模态信息输入下的智能体策略优化。

3.6 实验验证

本文将SMMN、HA以及基于多信息流的LSTM结构三个新机制与DQN算法、DDPG算法以及A3C算法结合，分别提出了MM-DQN、MM-DDPG、MM-A3C算法，并使用水牢迷宫环境，迷宫逃生环境以及自动驾驶环境三个含有多模态感知信息的实验来论证新机制对强化学习算法的效果提升。在三个实验环境中，智能体只有通过有效地利用多模态信息来才能有效地完成任务。

具体来说，第一个水牢迷宫环境是一个充分信息的多模态问题，即智能体通过视觉输入信息已经足够完成任务，但如果能同时结合听觉输入信息，则能够更高效的完成任务。第二个实验迷宫逃生环境是一个信息不充分的多模态问题，若仅使用视觉输入信息是不足以有效地进行策略优化，智能体需要同时利用多模态信息输入才能有效完成任务。最后一个是复杂的自动驾驶任务要求智能体不仅需要有效的利用多模态输入信息进行安全的车辆驾驶，还要求智能体对不同模态信息的重要性进行学习，实现高效的车辆驾驶。

综上，本文通过三个由易到难的实验，循序渐进地验证所提出的新机制能否有效的实现对多模态信息的处理，使得智能体进行高效地策略优化。

3.6.1 充分信息的多模态问题

在现实中很多场景对安全有着较高的要求，例如自动驾驶问题中对行驶安全的要求是十分严格的，又比如对一个控制机械臂的智能体而言，应当保证机械臂不会与自身或者是周围其它物体发生碰撞，否则会引起严重的安全问题。一般来说，研究人员会对智能体在真实环境中需要遵循一定的安全准则。但现实情况是当智能体在学习时，它并不了解环境，无法预测其行为之后的后果。单个模态的信息通常有限，并不足以提供例如安全示警之类的信息。为确保安全性，可以采用多模态的信息，来完成策略优化。

3.6.1.1 环境介绍

首先，本文使用名为水牢迷宫的游戏来验证算法性能。具体来说，环境中智能体（机器人）需要通过视觉模态信息进行策略优化，完成迷宫任务；而当智能体接近危险区域的时候，会收到危险信息预警。因此，智能体学习期间，如何合理利用额外声音模态信息更全面的了解周围环境，从而使得其自身在更安全的状态空间上运行将是本例所关注的重点。

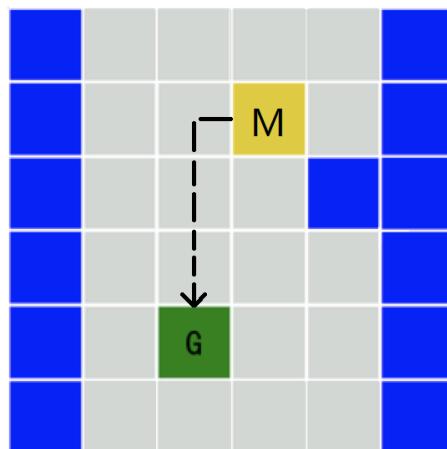


图 3.9 水牢迷宫 (Water Maze)

图3-9展示了水牢迷宫环境，智能体M在环境中巡游，每次都从图中同样的位置出发，目的是到达目标G点为止。智能体的动作包含上下左右四个操作。在不碰壁的情况下，每次智能体都会在相应的方向上移动一个方格。若智能体所采取的动作是碰壁，则其保持原地不动。图中浅色（灰色）方块是安全的格子，深色

(蓝色) 方块表示水域。智能体若进入水域，则智能体(机器人)将遭受不可逆的严重损坏，当前回合终止。

为了避免危险，辅智能体 M 处于水域相邻的位置时候(距离为1)，会收到声音警报信号，而在其余时刻收到的是噪声。智能体 M 每次运动将获得-1的奖励，到达目标方格 G 获得的奖励是40，如果在100步之内智能体都没有到达终点，则游戏强行结束。这个实验中，智能体的输入是视觉感知信息(当前图片)，同时还包括声音模态。注意到环境中存在着随机性，每次智能体选择执行动作 a 的时候，环境只有80%的概率会执行动作 a ，有20%的概率会在其余动作中随机选择一个执行。这使得智能体应该尽量避免靠近水域的区域，因为靠近水域的区域始终都存在一定的可能会调入到水域中。如图3-9所示，本问题只存在一条最优路径，本文期望使用了新机制的强化学习算法能够快速学习到最优路径。

3.6.1.2 实验结果分析

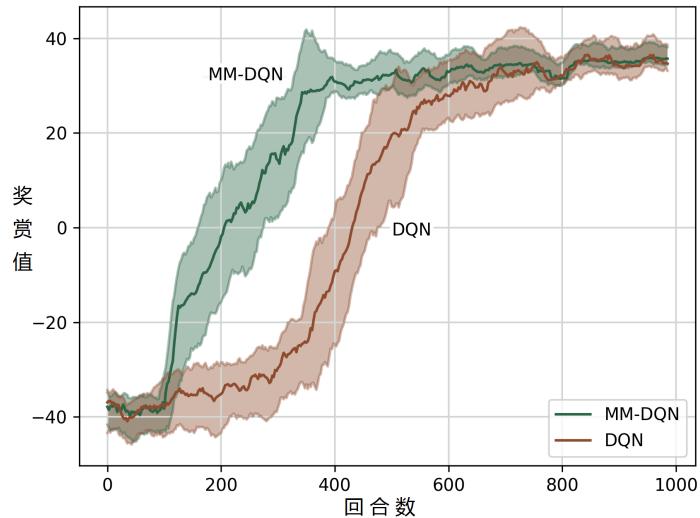


图 3-10 水牢迷宫算法结果对比图

本实验主要对比了DQN算法与MM-DQN算法在学习过程中的学习效率以及算法性能表型。智能体的行为目标是最大化奖赏值，但从安全性角度出发，即使是在学习期间，如果能更快地学习到避开危险区域的方法将是十分有益的。这意味着智能体将在远离水的同时运行到目标。由于DQN算法只使用视觉模态信息，无法高效地感知到环境中的危险，而MM-DQN算法能够借助辅智能体的声音模态信

息，更全面的了解到环境信息，望提升智能体在学习过程中的安全性。

图3-10展示了MM-DQN与DQN算法的效果对比，相比DQN算法，MM-DQN取得了更快的训练速度以及更高的奖赏值。图中也可以观察到，纵使DQN算法最后能学习到最优路径，但是需要花费较长的时间，而MM-DQN只需要较少的训练时间。本文推论这得益于新机制对DQN算法带来了性能提升。

水牢迷宫游戏中，视觉感知信息其实包含了足够学习到最优路径的信息，这是由于水方格的颜色和其它方格颜色不同，虽然学习稍慢，但是确实也能学习到最优策略。因此下一个实验中，本文将考察模态中存在信息确实的问题来考察新机制的效用。

3.6.2 不充分信息的多模态问题

3.6.2.1 环境介绍

本实验考虑迷宫逃生问题，属于信息不充分的多模态问题。相比之前的环境，本环境中单模态信息不足以完成任务，因此需要结合多模态的信息实现多智能体间的协作完成任务。具体来说（见图3-11），考察一个 $N \times N$ 方格内的迷宫问题。每一个回合开始智能体M出现在黄色的区域，并且需要使用正确的把钥匙（黄色钥匙）打开房间的门（绿色格子）。智能体M可以沿着上下左右四个方向运行，图

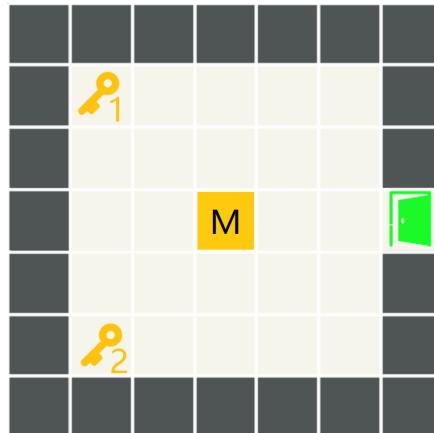


图 3-11 迷宫逃生问题

中浅色是智能体可以行走的格子。深灰色的格子代表墙壁，朝着墙壁方向的运动将被阻止，智能体将保持原地不动。智能体的每步运动将收到-1的奖励。游戏在智能体到达门时停止，若拿到正确的钥匙开门成功则获得+40的奖励，否则奖励

为0。如果在40步之内智能体都没有到达终点，则游戏强行结束。本问题的难点在于房间内有两把钥匙，但每一个回合只有一把能正确打开房门。而两把钥匙从视觉感知上是无法区分的，致使仅使用视觉模态信息是无法完成此任务。同时，当智能体M接近正确的钥匙时（1格距离），会听到特殊的声音模态信息，而其他时候收到的是噪声。

3.6.2.2 实验结果分析

这个环境中，智能体M无法使用单一视觉模态信息有效地完成任务，而需要与声音模态信息相结合，综合考虑多模态信息来有效地完成任务。

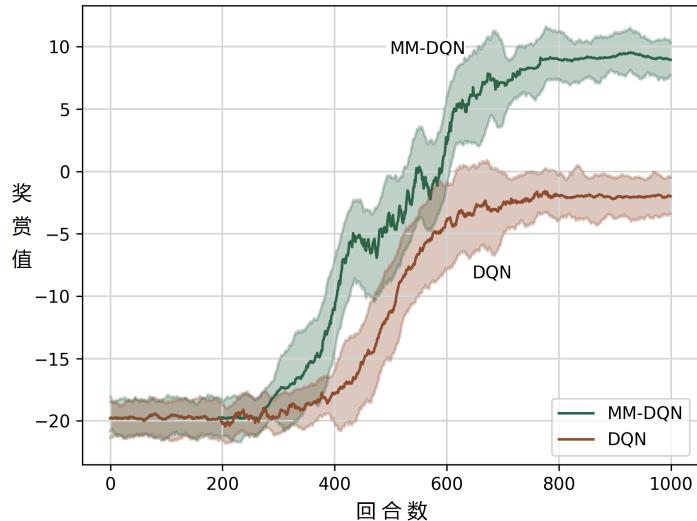


图 3-12 迷宫逃生实验算法结果对比图

图3-12展示了DQN算法与MM-DQN算法在迷宫逃生问题上的学习效果与算法表现，我们看到DQN算法学习较快，但是算法性能并没有持续上升，这是由于DQN算法只使用了视觉模态信息，虽然学习到了需要先捡起钥匙再开门这一知识来完成任务，但是由于视觉模态信息无法提供应该捡哪一把钥匙这一关键信息，因此DQN算法最后只有50%的概率能够正确完成任务。反观MM-DQN算法，其训练初期训练速度相对较慢，这是由于MM-DQN同时需要考虑了视觉与声音两种模态信息，导致增加了训练的难度。但随着训练的进行，智能体M逐渐开始利用其感知到的声音模态信息，有选择性的选择正确的钥匙，最后，完全解决了迷宫逃生问题。

上述实验说明仅使用单模态信息有时无法很好的完成任务，特别是单模态信息存在信息缺失的时候。此时需要使用多模态的信息融合，实现信息互补来协同完成任务。实验结果表明MM-DQN算法相比DQN算法能够有效的利用声音模态信息，弥补信息缺失问题，学习到完成任务所需要的知识（选择正确的钥匙）。该实验结果也证明，使用了新机制的MM-DQN算法能够有效地利用多模态信息，促进策略优化并寻找到最优策略。

3.6.3 多模态自动驾驶问题

3.6.3.1 环境介绍

本小节使用自动驾驶问题来验证算法处理多模态数据的有效性。本实验使用的是TORCS，一个开源的赛车模拟器环境。目前，TORCS已经逐步发展成为流行的自动驾驶的基准环境。在这个环境中，智能体将作为赛车手控制高性能的赛车进行竞速赛。智能体学习的目标是在避免碰撞的同时，尽可能快速的完成赛道。下图3-13展示了TORCS环境的部分鸟瞰图。



图 3-13 TORCS自动驾驶环境

本使用对TORCS环境进行了改造，改造后TORCS以100ms为周期向外部接口提供数据。智能体通过外部接口完成环境状态的读取以及动作的实施。TORCS提供了两种类型的状态数据：图像数据和传感器数据。图像数据是一张84*84的图片，展示的是赛车运行过程中，驾驶员所观察到的车前路况。传感器数据提供了赛车的各项传感器指标，包含了车辆速度、车辆与路中心的距离、发动机转速、轮胎转速、车辆前方180度激光雷达测量距离等。

本环境中智能体主要通过感知的视觉模态信息和听觉模态信息进行车辆的安全自动驾驶。具体来说，智能体每个时刻都能收到视觉感知输入，而只有在车辆将要发生车祸的时候才会收到警报信号（听觉感知输入），其余时刻都是收到的噪声。智能体通过一个由三个实数值组成的三维向量来控制车辆的驾驶，三个实数分别代表方向、油门以及刹车，其中方向的取值范围是[-1.0,1.0]，油门和刹车的取值范围是[0,1.0]。

3.6.3.2 奖赏值函数

在赛车竞速类游戏中，游戏的目标是通过合理有效的操作尽可能提高赛车完成每一圈的速度。在本例中，奖赏值函数定义如下：

$$R_t = V_x * \cos(\theta) - V_x * \sin(\theta) - V_x * |D_f|$$

其中 V_x 是赛车的车头方向的速度， θ 是汽车车头方向与赛道之间的夹角， D_f 是赛车重心距离路中心之间的距离。如图3-14所示，智能体的奖励函数鼓励汽车沿着赛道获取最大速度，惩罚其不在赛道方向的行驶以及偏离赛道的行为。

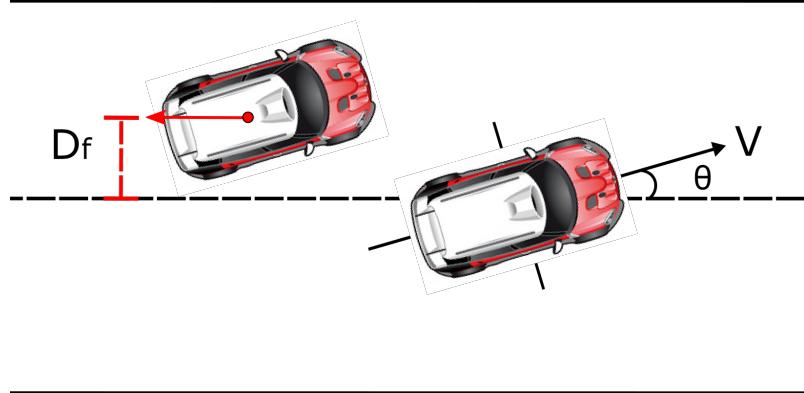


图 3-14 行车间奖赏值函数定义图

3.6.3.3 实验设置

在TORCS环境中，智能体主要使用视觉模态信息进行车辆行驶策略的学习。一般来说，智能体能够基于视觉信息学习出一定的转弯策略，但视觉感知输入的观察信息毕竟有限，无法保证智能体学习到完全准确地转弯策略。因此，在驾驶过程中，智能体还需要合理利用传感器数据，有效地提升策略优化效果。

考虑到现实驾驶过程中，车上的传感器会在车距离道路两侧过近时发出危险

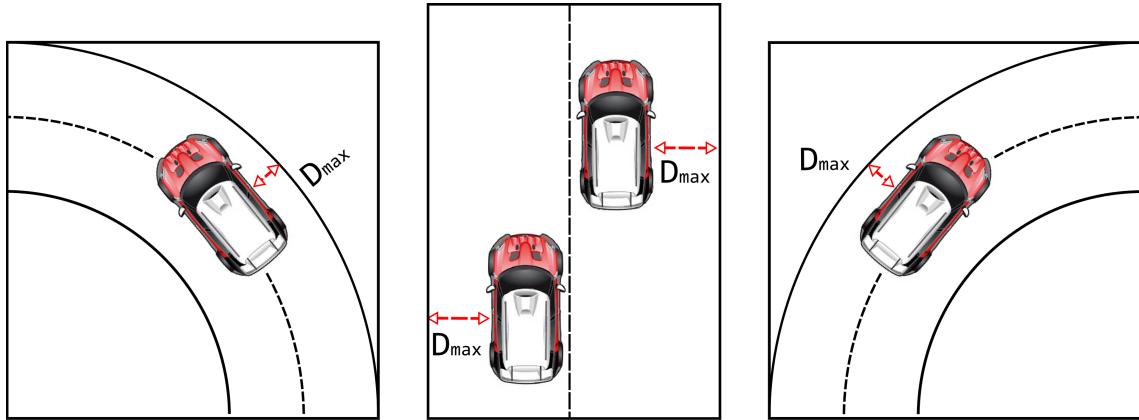


图 3-15 车辆传感器预警示意图

信号的提示。本实验相同的设置，当TORCS环境中返回的距离传感器数据 d （车辆与两边的路距）小于预先设置好的安全阈值 D_{max} 时候，智能体会收到警报信息。具体如图3-15所示，无论在直行车道，或者是转向道中，如果 $d < D_{max}$ ，则认为当前车辆距离道路边缘过近，当前车辆的状态不安全。智能体在车辆处于不安全的状态时，会收到危险的声音模态信息，其它情况下，收到的是噪声信息。

3.6.3.4 实验结果分析

由于该问题是连续空间决策问题，因此本实验使用深度确定性梯度下降算法（DDPG）和异步的行动者-评论家（A3C）算法作为基准算法。通过将MM-DDPG算法和MM-A3C算法与DDPG、A3C基础算法进行对比，可以对新机制的效果进行量化的检验。后续实验主要从自动驾驶任务中智能体获得的长期收益作为评估指标。

图3-16展示了MM-DDPG与DDPG的性能对比效果图（本实验使用10个随机种子进行）。横坐标代表智能体训练经过的帧数，纵坐标表示其获得的奖励。深色线条表示奖励的均值，浅色范围表示奖励的标准差。从图中可以得出，MM-DDPG相比于DDPG获得了更好地奖励（均值更高，标准差更低）。MM-DDPG在学习速度上也体现出来更高的水准，在2500000帧左右次训练时已经获得了大约50000的平均奖励，而相同训练规模下DDPG仅达到大约10000的水平。MM-A3C与A3C的对比实验也论证了类似的结果，使用了新机制的MM-A3C算法学习效率更高，获得奖赏值也更大。

综上所述，从实验结果上看，本章针对多模态信息输入提出的新机制能够有

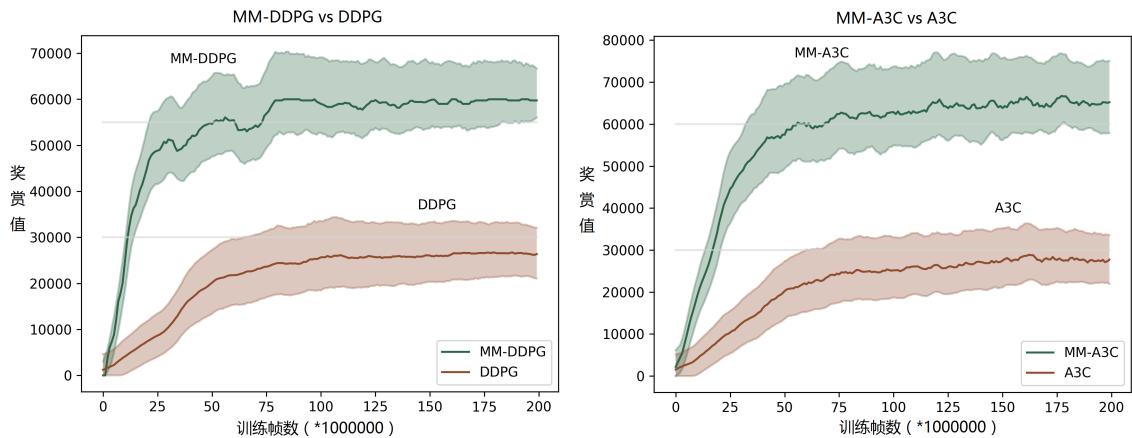


图 3-16 MM-DDPG、MM-A3C与DDPG、A3C在多模态自动驾驶任务上的实验结果对比

效的提升传统强化学习的性能和学习效率。第一个水牢迷宫实验结果论证了，当单模态信息足够描述整个环境的时候，传统强化学习算法也是能够寻找到最优策略的，但在优化效率上不如使用了多模态信息的强化学习算法。第二个迷宫逃生游戏中，由于单模态信息并不足以描述环境特征，因此传统强化学习算法无法找到最优策略，有效利用多模态信息可以促进智能体进行策略优化，寻找到最优策略。最后一个自动驾驶实验，更是论证了新机制与DDPG和A3C等算法的结合，能够更加全面准确的感知到当前环境，做出更优的决策，以提高长期利益最大化。

3.7 本章小结

本章主要提出了基于分离式多模态输入的强化学习框架、层次注意力机制以及基于多信息流的LSTM网络。分离式多模态输入的强化学习框架借助多模态学习的经验，能够有效地处理多模态信息的输入，拓展了强化学习处理多模态信息的能力。层次注意力机制不仅实现了模态间的注意力权重分配，也实现了模态内关键信息的注意力分配，实现了高效的特征提取。此外，基于多信息流的LSTM网络拓展了LSTM网络处理多模态信息输入的能力。

实验证明，基于新机制的多模态信息的强化学习算法（MM-DQN、MM-DDPG、MM-A3C）都相比其对应的原始算法（DQN、DDPG、A3C）取得了更快的策略优化速度，以及更高的算法性能。

综上，本章针对面向多模态信息输入的智能体策略优化问题展开研究，提出的三种新机制，有效提升一般强化学习算法处理多模态信息输入的能力，实现高

效的策略优化。由于感知信息对于智能体决策有着决定性的影响，因此，本章针对多模态感知信息的研究，是后续深度强化学习以及多智能体策略优化研究的基础，针对后续研究的展开起到了过渡作用。

第4章 面向噪声环境下独立学习智能体的策略优化研究

本章主要研究面向噪声环境下独立学习智能体的策略优化问题。从环境、算法以及多智能体系统层面，详细分析了实现多智能体协同策略优化所面临的问题与挑战。针对既有算法存在的缺陷进行了详细的分析，并提出了新算法来解决这些缺陷。详细解释了新算法能够解决有效解决此类问题的原因，并通过实验论证了新算法的有效性。

4.1 引言

独立学习智能体由于定义相对简单，在策略更新过程中不受限于环境与其它智能体，更具有普遍性，能运用于大部分真实问题中。因此，针对基于独立学习智能体间的多智能体策略优化问题的研究具有十分重要的理论及实践意义。

近年来，随着深度强化学习的发展，研究者们开始尝试使用深度强化学习技术解决独立学习智能体的策略优化的研究问题^[29,77,78]。然而，基于独立学习智能体的策略优化研究依然面临着潜在的难题与挑战。其中一个难题是来自环境层面，即环境的反馈奖赏值可能存在噪声，这将影响智能体的策略优化过程。另一个难题来自于强化学习算法本身存在估值偏差问题以及策略优化的效率问题，这将极大地影响策略优化的效果。最后一个挑战则来自于多智能体系统本身，即由于独立学习智能体把环境中共存的其余智能体当作环境的一部分，因此其余智能体的动态行为产生的随机性可能影响智能体策略优化过程，甚至产生不是策略优化不收敛的问题。

本章节主要研究在带噪声的合作式多智能体环境下，独立学习智能体的策略协同优化问题。主要针对上述强化学习算法本身存在值偏差问题、环境中存在不稳定噪声的问题、独立学习的智能体之间的协同策略优化问题以及经验回放机制导致策略优化效率下降的问题展开了研究，并提出了相应的解决方法。

总体来说，本文提出了基于双权深度Q网络（Weighted Double Deep Q-Network，WDDQN）算法来有效地促进独立学习智能体在带噪声的环境下有效

的进行策略协同优化，并促进智能体收敛到纳什均衡，乃至帕累托最优纳什均衡。概括来说，WDDQN通过使用两个DQN网络的线性组实现有效的估值纠偏；此外，WDDQN使用奖赏值网络（Reward Network, RN）解决环境中存在噪声的问题；同时，WDDQN使用宽容的奖赏值网络（Lenient Reward Network, LRN）促进独立学习体之间的有效合作。最后，WDDQN使用调度经验重放策略（Scheduled Replay Strategy, SRS）有效地实现重要样本的优先级采样，增加了网络训练的效率，促进智能体策略优化效率。本章最后使用了带噪声和不带噪声的实验、带有确定性奖赏和随机奖赏等多种实验对WDDQN算法、LRN机制以及SRS机制的有效性分别进行了论证，并在每个实验的结尾，对实验结果进行了详细的分析。

4.2 基于双权估计的多智能体策略优化算法框架

图4-1描述了WDDQN算法的总体框架。WDDQN算法包括了如下四大关键模块，分别是双权深度Q网络（Weighted Double Deep Q-Networks）、奖赏值网络Reward Network（RN）、宽容的奖赏网络（Lenient Reward Network, LRN）、调度经验重放策略（Scheduled Replay Strategy, SRS）。这四者的关系如下：

(1) 双权深度Q网络(Weighted Double Q-networks, WDDQN)

为了解决前文所述的强化学习算中存在估值偏差的问题，本文基于双权估计器机制^[26]，结合深度神经网络，提出了带双权的深度Q网络算法（WDDQN），有效地估值偏差修复，以解决基于值函数估计的强化学习算法中Q值估计不准确的问题。具体来说，WDDQN算法通过使用双权估计器机制，理论上能够有效地权衡双估计器产生的高估与低估，以实现估值修复；此外借助深度神经网络，WDDQN能够有效的解决含有大规模状态空间的复杂问题。详细的研究内容在4.3节中展开。

(2) 奖赏网络(Reward Network, RN)

为了解决前文所述的环境的反馈信息中存在噪声干扰的问题，本文提出使用奖赏网络(Reward Network, LRN)对环境返回的奖赏值建模，通过加权平均的方法降低噪音，提高训练稳定性和性能，提升算法学习效率与鲁棒性。详细的研究内容在4.4节中展开。

(3) 宽容的奖赏网络(Lenient Reward Network, LRN)

为了解决前文所述的独立学习智能体的策略协同优化问题，促使独立学习智

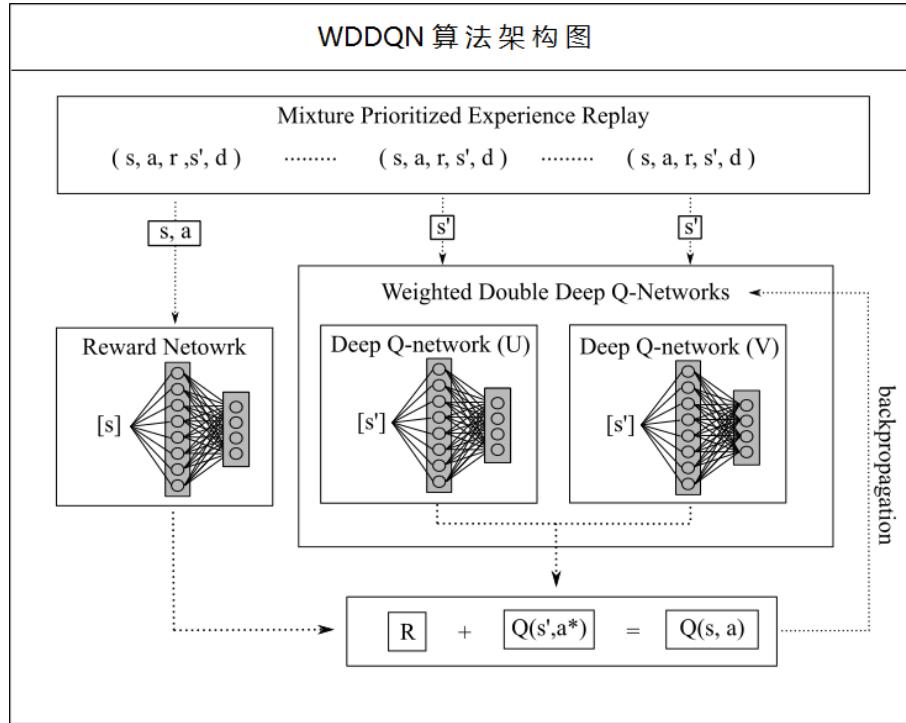


图 4-1 双权深度Q网络（WDDQN）算法的总体框架图

能体有效的进行策略协同优化，收敛到纳什均衡乃至帕累托最优纳什均衡。本文提出了宽容的奖赏网络(Lenient Reward Network, LRN)，将宽容机制^[81]与奖赏网络结合，促进独立学习智能体之间实现高效的配合，提高智能体之间的策略收敛效率，有效地促进策略收敛到帕累托最优纳什均衡。详细的研究内容在4.4节中展开。

(4) 调度经验重放策略(Scheduled Replay Strategy, SRS)

为了解决多智能体环境下优先级经验回放机制可能存在策略优化效率的问题，本文提出了调度经验重放策略（Scheduled Replay Strategy, SRS）来提升策略优化的效率与算法性能。SRS策略在抽样样本进行网络训练时，通过修正不同样本的抽样概率，有效地抽样重要的样本进行训练，实现算法效率的提升，并且促进智能体策略收敛到帕累托最优纳什均衡策略。此外，作为SRS的底层实现，本文提出了混合优先级采样池（Mixture Prioritized Experience Replay）以解决重要样本优先采样的问题，通过使用双经验池针对样本进行区别存储，结合SRS采样策略保证重要样本优先级的分配制度，确保重要样本优先采样，提升策略优化的效率。详细的研究内容在4.5节中展开。

算法4: WDDQN算法

Input: 回合数: E , 最大步数: S , 全局/单局经验池: D^G, D^E , 奖赏值

网络: R^N , 深度Q网络: Q^U, Q^V

```

1 for  $episode = 1$  to  $E$  do
2   初始化经验池  $D^E$ 
3   for  $step = 1$  to  $S$  do
4      $a \leftarrow \max_{a'} \frac{Q^U(s, a') + Q^V(s, a')}{2}$  (with  $\varepsilon$ -greedy)
5     执行 $a$ 并将样本存储到 $D^E$ 中
6     从 $D^G$ 采样( $s, a, r, s'$ )
7     随机选取 $Q^U$ 和 $Q^V$ 之一进行更新
8     if 如果选择更新 $Q^U$  then
9        $a^* \leftarrow \arg \max_a Q^U(s', a)$ 
10       $Q_U^w(s', a^*) \leftarrow \beta Q^U(s', a^*) + (1 - \beta) Q^V(s', a^*)$ 
11       $Q^{\text{Target}}(s, a) \leftarrow R^N(s, a) + Q_U^w(s', a^*)$ 
12      使用 $Q^{\text{Target}}$ 来更新 $Q^U$ 网络参数
13    else
14       $a^* \leftarrow \arg \max_a Q^V(s', a)$ 
15       $Q_V^w(s', a^*) \leftarrow \beta Q^V(s', a^*) + (1 - \beta) Q^U(s', a^*)$ 
16       $Q^{\text{Target}}(s, a) \leftarrow R^N(s, a) + Q_V^w(s', a^*)$ 
17      使用 $Q^{\text{Target}}$ 来更新 $Q^V$ 网络参数
18    根据 $D^G$ 中的样本更新 $R^N$ 
19  将 $D^E$ 存储在 $D^G$ 中

```

算法4描述了WDDQN算法的整理逻辑流程图。WDDQN算法的输入包括训练局数 E , 每局最大步数 S , 用于存放样本的全局经验池 D^G 和 D^E , 奖赏值网络 R^N , 以及基于神经网络实现的双估计器 Q^U 和 Q^V 。WDDQN整体算法流程基于DQN算法, 总共训练 E 局, 并且每局最多执行 S 步。算法一开始首先初始化经验池 D^E (第2行)。每一次决策, 智能体使用双估计器 Q^U 和 Q^V 一同选择最优动作 $a = \max_{a'} \frac{Q^U(s, a') + Q^V(s, a')}{2}$ (第4行)。接着智能体执行动作后, 将观察到的环境返回存储到经验池 D^E 中 (第5行)。随后智能体从全局经验池 D^G 中采样样本进行网络的训练 (第6行), 采样后随机选择双估计器中的一个选择最优动作, 而另一个则用

于估值，二者协同进行网络更新（第8-17行）。注意到第10行和第15行使用的是基于双权估计器来更新Q值，如此一来能够有效降低估值偏差（详见4.3节）。同时我们使用奖赏网络对环境反馈的奖励进行拟合，以此来降低奖赏值中的噪音（详见4.4节）。通过将宽容机制与奖赏网络结合，使智能体间实现有效的协作（详见4.4节）。最后通过改进优先级经验回放策略，提出预定回放策略实现提高重要样本的抽样概率，提升算法训练效率（详见4.5节）。

4.3 双权深度Q网络

4.3.1 估值偏差的机理分析

基于值的强化学习算法的一个基本思想是基于时差分的动态规划思想进行值更新^[21]。该思想通过迭代式更新策略来进行Q值的更新，但此类更新方式会产生估值不准的问题，进而影响强化学习算法的表现性能。这里主要讨论在Q值的中使用单估计器所带来的估值偏高以及使用双估计器带来的估值偏低问题进行讨论，并提出采用加权的双估计器来进行估值修复。

首先论述单估计器中的高估问题。Q-learning是一种典型的使用单估计器的算法，其Q值更新策略如下：

$$Q(s, a) \leftarrow Q(s, a) + \alpha[r + \gamma \max_{a'} Q(s', a') - Q(s, a)] \quad (4-1)$$

其中 r 是环境的反馈信息， α 代表更新率，一般介于0到1之间。上述公式中，使用了 $\max_{a'} Q(s', a')$ 来选择最优动作 a' ，并用于更新 $Q(s, a)$ 值。根据Smith的研究成果，将单个估计器同时用于Q值的估计与更新，会导致最大Q值的期望大于真实的最优Q值^[99]，具体如下公式所示：

$$E\{\max_{a'} Q(s', a')\} \geq \max_{a'} E\{Q(s', a')\} \quad (4-2)$$

由于理论上应该用 $\max_{a'} E\{Q(s', a')\}$ 的值来更新Q值，但是实际上使用的却是 $E\{\max_{a'} Q(s', a')\}$ 这导致了Q-learning算法在更新Q值的时候，使用了过大的Q值，产生了高估问题，进而影响算法效果。

接着探讨Q值低估的问题，在double Q-learning算法中，使用了两个估计器来缓解Q-learning算法中的高估问题。具体来说，它使用了基于 θ 和 θ' 参数化的两个估计器 Q^U 和 Q^V 来分别估计 $Q(s, a; \theta)$ 和 $Q(s, a; \theta')$ 。同时在更新Q值的过程中，将公式4-1中的 $\max_{a'} Q(s', a')$ 部分替换为 Q^U 和 Q^V 两个估计器的组

合 $Q^U(s', \arg \max_{a'} Q^V(s', a'))$ 。具体来说，

其中一个估计器根据其估计的Q值，选择出最优动作 a ，接着在更新的时候使用的却是另一个估计器估计的Q值作为更新过程中的目标值。例如每次 $Q(s, a, \theta)$ 在更新的时候，使用的目标值计算如下所示：

$$Y_t^Q \equiv R_{t+1} + \gamma Q(s_{t+1}, \arg \max_a Q(s_{t+1}, a, \theta_t); \theta'_t) \quad (4-3)$$

如此一来能够有效缓解单估计器产生的高估问题，并且这种双估计器的设计理念也很容易实现在深度强化学习的场景下(在深度强化学习的实现中，Q值的估计往往使用神经网络来实现)。但不幸的是，根据Hasselt^[46]的研究，使用双估计器虽然能够环节Q值高估的问题，但同时引入的低估的问题，低估的原因如下公式：

$$E\{\max_{a'} Q(s', a')\} \leq \max_{a'} E\{Q(s', a')\} \quad (4-4)$$

当使用双估计器的时候，会产生用于Q值更新的 $E\{\max_{a'} Q(s', a')\}$ 小于理论实际值 $\max_{a'} E\{Q(s', a')\}$ 的情况出现，此时就会导致更新后Q值低估的情况出现。总体来看，虽然基于双估计器的算法也存在低估的问题，但是相比于使用单估计器带来的高估问题，双估计器一般能够取得较好的效果。例如double Q-learning以及double Q-networks的算法效果都要优于Q-learning以及deep Q-networks。

4.3.2 基于双权估计器的估值纠偏

仅使用单估计器或者双估计器始终会带来估值过高或者估值过低的难题，为了解决这类问题，需要设计一种能够算法在高估和低估之间进行自适应的权衡。zhang等^[26]提出了一种基于双权估计器的Q-learning算法(weighted double Q-learning)进行估值纠偏。具体来说，单估计器存在高估问题，而双估计器存在低估问题，因此无偏估计的真实值一定存在着二者之间。基于这种设想，研究者提出了使用超参数 β 对可能高估的单估计器估值 Q^V 和可能存在低估的双估计器估值 Q^U 进行权衡，加权公式如下：

$$Q(s, a)^{U, WDO} = \beta Q^U(s, a^*) + (1 - \beta) Q^V(s, a^*) \quad (4-5)$$

这里使用了一个基于 Q^U 和 Q^V 的线性组合进行Q值估计。注意到 a^* 是根据 Q^U 的估值选取出来的最优动作(例如 $a^* \in \arg \max_a Q^U(s, a)$)，因此如果用 Q^U 来估计 a^* 的估值会产生 $Q^U(s, a^*)$ 的高估，而用 Q^V 进行估值会产生 $Q^V(s, a^*)$ 的低估。同样如果 a^* 是根据 Q^V 的估值选取出来的，则情况相反。公式4-5显示，因为 $\beta \in [0, 1]$ ，所以估值 $Q(s, a)^{U, WDO}$ 介于 Q^U 和 Q^V 之间，基于这种加权双估计器的机制，再使用权重 β ，

能够有效地平衡估值偏高与偏低的问题。此外此参数 β 能够实现动态自适应的调节，根据样本本身的特性，实现实时调节。

注意到基于双权估计器的机制在传统的基于表格的强化学习问题中被提出，并且一定程度上能够进行估值纠偏，但是其有效性在复杂的深度强化学习场景下，特别是基于图片输入这种巨大的状态空间问题，仍然需要进一步的研究论证。

通过结合深度神经网络，本文将双权估计器的思想推广到深度环境下用于解决状态空间无穷的深度强化学习问题。具体来说，如算法4-5所示，本文使用深度神经网络作为Q值估计器，为了降低估值偏差，WDDQN使用了两个神经网络估计器的线性组合(第10行与第15行)。 $\beta Q^U(s', a^*) + (1 - \beta)Q^V(s', a^*)$ 这种线性组合通过使用超参数 β 实现高低估之间的权衡，降低估值偏差。同时用于网络更新的 $Q^{\text{Target}}(s, a)$ 也是用基于双权的估计器计算得出(第11行与第16行)

此外需要注意的是，相比传统DQN算法或者double DQN算法，最优动作 a^* 的选择有所变化，在WDDQN中，最优动作 $a^* = \max_{a'} \frac{Q^U(s, a') + Q^V(s, a')}{2}$ (算法4-5, 第9、14行)。

4.4 奖赏值网络与宽容机制

本节主要论述针对环境返回的带噪声的奖赏值的解决方法。由于环境中存在不确定的噪声，导致智能体接收到的奖赏值不一定是准确的，如果用不准确的奖赏值来更新策略网络，则会导致策略网络往错误的方向更新，直接导致策略性能的下降。

为了尽可能降低环境中的噪声，本文提出了奖赏值网络(Reward Network, RN)来拟合环境反馈的奖赏值函数。这里假定环境中的噪声符合高斯分布 $N(0, 1)$ ，针对该噪声，只要采样的样本数量足够大，样本的算术平均是真是平均的无偏估计。因此如果能够拥有足够的奖赏值样本，只需要对奖赏值求均值，就能够过滤掉奖赏值中的高斯分布噪声。基于这种思想，本文尝试通过神经网络来拟合奖赏值函数，而为了尽可能降低奖赏值中的噪声，本文使用大量的奖赏值样本均值来训练奖赏值网络。如此一来，奖赏值网络就能够尽可能的剔除环境中的噪音干扰，拟合出环境中真是的奖赏值函数。

4.4.1 奖赏值网络

(1) 奖赏值函数拟合

具体来说，Q-learning算法在使用时查分思想进行Q值更新的时候(4-1), 使用的是环境返回的即使收益 $r(s, a)$, 但由于此时的存在高斯噪声, 则环境返回的奖赏值是带偏差的, 这里称为 $\hat{r}(s, a)$ 。这里我们考察带高斯噪声的随机环境, 则有如下公式:

$$\hat{r}(s, a) = r(s, a) + \mathcal{N}(0, 1) \quad (4-6)$$

其中 $\mathcal{N}(0, 1)$ 是满足标准高斯分布的噪声。由于存在噪声, 因此使用 \hat{r} 来更新Q值会导致更新过程不准确, 进而影响策略优化。为了解决此问题, 本文提出使用奖赏值网络来拟合 $r(s, a)$, 使用奖赏值网络能够有效地降低单次采样中存在不稳定噪声产生的估值偏差, 同时能够使网路训练更加高效平稳。WDDQN算法4 (第11行) 在使用 Q^{Target} 进行DQN网路训练的时候, 本文提出使用奖赏值网络的的估值来替代环境返回的即时奖赏值, 具体公式如下:

$$Q^{\text{Target}}(s, a) = R^N(s, a) + Q_U^w(s', a^*) \quad (4-7)$$

注意到这里和算法4 (第16行) 是类似的, 虽然双权估计器的时候分别使用了 Q_U^w 和 Q_V^w 两种估计器来进行更新, 但无论使用哪个估值, 使用的都是估值网络估计的奖赏值 $R^N(s, a)$ 来替代即使收益 $r(s, a)$ 。

(2) 奖赏值网络的更新

本节考察奖赏值网络RN的更新过程, 为了降低环境反馈的奖赏值中的噪声, 需要有大量的样本来更新策略网络RN。如算法4 (第19行) 所示, 全局经验池中 D^G 中的样本通过均匀采样的方式被抽样出来用于奖赏值网络RN的更新。具体来说, 这里采样出来的样本是由 (s, a, \hat{r}) 组成, 为了将同样 (s, a) 的 \hat{r} 分类在一起进行有效的样本均值计算, 需要一种有效的编码方式针对无穷的状态空间 s 进行编码, 本文提出使用微分自编码器 (Autoencoder)^[100]对状态空间 s 进行编码操作, 先将高维的状态空间压缩到低维空间。接着本文使用哈希编码 (xxhash) 对低维的编码空间进行哈希映射得到唯一编码, 如此一来就能够有效地将高位空间 s 聚类在一起进行样本均值计算。每一个 s 的唯一样本编码计算公式如下:

$$\text{xxhash}(\text{Encoder}(s)) \quad (4-8)$$

这里Encoder是使用Autoencoder对 s 进行编码操作, 接着使用xxhash进行哈希操作。如此一来, 针对拥有相同编码值的样本中带噪声的奖赏值进行均值计算, 计算公

式如下：

$$r^{\text{Target}}(s, a) = \sum \hat{r}_i(s, a) \quad (4-9)$$

这里 $\hat{r}_i(s, a)$ 是拥有相同编码值(xxhash(Encoder(s)))的带噪声的奖赏值。在样本足够充分的前提下，通过算数平均后的奖赏值 $r^{\text{Target}}(s, a)$ 中的噪声，能够得到有效地减少，如下公式所示：

$$E\{\hat{r}_i(s, a)\} = E\{r_i(s, a) + N(0, 1)\} = E\{r_i(s, a)\} + E\{N(0, 1)\} = E\{r_i(s, a)\} \quad (4-10)$$

在样本足够的情况下，奖赏值中的噪声能够有效地通过算数平均操作进行降低。如算法4(第19行)所示，使用降噪处理后的奖赏值 r^{Target} 进行奖赏值网络的训练，能够使得奖赏值网络输出的奖赏值接近环境的真是奖赏值，这样能够提高智能体策略优化的效率以及提升智能体策略的效果。

4.4.2 宽容的奖赏值网络

本节主要论述使用宽容机制结合奖赏值网络以促进智能体间实现有效地合作。一般来说，在多智能体环境中，智能体之间需要相互配合以实现群体利益最大化，由于多个智能体同时更新自己的策略，使得多智能体共同学习到合适策略变得困难。本文着重考察的是独立学习的多智能体问题，在这种环境下，智能体在进行策略优化的过程中，会把其余的智能体当做是环境中的一部分。在这类环境中，要求智能体独立学习，且不能使用观察到的其余智能体的任何信息，这使得多智能体的策略优化变得更加困难。

具体来说，除了前文所提到的环境中产生的 $\hat{r}(s, a)$ 存在噪音会干扰到策略优化的过程外，在一个合作式的独立学习多智能体系统中，其余智能体行为的动态性也会使得 $r(s, a)$ 中产生偏差。举例来说，假定多个智能体已经学习到了最优策略后，但由于一个智能体改变了其行为，会导致其余智能体尽管采取了最优策略，但收获的环境奖赏值却不是最优的。这时智能体接收到的 $r(s, a^*)$ 实际上是小于真实的 $r(s, a^*)$ 的。为了解决这个难题，本文借助了宽容的概念^[79]，并提出了宽容的奖赏值网络(Lenient Reward Network, LRN)。这种宽容的机制使得智能体在初期探索阶段能够保持乐观，也就是说针对Q值的负更新采取谨慎的策略。但伴随着智能体探索的更加充分，智能体对Q值的更新逐渐回归理性。如此，能够提高的促进多个智能体协作的似然率，使多智能体同步收敛到使群体利益最大化的策略。

(1) 宽容的Q学习算法 (Lenient Q-learning)

宽容(Leniency)的概念是由Potter提出，并与Q-learning相结合，提出了宽容的Q学习算法(Lenient Q-learning)。Lenient Q-learning算法的目的在于能够帮助多个独立学习的智能体在其优化策略的过程中，协同多个智能体同时朝着使群体利益最大化的方向更新各自的策略。简单来说，Lenient Q-learning是通过让多个智能体在策略优化初期，以一定的概率拒绝对Q值的负值更新，而对正值更新采取始终接收的策略。这种方式被证明能够有效地促进多智能体共同朝着使得群体利益最大化的方向去更新策略，同时能够避免智能体策略优化陷入到局部最优中，增加收敛到最优策略的似然性^[80-83]。

Lenient Q-learning算法的宽容机制具体原理如下，在训练过程中的 t 时刻，智能体对于每一对 (s, a) 都会记录一个温度函数 $T_t(s, a)$ 。这个温度函数在智能体开始探索之前都被设置成一个预定义好的最大值（超参数）。温度函数的目的在于衡量智能体对于 (s, a) 的宽容程度 $l(s, a)$ ，具体定义如下公式：

$$l(s_t, a_t) = 1 - e^{-K * T_t(s_t, a_t)} \quad (4-11)$$

上述公式中， K 是一个常数，决定了温度 T 对宽容程度 l 的衰减程度。在Wei^[83]研究中，温度 $T_t(s_t, a_t)$ 是逐渐衰减的，衰减影子表示为 $\kappa \in [0, 1]$ ，衰减公式如下：

$$T_{t+1}(s_t, a_t) = \kappa T_t(s_t, a_t) \quad (4-12)$$

由此表明，随着智能体不断在状态 s 中执行 a 动作，温度 $T(s, a)$ 是不断衰减的过程。此外在更新Q值的过程中，当给定更新误差 $\delta = Y_t^Q - Q_t(s_t, a_t; \theta_t)$ 时候，Lenient Q-learning针对Q值的更新公式具体如下：

$$Q(s_t, a_t) = \begin{cases} Q(s_t, a_t) + \alpha \delta & \text{如果 } \delta > 0 \text{ 或者 } x > l(s_t, a_t) \\ Q(s_t, a_t) & \text{否则} \end{cases} \quad (4-13)$$

上述更新公式中， $x \sim U(0, 1)$ 是一个均匀分布的随机变量，用于确保当针对Q值的更新是一个负值更新的时候($\delta < 0$)，只有 $1 - l(s_t, a_t)$ 的概率会执行更新操作。由于在初始阶段，每一个 (s, a) 访问次数很少，因此Lenient Q-learning会以很高的概率拒绝执行负值更新，让智能体对Q值保持一个相对乐观的状态。但是伴随着 (s, a) 访问次数的增加，智能体开始回归理性，逐渐接受负值更新。这种让智能体在探索初期保持相对乐观，再逐步回归理性的宽容机制，能够有效的促进智能体之间的策略优化朝着群体利益最大化的方向优化。

需要注意的是，对于一个多智能体系统问题，智能体初始化的位置会被频繁的访问到，这就导致了智能体初始化的位置的温度值可能会衰减的特别快。产生这个问题的原因是值考虑了当前状态 s 而忽略了时序上的可能存在的状态间以

来关系。因此解决该问题的一个自然而有效的方法是，在更新当前温度值的时候 $T(s, a)$ ，将下一个状态 s' 的温度值一起考虑。

首先我们考虑状态 s' 的平均温度 $\bar{T}(s')$ ，计算公式如下：

$$\bar{T}(s') = \frac{1}{|A|} \sum_{a_i \in A} T(s', a_i) \quad (4-14)$$

该指标考虑了状态 s' 下的所有可能的动作，并且进行平均温度求和。该指标能够有效的反应当前状态的一个平均访问情况。凭借对状态 s' 平均温度的衡量指标，就能够对状态 s 的温度进行计算的时候，同时考虑临近状态 s' 的平均温度^[83]，计算公式如下：

$$T_{t+1}(s_t, a_t) = \kappa \times \begin{cases} T_t(s_t, a_t) & \text{如果 } s' \text{ 终止状态} \\ (1 - \gamma) \times T_t(s_t, a_t) + \gamma \bar{T}_t(s') & \text{非终止状态} \end{cases} \quad (4-15)$$

该公式中， η 是一常量来控制 $\bar{T}(s')$ 融合到当前温度的权重。

宽容机制经过大量的实例论证确实有助于促进多智能体在各自策略优化的过程中，朝着使群体利益最大化的方向同步更新策略^[81,83]。因此本文将宽容机制与奖赏值网络RN结合，并提出一种新型奖赏值网络，试图来在合作式多智能体问题中，能够有效促进智能体之间的协作。

(2) 宽容的奖赏值网络 (Lenient Reward Network, LRN)

如前文所述，为了降低环境中的噪声，本文提出了奖赏值奖赏值网络，对环境中的奖赏值函数 $R(s, a)$ 进行显示拟合。针对每一个 (s, a) ，奖赏值网络通过查询存储在样本经验池中与 (s, a) 关联的所有样本 (s, a, r, s') ，计算即时奖赏值 r 的算数平均，并用计算的结果来训练奖赏值网络，实现有效地降低环境返回的即时收益 r 中的噪声。

但在多智能体系统中，其余智能体的行为变化也会给即使收益 r 带来而外的偏差，进而阻碍智能体收敛到合作策略。其本质原因是由于对手的行为变化降低了智能体执行最优策略所获取到的即时收益 $r(s, a^*)$ 。为了解决该问题，本文提出使用了宽容的奖赏值网络(Lenient Reward Network, LRN)。LRN将宽容机制与奖赏值网络结合，使得LRN在智能体探索初期保持相对乐观，对 $R(s, a)$ 的负更新以较大的概率拒绝执行。LRN能够有效地促进RN网络对真实的奖赏值函数进行拟合，更好的促进智能体寻找到实现群体利益最大化的协作策略。LRN网络的训练更新公式如下：

$$R_{t+1}(s_t, a_t) = \begin{cases} R_t(s_t, a_t) + \alpha \delta & \text{如果 } \delta > 0 \text{ 或者 } x > l(s_t, a_t) \\ R_t(s_t, a_t) & \text{否则} \end{cases} \quad (4-16)$$

这里 $R_t(s_t, a_t)$ 代表对状态 s 与决策 a 在 t 时刻的奖赏值估计。而 $\delta = \bar{r}_t^{(s,a)} - R(s_t, a_t)$ 代表奖赏值网络估值 $R_t(s_t, a_t)$ 与目标更新值 $\bar{r}_t^{(s,a)} = 1/n \sum_{i=1..n} r_i^{(s,a)}$ 之间的误差。注意这里 $\bar{r}_t^{(s,a)}$ 的计算是通过将存储在经验池中所有与 (s, a) 相关联的 $r_i^{(s,a)}$ 计算算数平均而得到的。此外，宽容值 $l(s_t, a_t)$ 和Lenient Q-learning算法中的定义相同(公式4-11)，拥有相同的含义，并且伴随着 (s, a) 访问的次数增加而逐渐衰减。LRN能够有效的降低因为其余智能体行为变化而引起估值偏差，进而帮助智能体在合作式马尔科夫游戏中，共同找到联合最优策略，实现群体利益最大化。

还有一点值得注意的是，LRN为每个 (s, a) 存储宽容值 $l(s, a)$ 和 (s, a) 是一一对应的；此外用于LRN训练的平均奖励 $\bar{r}_t^{(s,a)}$ 也依赖于 (s, a) ，因为它是使用 $r(s, a)$ 计算的。但是，当面对高维或无穷的状态空间时，使用表格方法将 l 或 r 映射到 (s, a) 是不可行的。为了解决这个问题，本文使用自动编码器（如^[101]中所建议的）对状态 s 进行自动聚类。相当于一种“离散化”操作，将高维状态空间映射到低可数状态空间。在这样做后， $l(s, a)$ 和 $r(s, a)$ 可以与特定的 (s, a) 相关联，这使得LRN可以应用于具有无穷的连续状态空间的问题。

(3) 帕累托最优纳什均衡

本节主要针对多智能体环境下的帕累托最优纳什均衡策略进行一定的探讨。首先在多智能体环境下可能存在多种合作式的纳什均衡策略。而这些策略之间必然存在一个或者多个最优纳什均衡策略，这种策略我们称之为帕累托最优纳什均衡策略^[29]。其次，在多智能体环境下，特别是独立学习的合作式多智能体环境下，如何促使多智能体有效的寻找到纳什均衡一直以来是学术界的研究热点以及未解难题。目前既有的基于独立学习的多智能体策略优化算法并没有理论保证一定能够收敛到纳什均衡^[29]，而帕累托最优纳什均衡策略的学习更是难上加难。

尽管如此，WDDQN算法仍然能够在随机的马尔科夫环境下，帮助独立学习智能体高效的学习到合作策略，并且加大了智能体的策略收敛到帕累托最优纳什均衡的可能性。后续章节的实例证明WDDQN算法借助宽容的奖赏值网络能够有效地引导智能体策略朝着帕累托最优纳什均衡的方向进行优化。

4.5 调度经验重放策略

本节主要从以下几个方面论述了经验重放机制在随机多智能体环境下应用的难点挑战以及解决方案。首先分析了优先级经验重放机制的算法原理，以及在多

智能体环境下使用优先级经验重放机制可能存在的问题，并提出了相应的解决方案。其次，论述了在多智能体环境这类稀疏奖赏值的环境下，优先级采样策略的优先级分配策略存在一定的不足，并提出了一种调度经验重放策略来弥补该不足。最后介绍了一种混合经验池重放设计，针对样本进行分类存储，进而更高效的提升策略优化效率。

4.5.1 优先级经验重放

优先级经验重放机制(Prioritized experience replay, PER)^[84]能够有效地提高DQN算法训练效率，原因在于PER能够对经验池中的样本分配了不同的采样优先级。具体来说，每个样本的优先级是根据样本计算出来的估值和网络估值绝对值差异进行衡量的，计算公式如下：

$$\delta = \|r(s, a) + Q(s', a^*) - Q(s, a)\|_2 \quad (4-17)$$

上述公式中， $Q(s, a)$ 是DQN网络针对 $Q(s, a)$ 的估计，而 $r + Q(s', a^*)$ 是基于样本 (s, a, r, s', d) 计算出来的用于网络的更新的目标值。二者之差用 δ 表示，衡量了网络估值和实际计算的Q值间的差异。差异越大，说明网络针对该 $Q(s, a)$ 的估值越不准，反之说明网络估值与真实值接近。因此 δ 作为刻画网络估值准确的一种度量方式，也能够用作于衡量经验池中样本采样的优先级。直观上理解， δ 越大说明，网络针对该 $Q(s, a)$ 的估值越不准确，因此该样本需要以尽可能被选取出来，用于更新并纠正Q估值网络。注意到这里使用均匀采样算法，也就是说 δ 越大的样本，越有可能被选择到用于更新网络。

然而，在随机多智能体环境下，由于奖赏值中存在噪声以及对手行为一直在不断变化，会导致智能体接收到的环境反馈的奖赏值存在偏差。而PER是根据 δ 对样本进行优先级分配的，这里的 δ 是通过 r 计算出来的。因此当 r 存在偏差的时候，势必会导致 δ 不准确，进而影响PER的采样效果。

一种可能的结果是PER会大概率选择 δ 较大的样本用于网络更新，而 δ 过大的原因并不止是因为网络存在估值不准确的原因，还有可能因为环境返回的 r 存在较大的噪声，亦或是因为对手改变了起行为，导致 r 突变，产生较大的 δ 。然而PER并无法感知这两种情况，且只会根据 δ 的大小进行样本抽样。假设给定一个训练样本 (s, a, r, s', d) ，其 δ 值过大是因为 r 中存在较大的噪声，而不是网络估值不准确。面对这样的样本PER会考虑到 δ 较大，而不断的选该样本进行网络更新。这势必会导致Q网络估值更加的不准确，恶化算法性能影响智能体策略的收敛性。

为了使得PER能够在随机多智能体环境下有效地选择重要样本来训练Q网络，首先要解决 r 值中可能存在的干扰。针对 r 中的噪音干扰和其余智能体行为变化产生的干扰，都可用前文所述的宽容奖赏值网络来解决，本文提出在对样本计算优先级的时候，使用宽容奖赏值网络的估值 $R^N(s, a)$ 来替代环境返回的即时奖赏值 $r(s, a)$ ，具体公式如下：

$$\delta = \|R^N(s, a) + Q(s', a^*) - Q(s, a)\|_2 \quad (4-18)$$

上述公式中使用LRN的估值 $R^N(s, a)$ ，能够有效的降低 δ 的偏差，进而能够是的PER分辨出真正重要的样本进行采样用于网络更新，提升智能体策略优化的效率。

4.5.2 调度经验重放策略

合作式多智能体问题中，稀疏奖赏值是一个常见问题与挑战，因为大部分样本中环境反馈的即时奖赏值是0。考虑到非零样本很难探索到，拥有较大的实际意义，一种直观的想法是尽可能多使用非零样本来更训练网络。但由于PER机制中针对新的轨迹数据中每一个样本的优先级分配策略是平等分配优先级，这导致本身就极低概率能探索到的非零样本，和同一条轨迹中其它样本拥有相同的采样优先级，这增加了其被采样出来进行网络训练的难度。

具体来说，图4-2展示了优先级经验重放机制PER对一个新探索的轨迹数据分配优先级的策略(图中蓝色部分)。上图中PER策略会跟踪当前样本池中的虽高的抽样优先级(红色样本)，并对最新的轨迹中的所有样本分配都分配此最高优先级作为样本的抽样优先级存储在样本池中。这种做法的出发点在于PER认为，相比样本池中既有的所有样本，最新轨迹里的所有样本都含有最新的信息，更应该用于网络更新¹。直觉上PER这种优先级分配策略能够有效的提取最新的样本用于网络训练。大量的实验论证了PER确实能够提升网络训练效率^[84]。

在合作式多智能体环境下，特别是当奖赏十分稀疏时，如果使用PER进行网络训练，存在一个潜在的问题。考虑如下场景，在合作式多智能体问题中，智能体之间能够成功协助的情况是很难探索到的，而在成功合作的智能体轨迹中，解决终点状态的 s 相比轨迹的初始状态 s 更加有价值，尤其是基于稀疏奖赏值的多智能体问题。除此之外，根据 $Q(s, a) = r + Q(s', a^*)$ 公式表明，针对靠近离终点状态的 (s', a^*) ，其估值 $Q(s', a^*)$ 如果已经存在较大的估值偏差的时，远离重点状态

¹参考OpenAI的开源代码实现：<https://github.com/openai/baselines>

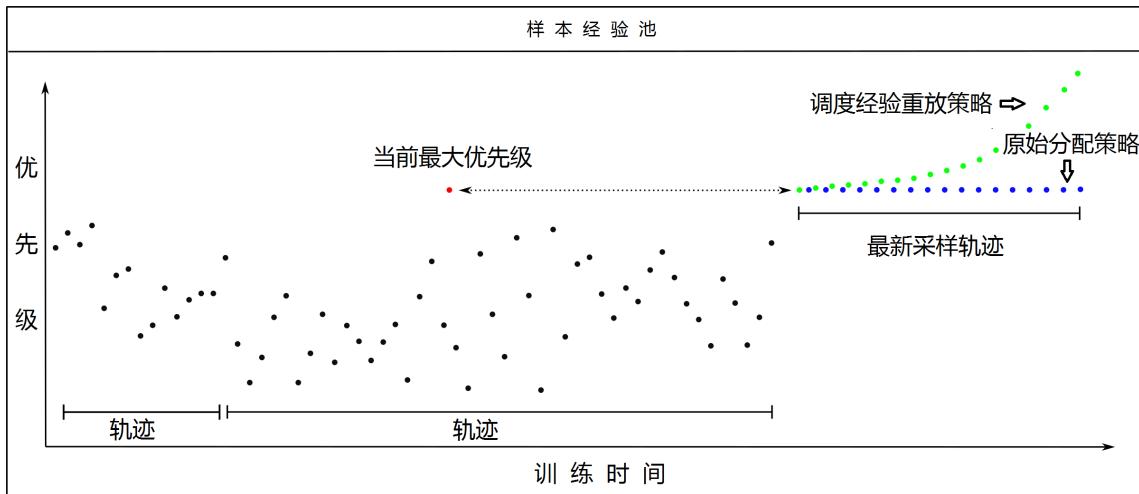


图 4-2 经验重放机制-优先级分配策略对比

的 (s, a) 的估值偏差会恶化的更加严重。这是由于 $Q(s, a)$ 的更新是依赖于 $Q(s', a^*)$ 的，而此时 $Q(s', a^*)$ 已经存在较大的估值偏差。这种估值偏差会沿着样本轨迹从末端状态一直蔓延到轨迹的初始状态，并且在传递期间不断加剧。这两个PER的特质都说明越是接近终点的样本数据越发的重要，相比原理重点状态的样本拥有更丰富的信息，应该被尽可能的采样，用于网络更新。

为了凸显靠近终点状态的重要性，使其在网络训练过程中，尽可能的被抽样到，我们需要赋予其相对较高的优先级。为此，本体提出了一种调度经验重放策略(Scheduled Replay Strategy, SRS)。SRS使用了一个提前预先计算好的上升序列，由一连串 n 个权重 w_i 组成，形式如下：

$$[w_0, w_1, \dots, w_n] \quad (4-19)$$

上述公式中的 w_i 和最新样本轨迹 $(s_0, a_0, \dots, s_i, a_i, \dots, s_n, a_n)$ 中第 i 个样本对应，代表了给赔给第 i 个样本的优先级权重系数。这里优先级权重系数 w 的计算公式如下：

$$w_i = e^{\nu * u^i} \quad (4-20)$$

上述公式中 ν 是一个常量， $u > 1$ 代表了指数函数的上升速度， $0 \leq i < n$ 与轨迹中的样本位置形成一一对应关系。直观上， w_i 依赖指数函数， ν 决定了指数函数上升的起点， $u > 1$ 决定了上升的快慢。在优先级分配的过程中，给定一个最新轨迹 $(s_0, a_0, \dots, s_i, a_i, \dots, s_n, a_n)$ ，分配每一个样本的优先级计算公式如下：

$$p_i = p_{\max} \times w_i \quad (4-21)$$

其中 p_{\max} 表示当前经验池中存储的最大的优先级(图4-2中红色样本的优先级)。上

述公式表明SRS在对轨迹中的样本分配优先级的时候，会根据样本在轨迹中的位置分配不同的优先级。越接近重点状态的样本 (s_i, a_i) ,由于*i*较大，对应的 w_i 也越大，因此分配给它的优先级 $p_i = p_{\max} \times w_i$ 会越大。这样SRS借助预先计划好的上升优先级权重序列，就能够对轨迹中的样本进行权重变化，对接近重点状态的样本分配较高的优先级，增加其被抽样的概率。

图4-2可视化地展示了SRS权重分配策略过程，图中每一个点都代表了一个样本，多个连续的点组成了样本轨迹。蓝色的点表示的是原始PER的权重分配策略，也就是PER给最新轨迹中的每一个样本分配相同的优先级。相比之下，SRS使用上升权重序列，给最新轨迹中点样本分配不同的权重（绿色的点），使得接近终点状态的样本拥有较高的优先级。SRS能够有效的增加重要样本的优先级，增加其被抽样的概率，提高网络的训练效率和算法效果。详细的实验结果会在后续的实验章节论述。

4.5.3 混合优先级经验重放池

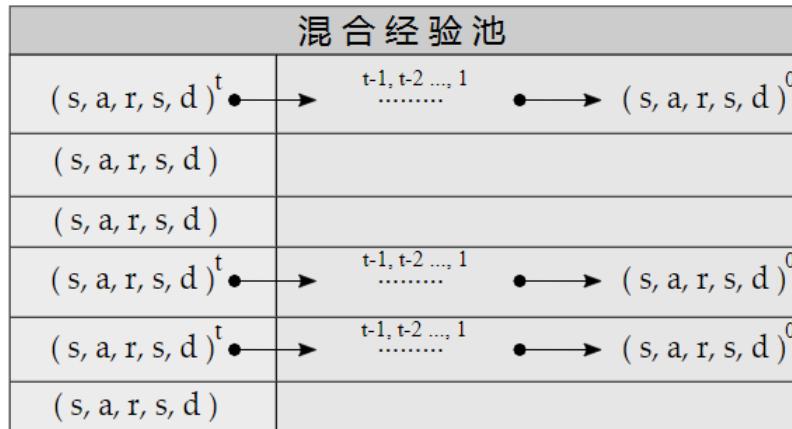


图 4-3 混合优先级经验重放池结构示意图

考虑到多智能体环境下，智能体通过随机探索实现合作的样本轨迹需要很长时间才能被找到。这一类成功合作的样本轨迹具有重要的价值。因此如算法4所示，这里使用两种不同尺度的经验池来储存样本数据。一个是全局样本池 D^G 用于存储所有的样本数据 (s, a, r, s', d) ，另一个 D^E 是专门用于存储成功的整条轨迹数据。注意到两个经验池同时用于训练。这背后的直观解释是鼓励智能体在以往成功的轨迹附近进行探索，这在某种程度上是一种自我模仿。通过这种方式，智能体可以快速掌握一些技能，特别是当遇到可能存在多个子任务的复杂问题时。

4.6 实验论证

本节进行实证评估以验证WDDQN的有效性，并将WDDQN与相关算法进行算法效果的对比。

首先，本文主要考察算法在估值纠偏，学习速度和性能方面的表现。具体来说，本实验给出了double DQN (DDQN) 和WDDQN的算法比较，并考察LRN与SRS机制的有效性。这里WDDQN使用了两个变种，一个是不使用LRN和SRS的简化版(WDDQN w.o. LRN+SRS)，以及使用了LRN和SRS的完整版(WDDQN)。实验环境使用的是基于原始视觉输入的单智能体的吃豆人游戏(Pacman-like gridworld)。

接着，本文使用多智能体环境来考察算法的效果，主要使用独立学习的合作式马尔科夫博弈游戏，在这种设定中，智能体需要在不知道其他智能体的策略的情况下协同解决任务。本文使用三种实验环境来验证算法效果，主要包括爬山游戏(Climbing game)^[29]、捕食者游戏(Predator-Prey game)^[102,103]和一个复杂的运输问题(CMOTP)^[81,104]。这里主要考察WDDQN能否提升多智能体的策略优化过程以实现智能体间的有效合作。每个实验的最后都针算法表型给出了详细的评估和讨论。

本文将WDDQN中 β 里的常量 c 设置为0.1，宽容的Q学习算法中的参数 K, κ, η 分别为2、0.95、0.6。此外，DDQN网络以及宽容的Q学习的网络训练学习率 α 设置为0.0001。

表 4-1 WDDQN算法中的网络架构

# 网络	视觉输入维度	卷积核 1/2/3维度	全连接的维度
DQN	84 * 84 * 3	32/64/64	512
LRN	84 * 84 * 3	16/16/16	128

表4-1描述了WDDQN中深层Q网络和LRN的网络结构。这里使用三个隐藏的卷积层(每两个连续层之间使用的是Relu激活函数)和一个完全连接层。DQN和LRN的输出分别代表单独的估值 $Q(s, a)$ 和奖励 $R(s, a)$ 。出于探索目的，DQN使用了(ϵ -greedy)的探索机制，在前10000步中， ϵ 线性地从1降低到0.01。我们使用了具有0.0001学习率的Adam算法和大小为32的批训练样本。我们训练了总共2500轮，并使用了大小为8192的记忆池储存最新的样本。最后，为了公平起见，

LRN中的 K, κ, η 与宽容Q学习相同，而SRS中的 ν 和 mu 设置为0.2和1.1。

4.6.1 估值纠偏实验

本节旨在研究如下问题：“直接使用双权估计器是否足以有效地解决环境中的估值纠偏？如若不行，是否需要额外的机制？”。

4.6.1.1 实验设置

本实验环境是 $N \times N$ 方格盘内的吃豆人游戏，智能体需要通过四个方向（上、下、左、右）的移动操作从初始状态（左上方方格）移动到目标方格（图中右下角的粉红色点）。在碰壁的情况下，智能体将不移动，除此之外每一次操作后将向目标方向移动一个方格。方格盘内目标方格随机出现，智能体在每次进入目标方格结束一阶段的操作后有等概率的机会获得范围在[-30, -40]的随机奖励。在未达目标方格的情况下，选择向北或向西的奖励是-10或+6，选择向南或者向东的奖励是-8或+6。由于环境返回的奖赏值存在不确定性，因此本实验的环境是有噪的。

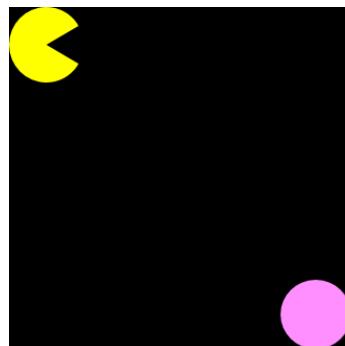


图 4-4 吃豆人游戏(Pacman-like Grid World)

4.6.1.2 实验结果分析

根据图4-5展示的实验结果可知，在含有噪声的随机的环境下，DDQN算法需要较长的时间实现优化策略，而使用双权估计器的WDDQN w.o LRN+SRS相对来说取得了较好的效果。然而，纵使WDDQN w.o LRN+SRS优于DDQN，但在12*12、14*14、16*16的吃豆人游戏中，都存在较大的震荡，甚至可能出现长时间训练后仍不收敛的情况。由此说明，仅使用双权估计器并不足以进行有效的

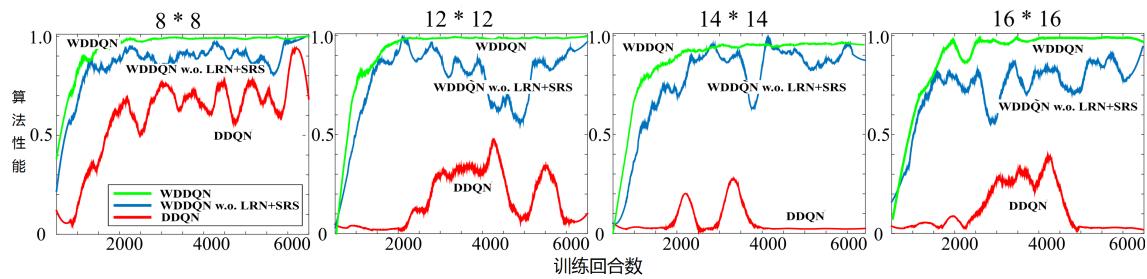


图 4-5 使用四种不同尺寸的吃豆人游戏对DDQN算法、WDDQN w.o. LRN+SRS和WDDQN算法进行试验对比。X轴代表训练的回合数，Y轴代表算法所需步数与最优步数的比值（越接近1说明越接近最优策略）。

策略优化。另一方面，实验结果证明，使用了LRN与SRS机制的WDDQN算法在稳定性、学习效率以及算法性能上都取得了较好的表现，这说明LRN和SRS能够有效地降噪以及估值修复，以实现智能体策略优化。由此得出结论，WDQ能够有效地与DRL算法结合，但不足以有效地解决环境中的估值纠偏以及策略优化，而借助了LRN与SRS机制的WDDQN实现了有效地估值纠偏以及策略优化。

4.6.2 合作式多智能体实验（离散动作空间）

本节旨在研究如下问题：“WDDQN能否有效促进独立学习智能体实现有效的合作？如果可以，WDQ，LRN和SRS分别起到了什么作用？如果是在随机环境中，WDDQN能否收敛到纳什均衡解，甚至是帕累托最优纳什均衡解”。

4.6.2.1 实验设置

本实验改编自Matignon^[29]提出的爬墙游戏。图4-6中的两个机器人代表两个智能体。两个智能体需要同时进入同一个目标状态以实现有效合作。带有字母S的方格是次优的目标，收益为+10，带有字母G的方格是全局最优目标，收益为+80。图中灰色部分是一面墙，将平面分割成两部分。在每一阶段，两个智能体从最左下/右下的方格开始，尝试同时走到绿色的目标方格。每一个智能体有东西南北四种操作。在碰到中间的墙时，智能体将不移动，除此之外每一次操作后将向目标方向移动一个方格。未达到目标状态的每一次操作收益为0。智能体同时进入目标状态结束一个阶段的操作将得到正收益，如果两个智能体进入了不同的目标状态，则双方的收益均为-1。

本实验存在如图4-6所示的两种纳什均衡策略，分别是双方共同进入S和G，其

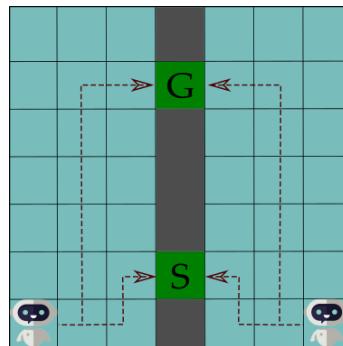


图 4-6 合作式爬山游戏(Cooperative Climbing Game)

中进入目标方格S次优解，而进入目标方格G是全局最优解，即帕累托最优纳什均衡解。

4.6.2.2 实验结果分析

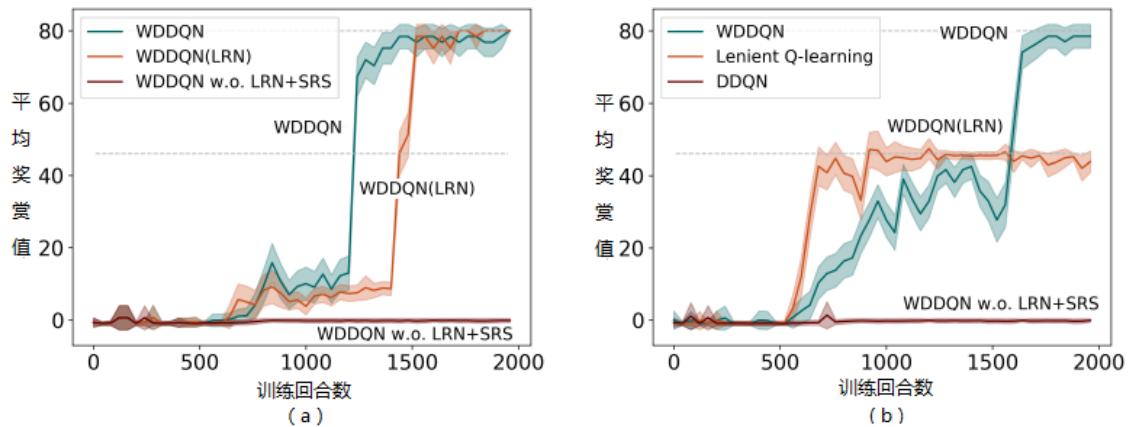


图 4-7 在确定性奖赏值的合作式爬山环境下，WDDQN与其变种算法的对比试验（左侧）以及在随机奖赏值的合作式爬山环境下，WDDQN与其它相关算法的对比试验（右侧）。注意到X轴代表训练回合数，每一个点代表50个训练回合，Y轴是对应50回合的平均奖赏值，阴影区域代表50个回合中得最低奖赏值和最高奖赏值

为了回答本节一开始提出的前两个问题，本文首先使用带确定性奖赏值爬墙游戏，分别对WDQ，LRN和SRS进行研究。图4-7(a)是WDDQN w.o. LRN+SRS, WDDQN(LRN)²和WDDQN算法在爬山游戏中取得的平均收益。

实验结果显示WDDQN w.o. LRN+SRS算法奖赏值接近0，独立学习智能

²WDDQN(LRN)只使用了LRN，和WDDQN w.o. SRS代表相同的算法

体之间无法实现有效的策略合作。这是由于在独立学习的多智能体环境下，策略优化的收敛性已经无法得到保证，因此无法有效地进行策略优化。由此说明，直接将WDQ与DRL算法结合并不能有效地促进独立学习智能体之间的合作，不足以解决独立学习智能体之间的策略优化问题的。然而，使用了LRN的WDDQN(LRN)却能够有效的促进独立学习智能体实现有效的合作。此外，使用了LRN与SRS的WDDQN不仅能够促进智能体之间的合作，其优化策略的效率明显优于其他算法。该实验结果证明，使用了LRN和SRS的WDDQN算法能够有效的促进独立学习智能体之间的合作，并且取得较高的策略优化效率。

为了回答本节一开始提出的最后一个问题，本文对前叙实验环境进行了调整，并验证了WDDQN、DDQN以及Lenient Q-learning算法的效果。具体来说，本节使用了带有随机奖赏的爬山游戏，即智能体在到达目标方格S时分别有60%和40%的概率获得+10和+100的收益，到达目标方格G可以获得既定收益+80。在这种情况下，目标方格S仍是次优的，因为平均收益是46。由于在S处有可能产生+100的随机奖赏，这有可能误导智能体朝着次优的方向进行策略优化，对算法性能产生较大的影响。

图4-7(b)展示了不同算法的平均收益结果，两条虚线分别表示纳什均衡和帕累托最优纳什均衡，收益分别是+80和+46。就收敛速度和平均收益来说，WDDQN和宽松Q-learning优于DDQN，由此说明，直接使用DRL算法是无法有效解决多智能体问题的。值得注意的是，在所述的含有随机奖赏值的多智能体环境中，WDDQN相较于宽松Q-learning取得了更高的平均收益。该结论WDDQN可以更好地找到纳什均衡解，甚至是帕累托纳最优解。

4.6.3 合作式多智能体实验（连续动作空间）

本节旨在研究如下问题：“使用LRN和SRS机制的DRL算法(DDPG)是否有助于独立学习智能体寻找到纳什均衡策略（特别是帕累托优化纳什均衡策略）？”。

4.6.3.1 环境设置

本节使用一个被称为捕食者游戏的合作式多智能体环境，该环境最早由Benda^[102]提出，后续Ryan^[103]也是用该环境用于研究合作式多智能体问题。为了增加问题的难度，本文增加了下述修改：首先，智能体的动作空间变为连续空

间；其次，在奖赏值函数 $R(s, a)$ 中加入正态分布 $N(0, 1)$ （均值为0，标准差为1）的环境噪声，增加策略优化的难度；第三，智能体到达部分终点目标时获得的奖赏值是随机的。

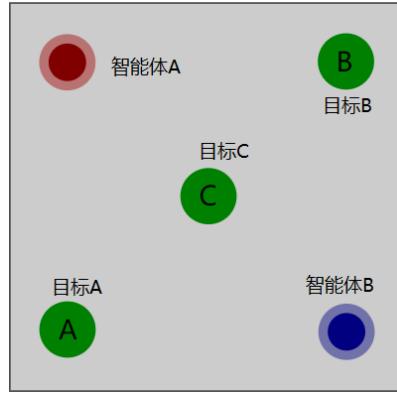


图 4-8 合作式的捕食者游戏(Cooperative Predator-Prey Game)

如图4-8所示，在每回合开始时，智能体A和智能体B随机出现在不同位置，并且需要同时移动到同一目标A、B或者C中。智能体彼此不知道对方的策略，通过上下左右四个方向加速（范围为-1到1）。每当两个智能体同时抵达同一个目标时，彼此都获得相同的正奖赏值，奖赏值的大小根据目标不同而不同，如果两个智能体分别进入了不同的目标点，则会收到相同的负奖赏值。

表4-2展示了智能体选择不同策略产生的收益矩阵。如果两个智能体同时到达A，将获得+11的奖赏值，同时到达B的平均收益则是+7（等概率产生+14或0的收益）。如果一个智能体已经到达而另一个仍未到达，则两个智能体都会收到非正奖赏值作为惩罚。可以看出两个智能体同时选择移动到A或者B均为纳什均衡解，而移动到目标A是帕累托最优纳什均衡解。

表 4-2 合作式的捕食者游戏的收益矩阵

		Agent 2			
		移动到A	移动到B	移动到C	移动到D
Agent 1	移动到A	11	-30	0	-30
	移动到B	-30	14/0	6	-10
	移动到C	0	6	5	0
	移动到D	-30	-10	0	0

4.6.3.2 实验结果分析

一般来说，有几个因素会阻碍独立学习者收敛于优化均衡。第一，独立学习智能体之间协同学习，给彼此带来了扰动，指示彼此都认为是环境中存在不稳定噪声。其次，在缺乏交流与协作机制的情况下，目标B产生的较大的随即奖赏值可能会误导智能体，妨碍其学习到帕累托优化纳什均衡解（朝A移动）。第三，独立学习智能体有可能因为其余智能体的探索行为，收到环境返回的惩罚，这种情况被称为“变更探索”（alter-exploration）问题。

为回答本节开始所提的问题，本文将LRN与SRS机制与DDPG结合，称为DDPG(LRN+SRS)，并与DDPG^[105]算法，MADDPG^[103]算法和Lenient Q-learning^[81]算法进行对比。由于Lenient Q-learning和LRN与SRS机制都是基于离散动作空间的设计，本文在评估时将(-1,1)的连续空间离散化，每0.01划分一个动作，得到共2000个离散的动作。DDPG与MADDPG仍旧使用原有的网络架构，而动作输出改为四个方向上加速的实数值。此外，DDPG与MADDPG中的Critic也保留原有的网络结构用于评估 $Q(s, a)$ 。

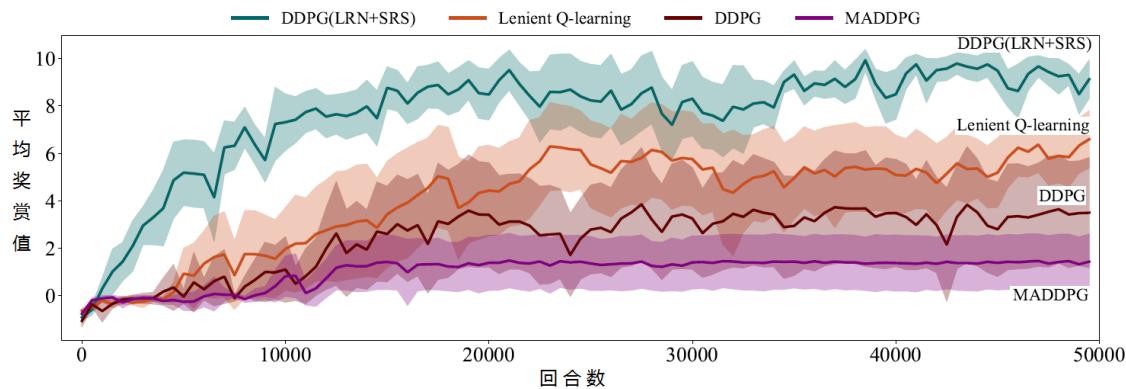


图 4-9 在使用了10个随机种子的合作式的捕食者游戏中，DDPG(LRN+SRS), DDPG, MADDPG, Lenient Q-learning 算法的效果对比试验。注意到X轴代表训练回合数，每一个点代表500个训练回合。Y轴是对应500回合的平均奖赏值，阴影区域代表5000个回合中得最低奖赏值和最高奖赏值(10个随机种子*500回合/种子)

评估结果如图4-9所示，一个出乎意料的发现是MADDPG算法表现最差。本文推测这是由于MADDPG算法中使用的Critic是基于多个智能体的联合动作进行Q值估计，只适用于奖赏值不含噪声的多智能体问题。具体来说，由于本实验中目标B的奖赏值变为随机奖赏，且随机性较大，导致基于集中式智能

体动作的Critic可能误导智能体朝着目标点C进行策略更新，甚至无法引导智能体进行有效的策略更新，致使策略优化不收敛。反观基于独立学习智能体的DDPG算法，由于没有集中式的Critic，DDPG获得相对较高的平均收益。另一个研究发现是Lenient Q-learning算法由于使用了宽松的机制，其策略优化效率以及获得的评价奖赏值均优于DDPG与MADDPG算法。此外，所有基准算法中，DDPG(LRN+SRS)算法在学习速度和平均收益方面都表现出了最好的性能。本文推论这是由于LRN能够有效降噪并促智能体之间的合作，并且SRS通过选择更合适的样本用于策略优化，有效的促进了学习速度。

综上，实验结果证明了本节开始的的猜想，即LRN和SRS确实有助于独立学习智能体寻找到纳什均衡策略（特别是帕累托优化纳什均衡策略）。

4.6.4 基于层次任务的合作式多智能体实验

本节旨在研究“使用了LRN与SRS机制的WDDQN能否在含有随机收益以及子任务的CMOTP问题中学习到最优策略？”。

4.6.4.1 实验设置

本章最后使用一个基于层次任务的合作式多智能体来验证算的有效性，该实验是一个基于合作式的物体运输任务，最早由Busoniu^[104]提出用于合作式多智能体系统的研究。Palmer^[81]也使用该环境验证深度多智能体强化学习算法的验证。如图4-10所示，环境中存在两个用字母A表示的智能体（左下方与右下方）在智能体需要相互配合将用字母T表示的物体一共运送到目标区域。环境中存在两个目标区域，目标G代表了全局最优解，而目标S代表了次优解。智能体需要分别位

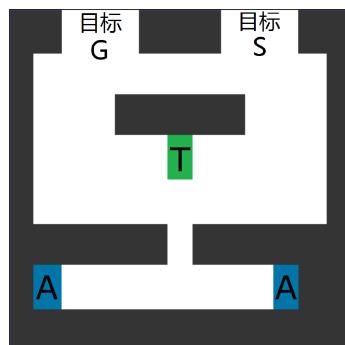


图 4-10 基于合作式的物体运输任务(Coordinated Multiagent Object Transportation)

于物件T的两侧，才能成功挪动物体，若只有一方位于T的一侧，则智能体无法移动，直到等待另一方到达T的另一侧。智能体动作空间包括朝上下左右四个方向移动以及原地不动五个动作。此外，当两个智能体处于T的两侧时，只有当双方选择想通的方向进行移动时，才能挪动物体朝着该方向移动，否则双方将呆在原地，不发生任何移动。最后，只有当物件T被移动到目标区域S或者G时，智能体获得相应的奖赏值，游戏终止。当智能体未进入目标区域之前，任何动作都会导致智能体收到-0.1的奖赏值。但成功拿起物品获得+10的奖赏值，进入区域G的奖赏值为+80，进入区域S的奖赏值为+120或0（等概率）。相较于之前的实验环境，CMOTP是一个较为复杂的问题，除了环境中存在的随机奖赏值之外，还存在层次化的子任务。智能体需要相互配合协同完成子任务，才能完成整个任务。

4.6.4.2 实验结果分析

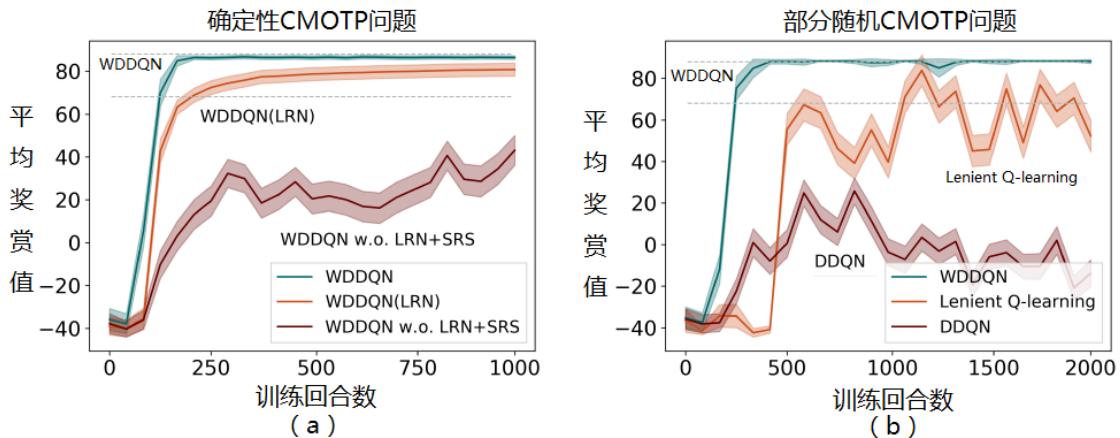


图 4-11 在确定性奖赏值的合作式物体运输任务的的环境下，WDDQN与其变种算法的对比试验（左侧）以及在随机奖赏值的合作式物体运输任务的的环境下，WDDQN与其它相关算法的对比试验（右侧）。注意到X轴代表训练回合数，每一个点代表50个训练回合，Y轴是对应50回合的平均奖赏值，阴影区域代表50个回合中得最低奖赏值和最高奖赏值。两天灰色的线分别代表通往G的最优策略和通往S的侧有策略的奖赏值

第一个实验从平均奖赏值，策略优化速度两方面，考察WDDQN算法，LRN机制以及SRS机制的有效性。本实验使用确定性CMOTP问题，即该环境中的奖赏值是确定性的。具体来说，智能体到达区域S获得确定性+60的奖赏值，到达区域G获得确定性+90的奖赏值。因此，本实验存在的两个纳什均衡解（移动到S与G）对应的总收益是+68.3与+88.3。为了保持符号统一，WDDQN、

WDDQN(LRN)、WDDQN w.o. LRN+SRS的涵义与前文相同。图4-11(a)展示了实验结果，与之前实验结论相似，WDDQN w.o. LRN+SRS的震荡比较频繁取得的平均奖赏值也较低，但同时，策略优化过程中的平均收益呈现总体上升趋势，说明WDQ仍然有助于独立学习智能体学习到合作策略。另一个发现是，LRN确实有助于提升策略优化效率并减少训练过程中的震荡，有效促进智能体学习到最优策略。最后，WDDQN在平均收益和学习速度上表现最好，这主要因为LRN在探索中可以使用了宽容的机制，有效促进智能体之间的合作；同时SRS机制也有效的提升了智能体策略优化的效率。

第二个实验使用了部分随机的CMOTP环境验证WDDQN算法的有效性，具体来说，当物件T被搬运到目标S区后，智能体有50%的机率获得+120收益，有50%的机率获得0收益。尽管区域S仍为次优解（平均收益为60），但其随机生成+120的高奖赏值，会误导智能体朝着次优解的方向进行策略优化，特别是像Lenient Q-learning此类使用了宽容机制的强化学习算法。图4-11(b)中的实验结果显示，WDDQN和Lenient Q-learning在平均收益和策略优化效率上明显优于DDQN算法，更适用于解决含有随机奖赏值的多智能体问题。此外，DDQN较差的表现说明了基于单智能体的强化学习算法无法直接拓展并解决基于独立学习的多智能体问题。另一方面，使用了宽容机制的Lenient Q-learning相比DDQN取得了更好的平均收益，但在学习过程中震荡的比较剧烈，并且没有学习到全局最优策略。这一现象说明了，宽容机制确实有助于独立学习智能体进行策略优化实现多智能体之间的有效合作。然而，基于宽容机制的强化学习算法对于奖赏值十分敏感，在含有随机奖赏之的环境中，无法有效的学习到帕累托最优纳什均衡策略。对比之下，使用了LRN的WDDQN算法在实验中能够有效应对环境中的噪声以及随机奖赏值的挑战，策略优化过程中表现出了更好的稳定性，并且能够有效的促进独立学习智能体学习到帕累托最优纳什均衡策略。此外，SRS机制提升了WDDQN算法在策略优化过程中的学习效率。综上，实验结果证明，基于LRN与SRS机制的WDDQN算法能够有效的帮助独立学习智能体进行策略优化，学习到纳什均衡策略乃至帕累托最优纳什均衡策略。

4.7 本章小结

本章针对噪声环境下独立学习智能体的策略优化问题展开研究，提出了基于

双权估计器的WDDQN算法，有效地解决了值函数的估值偏差问题，可用于解决基于视觉输入的复杂多智能体问题。此外，本文提出了奖赏值网络RN，有效地实现了奖赏值噪声的降噪，并将其与宽容机制相结合，提出了宽容奖赏值网络LRN，有效地促进了多智能体间的策略协同优化。最后，本文提出的调度经验回放策略SRS，有效地实现重要样本的优先级采样，增加了网络训练的效率，促进智能体策略优化效率，实现了多智能体间的有效合作。

一方面，实验证明，WDDQN(LRN+SRS)不仅能够有效的降低估值偏差，相比double Q-learning以及lenient Q-learning，能够实现更高效的多智能体策略优化，提升了多智能体学习到帕累托最优纳什均衡的可能性。

另一方面，WDDQN(LRN+SRS)存在一定的不足。例如其只使用了单线程来进行策略优化，这在大规模的强化学习问题中，将会使得探索过程变得低效，进而降低算法的有效性。在未来考虑通过使用A3C^[52]算法的异步加速机制与WDDQN算法相结合，以提高WDDQN的算法性能。

第5章 面向非静态对手环境下的多智能体策略优化研究

本章节主要研究面向非静态对手环境下的多智能体策略优化问题。主要从独立学习智能体的角度出发，尝试解决当面对不断变换行为对手时，策略优化过程中所面临的问题，以实现有效的策略优化，最大化长期收益。具体来说，主要从算法层面以及多智能体系统层面，详细分析了既有算法的缺陷，并提出了深度贝叶斯策略重用算法来解决缺陷。此外，对新算法设计的本质进行了详细的分析与论述，并使用包括合作式和一般式等多种含有非静态对手的多智能体环境对算法的有效性进行了验证。

5.1 引言

基于独立学习的多智能体策略优化研究工作中，一个重要的挑战来自于环境中非静态智能体（不断变化行为的智能体）^[30]。具体来说，通常在进行策略优化时，独立学习智能体把环境中共存的其余智能体（对手）当作环境的一部分，因此随着对手不断变化行为，从独立学习智能体的角度看来，相当于环境中发生了变化。由于环境发生了变化，等同于间接改变了智能体策略的优化方向，增加策略优化的难度，甚至导致策略优化无法收敛。尽管存在一定的难题与挑战，但由于非静态对手的定义更加一般化，并且更加符合解决现实问题，因此涌现了大量的相关的研究工作^[106,107]。一定程度上，面向非静态对手环境下的多智能体策略优化的研究能够指导并解决真实问题，具有重要的理论与实践意义。

近年来，面向非静态对手环境下的多智能体策略优化研究依然面临的最大难题依然是如何应对不断变化行为的对手。既有的研究工作主要分为两大类，第一类是直接使用深度强化学习算法来学习应对非静态对手的策略^[106,108,109]，此类算法在进行策略优化过程中，需要使用可用于描述对手信息的特征，一般需要丰富的领域知识，并在算法应用性上存在一定的局限性。另一种思路基于显示的对手分类，通过在线检测对手的策略类型，并执行相应的应对策略¹，实现长期收益最

¹本文将响应对手策略的最优策略称为应对策略（Response Policy）。

大化^[107]。此类算法的局限性跟小，易于实现并且拥有较好的算法性能。因此，本文针对显示的对手分类算法展开研究，对既有算法在策略优化过程中存在的缺陷进行详细的分析，并提出相应的解决方法。

既有的显示对手分类算法^[107]存在一定的缺陷与挑战。第一个挑战来自于算法层面，即基于贝叶斯策略重用的算法只使用奖赏值进行对手类型判别可能存在准确度低的问题，这将严重影响策略重用的效果。第二个挑战来自于既有算法在面对未知对手时，需要重头学习针对该未知对手的应对策略，造成算法总体效率的下降。第三个挑战在于既有算法将多个应对策略分别进行保存，需要较大的存储空间，造成空间使用率低的缺点。最后一个挑战在于既有算法只使用了简单的环境进行了算法效果验证，其有效性并未在以原始图片为输入的复杂问题上进行验证。

本章节主要研究面向非静态对手环境下的，如何有效的进行多智能体策略优化问题。主要针对上述包括对手策略检测不准确的问题、针对未知对手的应对策略优化问题、应对策略保存的问题以及在复杂问题下算法有效性的问题，并提出了相应的解决方法。具体来说，首先，本文提出了深度贝叶斯策略重用机制（deep BPR+），针对不同的对手使用重用不同的策略来实现长期收益最大化。deep BPR+实现了有效的对手策略变化检测以及准确的对手类型判别，通过快速切换并执行针对该对手的最佳应对策略来实现长期收益最大。其次，为了提高对手策略检测度的准确性，本文提出了修正置信模型（rectified belief model, RBM）。RBM使用对手建模机制（opponent modeling）来修正贝叶斯策略重用算法中的原始策略检测机制，进一步提高了检测准确度。最后，本章提出了使用策略蒸馏算法（policy distillation），将多个应对策略蒸馏得到单一的策略蒸馏网络（distilled policy network, DPN）。DPN作为策略库用于存储学习好的应对策略，不仅能够实现快速的策略切换，有效地降低了空间使用率，还显著提升了针对未知对手进行策略优化时的优化效率。在本章的最后，分别使用了多个含有非静态对手的合作式以及竞争式多智能体环境来检验deep BPR+算法、RBM机制以及DPN机制的有效性，并在每个实验最后都进行了详细的对比分析。

5.2 基于深度贝叶斯策略重用的多智能体策略优化算法框架

本章提出了深度贝叶斯策略重用算法（deep bayesian policy reuse, deep BPR+）

来应对非静态对手，通过准确的对手类型检测，实现高效地应对策略重用，达到多智能体策略优化的目标。具体来说，Deep BPR+算法包括三个关键模块，分别是贝叶斯策略重用（Bayesian Policy Reuse），修正置信模型（rectifice belief model, RBM）、蒸馏策略网络（distilled policy network, DPN）。这三者的关系如下：

(1) 贝叶斯策略重用（Bayesian policy reuse, BPR）

为了应对非静态对手，本文提出显示地对对手进行分类，并根据对手的类型选择相应的应对策略，以实现多智能体策略优化。为实现这一目标，本文基于贝叶斯策略重用理论，提出了deep BPR+算法，实现高效的策略重用来应对非静态对手。详细的研究内容在5.3节中展开。

(2) 修正置信模型（rectifice belief model, RBM）

为了解决前文所述的传统BPR算法对手类型判断不准确的缺点，本文提出了修正的置信模型（Rectified Belief Model, RBM）。具体来说，BPR使用环境反馈的奖赏值作为对手类型判别的唯一依据。在多智能体环境下，这种原始设计会导致对手类型判断不准确的缺陷。针对该缺陷，BRM模型使用奖赏值与对手建模（Opponent Modeling, OM）两种信号，并结合两种信号的贝叶斯后验概率，进行对手类型的判断，有效地提高了对手类型检测的准确度。详细的研究内容在5.4节中展开。

(3) 蒸馏策略网络（distilled policy network, DPN） 为了解决前文所述的针对未知对手的应对策略优化效率的缺陷以及应对策略存储利用率低的缺陷，本文提出了蒸馏策略网络（Distilled Policy Network, DPN）。具体来说，策略优化效率缺陷的产生是由于一般BPR算法在面对新未知对手时，需要重头学习应对策略，导致学习效率低下。为解决该缺陷，本文提出使用DPN来初始化一个策略，作为策略优化的起始策略。经过实验论证，使用DPN初始化的起始策略能够有效地提高策略优化的过程。另一方面，策略存储利用率低的缺陷是因为一般BPR中的策略库直接存储所以应对策略，这需要大量的存储空间。为了解决该缺陷，本文使用策略蒸馏算法，将多个应对策略蒸馏到单独的DPN中。DPN能够实现快速的应对策略切换以及极大地提高了空间使用率。实验结果证明，DPN通过策略蒸馏实现高效的在线策略学习以及策略重用。详细的研究内容在5.5节中展开。

最后，Deep BPR+继承了BPR中所有的优点，实验结果表明，Deep BPR+在对手策略检准确度、累积奖赏值、以及策略优化速度上相比既有的算法，表现出了较好的性能。此外，基于BPR的传统算法只能解决状态空间较小的简单问题，而Deep BPR+算法使用深度神经网络来作为值估计器，能够有效解决以原始图片

为输入的复杂问题，在多个合作式与竞争式的马尔可夫博弈游戏中都取得了较好的结果。

算法 5: Deep BPR+算法

Input: 回合数 K , 应对策略库 Π , 已知对手策略集合 \mathcal{T} , 性能模型 $P(U|\mathcal{T}, \Pi)$

```

1 使用均匀分布初始化置信模型  $\bar{\beta}^0$ 
2 for  $t = 1 \dots K$  do
3   if 执行策略重用阶段 then
4     根据  $\bar{\beta}^{t-1}$  选择策略  $\pi^t$  执行，并接收奖赏值  $u^t$ （见公式5-14）
5     根据观察到的对手行为，进行对手策略  $\hat{\pi}_o^t$  建模
6     使用  $u^t$  and  $\hat{\pi}_o^t$  更新修正置信模型  $\bar{\beta}^t$ （见公式5-13）
7     if 如果移动平均奖赏值检测到当前对手正使用未知策略 then
8       使用蒸馏策略网络初始化策略  $\pi^t$ ，在下一回合切换到应对策略
9       学习阶段
10    else if 执行应对策略学习阶段 then
11      使用DQN算法对  $\pi^t$  进行策略优化，并进行对手策略  $\hat{\pi}_o^t$  建模
12      if 应对策略优化完成 then
13        更新  $\mathcal{T}$ ,  $\Pi$  和  $P(U|\mathcal{T}, \Pi)$ ，并在下一阶段切换到策略重用阶段

```

算法5概述了deep BPR+的总体流程，主要包含两个阶段：首先是重用阶段（第3-8行），其关注选择最优的策略执行；第二是学习阶段（第9-12行），其关注学习针对新对手的最佳应对策略。在每一轮游戏中，只会执行一个阶段。

在重用阶段，deep BPR+使用由对手模型纠正过的置信模型 $\bar{\beta}^t$ 选择应对策略 π^t ，获得累积奖励 u^t ，估计对手的在线政策 $\hat{\pi}_o^t$ ，并使用 u^t 和 $\hat{\pi}_o^t$ 更新 $\bar{\beta}^t$ 。在该阶段的最后，deep BPR+检查对手是否正在使用从未见过的策略。如果检测结果显示对手正在使用一个未知策略，deep BPR+在下一轮将会切换到学习阶段，并使用DPN初始化的 π^t 作为起始策略，开始策略优化，学习如何应对新对手策略。

在学习阶段，任何DRL算法都可用于在线学习应对策略。一旦学习阶段结束，拟合得到的对手政策 $\hat{\pi}_o^t$ 将被添加到已知的对手政策集合 \mathcal{T} 中。相应的应对策略也将被添加到策略库 Π 中，同时性能模型 $P(U|\mathcal{T}, \Pi)$ 也将使用相应的收到累积奖励进行更新。最后，deep BPR将在下一回合中切换回策略重用阶段应对非静态对手。

注意，deep BPR+目前仅在每个回合开始时改变其策略，因为用于检测的奖励

信号是基于每个回合的。然而，如果使用即时奖励作为置信模型中的更新信号的话（第4行），deep BPR+也可以在每个步骤执行策略更新。总之，信号的选择是根据特定问而决定的，且直接决定了策略切换频率的粒度。

接下来的章节主要从以下几个方面进行描述：首先，针对贝叶斯策略重用理论yii相关概念进行介绍；其次，针对对手策略检测过程的难点以及BPR存在的缺点进行论述，并且介绍对手建模以及修正置信模型。然后，介绍策略蒸馏网络的设计原理及优点。最后实验论证部分，使用三个复杂的多智能体问题来验证deep BPR+的算法性能。

5.3 贝叶斯策略重用理论

(1) 贝叶斯策略重用 (BPR)

贝叶斯策略重用算法为智能体在面对未知任务时该如何选择最佳应对策略提供一个有效的解决方案。在BPR中，一个任务 $\tau \in \mathcal{T}$ 一般被定义成一个马尔可夫问题，策略 $\pi(s)$ 根据 s 给出执行动作 a 的概率。策略 π 在一个MDP环境中，经过 k 次交互后的回报（累计收益）的定义如下：

$$u = \sum_{i=1}^k r_i \quad (5-1)$$

其中， r_i 是在第 $i - 1$ 时刻执行完都做后的即时收益。

此外，BPR中维护一个性能模型 $P(U|\tau, \pi)$ ，这是一个关于效用 U 的概率分布，用于描述策略 π 在任务 τ 上的表现效果。给定一组先前解决的任务 \mathcal{T} ，置信度 $\beta(\tau)$ 是一个关于 \mathcal{T} 的概率分布，它测量到当前面临任务 τ^* 和已知任务 τ 之间的相似度。该相似度是根据奖励信号（即累积奖励 u ）计算得出的。注意这里置信模型 $\beta^0(\tau)$ 是用先验概率初始化，并在 t 时刻使用贝叶斯规则进行更新，更新规则如下：

$$\beta^t(\tau) = \frac{1}{\eta} P(u^t|\tau, \pi^t) \beta^{t-1}(\tau) \quad (5-2)$$

其中， $\eta = \sum_{\tau' \in \mathcal{T}} P(u^t|\tau', \pi^t) \beta^{t-1}(\tau')$ 是一个正则化因子。基于置信模型，为了最大化效用，BPR使用“期望提升概率”作为指标，通过指标能够在策略重用库中选择最合适的策略 Π 进行重用^[93]。该指标考虑了当前最佳政策可以实现的效用提升的期望。

假设 \bar{U} 是当前置信下最佳当前政策的预期效用，则 \bar{U} 计算公式如下：

$$\bar{U} = \max_{\pi \in \Pi} \sum_{\tau \in \mathcal{T}} \beta(\tau) \mathbb{E}[U|\tau, \pi] \quad (5-3)$$

因此，BPR选择最佳的候选策略 π^* ，即最有可能导致预期效用提升的策略，衡量公式如下：

$$\pi^* = \arg \max_{\pi \in \Pi} \int_{\bar{U}}^{+\infty} \sum_{\tau \in \mathcal{T}} \beta(\tau) P(U|\tau, \pi) dU. \quad (5-4)$$

BPR+算法^[107,110]扩展了BPR算法，用于处理多智能体问题中的非静态对手，并以在线方式学习新的性能模型。注意到，BPR中的任务和策略分别对应于BPR+中的对手策略和针对这些策略的最佳应对策略。尽管BPR+能够检测对手策略的切换，并以在线调整当前策略，但其有效性仅在使用表格表示的单状态迭代矩阵游戏中得到验证，而并没有用基于原始图片输入的复杂游戏进行过验证。

(2) 深度贝叶斯策略重用 (deep BPR+)

deep BPR+使用深度神经网络对BPR+算法进行了拓展，能够处理状态空间巨大的复杂问题。同时，解决了在扩展过程中存在的以下两个主要缺点：

一方面，公式5-2中的置信模型 $\beta(\tau)$ 的准确性高度依赖于性能模型 $P(U|\tau, \pi)$ 。这里 $P(U|\tau, \pi)$ 衡量了应对策略 π 在应对对手策略 τ 时的性能表现。然而，针对不同策略的应对策略的性能模型在多智能问题中可能是相同的，导致置信模型的不可区分性，从而导致检测不准确。为了解决这个问题，我们使用基于奖赏信号的置信模型和基于观察的对手模型同时检测对手的策略。

另一方面，在学习针对新对手的应对策略（例如DQN）时，BPR+会从头开始学习，这需要较长的学习时间，极大的降低学习效率。因此，我们提出蒸馏策略网络(distilled policy network, DPN)。这是一种使用策略蒸馏将多个应对策略组合到一个策略中的技术。此外，在学习新的应对策略时，可以使用提取的策略网络来初始化初始策略以加速学习过程，从而显着提高快速响应能力。

5.4 基于置信模型的对手策略检测

5.4.1 置信模型

检测对手策略是deep BPR+的关键组成部分，因为更高的检测准确性可以实现更有效的策略重用，从而提高性能。然而，传统BPR+算法中的置信模型是针对迁

移学习所设计的，其目的在于衡量不同任务之间的相似度。如果直接使用BPR+算法来解决多智能体问题，可能会遇到对手策略检测不准确的问题。

具体来说，公式5-2中的 $\beta^k(\tau) \equiv \beta^k(\tau|u^k, \pi^k)$ 描述的是，在第 k 个回合，当智能体使用策略 π 并接收到奖赏值 u^k 时，对手使用策略 τ 的概率。在 $k + 1$ 回合的开始，智能体通过推理对手的政策 τ^* 来选择最合适的应对策略。对手策略 τ^* 的推测公式如下：

$$\tau^* = \arg \max_{\tau} \beta^k(\tau) \quad (5-5)$$

上述公式中， τ^* 是唯一解的前提是当且仅当对于所有 $\tau_i \neq \tau^*$ 的满足 $\beta(\tau^*) > \beta(\tau_i)$ 的条件。但是，这种情况并不总是成立，因为置信模型仅使用公式5-2中的性能模型进行更新，这使得置信模型与性能模型成正比例关系，并高度依赖性能模型，具体见下公式：

$$\beta^k(\tau) \equiv \beta^k(\tau|u^k, \pi^k) \propto P(u^k|\tau, \pi^k) \quad (5-6)$$

假设在完全合作式环境中，一个智能体使用 π_1, \dots, π_n 分别与对手的策略 τ_1, \dots, τ_n 实现有效合作。在这种设定下，任何不匹配的策略组合都会导致奖赏值为0，如下列公式：

$$P(0|\tau_i, \pi_j) \simeq 1, \text{ 当 } i \neq j \text{ 且 } i, j \in [1, n] \quad (5-7)$$

假设在第 k 个回合，当一个智能体使用策略 π_j ，而其对手使用策略 τ_i ，则无法实现合作。考虑到关于每一个无法实现合作的策略组合 (τ_i, π_j) ，其性能模型是无法区分的，具体原因如下公式：

$$P(u = 0|\tau_1, \pi_i^k) = \dots = P(u = 0|\tau_{i-1}, \pi_i^k) = P(u = 0|\tau_{i+1}, \pi_i^k) = \dots = P(u = 0|\tau_n, \pi_i^k) \simeq 1 \quad (5-8)$$

这导致针对不同对手策略 $\tau_j (j \neq i)$ 的置信模型也是不可区分的，原因如下：

$$\beta^k(\tau_1) = \dots = \beta^k(\tau_{i-1}) = \beta^k(\tau_{i+1}) = \dots = \beta^k(\tau_n). \quad (5-9)$$

因此，在公式5-5中可能存在多个 τ^* 解。此时，将从多个解中随机选择一个执行，这将导致在随后连续的几个回合中始终无法有效合作。因此，仅从一个单一的角度并不足以准确推断对手的策略。

5.4.2 对手建模

为了克服这一问题，本文提出由 θ 参数化的“对手模型” $\hat{\tau}$ 。对手模型由神经

网络实现，用于拟合对手当前的真实策略 τ 。对手模型通过分析对手过去的行为，以实现对其策略的估计。

对手模型对于识别对手的策略类型是至关重要的^[111]。类似的想法也在近期的一些多智能体强化学习算法中有所体现（例如DRON^[106]和MADDPG^[108]）。然而，DRON需要使用手工制作的行为特征，而MADDPG需要观察所有智能体的行为数据，以便在没有明确对手分类的情况下策略学习。DPIQN^[109]算法对不同的对手策略都学习单独的特征信息，并用于训练一个泛化的Q网络，而不是直接复用一个更具有优势的响应网络。LOLA^[112]考虑对手也会同时在学习并优化器策略，这属于^[113]中定义的“心智模型”，该算法的策略优化求解需的计算复杂度较大，需要较大的计算资源。

在deep BPR+中，本文提出对对手的策略进行显示分类，并尝试通过重用最佳应对策略来实现更好的性能。假定 $(s_0, a_0, \dots, s_t, a_t, \dots)$ 是与使用策略 τ 的对手交互时的观察结果。此时对手模型 $\hat{\tau}$ 可以使用最大似然算法得到。但是，仅使用抽样观察可能很容易引起过拟合问题。此外，观察可能在不同回合之间变化很大，导致高方差。为了缓解这种情况，本文提出在损失函数中引入了熵正则化项，具体如下公式：

$$\mathcal{L}(\theta) = -\mathbb{E}_{s_i, a_i} [\log \hat{\tau}(a_i|s_i) + H(\hat{\tau})] \quad (5-10)$$

这里 H 代表策略 $\hat{\tau}$ 的熵，而 (s_i, a_i) 代表从观察中采样出来用于训练的样本。基于对手模型 $\hat{\tau}$ ，对手使用策略 τ_1 与使用策略 τ_2 之间的相似度就可以通过KL散度来衡量，衡量公式如下：

$$D_{KL}(\hat{\tau}_1, \hat{\tau}_2) \approx \mathbb{E}_{(s,a)} \log \left\{ \frac{\hat{\tau}_1(s,a)}{\hat{\tau}_2(s,a)} \right\} \quad (5-11)$$

因此，针对使用置信模型无法区分的不同对手策略，这里可以用对手模型进一步区分。具体来说，在和对手进行在线交互过程中，算法5（第5行）对对手在线策略进行了估计得到 $\hat{\tau}_o$ ，并用于计算对手当前使用策略 τ 的后验概率，计算公式如下：

$$p(\tau) \equiv p(\hat{\tau}|\hat{\tau}_o) = \sum_{\hat{\tau}_i \in \mathcal{T}} D_{KL}(\hat{\tau}_o, \hat{\tau}_i) / D_{KL}(\hat{\tau}_o, \hat{\tau}) \quad (5-12)$$

这里 $\hat{\tau}_i$ 是先前学习好的对手策略 $\tau_i \in \mathcal{T}$ 的近似。此外，KL散度的计算使用的是观察到的状态动作数据。注意到，由于对手的不同政策之间的相似程度与KL散度值成反比关系。因此上述公式使用KL散度的相对比例的倒数。

5.4.3 修正置信模型

直觉上，置信模型和对手模型都可以理解为衡量对手策略的后验概率，前者基于收到的奖励信号 u 进行概率判断，后者基于观察到的在线行为 $\hat{\tau}$ 进行概率判断。 $\beta(\tau)$ 和 $p(\tau)$ 是相互独立的，这是因为他们分别只依赖 u 和 $\hat{\tau}$ 。因此，本文提出一种修正置信模型用于衡量对手当前的策略是 τ 的概率。具体做法是将置信模型和对手模型相乘来获得一个更准确的预测模型，又叫做修正置信模型（Rectified Belief Model, RBM），计算公式如下：

$$\bar{\beta}(\tau) = \frac{1}{\eta} p(\tau) P(u^k|\tau, \pi^k) \beta^{k-1}(\tau) \quad (5-13)$$

这里 $\eta = \sum_{\tau' \in \mathcal{T}} p(\tau') P(u^k|\tau', \pi^k) \beta^{k-1}(\tau')$ 是归一化因子。RBM使用对手模型能够有效修正性能模型，并且选择更加合适的策略 π^* 以取得更高的长期累积收益，策略选择的公式如下：

$$\pi^* = \arg \max_{\pi \in \Pi} \int_{\bar{U}}^{+\infty} \sum_{\tau \in \mathcal{T}} \bar{\beta}(\tau) P(U|\tau, \pi) dU \quad (5-14)$$

最后，deep BPR+使用了BPR+中的移动平均回报来检测是否运上了一个未知策略(算法5第8行)。具体来说， r_π^t 表示智能体在回合 t 中的使用策略 π 所得到的累积奖赏值。接着，当对手使用策略 τ 时，获得奖赏值 r_π^t 的概率为 $p_t^\tau = P(r_\pi^t|\tau, \pi)$ 。如果在连续的 n 个回合内，对于所有的 $\tau \in \mathcal{T}$ ， $\sum_{i=t-n+1, \dots, t} p_i^\tau / n$ 的值都小于一个预定义的阈值 \mathcal{P}_{thr} ，那么deep BPR+认为对手正在使用一个未知的策略，然后deep BPR+会切换到学习阶段进行策略优化，旨在能够学习到最优的应对策略。直觉上， \mathcal{P}_{thr} 代表了对手正在使用一个已知策略的下界。一定程度上，该值也反映了deep BPR+算法对于对手正在使用一个位置策略的敏感程度。

5.5 基于策略蒸馏的策略优化

deep BPR+继承了使用在线学习的方式来学习如何对抗一个未知新对手的能力，然而传统的BPR+在遇到新的对手时每次都需要重新学习，这就导致了学习效率极度低下。对手策略之间存在着一定的相似性，这决定了相应的应对策略之间也存在一定的相似性。考虑到已经学习到的对手策略和当前对手使用的位置策略之间存在一定的相似性，因此在面对未知对手策略的时候，使用已经学习好的应对策略作为初始策略，并用于后续策略优化是比较合理的。

一种直接的实现方法如下：给定对手模型 $\hat{\tau}_o$ ，可以直接重用对手策略 π_i 作为初始策略，注意这里 π_i 是对手模型 $\hat{\tau}_i$ 的应对策略，且 $\hat{\tau}_i$ 有着最高的相似度值 $D_{KL}(\hat{\tau}_o, \hat{\tau}_i)$ 。使用学习好的应对策略作为厨师策略进行策略优化，有助于提升学习的效率。但也会导致探索不充分，最终得到次优的解决方案。为解决这一问题，本文提出使用蒸馏策略网络进行有效的策略优化，同时避免收敛到次优解。

5.5.1 策略蒸馏

策略蒸馏(Policy distillation, PD)^[114]常用于将多个用于解决不同任务的策略压缩到一个单独的策略网络中。主要思想是通过知识迁移（例如Q值）将知识从教师模型 ψ 中迁移到学生模型 ϕ 中。教师模型 ψ 用于生成数据集 $\mathcal{D}^\psi = \{(s_i, \mathbf{q}_i)\}_{i=0}^N$ ，这些数据集中每一个样本都由一个状态 s_i 和向量 \mathbf{q}_i 组成。而 \mathbf{q}_i 是由每个动作 a 对应的Q值组成。PD使用监督学习来训练学生模型 ϕ ，优化目标是使学生模型 ϕ 与教师模型 ψ 拥有相似的输出，使用的训练样本 (s, a, r, s') 采样自 \mathcal{D}^ψ 。此外PD使用带温度 t 的KL-散度来衡量学生模型 ψ 与教师模型 ϕ 之间的差异，具体计算公式如下：

$$Loss_{KL}(D^\psi, \theta_\phi) = \sum_{i=1}^{|D^\psi|} \text{softmax}\left(\frac{\mathbf{q}_i^\psi}{t}\right) \ln \frac{\text{softmax}\left(\frac{\mathbf{q}_i^\psi}{t}\right)}{\text{softmax}(\mathbf{q}_i^\phi)} \quad (5-15)$$

这里假定 \mathbf{z} 表示的是一个向量，则 $\text{softmax}(\mathbf{z})$ 中的第 i^{th} 个元素的定义是：

$$\exp \mathbf{z}(i) / \sum_j \exp \mathbf{z}(j) \quad (5-16)$$

5.5.2 蒸馏策略网络

图5-1是DPN的总体架构图，DPN由一个共享的卷积层和多个独立分离的控制层组成。每一个控制层都是针对具体某一个应对策略进行训练而得到的。运行时会将不同的表情输入到DPN中来实现应对策略的切换，具体来说根据不同的标签，DPN会将共享卷积层和与标签对应的控制层相连接，实现快速的策略切换。具体来说，针对 n 个不同对手的应对策略是分开训练的并且用不同的标签加以区分。由不同应对策略生成的、用于训练的样本（例如Q值）也都和对应的标签相关联上。为了将多个应对策略压缩到一个单独的蒸馏策略网络中，本文使用了监督学习来最小化蒸馏误差，整个监督学习同时过程使用了 n 种不同不同标签的训练样本。

直观上说，相较于为每一个应对策略训练一个独立的卷积层，这个架构有助于使得卷积层学习到更加一般化的特征，更好的刻画环境，并以此来对抗不同的

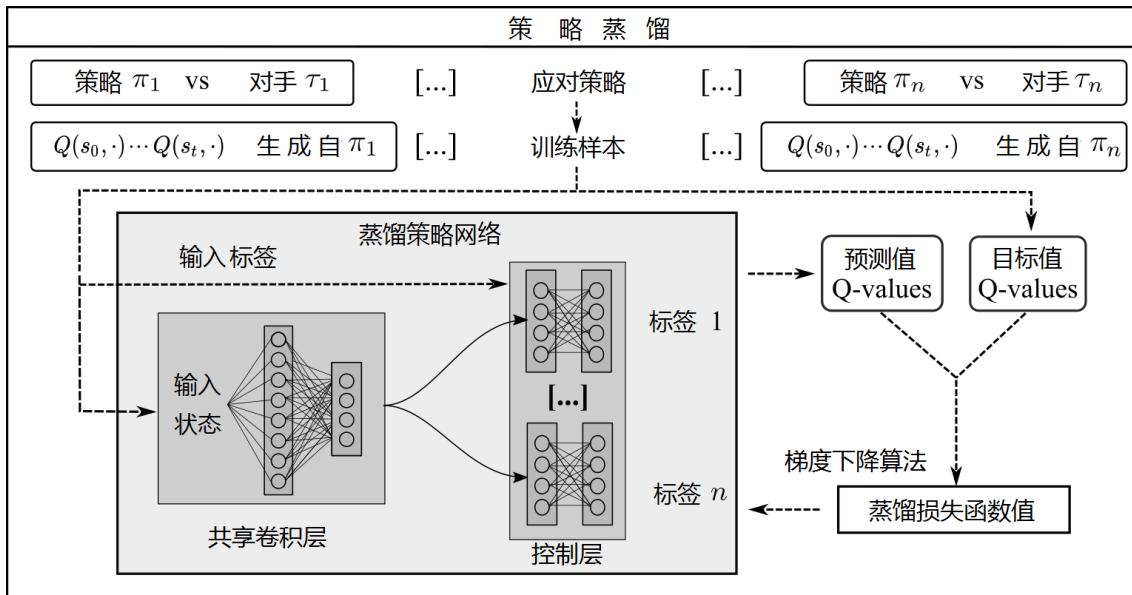


图 5-1 DPN的总体网络架构图

对手。图5-2所示，当遇到新的对手时，起始策略是通过将已学习的共享卷积层与DPN用一个随机初始化的控制层进行连接从而初始化的。与直接重复使用应对策略相比，deep BPR+使用起始开始策略优化，获得了更高的学习效率，此外也呈现了较强的鲁棒性。学习到应对策略后，需要使用策略蒸馏来更新蒸馏策略网络。注意到，deep BPR+在线学习的过程中是不需要存储训练数据的，因为训练数据可以使用DPN重新生成（当需要使用策略来蒸馏来处理新的应对策略时）。另一优点是，DPN使用了共享卷积层，这使得其空间使用率较高，适用于对空间相对敏感的问题。

值得注意的是，deep BPR+与DRON^[106]算法存在一些不同，主要包括如下几点：首先，deep BPR+使用DPN来保证有准确的一对一应对策略，而DRON使用端对端训练响应的子网络，这使得DRON不能保证能够准确高效的应对特定类型的对手并取得较高的长期奖赏回报。其次，在DRON中，可应对的对手数量K是固定的，因此不能够应对对手数量动态变化的情况。相反，deep BPR+具有较高的灵活性，可以增加任意新的应对策略至策略库中，以应对位置的新对手，通过在线的形式，实现持续性的性能提升。最后，在策略切换方面，deep BPR+更具有泛化性，因为deep BPR+不需要除对手过往行为之外的任何信息。然而DRON一般需要额外的人工特征，而这些人工特征很难在不同的问题都保持良好的效果。

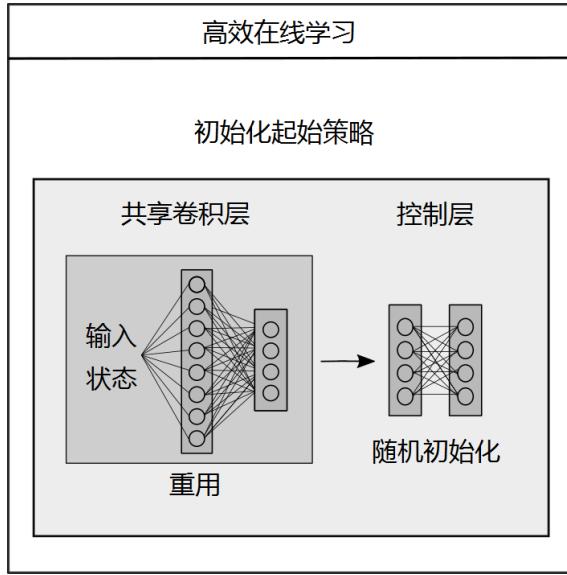


图 5-2 在线学习其实策略初始化示意图

5.6 实验论证

本节对文献^[115]的格子世界游戏、文献^[116]的导航游戏和文献^[87,106]的足球游戏改造后进行了实证实验，并通过对比BPR^[93]、BPR+^[107]和deep BPR+的效果来验证其性能。在对使用图片作为输入的多智能体强化学习算法进行比较时，为保证对比公平，BPR和BPR+也将使用神经网络作为函数估计器。本文使用一种全知智能体作为基准，该智能体拥有应对非静态对手的最佳响应策略，能够对非静态对手的所有行为作出最佳响应。在所有涉及非静态对手的多智能体的环境中，deep BPR+在对手检测准确率、累计收益和新应对策略学习速度等方面的有效性都得到了实证验证。

5.6.1 环境描述

图5-3为格子世界游戏示意图，其中，两个智能体（A和O）需要在避免冲突的前提下进入各自的目标单元（G(A)和G(O))。每个智能体拥有五个行动选项：北，南，东，西，不动。除了遇到网格边缘或厚壁上不会移动之外，每次移动都会使得智能体以相应的方向移动到相邻网格中。一旦进入一个目标单元，智能体将获得+5的正奖励，并将一直停留在那里直到该回合结束。进入非目标位置的得分为0。此外，如果两个智能体试图进入相同的单元，他们的位置将会保持不变，并

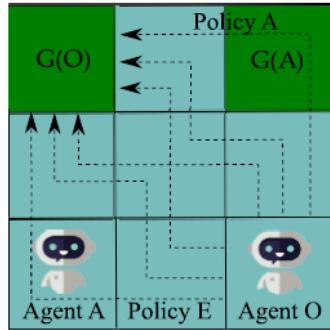


图 5-3 合作式格子游戏

且得到-1的分数作为惩罚。只有当两个智能体都到达他们的目标单元，这一回合才会结束。

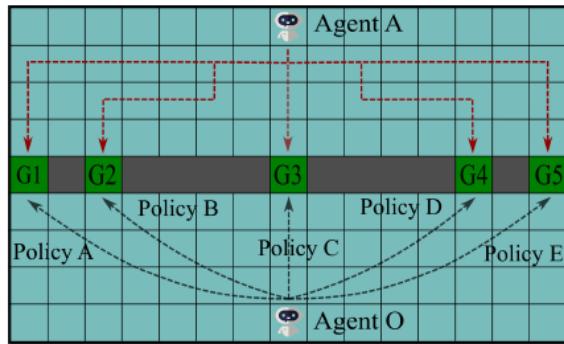


图 5-4 合作导航游戏

图5-4合作导航游戏的示意图。游戏设置与格子游戏总体相同，仅存在以下两点区别：首先，有一堵厚墙（灰色）将区域划分为上下两部分，两个智能体需要进入相同的标有“G”的目标单元（图中绿色单元）。其次，如果两个智能体同时进入相同的目标单元，将获得正的奖励，该回合结束；否则，会收到一个微小的负奖赏值作为没有合作成功的惩罚（该奖赏值的大小和游戏结束时两个智能体之间的欧几里得距离成正比）。

图5-5描述了一个 5×5 足球游戏，两个智能体都试图从对方那抢到球（图中灰色圆形）并将球带到各自的得分区域。与前文设置不同的是，如果两个智能体都移动到同一单元，球的占有权将发生互换，而两个智能体的位置没有发生改变。当一个智能体将球带到它的分区，那么它将获得+10的奖励，而另一方将获得-10的惩罚，该回合结束，游戏重置为智能体O拥有持球权初始状态。

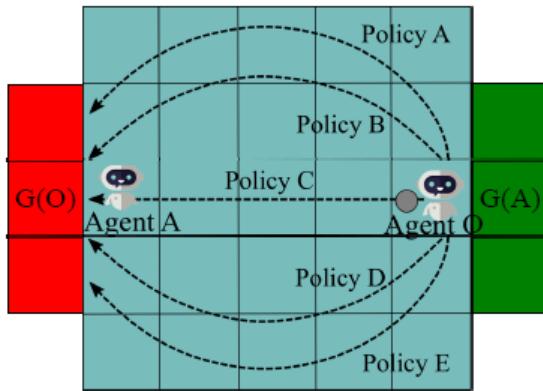


图 5-5 竞争式足球游戏

直觉上，所有的测试环境中，智能体A能通过准确地检测对方的意图，从而实现更高的算法性能。

5.6.2 非静态对手的策略检测

在图5-3、图5-4和图5-5所示的三个游戏中，智能体O拥有6, 5, 5个随机初始策略（例如图5-4中有策略A、B、C、D、E五个策略），并且每隔几个回合进行策略切换。智能体A具有预先训练好的针对该对手的应对策略，并且其目标是通过观测智能体O的行为，从而重用最适合对抗智能体O的应对策略。为了提高对手策略检测度的准确性，deep BPR+(D)在面对非静态对手时候，使用了RBM机制进行对手策略检测以及应对策略的选择。

图5-6展示了针对对手策略检测的平均准确度，相比于BPR+，deep BPR+(D)的表现更接近于全知智能体，并且检测准确度更高。值得注意的是，当一个非静态

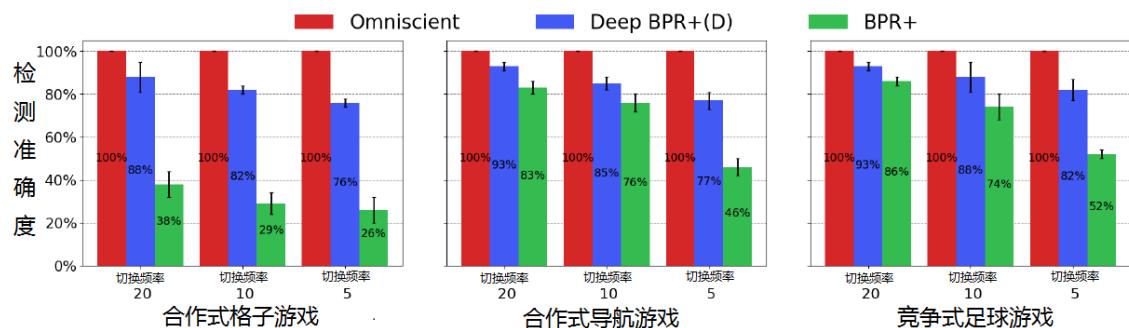


图 5-6 面对非静态对手时各个算法的平均检测准确度对比实验（使用了10个随机种子）

智能体更频繁地转换它的策略时，BPR+的检测准确度急剧降低，而deep BPR+的准确度则保持在相对较高的水平。RBM之所以能够克服传统BPR+的缺陷从而提高检测准确度的主要原因包括：1) 一旦原始的BPR+判断对手策略失败，就需要遍历对手的所有已知策略，这将耗费大量时间；2) 在传统BPR+算法找到正确的应对策略之前，对手可能再次转换它的策略，这将扰乱内部信念模型的更新，导致后续检测准确度的降低。实验证明，使用RMB，deep BPR+(D)能够通过准确选择并执行最佳的应对策略，从而得到更好的表现，尤其是当对手频繁地转换策略时。

5.6.3 针对未知策略的有效学习

在本次实验中，本文旨在检验在对抗使用新未知策略的智能体O时deep BPR+的学习效率。与前文设置不同之处在于，起始时智能体A只知道如何应对智能体O除了策略A和E之外的任何策略，并且需要以在线的形式学习新的应对策略。根据文献^[107]的既有研究，本文假设在智能体A学会新应对策略前，智能体O不会转换它的策略。在其他情况下，智能体O在执行一个策略5-20次时将会随机转换策略。图5-7为在线学习速度和累计奖励的对比情况。

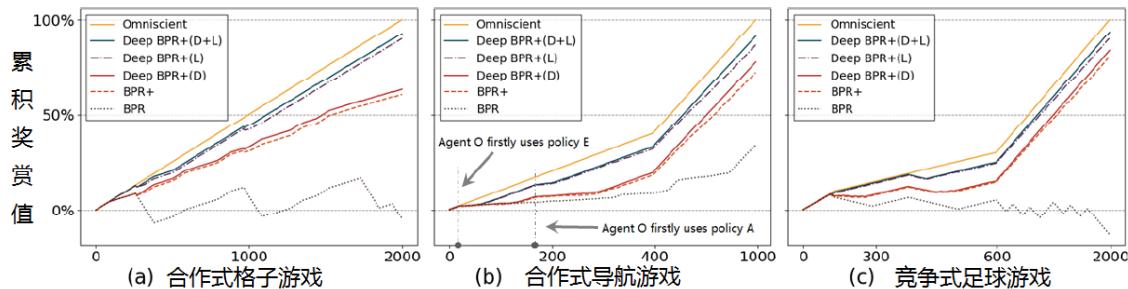


图 5-7 针对基于BPR的算法对比，包括BPR，BPR+，deep BPR+(D)（只在检测阶段使用了RBM机制），deep BPR+(L)（只在学习阶段使用了DPN），deep BPR+(D+L)（同时使用了RBM和DPN机制）。所有算法获得的奖赏值都是使用全知智能体获得的奖赏值进行了标准化

BPR在三个游戏中的表现较差，因为其无法对智能体O的未知策略做出正确的检测和响应，而其他两者则能够检测并学习相应的应对策略。值得注意的是，由于DPN能够提高学习效率，deep BPR+(D+L)和deep BPR+(L)的得分比deep BPR+(D)和BPR+高。例如，见图4(b)所示，智能体O在大约第20回合的时候首先使用了智能体A未知的策略E，deep BPR+(D+L)和BPR+(L)能够检测到这个未知

策略，并且利用DPN初始化起始策略，从而高效学习相应的应对策略；而deep BPR+(D)和BPR+的策略优化过程耗时则相当长。类似的情况发生在第175回合，智能体O采用了另一个未知策略A。同样的，在随后的交互中，无论智能体O重新使用策略A或E，DPN都能高效地重复利用学习到的应对策略。此外，无论是否使用DPN，RBM都能够取得比原始的置信模型更好的效果。

为了进一步证明DPN对于学习对抗未知策略的作用，本文对比了直接基于DPN初始化的起始策略进行优化和直接基于学习到的应对策略进行优化两种模式，如图5-8所示。为了进行公平的对比，本文不重复利用所有学习到的应对策略，而是仅重复利用它们的卷积层。实验结果证明，无论是面对未知策略A或E，相比于使用应对策略直接优化而言，使用DPN初始化的起始策略的结果更优且更稳定（方差较小）。实验的另一个结论是，即使有时候使用一个已知应策略作为初始策略可能得到相似的结果（例如，图5(b)的应对策略 π_A ），然而一旦选择了不合适的应对策略都将显著降低在线学习的效果（例如，图5的 π_B , π_C 或 π_D ）。此外，如何选择应对策略作为起始策略也十分困难。相反的，DPN能够提供更普适和简练的方式来初始化起始策略，获得前景更好的表现，而无须考虑选择哪种应对策略。

上述结果表明，DPN面对不同未知策略时的表现和鲁棒性更优。为了探究其中的内在机理，图5-9可视化地展示了，当面对未知策略A和E时，使用不同应对策略 π 以及DPN作为初始策略进行策略优化过程的最后200回合的轨迹。公平起见，学习过程中的探索率（exploration rate）都是相同的。直觉上，利用已学习的策略作为作为初始策略趋向于困在局部最优解，而DPN能够避免这种情况，从而获得更好的表现。例如，当学习针对未知策略A的应对策略时，应对策略 π_B , π_C , π_D 和 π_E 似乎都在探索错误的方向，导致了无效的探索（图5-9(a-d)）。策略 π_A , π_C 和 π_D 在学习针对策略E的应对时也出现了类似的情况（图5-9(f, h, i)）。相反，

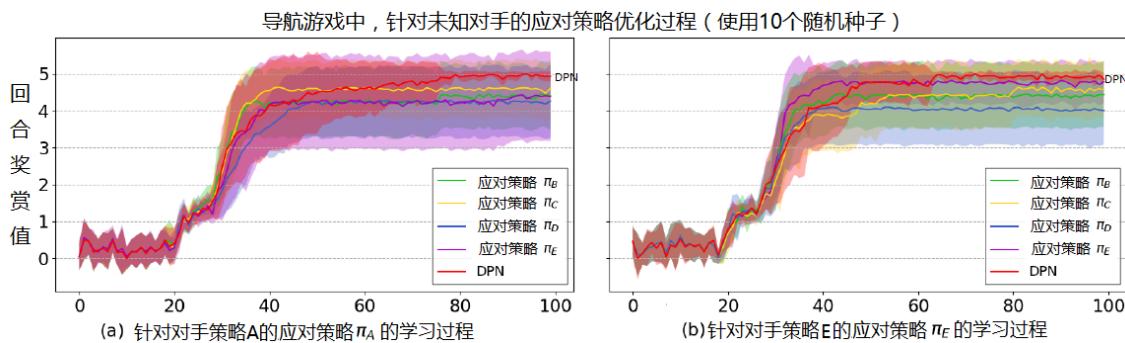


图 5-8 使用不同策略作为起始策略进行学习的效率对比实验图

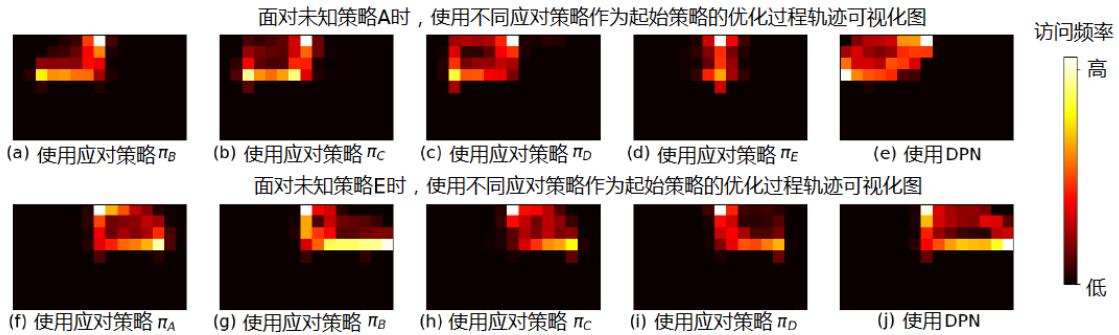


图 5-9 在导航游戏中面对一个未知对手时的策略优化路径可视化结果图

使用相同的探索率，DPN能够获得更高的探索效率，从而获得更好的结果。本文认为这是由于网络结构允许共享卷积层学习到了更具有普适性的特征，以应对不同的对手，更好地描述环境并且引导智能体朝正确的方向探索。

5.7 本章小结

本章提出了一种面向非静态多智能体的deep BPR+算法，该算法主要使用神经网络与贝叶斯策略重用相结合，实现快速高效的对手策略检测，并通过重用既有的应对策略来应对对手，实行长期受益最大化。

针对传统BPR+算法只使用奖赏值来更新内部置信模型可能会引起对手策略检测不准确的问题，deep BPR+提出了对手模型，并用于修正传统的置信模型，提出了新的修正置信模型RBM。RBM同时从接收到的奖赏值信号和对手的行为两个角度实现准确的策略检测。

此外本章还提出使用蒸馏策略网络DPN作为策略库，DPN网络充当了deep BPR+算法中策略库的决策，能够实现对既有应对策略的快速切换，同时也是一個高效的策略存储方案。另一方面，使用DPN来初始化起始策略用于在线学习，极大的提高了学习效率，减少了策略优化的时间。

实验结果表明，在三种复杂的马尔可夫博弈中，相比现有的其它基准测试方法，deep BPR+算法确实取得了较好的性能。

在未来的工作中，我们将研究当面对更加复杂的动态智能体（其行为随着时间一直在不断变化，而不是目前只在回合之间变化）如何有效地做出最优决策。

第6章 总结与展望

6.1 研究内容总结

本文聚焦于基于深度强化学习的多智能体策略优化研究，从环境、强化学习算法以及多智能体三个角度展开，对既有算法存在的局限性进行分析，并提出相应的解决方法。论文的主要工作内容如下：

首先，本文针对面向多模态信息输入下的智能体策略优化问题展开研究，主要论述了如何将多模态学习与深度强化相结合，以改进传统强化学习算法使用单模态输入可能潜在的信息不足的问题。首先，提出了分离式多模态输入的强化学习框架，拓展了一般强化学习算法处理信息输入的能力，实现了针对多种模态信息的有效处理。针对多模态强化学习中存在的注意力分配问题，其次，提出了层次注意力机制，实现了在多模态信息间以及各个模态信心内部的注意力权重分配方法，弥补了一般注意力机制只处理单模态信息的缺陷，实现了对多模态信息输入的层次化注意力分配，有效提高策略优化效率。最后，针对多模态输入信息融合的问题，提出使用基于LSTM网络的多模态融合机制进行信息融合，有效提高了算法处理多模态信息的能力。

其次，本文针对面向噪声环境下的独立学习智能体策略优化问题展开研究，主要论述了在带噪声的多智能体环境中，如何有效的降低环境中的噪声并促进独立学习智能体之间有效地进行策略协同优化，并提高收敛到帕累托最优纳什均衡策略的概率。本章解决了策略优化过程中存在的噪声挑战、估值偏差挑战以及独立学习智能体协同优化的挑战。首先，针对环境中噪声的挑战，提出了奖赏值网络对环境信息进行拟合，有效解决环境中的噪声问题。其次，针对估值偏差的挑战，提出了使用基于双权Q网络的强化学习算法来降低估值偏差，有效地进行估值纠偏。再次，关于独立学习智能体之间协同策略优化的挑战，本章将宽容的思想与奖赏值网络想结合，提出宽容奖赏值网络，有效地促使独立学习的多智能体之间有效的进行协同策略优化，提升了学习到帕累托最优纳什均衡的概率，实现群体利益最大化。最后，针对策略优化的效率问题，提出了调度经验回放策略，

有效地提高了多智能体策略优化的效率。

最后，本文针对面向非静态智能体的策略优化问题展开研究，主要论述了在多智能体环境下，如何应对不断变化行为智能体的挑战，以实现长期收益最大化。首先，基于贝叶斯理论提出了深度贝叶斯策略重用算法，实现了有效的对手策略变化检测以及准确的对手类型判别，快速切换并执行针对该对手的最佳响应策略。其次，为了提高对手策略检测度的准确性，提出的使用对手建模机制，用于修正贝叶斯策略重用算法中的原始策略检测机制，进一步提高了检测准确度。最后，本章提出了使用策略蒸馏算法，对多个相应策略蒸馏得到单一的策略整流网络。策略整流网络不仅能够实现快速的策略切换，有效地降低了空间使用率，还显著提升了针对未知对手进行策略优化时的优化效率。实验结果表明，在三种复杂的马尔可夫博弈中，相比现有的其它基准测试方法，深度贝叶斯策略重用算法能够取得较好的性能。

6.2 展望

本文主要针对深度强化学习与多智能体系统展开研究，尝试解决智能体的策略优化问题，对既有算法存再的局限性，提出了解决方法呢，基于本文的研究结论，未来将在以下几方面进行更深入的研究：

首先，本文提出的基于多模态信息输入的强化学习框架的有效性目前只使用了单智能体环境进行验证，未来需要考虑使用更加复杂的多智能体场景来进一步验证其有效性。

其次，本文提出的WDDQN(LRN+SRS)算法，虽然能够有效的降低估值偏差，实现多智能体策略优化，但仍存在一定的局限性，例如其只使用了单线程来进行策略优化，难以适应大规模的强化学习问题对算法有效性的需求。未来可考虑改进WDDQN算法的加速机制，以提高WDDQN的算法性能。

再次，本文针对非静态对手而提出的deep BPR+算法在多种复杂马尔可夫博弈中均表现出了较好的性能，但目前只实现了智能体在回合之间变化。未来可考虑针对更加复杂的动态智能体，研究如何更有效地做出最优决策。

最后，本文的工作大部分是针对独立学习智能体的设定展开，而真实情况中，智能体往往可以拥有一定关于其余智能体的观测信息，并且这有助于智能体进行策略优化，因此，未来的工作可考虑在独立学习智能体的基础上，如何有效地利

用有限的观测信息，进行多智能体策略优化。

参考文献

- [1] Silver D, Schrittwieser J, Simonyan K, et al. Mastering the game of Go without human knowledge[J]. *Nature*, 2017, 550(7676):354–359.
- [2] Silver D, Hubert T, Schrittwieser J, et al. A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play[J]. *Science*, 2018, 362(6419):1140–1144.
- [3] Mnih V, Kavukcuoglu K, Silver D, et al. Human-level control through deep reinforcement learning[J]. *Nature*, 2015, 518(7540):529–533.
- [4] Hu Y, Da Q, Zeng A, et al. Reinforcement learning to rank in e-commerce search engine: Formalization, analysis, and application[C]. *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery Data Mining*, 2018:368–377.
- [5] Selvaraju R R, Cogswell M, Das A, et al. Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization[C]. *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017:618–626.
- [6] Caicedo J C, Lazebnik S. Active Object Localization with Deep Reinforcement Learning[C]. *2015 IEEE International Conference on Computer Vision (ICCV)*, 2015:2488–2496.
- [7] Andreas J, Rohrbach M, Darrell T, et al. Learning to Compose Neural Networks for Question Answering[C]. *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2016:1545–1554.
- [8] Li J, Monroe W, Ritter A, et al. Deep Reinforcement Learning for Dialogue Generation[C]. *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, 2016:1192–1202.
- [9] Deisenroth M P, Fox D, Rasmussen C E. Gaussian Processes for Data-Efficient Learning in Robotics and Control[J]. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2015, 37(2):408–423.
- [10] Zhang F, Leitner J, Milford M, et al. Towards vision-based deep reinforcement learning for robotic motion control[J]. *arXiv preprint arXiv:1511.03791*, 2015.
- [11] Ebert F, Finn C, Dasari S, et al. Visual Foresight: Model-Based Deep Reinforcement Learning for Vision-Based Robotic Control.[J]. *arXiv preprint arXiv:1812.00568*, 2018.
- [12] Shalev-Shwartz S, Shammah S, Shashua A. Safe, Multi-Agent, Reinforcement Learning for Autonomous Driving.[J]. *arXiv preprint arXiv:1610.03295*, 2016.

- [13] Sallab A E, Abdou M, Perot E, et al. Deep Reinforcement Learning framework for Autonomous Driving[J]. electronic imaging, 2017, 2017(19):70–76.
- [14] Reddy P P, Veloso M M. Strategy learning for autonomous agents in smart grid markets[C]. IJCAI'11 Proceedings of the Twenty-Second international joint conference on Artificial Intelligence - Volume Volume Two, 2011, 2011:1446–1451.
- [15] Yang Y, Hao J, Sun M, et al. Recurrent Deep Multiagent Q-Learning for Autonomous Brokers in Smart Grid[C]. IJCAI 2018: 27th International Joint Conference on Artificial Intelligence, 2018:569–575.
- [16] Yang Y, Hao J, Wang Z, et al. Recurrent Deep Multiagent Q-Learning for Autonomous Agents in Future Smart Grid[C]. Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems, 2018:2136–2138.
- [17] Sadek A, Basha N. Self-Learning Intelligent Agents for Dynamic Traffic Routing on Transportation Networks[J]. 2010:503–510.
- [18] 郭禹, 朱大鹏. 基于多目标规划的城市垃圾运输车辆调度问题研究 Study of City Garbage Transportation Vehicle Scheduling Problem Based on Multiple Objective Programming[C]. [S.l.]: Scientific Research Publishing, 2014, 2014.
- [19] Li Z, Liu Y, Tang P, et al. Stability of Generalized Two-sided Markets with Transaction Thresholds[J]. adaptive agents and multi agents systems, 2017:290–298.
- [20] Zhao X, Zhang L, Ding Z, et al. Deep Reinforcement Learning for List-wise Recommendations.[J]. arXiv preprint arXiv:1801.00209, 2018.
- [21] Sutton R S, Barto A G. Reinforcement Learning: An Introduction[J]. 1988.
- [22] Shoham Y, Leyton-Brown K. Multiagent Systems: Algorithmic, Game-Theoretic, and Logical Foundations[M], 2008.
- [23] Busoniu L, Babuska R, Schutter B D. A Comprehensive Survey of Multiagent Reinforcement Learning[J]. systems man and cybernetics, 2008, 38(2):156–172.
- [24] Tampuu A, Matiisen T, Kodelja D, et al. Multiagent cooperation and competition with deep reinforcement learning[J]. PLOS ONE, 2017, 12(4).
- [25] Omidshafiei S, Kim D K, Pazis J, et al. Crossmodal Attentive Skill Learner[C]. Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems, 2018:139–146.
- [26] Zhang Z, Pan Z, Kochenderfer M J. Weighted Double Q-learning[C]. International Joint Conference on Artificial Intelligence, 2017:3455–3461.
- [27] Thrun S, Schwartz A. Issues in using function approximation for reinforcement learning[C]. Proceedings of the 1993 Connectionist Models Summer School Hillsdale, NJ. Lawrence Erlbaum, 1993.

参考文献

- [28] van Hasselt H, Guez A, Silver D. Deep reinforcement learning with double Q-Learning[J]. national conference on artificial intelligence, 2016:2094–2100.
- [29] Matignon L, Laurent G J, Le Fort-Piat N. Independent reinforcement learners in cooperative markov games: a survey regarding coordination problems[J]. The Knowledge Engineering Review, 2012, 27(1):1–31.
- [30] Hernandez-Leal P, Kaisers M, Baarslag T, et al. A Survey of Learning in Multiagent Environments: Dealing with Non-Stationarity.[J]. arXiv preprint arXiv:1707.09183, 2017.
- [31] Sutton R S. Learning to Predict by the Methods of Temporal Differences[J]. Machine Learning, 1988, 3(1):9–44.
- [32] Yang G, SHIFU C, XIN L. Research on Reinforcement Learning Technology: A Review[J]. 城市规划, 2004(30(1)):86–100.
- [33] Bellman R E. A Markovian Decision Process[J]. Indiana University Mathematics Journal, 1957, 6(4):679–684.
- [34] Balaji P G, Srinivasan D. An Introduction to Multi-Agent Systems[J]. 2010:1–27.
- [35] Horling B, Lesser V R. A survey of multi-agent organizational paradigms[J]. Knowledge Engineering Review, 2004, 19(4):281–316.
- [36] Claus C, Boutilier C. The dynamics of reinforcement learning in cooperative multiagent systems[C]. AAAI Conference on Artificial Intelligence, 1998:746–752.
- [37] Melo F S, Lopes M C. Convergence of Independent Adaptive Learners[C]. Progress in Artificial Intelligence, 13th Portuguese Conference on Artificial Intelligence, EPIA 2007, Workshops: GAIW, AIASTS, ALEA, AMITA, BAOSW, BI, CMBSB, IROBOT, MASTA, STCS, and TEMA, Guimarães, Portugal, December 3-7, 2007, Proceedings, 2007:555–567.
- [38] Finn C, Abbeel P, Levine S. Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks.[J]. international conference on machine learning, 2017:1126–1135.
- [39] Vinyals O, Blundell C, Lillicrap T P, et al. Matching networks for one shot learning[J]. neural information processing systems, 2016:3637–3645.
- [40] Andrychowicz M, Denil M, Gomez S, et al. Learning to learn by gradient descent by gradient descent[J]. neural information processing systems, 2016:3981–3989.
- [41] Nagabandi A, Kahn G, Fearing R S, et al. Neural Network Dynamics for Model-Based Deep Reinforcement Learning with Model-Free Fine-Tuning[C]. 2018 IEEE International Conference on Robotics and Automation (ICRA), 2018:7559–7566.
- [42] Chebotar Y, Hausman K, Zhang M, et al. Combining Model-Based and Model-Free Updates for Trajectory-Centric Reinforcement Learning[J]. international conference on machine learning, 2017:703–711.
- [43] Ebert F, Finn C, Lee A X, et al. Self-Supervised Visual Planning with Temporal Skip Connections.[J]. arXiv preprint arXiv:1710.05268, 2017:344–356.

-
- [44] Watkins C J C H, Dayan P. Technical Note : Q -Learning[J]. Machine Learning, 1992, 8(3):279–292.
 - [45] Asis K D, Hernandez J, Holland G, et al. Multi-Step Reinforcement Learning: A Unifying Algorithm[J]. national conference on artificial intelligence, 2018:2902–2909.
 - [46] Hasselt H V. Double Q-learning[C]. Advances in Neural Information Processing Systems, 2010:2613–2621.
 - [47] Williams R J. Simple Statistical Gradient-Following Algorithms for Connectionist Reinforcement Learning[J]. Machine Learning, 1992, 8(3):229–256.
 - [48] Silver D, Lever G, Heess N, et al. Deterministic policy gradient algorithms[C]. Proceedings of the 31th International Conference on Machine Learning, 2014.
 - [49] Konda V, Tsitsiklis J N. Actor-critic algorithms[M], 2002.
 - [50] Mnih V, Kavukcuoglu K, Silver D, et al. Playing Atari with Deep Reinforcement Learning[J]. arXiv preprint arXiv:1312.5602, 2013.
 - [51] Lillicrap T P, Hunt J J, Pritzel A, et al. Continuous control with deep reinforcement learning[J]. International conference on learning representations., 2016.
 - [52] Mnih V, Badia A P, Mirza M, et al. Asynchronous methods for deep reinforcement learning[C]. International conference on machine learning, 2016:1928–1937.
 - [53] Schulman J, Moritz P, Levine S, et al. High-Dimensional Continuous Control Using Generalized Advantage Estimation[J]. international conference on learning representations, 2016.
 - [54] Babaeizadeh M, Frosio I, Tyree S, et al. Reinforcement Learning through Asynchronous Advantage Actor-Critic on a GPU[J]. international conference on learning representations, 2017.
 - [55] Todorov E, Erez T, Tassa Y. MuJoCo: A physics engine for model-based control[J]. 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2012:5026–5033.
 - [56] Hausknecht M J, Stone P. Deep Reinforcement Learning in Parameterized Action Space[J]. International Conference on Learning Representations, 2016.
 - [57] Ngiam J, Khosla A, Kim M, et al. Multimodal deep learning[C]. Proceedings of the 28th international conference on machine learning (ICML-11), 2011:689–696.
 - [58] Chambers A, Scherer S, Yoder L, et al. Robust multi-sensor fusion for micro aerial vehicle navigation in GPS-degraded/denied environments[C]. 2014 American Control Conference, 2014:1892–1899.
 - [59] Lynen S, Achtelik M W, Weiss S, et al. A robust and modular multi-sensor fusion approach applied to mav navigation[C]. 2013 IEEE/RSJ international conference on intelligent robots and systems, 2013:3923–3929.
 - [60] Nobili S, Camurri M, Barasuol V, et al. Heterogeneous Sensor Fusion for Accurate State Estimation of Dynamic Legged Robots[J]. 2017, 13.

- [61] Beal M J, Attias H, Jojic N. Audio-video sensor fusion with probabilistic graphical models[C]. European Conference on Computer Vision, 2002:736–750.
- [62] Bengio S. An asynchronous hidden markov model for audio-visual speech recognition[C]. Advances in Neural Information Processing Systems, 2003:1237–1244.
- [63] Srivastava N, Salakhutdinov R R. Multimodal learning with deep boltzmann machines[C]. Advances in neural information processing systems, 2012:2222–2230.
- [64] Eitel A, Springenberg J T, Spinello L, et al. Multimodal deep learning for robust RGB-D object recognition[C]. 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2015:681–687.
- [65] Kiros R, Salakhutdinov R, Zemel R S. Unifying Visual-Semantic Embeddings with Multimodal Neural Language Models[J]. arXiv preprint arXiv:1411.2539, 2014.
- [66] Bahdanau D, Cho K, Bengio Y. Neural Machine Translation by Jointly Learning to Align and Translate[J]. international conference on learning representations, 2015.
- [67] Luong T, Pham H, Manning C D. Effective Approaches to Attention-based Neural Machine Translation[C]. Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, 2015:1412–1421.
- [68] Caglayan O, Barrault L, Bougares F. Multimodal Attention for Neural Machine Translation.[J]. arXiv preprint arXiv:1609.03976, 2016.
- [69] Hausknecht M J, Stone P. Deep Recurrent Q-Learning for Partially Observable MDPs.[J]. national conference on artificial intelligence, 2015:29–37.
- [70] Sorokin I, Seleznev A, Pavlov M, et al. Deep Attention Recurrent Q-Network[J]. arXiv preprint arXiv:1512.01693, 2015.
- [71] Mnih V, Heess N, Graves A, et al. Recurrent Models of Visual Attention[J]. neural information processing systems, 2014:2204–2212.
- [72] Ba J L, Mnih V, Kavukcuoglu K. Multiple Object Recognition with Visual Attention[J]. international conference on learning representations, 2015.
- [73] Ke N R, Goyal A G A P, Bilaniuk O, et al. Sparse Attentive Backtracking: Temporal Credit Assignment Through Reminding[J]. neural information processing systems, 2018:7651–7662.
- [74] Oh J, Chockalingam V, Singh S P, et al. Control of memory, active perception, and action in minecraft[J]. international conference on machine learning, 2016:2790–2799.
- [75] Harb J, Bacon P L, Klissarov M, et al. When Waiting is not an Option : Learning Options with a Deliberation Cost[J]. national conference on artificial intelligence, 2018:3165–3172.
- [76] Vezhnevets A S, Osindero S, Schaul T, et al. FeUdal Networks for Hierarchical Reinforcement Learning[J]. international conference on machine learning, 2017:3540–3549.

- [77] Gupta J K, Egorov M, Kochenderfer M J. Cooperative Multi-agent Control Using Deep Reinforcement Learning[C]. International Conference on Autonomous Agents and Multiagent Systems, 2017:66–83.
- [78] Lanctot M, Zambaldi V, Gruslys A, et al. A unified game-theoretic approach to multiagent reinforcement learning[C]. Advances in Neural Information Processing Systems, 2017:4193–4206.
- [79] Potter M A, Jong K A D. A Cooperative Coevolutionary Approach to Function Optimization[C]. PPSN III Proceedings of the International Conference on Evolutionary Computation. The Third Conference on Parallel Problem Solving from Nature: Parallel Problem Solving from Nature, 1994:249–257.
- [80] Bloembergen D, Tuyls K, Hennes D, et al. Evolutionary dynamics of multi-agent learning: a survey[J]. Journal of Artificial Intelligence Research, 2015, 53(1):659–697.
- [81] Palmer G, Tuyls K, Bloembergen D, et al. Lenient Multi-Agent Deep Reinforcement Learning[C]. Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems, 2018:443–451.
- [82] Panait L, Sullivan K, Luke S. Lenient learners in cooperative multiagent systems[C]. Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems, 2006:801–803.
- [83] Wei E, Luke S. Lenient learning in independent-learner stochastic cooperative games[J]. Journal of Machine Learning Research, 2016, 17(1):2914–2955.
- [84] Schaul T, Quan J, Antonoglou I, et al. Prioritized Experience Replay[C]. International Conference on Learning Representations, 2016.
- [85] Brownlee O H, Koopmans T C. Activity Analysis of Production and Allocation[J]. Econometrica, 1952, 20(1):111.
- [86] Bowling M H, Veloso M M. Multiagent Learning Using a Variable Learning Rate[J]. Artificial Intelligence, 2002, 136(2):215–250.
- [87] Littman M L. Markov games as a framework for multi-agent reinforcement learning[M]//ICML'94 Proceedings of the Eleventh International Conference on International Conference on Machine Learning, 1994:157–163.
- [88] Hernandez-Leal P, Rosman B, Taylor M E, et al. A Bayesian Approach for Learning and Tracking Switching, Non-Stationary Opponents: (Extended Abstract)[J]. adaptive agents and multi-agents systems, 2016:1315–1316.
- [89] Gmytrasiewicz P J, Doshi P. A framework for sequential planning in multi-agent settings[J]. Journal of Artificial Intelligence Research, 2005, 24:49–79.

参考文献

- [90] Da Silva B C, Basso E W, Bazzan A L, et al. Dealing with non-stationary environments using context detection[C]. Proceedings of the 23rd international conference on Machine learning, 2006:217–224.
- [91] Hernandez-Leal P, Zhan Y, Taylor M E, et al. Efficiently detecting switches against non-stationary opponents[J]. Autonomous Agents and Multi-Agent Systems, 2017, 31(4):767–789.
- [92] Brafman R I, Tennenholtz M. R-max-a general polynomial time algorithm for near-optimal reinforcement learning[J]. Journal of Machine Learning Research, 2002, 3(Oct):213–231.
- [93] Rosman B, Hawasly M, Ramamoorthy S. Bayesian policy reuse[J]. Machine Learning, 2016, 104(1):99–127.
- [94] Vaswani A, Shazeer N, Parmar N, et al. Attention is All you Need[J]. neural information processing systems, 2017:5998–6008.
- [95] Zhang Q, Gu G, Xiao H. Image Segmentation Based on Visual Attention Mechanism.[J]. Journal of multimedia, 2009, 4(6).
- [96] Bakker B. Reinforcement learning with long short-term memory[C]. Advances in neural information processing systems, 2002:1475–1482.
- [97] Hochreiter S, Schmidhuber J. Long short-term memory[J]. Neural Computation, 1997, 9(8):1735–1780.
- [98] Graves A, Liwicki M, Fernández S, et al. A Novel Connectionist System for Unconstrained Handwriting Recognition[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2009, 31(5):855–868.
- [99] Smith J E, Winkler R L. The optimizer’ s curse: Skepticism and postdecision surprise in decision analysis[J]. Management Science, 2006, 52(3):311–322.
- [100] Vincent P, Larochelle H, Lajoie I, et al. Stacked Denoising Autoencoders: Learning Useful Representations in a Deep Network with a Local Denoising Criterion[J]. Journal of Machine Learning Research, 2010, 11:3371–3408.
- [101] Tang H, Houthooft R, Foote D, et al. # Exploration: A study of count-based exploration for deep reinforcement learning[C]. Advances in Neural Information Processing Systems, 2017:2753–2762.
- [102] Benda M, Jagannathan V, Dodhiawala R. On Optimal Cooperation of Knowledge Sources - An Empirical Investigation[R].[S.l.]: Boeing Advanced Technology Center, Boeing Computing Services, 1986.
- [103] Lowe R, Wu Y, Tamar A, et al. Multi-Agent Actor-Critic for Mixed Cooperative-Competitive Environments[C]. Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA, 2017:6382–6393.

- [104] Busoniu L, Babuška R, De Schutter B. Multi-agent reinforcement learning: An overview[J]. Innovations in multi-agent systems and applications-1, 2010, 310:183–221.
- [105] Chou P, Maturana D, Scherer S. Improving Stochastic Policy Gradients in Continuous Control with Deep Reinforcement Learning using the Beta Distribution[C]. Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6–11 August 2017, 2017:834–843.
- [106] He H, Boyd-Graber J L. Opponent Modeling in Deep Reinforcement Learning[C]. Proceedings of the 33rd International Conference on Machine Learning (ICML), 2016:1804–1813.
- [107] Hernandez-Leal P, Rosman B, Taylor M E, et al. A Bayesian Approach for Learning and Tracking Switching, Non-Stationary Opponents (Extended Abstract)[C]. Proceedings of the 15th International Conference on Autonomous Agents and Multiagent Systems (AAMAS), 2016:1315–1316.
- [108] Lowe R, Wu Y, Tamar A, et al. Multi-Agent Actor-Critic for Mixed Cooperative-Competitive Environments[J]. neural information processing systems, 2017:6379–6390.
- [109] Hong Z, Su S, Shann T, et al. A Deep Policy Inference Q-Network for Multi-Agent Systems[C]. Proceedings of the 17th International Conference on Autonomous Agents and Multiagent Systems (AAMAS), 2018:1388–1396.
- [110] Hernandez-Leal P, Kaisers M. Learning against sequential opponents in repeated stochastic games[C]. The 3rd Multi-disciplinary Conference on Reinforcement Learning and Decision Making, 2017.
- [111] Albrecht S V, Stone P. Autonomous agents modelling other agents: A comprehensive survey and open problems[J]. Artificial Intelligence, 2018, 258:66–95.
- [112] Foerster J N, Chen R Y, Al-Shedivat M, et al. Learning with Opponent-Learning Awareness[C]. Proceedings of the 17th International Conference on Autonomous Agents and Multiagent Systems (AAMAS), 2018:122–130.
- [113] Hernandez-Leal P, Kaisers M, Baarslag T, et al. A Survey of Learning in Multiagent Environments: Dealing with Non-Stationarity[J]. CoRR, 2017, abs/1707.09183.
- [114] Rusu A A, Colmenarejo S G, Gülcühre Ç, et al. Policy Distillation[J]. CoRR, 2015, abs/1511.06295.
- [115] Hernandez-Leal P, Kaisers M. Towards a Fast Detection of Opponents in Repeated Stochastic Games[C]. Proceedings of the 16th International Conference on Autonomous Agents and Multiagent Systems (AAMAS) 2017 Workshops, 2017:239–257.
- [116] Crandall J W. Just add Pepper: extending learning algorithms for repeated matrix games to repeated Markov games[C]. Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS), 2012:399–406.
- [117] Goodfellow I, Bengio Y, Courville A. Deep learning[M].[S.l.]: MIT press, 2016.

参考文献

- [118] Krizhevsky A, Sutskever I, Hinton G E. ImageNet classification with deep convolutional neural networks[J]. Communications of The ACM, 2017, 60(6):84–90.
- [119] Russakovsky O, Deng J, Su H, et al. ImageNet Large Scale Visual Recognition Challenge[J]. International Journal of Computer Vision, 2015, 115(3):211–252.
- [120] Graves A, rahman Mohamed A, Hinton G E. Speech recognition with deep recurrent neural networks[C]. 2013 IEEE International Conference on Acoustics, Speech and Signal Processing, 2013:6645–6649.
- [121] Cho K, van Merriënboer B, Gulcehre C, et al. Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation[J]. 2014:1724–1734.
- [122] Schulman J, Wolski F, Dhariwal P, et al. Proximal policy optimization algorithms[J]. arXiv preprint arXiv:1707.06347, 2017.
- [123] Stone P, Veloso M M. Multiagent Systems: A Survey from a Machine Learning Perspective[J]. Autonomous Robots, 2000, 8(3):345–383.
- [124] Lafferty J D, McCallum A, Pereira F C N. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data[C]. ICML '01 Proceedings of the Eighteenth International Conference on Machine Learning, 2001:282–289.
- [125] Bishop C M. Neural networks for pattern recognition[M], 1995.
- [126] Lin H, Wang Z. Flight scheduling for airport closure based on sequential decision[C]. 2018 4th International Conference on Information Management (ICIM), 2018.

发表论文和参加科研情况说明

(一) 发表的学术论文

- [1] **Zheng Y**, Meng Z, Hao J, et al. A Deep Bayesian Policy Reuse Approach Against Non-Stationary Agents, NIPS 2018. (**CCF A**)
- [2] **Zheng Y**, Meng Z, Hao J, et al. Weighted double deep multiagent reinforcement learning in stochastic cooperative environments. PRICAI 2018. (**CCF C**)
- [3] **Zheng Y**, Wang Z, Fan X, et al. Localizing multiple software faults based on evolution algorithm[J]. Journal of Systems and Software, 2018, 139: 107-123. (**SCI 二区, IF=2.278**)
- [4] **Zheng Y**, Li Y, Own C M, et al. Real-time predication and navigation on traffic congestion model with equilibrium Markov chain[J]. International Journal of Distributed Sensor Networks, 14(4), 1550147718769784. (**SCI 四区, IF=1.787**)
- [5] **Zheng Y**, Nie X, Meng Z, et al. Layered Modeling and generation of Pollock's drip style. The Visual Computer, 31(5), 589-600. (**SCI 四区, IF=1.036**)
- [6] **Zheng Y**, Meng Z, Xu C. A Short-Text Oriented Clustering Method for Hot Topics Extraction. International Journal of Software Engineering and Knowledge Engineering, 25(03), 453-471. (**SCI 四区, IF=0.397**)
- [7] Yang Y, Hao J, Yu C, **Zheng Y**, Large-Scale Home Energy Management Using Entropy-Based Collective Multiagent Deep Reinforcement Learning, IJCAI 2019. (**CCF A**)
- [8] Yang T, Hao J, Meng Z, Zhang C, Zheng Z, **Zheng Y**. Efficiently Detecting and Optimally Responding Towards Sophisticated Opponents. IJCAI 2019. (**CCF A**)
- [9] Yang Y, Hao J, **Zheng Y**, et al. Large-Scale Home Energy Management Using Entropy-Based Collective Multiagent Reinforcement Learning Framework. AAMAS 2019. (**CCF B**)

(二) 申请及已获得的专利

- [1] 第一申请人, 基于TF-IDF特征的短文本聚类以及热点主题提取方法（已授

权) 专利号: 2014103787856,。

(三) 参与的科研项目

- [1] 大数据算法及其应用, 天津市支撑计划项目, 课题编号: 2014F3-0022
- [2] 群体利益最大化驱动的多智能体协调关键技术研究, 国家青年科学基金项目, 课题编号: 61702362
- [3] 网络异常行为检测技术及定位方法, 国家自然科学基金重点项目, 课题编号: U1836214
- [4] 基于多智能体层次强化学习的广告展示优化, 阿里巴巴AIR计划项目

致 谢

我很感恩、庆幸能够来到天津大学，始于本科，终于博士。

感谢我的父母，十年以来一直支持我的选择，在不容易的博士求学生涯中，提供了莫大的包容以及关怀，最后能够顺利毕业离不开父母无私的付出。

感谢我的导师孟昭鹏教授，十年来的教诲与相处，让我学习了该如何为人处世，该如何严谨科研。在科研方向上，孟老师给予了我极大地自由与支持，由衷的感谢孟老师对我的关心与指导。

感谢郝建业副教与南京大学的章宗长副教授，二位老师严谨的科研态度、深厚的科研底蕴以及谦逊的处事原则都对我产生了深远的影响，是我未来科研工作以及生活中宝贵的经验与财富。也要感谢徐超教授对我的教诲，从徐老师身上我也学习到了很多宝贵的人生经验，时刻鞭策的未来的处事原则。

感谢我的女朋友李嫣然，在十一年的相处中，我们相互扶持，包容，理解，感恩你对我付出的一切，相信我们会有更美好的未来。

感谢学习生活中帮助过我的同学及朋友们，你们是我求学生涯中不可或缺的一群可爱的人，一路走来，能够认识你们我感到很幸运。

最后感恩生在最美好的时代，感恩所有关心我帮助我的人，谢谢。