北京大学

硕士研究生学位论文

题目：**一种用于国标麻将 AI 强化学习训练的课程学习方法**

姓　　名：	郑启帆
学　　号：	2101213044
院　　系：	计算机学院
专　　业：	计算机软件与理论
研究方向：	人工智能与机器学习
导师姓名：	李文新 教授

☑ 学术学位　　☐ 专业学位

二〇二四　**年**　五　**月**

版权声明

摘要

游戏作为人工智能（AI）研究的重要试验场，提供了清晰的问题定义和明确的胜负结果，使其成为 AI 算法测试的理想平台。国标麻将与其他牌类游戏相比，具有规则更复杂、奖励（reward）更稀疏、隐藏信息多、状态空间庞大以及存在多智能体相互博弈与干扰等特点，为 AI 研究提出了更多新挑战，具有相当大的研究价值。因此本设计选择国标麻将 AI 作为研究对象，试图探讨如何控制和理解国标麻将 AI 强化学习的过程，进而提出如何提升训练效率并增强训练过程的可解释性的方法。

国标麻将 AI 进行强化学习训练的过程中，对国标麻将 AI 决策表现影响最大的两个因素分别是：1）获胜目标种类繁多且相互干扰导致的决策目标不明确；2）国标麻将回合数多、决策点密集导致的单局复杂程度高，即该问题具有多目标长时决策的性质。为了克服国标麻将多目标长时决策性质带来的挑战，本设计提出了基于国标麻将场景的课程学习训练方法。通过将具有不同上听数的初始手牌作为训练的初始状态，以满足课程学习“从易到难”的基本要求。此外，本设计还探讨了构造新的马尔可夫过程来帮助控制训练进程，以期达到更好的训练效果，并通过对超参数和问题参数的调整初步解释基于神经网络的国标麻将 AI 的决策黑箱。

本设计实现了一个国标麻将的分布式强化学习框架，其中包括环境模块、强化学习模块、分布式训练模块以及与 botzone 交互入口模块。基于该框架，本设计通过强国标麻将 AI 的和牌对局对训练初始局面进行构筑，将其按照上听数划分为不同的初始局面载入到国标麻将强化学习环境中，以作为待训练课程。该框架创新性的在于国标麻将环境中提供了加载对局初始状态的功能，从而实现了对训练过程的极大程度控制。本设计的大部分强化学习组件具有的可复用性。

经过课程学习的强化学习 AI 在 Botzone 平台上取得了预期的效果。经过课程学习的强化学习 AI 在 botzone 天梯排名前 10%，并大幅优于使用相同实验配置和相同强化学习框架进行训练，但未使用课程学习训练方式的直接强化学习 AI。在与 2020 年 IJCAI 国标麻将 AI 比赛的代表性 AI 进行对比的中展现的水平也在当年比赛的 4 到 5 名之间，并比起其他使用神经网络的 AI 实现了所需算力规模的显著缩小和同等效果下的训练所需时间的缩短。本设计还通过修改奖励函数 Reward、折扣因子 γ 和特征维度等参数，完成了对国标麻将 AI 行为差异的分析与总结，为后续对课程学习工作的进一步拓展进行了先验的积累。

综上，该课程学习方案的提出，针对了国标麻将 AI 多目标长时决策的性质，解决了直接使用强化学习进行训练不稳定和算力要求高等问题，达到了良好的训练效果与稳定性，也为解释国标麻将强化学习的黑箱做出了有益的探索。

关键词：课程学习，强化学习，游戏，AI 可解释性

A Curriculum Learning Method for Official International Mahjong Reinforcement Learning AI

Kaifan Cheng (Computer Software and Theory)
Directed by Professor Wenxin Li

ABSTRACT

The game, as an important testing ground for artificial intelligence (AI) research, provides a clear problem definition and explicit win-loss outcomes, making it an ideal platform for testing AI algorithms. Compared with other card games, Official International Mahjong has more complex rules, sparser rewards, more hidden information, a vast state space, and the presence of multiple agents engaging in mutual competition and interference. This presents more new challenges for AI research and has considerable research value. Therefore, this thesis selects Official International Mahjong AI as the research object, attempting to explore how to control and understand the reinforcement learning process of Official International Mahjong AI, and then propose methods to improve training efficiency and enhance the interpretability of the training process.

During the reinforcement learning training process of Official International Mahjong AI, the two most significant issues affecting the decision-making performance of Official International Mahjong AI are: 1) the multitude of winning goals and mutual interference leading to unclear decision goals; 2) the high complexity of single rounds due to the high number of rounds and dense decision points, i.e., the problem has the nature of multi-objective long-term decision-making. To overcome the challenges brought by the multi-objective long-term decision-making nature of Official International Mahjong, this thesis proposes a curriculum learning training method based on the Official International Mahjong scenario. By dividing different initial hand tiles into training initial states based on the number of tiles needed to complete the hand, it meets the basic requirement of curriculum learning "from easy to difficult". Additionally, this thesis explores the construction of a new Markov process to help control the training process to achieve better training results and preliminarily explain the decision-making black box of the Official International Mahjong AI based on neural networks through adjustments to hyperparameters and problem parameters.

This thesis implements a distributed reinforcement learning framework for Official International Mahjong, which includes environment modules, reinforcement learning

modules, distributed training modules, and interaction entry modules with the botzone. Based on this framework, this thesis constructs initial game situations for training through winning hands of Official International Mahjong and divides them into different initial game situations based on the number of tiles needed to complete the hand, loading them into the Official International Mahjong reinforcement learning environment as the training curriculum. The framework also innovatively provides the function of loading the initial state of the game in the national standard mahjong environment , so as to control over the training process. Most of the reinforcement learning components in this thesis are reusable.

The reinforcement learning AI trained through curriculum learning achieved the expected results on the Botzone platform, ranking in the top 10% of the ladder with a smaller computational scale for training, significantly outperforming reinforcement learning AI using same training resource and reinforcement learning framework but without using curriculum learning. Compared with the representative AI of the 2020 IJCAI national standard Mahjong AI competition, the level shown in the competition was also between 4 and 5. Compared with other AI using neural networks, this AI achieves a significant reduction in the scale of computing power required and a reduction in the training time required for the same effect. This thesis also analyzes and summarizes the behavioral differences of Official International Mahjong AI by modifying parameters such as reward function Reward, discount factor γ , and feature dimensions, accumulating prior knowledge for further expansion of curriculum learning work.

In summary, the proposal of this curriculum learning scheme addresses the multi-objective long-term decision-making nature of Official International Mahjong AI, solves the problems of the instability and high computational requirements of training directly using reinforcement learning, achieves good training results and stability, and also makes a beneficial exploration to explaining the black box of Official International Mahjong reinforcement learning.

KEY WORDS: Curriculum Learning, Reinforcement Learning, Game, Explainability AI

目录

第一章　引言

游戏 AI 作为涉及各种人工智能算法的领域，具有十分重要的研究意义。本章将简述其发展历程，并针对游戏 AI 中的具有重要研究价值国标麻将 AI 的难以训练提出解决方案。

1.1　游戏在人工智能研究中的重要地位

人工智能（Artificial Intelligence, AI）自 1956 年达特茅斯会议提出以来，经过 60 余年的演进，人工智能正处于以深度学习为代表技术的第三次浪潮，并已经在语音识别、视觉识别、机器博弈、自动程序设计、大语言模型等诸多领域取得了无数令人震撼的突破。近年来，我国始终高度重视人工智能发展机遇和顶层设计，发布多项人工智能支持政策。国务院于 2017 年发布《新一代人工智能发展规划》，科技部等六部门于 2022 年印发《关于加快场景创新以人工智能高水平应用促进经济高质量发展的指导意见》落实规划。2024 年《政府工作报告》中提出开展“人工智能+”行动。由此可见，人工智能在社会中发挥的作用愈发突出，具有的地位日渐提升。

游戏起源于人类对决策模拟的需求。也正因如此，游戏通常具有清晰准确的问题定义，胜负结果明确目标唯一，容易与人类智能作对比。在整个游戏 AI 发展的历史中，每当有 AI 系统能够在更复杂的游戏中打败顶尖的人类选手，人们都认为 AI 技术达到了新的里程碑，更加接近人类的智能水平。随着算力的提升以及新算法的提出与应用，如今的游戏 AI 取得了很大的进步，已经能在许多非常复杂的游戏中打败人类选手，尽管这些游戏在过去被玩家们认为高水平的策略只有人类的智慧才能掌握。自从 AlphaGo 在围棋游戏中打败了职业选手[1]，引起了游戏 AI 研究的热潮之后，近几年许多游戏都被 AI 所攻克，包括棋类游戏如围棋、将棋和国际象棋[1][2][3]，牌类游戏如德扑[4][5][6]，以及视频游戏比如星际争霸[7]、Dota2[8]以及王者荣耀[9]。这些 AI 系统在越来越复杂的游戏中打败了顶尖人类选手，但这些游戏的类型不尽相同，而攻克它们的 AI 所使用的技术也不一样。所以，在人工智能算法的研究中，游戏一直被视为 AI 算法最得天独厚的试验田[10]。通过将难题建模成一个个经典的游戏问题，越来越多的游戏 AI 算法除了用于解决游戏内问题本身，还广泛应用于其他包含决策需求的领域，展现出了很好的延展性和泛化性。研究游戏 AI 可以让我们更深入的理解人工智能算法的性质，从而推动 AI 技术的进步。

1.2 游戏 AI 的发展历程

自从 1945 年最早的计算机 ENIAC 发明之后，让计算机程序玩游戏就已经成为了人工智能领域的重要问题。在 1951 年，Christopher Strachey 编写了一个跳棋程序，同年 Dietrich Prinz 编写了一个国际象棋程序[11]，这些是最早利用计算机玩游戏的程序。游戏 AI 早期的研究主要集中在经典的棋类游戏上，比如跳棋和国际象棋，这是因为这类游戏的游戏元素非常简洁，而规则也相对比较简单，但它们的复杂度却非常高，以至于人类已经研究了几百上千年但水平仍然有很大提升空间，离这些游戏的最优解还有很长的距离。AI 系统在这类游戏中打败人类职业选手总会被认为是游戏 AI 技术的新突破，象征着 AI 智力水平的提升。

在这些里程碑中，第一个引起人们广泛关注的是 1992 年由 Gerald Tesauro 开发的 TD-Gammon 程序[12]，这个程序在西洋双陆棋上打败了职业棋手。接着在 1994 年，一个名为 Chinook 的跳棋程序打败了当时的跳棋世界冠军 Marion Tinsley[13]。最著名也是人尽皆知的一个里程碑是 IBM 公司开发的深蓝程序，它在 1997 年的一场国际象棋人机大战中打败了当时的国际象棋特级大师 Kasparov[14]。而在近几年，更复杂的围棋也被 AI 所攻克。在 2016 年，谷歌旗下的 DeepMind 公司开发了 AlphaGo 程序，在一场五局的人机比赛中打败了已退役的世界冠军李世石[1]，而在 2017 年，新版本的 AlphaGo 程序在一场三局两胜的人机大战中以 3-0 横扫现世界冠军柯洁[2]。尽管可以构造出比围棋复杂度更高的棋类游戏，但那样的游戏并没有围棋和象棋这样的流行度，围棋已经是流行的棋类游戏中复杂度最高的游戏。游戏 AI 背后的算法，也从传统的 Alpha-Beta 剪枝、专家系统，逐渐演变到如今的模仿学习、强化学习。游戏 AI 水平也随着算法的不断革新，来到了全新的高度，同时也为游戏本身的发展起到了极大的推动作用。

在攻克围棋这一难题之后，人工智能的研究开始面向更加复杂的游戏问题。经典的棋类游戏在游戏 AI 领域算是比较容易解决的游戏，这是因为棋类游戏通常是回合制的，它们的状态表征非常规则，而游戏的全局信息也是所有玩家可见的。近几年，研究者们将目光投向更加有挑战性的问题，比如牌类游戏和视频游戏。这些游戏解决起来要困难得多，因为此类游戏大都有巨大的状态空间和动作空间，对局回合数比棋类游戏长得多，玩家的隐藏信息会造成信息不对称，同时也涉及到玩家之间的合作问题。但随着算力的提升和新 AI 技术的发展，AI 在这类问题上也取得了很好的进展。牌类游戏通常会涉及发牌引入的随机性以及不同玩家之间的信息不对称。

图 1.1 2017 年，时任世界第一柯洁对战 AlphaGo

在 2015 年，Bowling 团队攻克了双人有限注德州扑克，德州扑克的简化版本，并计算出了该游戏的近似最优解[15]。他们团队在 2017 年开发了 DeepStack 程序，在双人无限注德扑游戏中打败了人类职业选手[4]。而在 2018 年，Brown 团队开发了 Libratus 程序[5]，同样在双人无限注德扑游戏中打败了人类职业选手，但使用了和 DeepStack 不同的技术。他们团队在 2019 年进一步开发了 Pluribus，在六人的无限注德扑游戏中打败了人类职业选手[6]。

在各类牌类游戏中，由于集众多难点于一身，麻将的游戏 AI 研究是相当特别且重要的。麻将是一款多人非完美信息博弈，具有规则复杂、奖励函数 reward 稀疏、隐藏信息多、状态空间庞大和多智能体相互博弈与干扰等众多特点，十分具有研究价值。而国标麻将 AI 则由于其八番起和的复杂和牌条件，使得对局的随机性相对其他麻将有所减弱，从而更具研究价值。

1.3 国标麻将 AI 强化学习训练中遇到的问题简述

然而，对国标麻将 AI 的强化学习训练并不容易。3.1 中，我们将详细叙述国标麻

将 AI 强化学习的难点所在，包括在本实验室于 IJCAI 举办的国标麻将 AI 比赛中难以取得普遍性的好成绩、在开设的强化学习和 AI 基础课程中学生的强化学习训练往往取得不了基本的进展。以及在本实验室自己对国标麻将 AI 进行训练的过程中，也遇到了许多诸如智能体在训练过程中掌握不了基本的胡牌能力从而找不到提升的方向、无脑吃碰杠导致陷入局部最优等情况。

在对过去三届 IJCAI 国标麻将比赛中最强的的国标麻将智能体、游戏 AI 测评网站 botzone[16]以及人类麻将对局网站 MahjongSoft[17]的对局数据统计，笔者发现，影响国标麻将 AI 表现的问题主要是两个方面：一是获胜目标种类繁多且相互干扰导致的决策目标不明确，二是决策点多、决策点的影响持续整个对局，总结为国标麻将具有多目标长时决策的特点。

1.4 解决方案——课程学习

针对 1.3 中提到的国标麻将具有的多目标长时决策的特点以及前文提到的在实验室训练过程中遇到的问题，本文拟提出课程学习的方式来改善强化学习的训练流程，以期解决现存的问题。课程学习[18]（Curriculum Learning）是一种训练策略，模仿人类教育中有效的学习顺序，让模型先从容易的数据或子任务上进行训练，再慢慢转移到更困难的数据或者子任务上训练[19]。应用课程学习训练策略到诸多正式场景的优势主要集中以下几点。

a) 提升模型在目标任务上的性能

b) 加速训练过程

c) 容易使用，独立于模型本身的算法

通过设置较为简化的局面作为简单的课程，缩小了状态空间和待决策时长以减少训练复杂度和决策需要估计的回合数，从而能够稳定训练效果，这解决了 1.3 中长时决策特性对训练的影响问题；同时，开始训练的局面如果更接近某种和型很接近的位置，则是在待训智能体尚未具备和牌能力时帮助确定和牌目标，一定程度上防止了待训智能体在不具备基本和牌能力时面对过多的和牌目标陷入迷茫。

1.5 本文的主要工作和创新点

本文选取了国标麻将这一非完全信息牌类游戏作为研究对象，分析了多目标长时决策的特性对国标麻将 AI 强化学习训练造成的困难，并尝试解决这一问题。本文提出

了国标麻将的课程学习训练方式，通过划分初始训练局面来从易到难地分段训练国标麻将智能体，以期达到良好的训练效果和更优的训练稳定性，并在 Botzone 平台上对其进行了效果验证。此外，在国标麻将的课程学习训练过程中，本文还利用了国标麻将课程学习牌张可变范围小且易观测的特点，探究其可解释性方面的能力，为今后拓展现有的课程学习流程打下基础。

1.6 本章小结及后续章节安排

本章首先介绍了本文的研究背景，介绍了游戏在人工智能研究中的重要地位和游戏 AI 的发展历程，由此引出了本文的主要研究对象——国标麻将强化学习 AI 并简述了其在训练中遇到的问题，并对此做了性质上的拆解分析。最后对本文的主要工作和创新点进行了说明。

本文对后续章节的安排如下：第二章将介绍国标麻将的缘起、规则，游戏 AI，麻将 AI 以及引入课程学习的介绍和可行性分析；第三章将介绍本文提出的国标麻将的课程学习训练方式的设计方案以及课程学习对国标麻将可解释性的分析；第四章将展示国标麻将的课程学习训练框架的实现，包括强化学习框架部分、课程学习部署部分以及该框架实现过程中遇到的难点以及创新性和可复用性分析；第五章将展示该训练方式在国标麻将强化学习中得到的实验结果；第六章将总结本文并展望国标麻将课程学习未来可进行的更多训练模式以及对强化学习训练的帮助。

第二章　工作基础及相关工作的国内外研究现状

本章将介绍本文的工作基础以及相关工作的研究现状。首先介绍作为研究主体的国标麻将游戏的基本知识，包括其缘起和规则；之后介绍游戏 AI 的，包括基于人类经验的方法、基于规划方法和基于学习的方法；以及麻将 AI 的研究现状，包括基于专家经验及数据统计、监督学习和神经网络、深度强化学习等方法。最后介绍了课程学习的概念以及相关应用。

2.1　国标麻将的缘起

麻将是一种中国古代发明的博弈游戏，也是一种牌类娱乐用具，用竹子、骨头或塑料制成的小长方块，上面刻有花纹或字样，每副 136 张（有的地区 74 张）南方麻将多八个花牌，分别是春夏秋冬，梅竹兰菊，共计 144 张。不同地区的游戏规则稍有不同。麻将的牌式主要有“饼（文钱）”、“条（索子）”、“万（万贯）”等。在古代，麻将大都是以骨面竹背做成，可以说麻将牌实际上是一种纸牌与骨牌的结合体。与其他骨牌形式相比，麻将的玩法最为复杂有趣，它的基本打法简单，容易上手，但其中变化又极多，搭配组合因人而异，因此成为中国历史上最吸引人的博弈形式之一。由于其全面的文化内容以及有趣、有竞争性、有助于智慧和友谊的优点，近一个世纪以来，它一直是全世界人民的一种令人愉快的消遣。由于中国幅员辽阔，各地麻将玩法种类繁多，这并不利于作为一项新兴的竞技体育的麻将的发展。为了使麻将得到更好的推广，中国国家体育总局在咨询权威专家的基础上，以引导麻将科学化健康化为原则，于 1998 年制定了中国麻将竞赛规则（Mahjong Competition Rules）简称国标麻将[20]。

2.2　国标麻将的基本规则

国标麻将一共有 144 张牌，136 张常规牌和 8 张奖励牌，可以分为序牌和字牌两大类，序牌包括三种，万、筒、条，每一种都是由一到九的牌组成。字牌包括风牌和箭牌。风牌包括东、西、南、北。箭牌包括中、发、白。每张牌有 4 张相同的牌。8 张奖励牌包括梅、兰、竹、菊、春、夏、秋、冬，奖励牌不计入玩家的手牌。8 番起和，否则“和牌”无效并且有相应的惩罚。番种增加至 81 种。根据玩家和牌的难易程度，和牌时的番数也会不同。这些规定无形中增加了国标麻将和牌的难度，在提升了国标麻将游戏乐趣的同时，也使玩家需要借助科学的方法根据自己的手牌合理的

制定策略。国标麻将一局为 4 圈，每圈 4 盘不设连庄，也就是每局 16 盘，对局结束后计算总分排名。为了方便展开后续讨论，下文将介绍一些基本的术语、番种和计分方式。

图 2.1 麻将牌种类

2.2.1 术语

2.2.1.1 吃牌、顺子

吃牌，如下图 2.2 所示，是指上家上家出牌后的那张牌是下家正好需要的牌可以组成顺子，比如说上家玩家打出一万，下家玩家手中恰好有二万和三万，此时玩家可以亮出两张牌并拿走上家的牌组成一、二、三万。此时这三张牌不能放回原先的手牌中，也不可以使用其中的牌与手牌中的牌组合。

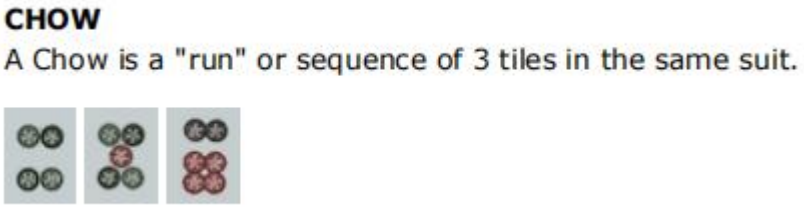


图 2.2 吃牌、顺子图示

2.2.1.2 碰牌、刻子

如下图 2.3 所示，如果任意玩家打出一张牌其余任何玩家中有这张牌的对子，该玩家可以亮出两张牌拿回玩家打出的那张牌组成刻子。

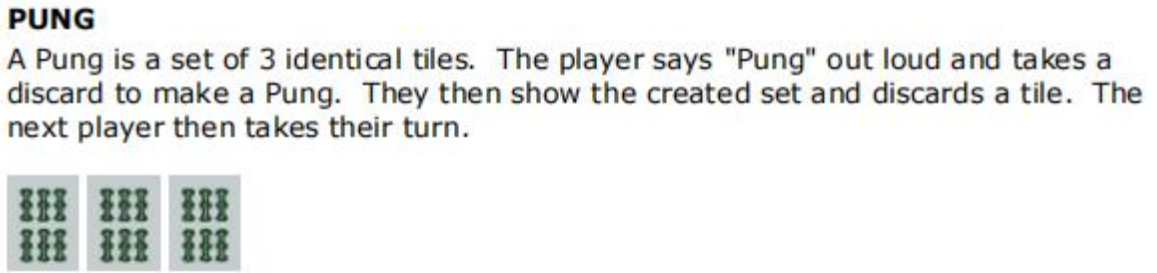


图 2.3 碰牌、刻子图示

2.2.1.3 杠牌、杠子

如下图 2.4 所示，如果任意玩家打出一张牌其余任何玩家中有这张牌的刻子，该玩家可以亮出三张牌并拿回玩家打出的那张牌组成杠子。如果玩家已有三张相同的牌当玩家摸到第四张的时候，此时称作暗杠。如果玩家在碰牌后得到一个刻子，之后自己又摸到了相同的牌，此时称作补杠。

KONG

A Kong is a set of 4 identical tiles. If formed from a discarded tile, the player declares "Kong" out loud and exposes the set. If drawn from the wall the player may retain it as concealed. The advantage in concealing a Kong is that the player can later split it and use one of the tiles to make a Chow if they wish.

图 2.4 杠牌、杠子图示

2.2.1.4 听牌

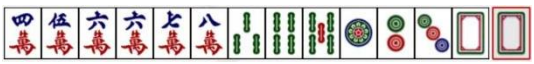
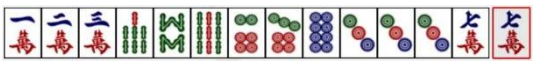
玩家通过摸排、弃牌、吃牌、碰牌、杠牌等一系列操作，保留有价值的牌，舍弃无用的牌，并差一张牌便可以组合成“四组一对”的情况叫做听牌。国标麻将中不需要报听，听牌后也可以换和其他牌。

图 2.5 听牌示意，此时差一张“南风”即可达到和牌状态

2.2.1.5 和牌

“和牌”是指玩家在听牌后，摸到或者是由其他对手打出的牌可以让该玩家组成“4 组 1 对”的组合并且至少有 8 番，此时该玩家和牌。如果该牌由别的玩家打出该和牌称作放炮或点炮，由自己摸到牌叫做自摸。

2.2.1.6 和牌牌型

和牌牌型	牌型具体含义	图示
11 123 123 123 132	4 副顺子+1 对将牌	
11 111 123 123 123	3 副顺子+1 副刻子+1 对将牌	

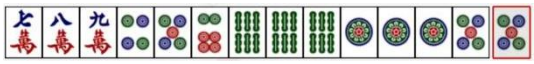
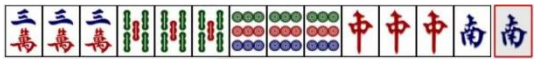
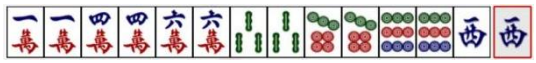
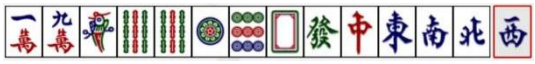
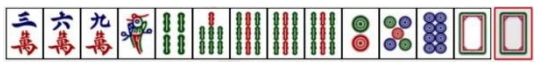
11 111 111 123 123	2 副顺子+2 副刻子+1 对将牌	
11 111 111 111 123	1 副顺子+3 副刻子+1 对将牌	
11 111 111 111 111	4 副刻子+1 对将牌	
11 11 11 11 11 11 11	七对	
十三幺	由万、筒、条的 1 和 9 牌组成再加上 7 张字牌，摸到任意一张和牌。	
七星不靠	由 7 张不同字牌, 和包含 147、258、369 等三组组间花色不同组内花色相同的 9 张序数牌中任意 7 张组成的和牌。	
全不靠	由 7 张不同字牌, 和包含 147、258、369 等三组组间花色不同组内相同的 9 张序数牌组成的 16 张牌中任意 14 张组成的和牌。	
组合龙	包含 147、258、369 等三组组间花色不同组内相同的 9 张序数牌组成的和牌。	

表 2.1 和牌牌型

2.2.2 番

从字面意思理解，番就是翻倍的意思。番是麻将中评估获胜玩家的最终手牌的机制，根据赢家最终手牌形成的难易程度，玩家手牌形成的难度越高获得的番数就越高。根据国标麻将规则，一共有 81 种不同的种类，每一种都有独特的名称，从 1 番到 88 番不等。由于 8 番起和的规定，玩家必须想方设法将手牌满足 8 番，否则“和牌”无

效并且会有相应的惩罚。玩家获得番数的方法有很多，大概可以分为 4 大类：手牌番种；和牌方式；花牌；特殊牌型。

2.2.2.1 手牌番种

玩家手牌是包含某些特定种类的牌，比如说玩家手牌中包含 4 副风刻子（东、西、南、北刻）再加任意一副将牌，该组合叫做大四喜。通过计算得出整副牌满足该条件的排列组合一共三十组，概率为 1.7016e-32。因为组成大四喜的几率很低，所以它值 88 番。

除了手牌中含有特定种类的牌，玩家还可以将手中的牌组合成某种特殊的序列，比如说三色三步高，通过字面意思不难理解，就是指手牌中包含 3 种序牌，每一种序牌依次递增。举个例子，玩家手牌中包括一、二、三万，二、三、四筒和三、四、五条，可以看出每一种序牌依次递增。

更有意思的是除了这两种情况之外，还可以根据麻将上符号的样子来组合成特定番种。比方说根据颜色分类的绿一色，可以发现二、三、四、六、八条和发，这六张牌上只有绿色，如果手牌中只包含这六张牌并且可以组成四组一对的情况和牌将得到 88 番回报。

2.2.2.2 和牌方式

增加番数的方法除了排列组合的方法之外，还可以通过一些规定的玩家和牌时的方式来增加番数。比如说如果玩家自摸的话，会有一番的奖励。又比如，如果玩家自摸到牌堆中的最后一张牌，该番种叫做妙手回春，会有 8 番的奖励。还有一种情况是，如果其他玩家打出的最后一张牌是该玩家和的牌这种情况叫做海底捞月，也会有 8 番的奖励。

2.2.2.3 花牌

国标麻将有 8 张花牌分别是春、夏、秋、冬、梅、兰、竹、菊。花牌不计入手牌，摸一张花牌会有一番的奖励。当玩家摸到花牌后将其放入一旁，然后再去牌堆中摸一张牌，这个过程叫做补花。

2.2.2.3 特殊牌型

在国标麻将中大部分和牌的情况都是“四组一对”，之前提到的“和牌”牌型中

可以发现除了通常的情况还有七对、十三幺、七星不靠、全不靠、组合龙这种情况。如果玩家一但决定要做这些特殊牌型，再更改为常规牌型就比较困难，所以玩家还需谨慎考虑。但是高风险意味着高回报，这些特殊牌型的番数基本都在 12 番以上，连七对和十三幺更是高达 88 番。

2.2.3 计分方式

国标麻将的计分十分重要，玩家根据自己当前的分数在不同的对局中采用不同的策略，如果玩家在 16 局的赛制中分数处于领先的位置，大多情况下玩家会打的相对保守，而落后的玩家会打的相对激进。在一盘对局中，只有获胜的玩家得分，输家全部扣分。获胜方在自摸和牌时，获得的分数为（底分 + 番种分数）× 3；而在通过其他玩家点炮而和牌时，获得的分数为（底分 × 3）+ 番种分数。失败方得分规则是：当获胜方为自摸和牌时，失败方的分数为 -（底分 + 番种数）；当获胜方通过其他玩家点炮而和牌时，非点炮玩家扣除底分，而点炮玩家则扣除 -（底分 + 番种数）。一般情况下，国标麻将采用的底分为 8。

国标麻将作为麻将赛事的主要规则，降低了随机性对对局公平性的影响而更具有科学性。番种丰富、和牌门槛更高的特点，使得选手需要用科学的方法制定有效的策略。上述国标麻将的介绍简单总结了游戏规则、术语、番和计分的概念，如需更深入地了解国标麻将可以参考国标麻将所有番种[20][21]的介绍。

2.3 游戏 AI 的研究现状

为了对不同类型的游戏构建玩游戏的 AI 系统，研究者们提出过许多 AI 算法。典型的游戏 AI 系统通常包含一些先验知识，要么是人类游戏专家所编写的、显式的知识规则，要么是通过一些预训练流程学到的知识。这些知识接着在实际玩游戏的过程中会被用到，可能再结合一些实时的规划和搜索算法。后续的算法叙述按照这样的分类方式展开，将游戏 AI 算法整体上分为三类。第一类是显式的人类知识，可以直接作为先验知识嵌入到 AI 系统中。第二类是实时的规划算法，这类算法通常会从当前的游戏状态展开一棵博弈树，预先评估一些未来可能遇到的状态，并基于先验知识选择最好的动作。然而，这类算法需要精确地知道游戏的状态转移模型，要么事先给定要么通过学习得到。第三类是学习算法，可以用于预训练一些参数来保存先验知识，为实时规划算法提供更好的搜索和指导方向。

2.3.1 基于人类经验的算法

在从零开始构建游戏 AI 系统时，引入显式的人类游戏经验是最直接的方法。在很多流行游戏中，游戏内非玩家控制的角色（NPC）以及电脑玩家并不需要非常复杂的智能，通常只基于固定的人工策略进行设计。即使在使用了学习算法的 AI 系统中，引入人类知识也会让训练的前期达到更快的收敛速度和更好的模型表现，尽管在训练后期当模型达到人类水平之后可能会拖累模型能达到的上限。

在构建游戏 AI 系统时，主要有三种嵌入显式的人类知识的方法。第一种是将人类知识用于特征提取。对于那些以图像形式呈现的游戏来说，人类玩家知道图像中的每个游戏元素的位置以及它们的含义。这种从游戏图像中提取出关键信息的过程可以被写入游戏 AI 系统中，比如利用图像处理的技术或者直接通过游戏提供的接口对关键信息进行查询。另一方面，给定游戏的原始特征信息，人类玩家可能还会进一步计算一些与具体策略相关的关键特征。比如在黑白棋（也被称为翻转棋，英文名 Othello）中，人类玩家通常会基于“稳定子”和“行动力”等数值进行决策[22]。在游戏 AI 系统中，将这些人类总结的高层次特征加入到模型输入中是一种非常普遍的利用显式人类知识的方式。

使用显式人类经验的第二种方法是给出基于明确规则的策略，通常写成 if-else 的形式。比如说，2022 年李凯团队构建了一套用于无限注德扑研究的平台 OpenHoldem[23]，平台上使用几个基于人工策略的德扑 AI 作为深度学习算法的基准，它们基于当前手牌的强度以及预定义的概率来选择动作。这样的策略通常会被建模成决策树，决策树中的内部节点表示对游戏状态的判断条件，基于条件往下选择指定的子节点，而每个叶子节点是一个应该选择的动作。某些决策树还会包含随机节点，根据概率选择其子节点。在游戏制作行业，一种更高级的决策树——行为树[24]被广泛用于建模游戏中非玩家角色的行为，相比决策树能够表达更复杂的人工策略。

第三种是给出基于人类经验的估值函数，来评估人类对游戏局面好坏的看法。这样的估值函数以游戏局面的特征为输入，输出一个分数，表示当前状态是好还是不好。关于如何选择特征以及如何打分就完全依赖于人类经验。比如说，Iago 是 1982 年由 RosenBloom 编写的一个黑白棋程序，使用了人工设计的估值函数，利用了“稳定子”和“行动力”等特征并人工选择了一些权重进行打分。这样的估值函数可以进一步和实时规划算法进行结合，来选择那些在后面的回合中倾向于到达更好的游戏局面的动作。

2.3.2 基于规划的算法

很多时候，游戏的状态空间都非常大，很难在游戏开始之前就算出所有游戏状态下的最优动作。如果游戏的状态转移模型是已知的话，可以在实际对局中只针对那些实际遇到的游戏状态进行实时的规划，计算出在当前局面下可选的最优动作。这样的计算过程被称为实时规划算法，对于那些选动作可以有思考时间的游戏非常有用。比如说，象棋程序中 AI 可以思考几秒甚至几分钟，这段时间就可以用于对当前局面进行深入的计算，而实时战略游戏中需要快速根据当前的游戏局面选择动作，一般就不会用到规划算法。

如果事先就有对游戏状态的估值函数，最简单的规划算法就是比较一下当前局面采取每个动作之后转移到的新状态的估值，然后选择那个估值最高的动作。像这样的规划可以往后看不止一步，从而展开一棵搜索树，评估各种不同动作序列之后的游戏状态，从而构建一个更加有远见的智能体。这类算法整体上被称为启发式搜索，其中最有名的算法之一是 A*算法[25]，使用估值函数来引导对未探索过的游戏状态的选择。总的来说，当估值函数不完美的时候，搜得更深通常能得到更好的侧脸，这是因为往后多看几步可以减少一些估值带来的误差。然而，搜索更深的节点需要花费更多的时间，因此实践中的搜索算法通常会限制一个固定的搜索深度，或者通过逐次增加深度直到达到时间限制来决定实际搜索的层数。

Minimax 算法是双人非合作博弈中经典的实时规划算法，被广泛用于棋类游戏的 AI 中[26]。这个算法的核心假设是每名玩家都想最大化自身的收益，因此在展开搜索树时，节点根据当前决策玩家的决策被划分成 max 节点和 min 节点。在到达了指定的搜索深度之后，用估值函数对搜索树的叶子节点进行评估，算出当前玩家的收益。对于那些在 minmax 机制下不可能影响父节点价值的子树，可以通过 Alpha-Beta 剪枝算法进行进一步优化。Minimax 算法能够自然地扩展到多人游戏中，只要让估值函数在叶子节点处计算出所有玩家的收益，然后每个节点最大化当前决策的玩家收益即可。

蒙特卡洛树搜索（Monte Carlo Tree Search，MCTS）是另一个实时规划算法，在棋类游戏上也取得了巨大的成功。可以说在围棋领域从 2005 年 AI 只能达到业余玩家水平，发展到 2015 年达到顶尖人类水平，主要归功于 MCTS 算法的应用[27]。MCTS 算法在遇到每个游戏局面时，会从当前局面开始模拟许多轮对局到游戏结束。每一轮模拟先从目前的搜索树中使用树策略选择一条路径，然后展开此处的叶节点，再使用 rollout 策略一直跑到游戏结束。每一轮结束时的分数会被用于更新路径上经过的局面的估值。这个算法的核心思想在于从当前的局面往下搜索时更加重视那些前几轮模拟时取得过高分的路线。一般来说，rollout 策略会选择一个比较简单的策略，从而减少每次模拟的时间代价，而树策略需要在探索和剥削中取得平衡，使搜索主要集中在那

些更有潜力的路线上，同时也不要漏掉搜索树那些未探索部分可能更好的动作。2006年 Kocsis 提出了 UCT 算法，将置信区间上界算法应用到 MCTS 中，取得了理论上最优的期望收益[28]。

MCTS 算法常被用于非完美信息博弈中。其中一种方法基于确定化（Determinization），在桥牌[29]和纸牌接龙（Klondike Solitaire）游戏[30]中都取得了一定的效果。其主要思想在于将游戏的随机性确定化，从当前所在的信息集中随机采样一个可能的游戏状态，并将后续的随机事件结果预先确定，将原问题转化为一个完美信息博弈之后再用朴素的 MCTS 求解。然而，这种方式所搜索的博弈树并不能真正捕捉原问题的非完美信息的本质，存在两个主要问题[31]。一方面，将随机性确定化意味着需要在当前信息集中采样许多状态，对每个状态搜索一棵完美信息博弈的博弈树，这会带来很大的时间代价，而且其中很多博弈树的子树都是相同的，带来了计算上的冗余。另一方面，非完美信息博弈中由于同一信息集中的不同状态无法被玩家区分，因此动作选择必须一致，但确定化之后不同状态下会产生不同的动作选择，违反了非完美信息博弈的本质。另一种方法叫做基于信息集的 MCTS（IS-MCTS）[32]，在当前玩家的信息集构成的博弈树上进行搜索而非实际游戏状态的博弈树，可以更加直接的分析游戏结构，保留了隐藏信息以及随机性可能带来的变化。实验表明在斗地主等非完美信息博弈中，IS-MCTS 能够取得比确定化 MCTS 更好的效果[33]。

另一类用于非完美信息博弈的实时规划算法叫做 Continual Re-solving，在无限注德州扑克上取得了非常好的效果[5]。这种规划算法是基于 CFR-D 框架[34]的，该框架将非完美信息博弈分解成若干子博弈并对每个子博弈分别运行安全的求解算法，最终能得到完整博弈的均衡解。在此之前，非完美信息博弈通常被看做一个不可分割的整体，无法分解成子问题进行求解，这是因为单独求解子问题计算出的子问题均衡解很可能不是整个博弈的均衡解的一部分。CFR-D 框架利用玩家对对手私有信息分布的信念以及对手的反事实遗憾值作为子问题求解的约束条件，从理论上确保了求解子问题得到的策略并不比原问题上一轮求解的策略更坏。这里反事实遗憾值指的是当前游戏状态下玩家的期望收益，假设玩家采取的动作以最大的概率到达当前状态。Continual Resolving 算法进一步拓展了这个框架，将固定深度的搜索树与估值函数相结合进行实时规划，其中估值函数输入每个玩家对对手私有信息的信念并计算出各玩家的反事实遗憾值。例如，DeepStack 预先训练了深度反事实遗憾值网络，并用 Continual Re-solving 作为实时规划算法，在无限注德州扑克上取得了超过人类水平的表现。

2.3.3 基于学习的算法

和人类刚上手新游戏时需要熟悉规则、经过练习才能玩得好一样，大多数游戏 AI 系统也需要有预训练的过程，学习一些和游戏策略相关的先验知识。这样的先验知识可以被保存在模型中，比如策略模型或者值模型，然后在实际对局时再和实时规划算法相结合计算出决策。大多数情况下，这样的学习过程是构建游戏 AI 系统最重要的一环，也是游戏 AI 研究的核心。本节中分类总结了用于游戏 AI 系统的训练过程中各种学习算法，包括早期研究使用的进化算法、依赖对局数据的监督学习算法、通过对局自我提升的强化学习算法以及用于多智能体环境的学习算法。

2.3.3.1 进化算法

进化算法是受到进化论提出的“物竞天择，适者生存”的自然选择过程的启发而提出的，属于随机全局优化算法[35]。其基本思路是创造一系列个体组成的种群，让种群中适应度更高的个体有更高的概率繁衍并继承其父代的大部分性质。同时，遗传的过程中也需要引入变化，包括交叉互换和变异，从而使种群内的个体具备必要的差异性，这样随着时间的推移，选择压就能够使整个种群的平均适应度提高。这样的进化过程可以看做是对适应度函数的优化，使该函数越来越接近其最优值，所以进化算法可以在那些人类专家很难找到最优解的问题中作为优化算法被使用。

进化算法作为一类算法有各种不同的变体，但所有这些算法都有一些共同的组件。首先是关于问题解的表征或者编码，需要将原问题中复杂的解空间简化成规则的基因空间，从而可以在数学上定义变异和交叉算子。由于我们一般不清楚最优解实际长什么样，对解的编码方式最好能将所有可能的解都包含在内。另一个重要的组件是适应度函数的选取，它直接关系到种群中的个体适应环境的方向。在游戏 AI 的场景中，解空间一般指的是策略空间，而适应度函数通常被选为策略在游戏中的表现，比如得分或者能取得的收益。在不同的进化算法中，交叉、变异算子的具体实现、如何选择用于繁衍的亲代样本、以及如何选择哪些个体存活下来，都因具体算法而异。

进化算法中最有名的变体是基因算法（GA）和进化策略（ES）[36]。GA 算法将原问题的解编码为二进制串，将变异看做随机翻转串中的一位、交叉看做随机选取两个串的对应部分互换。亲代被选择繁衍的概率与适应度函数值成正比，所有的亲代都不会被保留，新的种群只由繁衍得到的下一代组成。ES 算法将原问题的解定义为浮点数向量，并将变异看做对向量的高斯扰动、交叉看做向量之间的插值。每一轮繁衍时完全随机的选择亲代个体，然后将亲代个体和繁衍得到的下一代个体全部放在一起，将适应度最高的那些个体选出来作为新的种群。

在多智能体博弈中，共同进化（Coevolution）算法[37]是最常用的进化算法。其核心思想在于将适应度函数定义为个体之间互相对战所能获得的相对分数，而非单智能体环境中每个个体独立与环境交互的得分。在实践中，共同进化算法既可以维护一个单独的种群，让种群里的个体之间互相对战来评估他们的适应度函数，或者建立多个种群，让不同种群之间的个体互相对战。研究者们认为正如自然界中共同进化的原理一般，在这样个体间频繁交互的环境中，竞争性的共同进化可以提升物种的适应度。基于共同进化的算法在许多游戏上都取得了成果，包括井字棋（Tic-Tac-Toe）[38]、博弈论中经典的追逃游戏 (Pursuit and Evasion) [39]、捕食者与猎物博弈 (Predator and Prey) [40]、实时战略游戏如夺旗（Capture The Flag）[41]和星际战争（Planet Wars）[42]，以及一款名叫炉石的卡牌游戏[43]。

2.3.3.2 监督学习

监督学习是一类数据驱动的算法，用来拟合数据以及对应特性之间的内在关系。在游戏 AI 的场景中，这里的数据通常指的是游戏状态或者游戏的可观测信息，而任务是学习一个策略模型或者值模型，可以根据当前遇到的游戏状态预测选择的动作或者评估期望收益。这样的算法需要许多以状态-动作对或者状态-价值对的形式提供的标注数据，这些数据通常来自于人类的对局数据或者同一个游戏中其他 AI 算法所产生的数据。一旦用监督学习学到了这样的策略模型或者值模型之后，它可以在实际推断过程中用做先验知识，可以直接使用或者进一步和实时规划算法相结合得到更好的策略。

一般来说，监督学习拟合函数是通过改变函数模型中的参数来实现的。这样的函数模型可以有很多种，比如支持向量机、决策树以及深度神经网络，每种模型修改参数的算法都是不同的。这里我们着眼于现代的深度神经网络算法，尽管在一些场合中，决策树之类的经典算法会更好，比如当我们需要一个可解释的函数模型时。大多数现代的游戏 AI 系统都用神经网络来表示策略模型或者值模型，是因为神经网络具有很强的表达能力[44]，并且在特征提取方面有着很强的适应性。神经网络中的一些变体，比如卷积神经网络（CNN）以及循环神经网络（RNN）十分常用，这是因为它们分别能够很好提取空间以及时序上的特征。

在现代游戏 AI 系统中，监督学习的应用可以根据其数据来源分为三类。最常见的一类是使用人类对局数据。针对人类数据进行监督训练可以从数据中学到隐式的人类经验，并将其储存在策略模型或者值模型中。然而，即使模型能够在训练集上达到100%的准确率，模型的泛化误差是不可避免的，因此模型的水平通常无法达到它所模仿的人的水平。除此之外，从人类数据学到的模型有可能被人类经验所误导，陷入人

类策略的局部最优中，尤其是围棋这类非常复杂的游戏，即使人类职业玩家的策略也可能和最优策略相差很多。通过人类数据监督学习的模型通常用于对其他学习算法如强化学习提供初始模型[1][7][45]，可以在训练的前期大幅提升训练效率。

除了使用人类数据进行监督学习以外，还有两类监督学习的应用是不依赖于人类数据的。一类称为知识提取，比如有一个游戏 AI 算法运行起来特别慢，无法用于实战中，但可以在后台生成源源不断的数据，那么监督学习可以针对这些数据训练一个网络模型，提取数据中隐含的知识，而这样学到的网络模型就可以进行快速推断。例如，DeepStack[4]在无限注德扑游戏对局的三个阶段分别用监督学习训练三个值网络模型，其数据来源是用搜索速度很慢的 Continual Re-solving 算法结合下一个阶段的值网络模型得到的。这些模型将不同游戏阶段的状态价值评估的知识保存起来，可以快速用于实时的游戏推断中。另一类不依赖于人类数据的应用叫做知识蒸馏[46]，通过监督学习训练一个较为轻量级的模型，用于克隆一个更大的模型的行为或者多个不同模型的统一行为。在知识蒸馏中，训练目标并非用状态对应的动作选择作为硬目标，而是使用一个相对更软的目标，即原模型输出的动作概率分布，使用香农熵之类的函数作为 loss 函数，从而确保原模型的泛化能力被保留下来。例如，在王者荣耀游戏中，腾讯的绝悟模型使用监督学习从多个基于不同英雄组合训练得到的老师模型中，学习一个统一的学生模型，从而提取出任意英雄组合下都能适用的通用策略模型[9]。

2.3.3.3 强化学习

强化学习是机器学习的一大领域，主要研究智能体如何在环境中决策从而最大化累计收益。与监督学习中从已标注的样本对中学习映射不同的是，强化学习处理的是控制类问题，试图学习一个从局面到动作的映射，但这里的动作并非以真实标签的形式提供，而是需要模型和环境交互，实际尝试这些动作并发现哪些动作可以带来更高的后续收益。强化学习的学习重心通常在于探索与利用的权衡上，一方面需要探索未知的状态动作从而发现可能更好的动作，另一方面需要利用已知的动作最大化收益。近年来强化学习一直是游戏 AI 领域最流行的算法之一，这是因为学习如何玩游戏本身就是一个控制问题，可以直接被强化学习的框架所建模。

一般来说，强化学习将环境建模为马尔科夫决策过程（MDP）。在每个时间步中，环境都处于一个状态，智能体需要从该状态选一个可行动作，环境会相应转移到一个新状态上并给智能体一个奖励。这里的状态转移概率和奖励都遵循马尔科夫性质，即他们只与当前的状态和动作有关。当环境转移模型已知的时候，通用策略迭代算法（Generalized policy interation）通过动态规划来求解最优策略以及对应的值函数这里值

函数指的是状态或者状态动作对的期望收益。这个算法最常用的变体是值迭代算法，该算法基于描述策略和值函数之间关系的 Bellman 方程来求解最优策略和值函数。

值迭代算法属于基于模型（model-based）的算法，因为它需要事先知道 MDP 的完整转移模型。然而在大多数情况下，环境转移模型是未知的，因此我们只能用无模型（model-free）的算法，通过直接和环境交互收集数据进行学习。强化学习中的无模型算法可以分为两类，基于值的方法和基于策略的方法。基于值的方法思路是先近似估计状态或者状态动作对的价值，然后基于这些价值选择更好的动作，从而优化实际的策略。对于那些状态空间有限的环境，值函数可以直接用数组来刻画，数组下标对应具体的游戏状态，这类算法被称为打表（tabular）算法[27]。根据如何更新值函数，可以将打表算法分为几类。蒙特卡洛（MC）算法计算从当前状态到对局结束的累计收益并用来更新当前的值函数，而时序差分（TD）算法基于这一步的奖励和下一个状态的价值估计未来的累计收益并更新值函数。在实践中，蒙特卡洛算法的方差很大，所以人们一般采用基于 TD 值目标的算法，比如 Q-learning 算法。而对于那些状态空间过大，无法以数组形式储存值函数的环境，可以使用近似函数来拟合值函数。例如，DQN 算法是 Q-learning 算法的改进，它使用深度神经网络来拟合状态价值函数，并在 Atari 游戏上达到了人类水平[47]。

基于策略的算法是另一类无模型算法，近年来随着深度学习的发展逐渐成为主流算法。这类算法使用梯度下降算法，通过某些评价指标的梯度，直接学习参数化的策略模型。这类算法中最早的是 REINFORCE[48]，先采样完整的对局轨迹，然后用蒙特卡洛算法中的值目标，即对局的实际累计收益来作为 loss 函数。然而，纯的基于策略算法会有很大的方差，研究者们进一步提出了 actor-critic 系列算法[49]，一边用 actor 来学习参数化的策略，一边用 critic 来学习值函数。这样，actor 在更新策略时，可以减去 critic 评估的状态价值，从而减小状态动作价值的方差。actor-critic 系列算法有许多变体。其中，DDPG 算法对连续动作空间的问题进行处理，通过对 actor 输出的策略增加特定分布的噪声来增加连续动作空间下模型的探索能力[50]。A3C 算法作为分布式算法，将多个 actor 并行的运行在不同进程中，同时与环境交互采集数据，分别在各自进程中计算梯度，然后再汇总到 learner 进程中，从而大幅提高了训练效率[51]。IMPALA 算法是另一个分布式算法，在 A3C 算法的基础上进行了改进[52]。其主要处理的问题是在分布式训练时，actor 实际采样使用的模型参数可能滞后于 learner 进程最新更新的模型参数，带来了样本梯度上的误差，为此 IMPALA 算法在 loss 函数中引入 V-trace，通过特定的比率对 loss 进行调整，从而弥补了参数延迟更新带来的梯度误差。TRPO 算法[53]和 PPO 算法[54]在 A3C 算法的基础上，使训练过程变得进一步稳定。在使用期望收益作为 loss 函数进行策略更新时，梯度的误差可能会大幅影响当前策略的表现，从而使训练变得不稳定。为此，TRPO 算法使用了一个替代目标函

数（surrogate loss），将新参数下的期望收益重写成基于当前策略参数下期望收益的增量，引入了新旧策略间的 KL 散度，并使用二阶优化算法求解策略参数。PPO 算法进一步简化了这一过程，通过直接对梯度进行裁剪，避免了训练过程中策略的突变。目前 PPO 算法由于其训练的稳定性，已经成为基于策略的强化学习算法中的首选。

2.3.3.4 多智能体学习算法

在多智能体环境中学习策略与单智能体环境很不一样，因为每个智能体的行为都会影响其他智能体的观测信息，使得从每个智能体的视角来看，环境都是动态变化的。和单智能体环境中求解最优策略不同的是，多智能体场景中的学习主要是为了寻找某些均衡解，比如纳什均衡[55]。有一些算法是专门针对多智能体场景设计的，这里我们列出了现代游戏 AI 系统中最常用的几类算法。

Regret Matching 算法[56]是求解正则形式博弈纳什均衡解的一个很简单也很直观的算法。这个算法中，玩家之间会反复地进行对局，并在每一局中记录自己如果选择其他动作的话会得到怎样的收益，与实际收益进行对比，计算出没有选其他动作的遗憾值，并在下一局中按与累计遗憾值成正比的概率选择各个动作。反事实遗憾最小化（CFR）算法[57]进一步将其扩展到了扩展形式博弈中，该算法已经成为求解非完美信息游戏的强大工具。然而，原始的 CFR 算法需要在每一轮迭代中遍历整棵博弈树，而且需要大量的迭代轮次才能收敛，计算代价很高，无法扩展到大规模的游戏中。研究者们提出了许多 CFR 算法变体来提高其计算效率。CFR+算法[58]和 Discounted CFR 算法[59]对之前迭代轮次中的遗憾值进行衰减，并且对每一轮迭代的策略进行不同形式的加权，从而减少训练收敛所需的迭代次数，加速了整个训练过程。MCCFR 算法[60]在遍历博弈树时只采样一部分路径，从而使 CFR 算法可以解决那些博弈树巨大，尤其是包含随机节点的游戏。尽管对博弈树进行采样带来了很大的方差，使算法需要更多的迭代轮次才能收敛，但这与完整遍历博弈树带来的时间开销相比可以忽略，整个训练过程仍然得以加速。VR-MCCFR 算法[61]在 MCCFR 算法的基础上引入一个用作基准的值模型评估每个信息集的期望收益，从而减少了 MCCFR 对博弈树进行采样带来的方差。还有一些 CFR 算法变体对游戏状态进行简化[62]，并使用诸如线性回归函数之类的近似函数[63]来减少 CFR 算法的时间和内存开销，但对游戏状态的简化以及对值函数的近似都需要专家知识的辅助。直到深度神经网络与 CFR 算法相结合[64][65][66][67]之后，状态的简化以及值函数的近似终于摆脱了人类经验。基于深度神经网络的 CFR 算法大多也使用了之前变体中提出的博弈树采样、遗憾值衰减以及迭代权重调整，使这类算法在复杂游戏中能够更快地收敛到更好的结果上。

在竞争性的多智能体环境中，将单智能体强化学习算法与自对弈技术相结合也可以求解纳什均衡。这一类算法中最早的是用于双人零和游戏的 Fictitious Play（FP）算法[68]，通过反复运行对局，让每个智能体都计算出针对对手过去平均策略的最优策略应对，从而最终逼近纳什均衡解。Fictitous self play 算法（FSP）[69]进一步将其扩展到了扩展形式博弈中，可以解决多步的回合制游戏。Neural FSP 算法（NFSP）[70]使用神经网络作为策略模型来处理更大规模的游戏，使用强化学习来计算最优策略应对，并用监督学习来计算平均策略的模型。Double oracle（DO）算法[71]从策略空间的一个子集开始，让每个玩家都计算出当前策略集合下的纳什均衡，然后将均衡策略加入到集合中，反复迭代就能得到完整游戏的纳什均衡解。PSRO 算法[72]对 FSP 和 DO 算法进行了形式上的统一，该算法的核心是维护一个策略池，然后可以用不同的方式从策略池中选择对手并训练出一个新策略，加入到策略池中，在这样的框架下，DO 和 FSP 都可以看做是 PSRO 算法的具体实例，只是选择对手和训练新策略的方式不同而已。而在实践中，训多智能体场景下训练 RL 算法会维护一个模型池，从模型池中采样对手进行对局来收集样本用于训练。这里从模型池中选择对手的具体策略有很多种，在实践中比较常用的有以下几种：

（1）朴素自对弈，即总是选择最新的模型作为对手；

（2）历史模型自对弈，从最近一段时间保存的模型中随机选择一个作为对手[73]；

（3）基于种群的自对弈，创建多个不同类型的种群，每个种群有选择性的使用自己或者其他种群中的模型作为对手[7][74]；

（4）基于模型表现的自对弈，根据对局的胜率以更高概率选择表现更好的模型作为对手[9]。

中心化训练、分布式执行（CTDE）框架是多智能体环境中另一类非常流行的算法框架，这类算法在训练智能体的时候使用中心化的方式，可以引入每个智能体观察不到的全局信息，从而更好地指导训练方向，而在实际执行的时候使用分布式的方式，每个智能体只使用自己可见的信息，从而遵循非完美信息博弈的规则。像这样一种训练时使用额外隐藏信息进行模型间沟通的机制主要是为了缓解各智能体独立训练时发生的不稳定现象，比如训练陷入到策略空间的环状结构中。其中，基于值的 CTDE 算法包括值分解网络（VDN）算法[75]和 QMIX 算法[76]，它们都是 DQN 算法用于处理多智能体合作问题的变体。这两个算法都用到一个中心化的状态动作价值函数，并将该函数分别看做是每个智能体自身值网络输出的求和结果或者更复杂的结合算子，用网络来建模这种算子。QMIX 算法还在结合算子的网络中引入了环境不可观测的实际状态。MADDPG 算法[77]是基于策略的 CTDE 算法，将 DDPG 算法扩展到了多智能体环境中。该算法中，每个智能体都拥有自己的策略网络和值网络以及自己的奖励函数，所以既可以解决纯合作或纯竞争的问题，也可以解决合作竞争并存的复杂问题。每个

智能体的值网络不仅将当前智能体的观测作为输入，还把其他智能体的观测和动作都作为输入，并以中心化的方式进行训练。COMA 算法[78]将朴素的 actor-critic 算法扩展到了纯合作的多人场景中，即所有智能体的奖励函数都是相同的。该算法训练一个共用的值网络，并使用一个反事实的基准值函数来为不同的智能体分配奖励，即用排除掉当前玩家之后所有玩家总收益的减少量来作为给当前玩家的奖励。

2.4 麻将 AI 的研究现状

麻将是一款多人非完美信息博弈，其本身具有规则复杂、奖励函数 reward 稀疏、隐藏信息多、状态空间庞大和多智能体相互博弈与干扰等众多特点，使得其具有相当大的研究价值。现有的麻将 AI 的种类大致包括基于专家经验及数据统计、监督学习和神经网络、深度强化学习等。

2.4.1 专家经验及数据统计的方法

Sato[79]通过分析人类顶尖玩家的对局数据，总结并量化表示了人类顶尖玩家的战术经验。该文章通过计算风险指标参数，评估当前局面下玩家应该进攻或者防守、判断其他玩家是否听牌、以及评估出牌风险。

当前玩家积分和剩余做庄次数玩家本局的危险性可用于决定该局应该偏向进攻还是防守。在日本麻将比赛中，每个玩家都可以做庄两次，每局对局结束后每个玩家都会知道当前的积分。通过与当前分数最高的玩家对比，将分差分为大、中、小三类分差，大于 12600 分为大分差，小于 12600 大于 1000 为中分差，小于 1000 分为小分差。该文章将进攻至防守的倾向用 1-10 表示，数字越小进攻性越大，数字越大防守性越大。当玩家没有做庄剩余数时，小分差、中分差、大分差对应的倾向参数分别为 8、9、7，即倾向于防守。当玩家还有两次剩余做庄机会时，小分差、中分差、大分差对应的倾向参数分别为 2、4、6，即倾向于进攻。当玩家还剩余一次做庄机会且与最高分玩家为大分差时，此时倾向参数为 1，即最倾向于进攻。

判断玩家是否听牌是评估玩家危险性的最直接表现。由于麻将规则中无法知道其他玩家的牌，但是可以通过观察玩家在摸牌后换手的次数可以评估出玩家听牌的可能性。通过公式 $T = 0.07a + 0.73b + 1.73c + 1.68d + 1.74e + 5.45$，记录玩家在不同回合中换牌的次数来评估玩家听牌的可能性。a 表示在 1-3 圈中玩家改变牌的次数，b 表示在 4-6 圈中玩家改变牌的次数，c 表示在 7-9 圈中玩家改变牌的次数，d 表示在 10-12 圈中玩家改变牌的次数，e 表示在 13 圈后中玩家改变牌的次数。T 越大玩家听牌的可能性越大。

当玩家通过上述方法预测玩家听牌的可能性后，通过计算麻将出牌危险性[17]可以知道在当前回合打哪张牌风险最大。总结如表 2.2，例如，当玩家要打 4、5、6 时，发现所有玩家打出的牌中没有与 4, 5, 6 有关系的牌时风险最高为 12.3。所谓关系就是 2、3、4、5 与 4 有关系 3、4、5、6 与 5 有关系。以此类推可以评估打出不同牌风险性。

表 2.2 打出牌风险评估[79]

Discarded tile of the player	Risk
4, 5, and 6 with no relation to discarded tiles of other players	12.3
4, 5, and 6 with half relation to discarded tiles of other players	7.0
3 and 7 with no relation to discarded tiles of other players	7.1
2 and 8 with no relation to discarded tiles of other players	7.0
1 and 9 with no relation to discarded tiles of other players	6.3
3, 7 with relation to discarded tiles of other players	5.5
2, 8 with relation to discarded tiles	other players 4.8
1, 9 with relation to discarded tiles other players	2.9
Winds and Dragons which can be used only for eye or meld	3.4
Winds and Dragons which can be used only for eye	0.9

2.4.2 监督学习和神经网络的方法

Mizukami[80]将麻将 AI 拆分为两个部分：单人麻将决策和对手建模，以分别负责进攻与防守两个方面。在单人麻将决策时，该 AI 只考虑自己的手牌和附露情况，即将问题简化为单人麻将做牌模式，并通过监督学习（Supervised Learning）的方法[81]计算自己的手牌能够如何更有效的逼近尽可能番数更大的和牌局面。开局时，AI 主要聚焦于单人麻将决策，使自己的牌型尽快达到上听的局面，并在上听之后进入对手建模模式。对手建模模式采用 Logistic 回归预测其他玩家可能听的牌，并使用另一个模型判断对应玩家潜在可能的和牌番数，此后结合前两者的预测进行蒙特卡洛搜索（Monte Carlo Search），给出防止点炮的防守向决策。文章通过对比职业玩家实际决策与该麻将 AI 判断的吻合率论证该麻将 AI 的决策水平，该吻合率达到了 62.1%。

Gao 通过结合监督学习和训练卷积神经网络[82]（Convolutional Neural Network），将不同的麻将决策模块使用独立的网络分别进行训练，在日麻中取得了优秀的效果。首先，文章将麻将牌张编码为 4 * 34 的 One-Hot 形式，并将对局中所有公共信息，包括玩家自己的手牌、牌河中打出过的牌张、所有玩家的附露以及四位玩家当前手牌数等，提取为如表 2.3 所示的训练所需的特征，并使用包含 3 层 5*2 卷积层和 2 层全连接层将其提取到网络。出牌、吃碰杠和听牌三个动作决策分别使用了三个独立网络进行训练，以防止彼此的决策相互干扰。该工作使用了日麻对战平台天凤[83]中的头部玩家的对局数据进行结果验证。与头部玩家的实际对局情况相比，该工作出牌的准确率达到了 68.8%，吃、碰牌的准确率分别达到了 90.4%和 88.2%。此后，该作者还通过变

更输入特征结构的层数和神经网络的优化[84]，进一步提升了出牌的准确率，使其达到70.44%。

表 2.3 输入特征

特征	通道
自己手牌	1
所有玩家牌河中弃牌	4
所有玩家附露	4
宝牌指示	1
立直玩家指示	3
圈风	1
门风	1
玩家 1 手牌	13
玩家 2 手牌	9
玩家 3 手牌	9
玩家 4 手牌	9

2.4.3 深度强化学习的方法

2020 年，微软亚洲研究院开发了 Suphx 程序[45]，用深度强化学习解决立直麻将，打败了大多数顶尖人类玩家。Suphx 训练了五个策略网络作为先验知识，这些网络被嵌入到一个人工策略的决策流程中并用于实战，决策流程如图 2.6 所示。整个训练流程包括两个阶段。第一阶段用监督学习训练被用于不同类型状态的 Resnet 策略网络，包括出牌、吃牌、碰牌、杠牌以及立直五个网络，五个网络分别使用了来自人类顶尖玩家对局的 $4 * 10^6$ 到 $1.5 * 10^7$ 个状态动作对。在第二阶段，出牌网络使用策略梯度算法的变体用强化学习训练，该算法在原始策略梯度算法的基础上增加了熵正则项以鼓励探索，并使用动态的熵权重让训练更加稳定。训练过程中，最新的出牌网络会被嵌入到决策流程中，并始终与当前最新的模型对战来收集对局数据。由于麻将的一局游戏包括许多轮，最终输赢是按各轮的累计得分排名，因此不同的轮中总分可能会很大的影响游戏策略，如总分落后时可能采取更激进的策略以追分。为此，强化学习训练中，每一轮的奖励并非只由当前轮的分数决定，而是用一个全局奖励预测网络，基于循环神经网络预测几轮之后玩家的得分。这个全局奖励预测网络也是事先用监督学习基于人类数据训练得到的。为了能够在非完美信息场景下加速训练，Suphx 还使用了一个叫做 oracle guiding 的技术，在训练的一开始将玩家不可见的信息也提供给

模型的输入，让模型能更好地把握数据规律，但随着训练的进行逐渐减少这些信息的权重，将一个有上帝视角的模型逐渐转变为一个遵守不可见信息规则的模型。

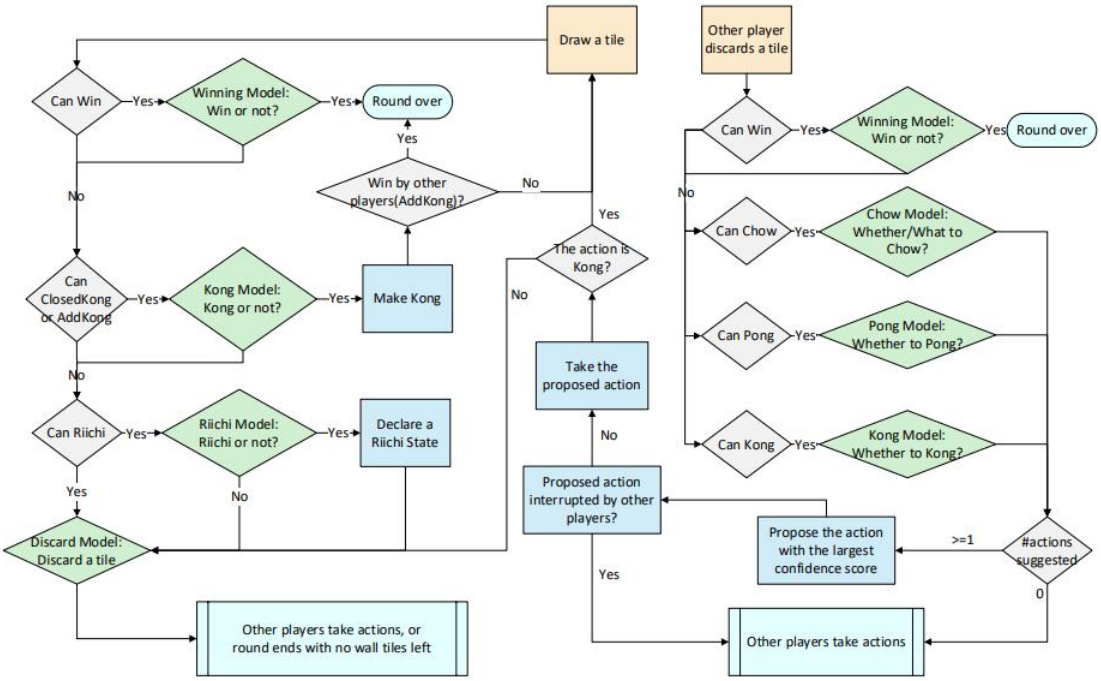


图 2.6 Suphx 程序中预定义的决策流程，绿色表示嵌入的策略网络模型[45][7]

在实时的对局中，Suphx 用预先定义的决策流程与五个策略网络相结合来选择动作。然而，由于麻将每一轮的起始手牌好坏会直接影响到本轮的策略是偏向激进还是保守，Suphx 在每一轮对局开始的时候使用 pMCPA 算法重新对策略模型进行微调。具体来说，对局开始时 Suphx 先随机猜测对手的几组手牌，然后依据这样的手牌生成一些完整对局，并对原始的策略模型进行策略梯度更新的微调，每局微调后的模型不会被继承到下一局中。实验表明，Suphx 在全局排名第一的概率高于其他麻将 AI 和人类专家，最后一名和点炮概率都要小于其他麻将 AI。而在最流行的立直麻将平台——天凤平台[85]上，Suphx 打败了 99.99%的人类选手，并且比职业选手取得了更高的稳定段位，成为了第一个击败职业选手的麻将 AI。

表 2.4 Suphx 实验结果[45]

	1 st Rank	2 nd Rank	3 rd Rank	4 th Rank	Win Rate	Deal-in Rate
Bakuuchi	28.0%	26.2%	23.2%	22.4%	23.01%	12.16%
NAGA	25.6%	27.2%	25.9%	21.1%	22.69%	11.42%
Top human	28.0%	26.8%	24.7%	20.5%	-	-
Suphx	29.3%	27.5%	24.4%	18.7%	22.83%	10.6%

2022 年腾讯 AI 实验室开发了绝将程序[86]解决双人麻将，将 actor-critic 算法和 CFR 思想相结合，打败了人类冠军玩家。2023 年，腾讯的绝艺 LuckyJ 程序[86]更是在日本麻将天凤平台特上房达到稳定段位 10.68 段，刷新了 AI 在麻将领域取得的最好成绩，此前还在国标麻将线下邀请赛中战胜了 6 位国标麻将职业选手，成为首个战胜国标麻将顶尖职业选手的麻将 AI。

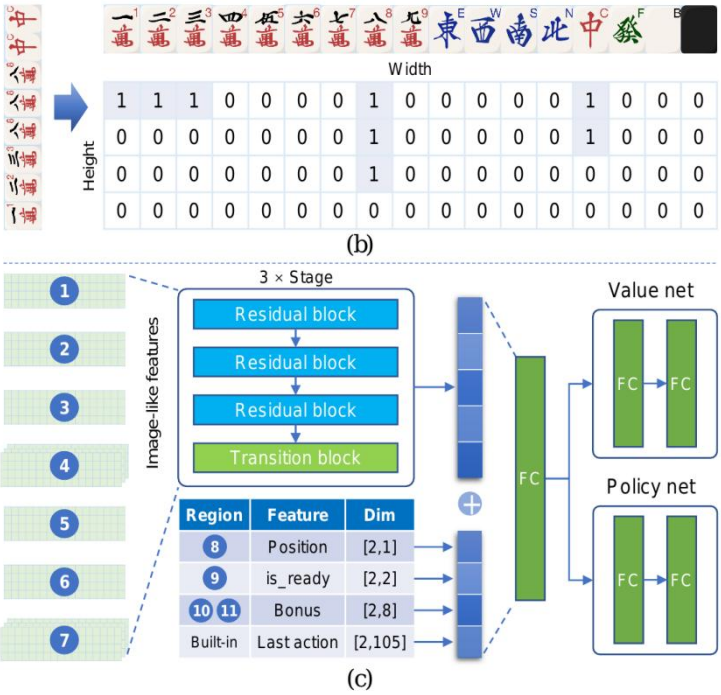


图 2.7 绝将结构[86]

绝将结构如图 2.7 所示，其独特之处在于训练了一个策略价值网络作为先验知识，并在实时对局中只进行单次的网络前向传播进行推理。其策略价值网络的训练基于 ACH 算法，是 Neural Weighted CFR 算法在实践中的变体，而后者在理论上被证明可以收敛到纳什均衡。具体来说，值网络的训练目标是为了拟合游戏结束时的得分，而策略网络的训练目标是为了最小化累计遗憾值，而非传统的 RL 训练中最大化累计收益。这里的遗憾值是根据值网络在当前游戏状态下的估值进行计算的，用遗憾值代替传统策略梯度算法中的优势函数，从而可以适用于经典的分布式 actor-critic 训练框架中。绝将在训练过程中只保留最新的模型，用最新的模型自对弈来生成训练数据。实验表明绝将与其他强化学习算法训练出的智能体相比，在面对最坏情况下的对手时表现更好，并且在与人类玩家一对一的比赛中取得了超过人类冠军的水平。

2.5 课程学习

课程学习（Curriculum Learning）由 Montreal 大学的 Bengio[18]教授团队在 2009 年的 ICML 上提出，是一种训练策略，模仿人类教育中有效的学习顺序，让模型先从容易的数据或子任务上进行训练，再慢慢转移到更困难的数据或者子任务上训练[19]。这种“由易到难” 的训练策略在人类教育中很常见，例如，一个孩子要从最简单的加减乘除概念入手，逐步学习方程、求导等，才能学会微积分。然而，传统机器学习算法往往采用随机的训练数据，忽略了样本的难度与模型的当前状态。课程学习正是希望从“设计一个更好的训练课程” 的角度，改进机器学习的训练策略。以图 2.8 图像分类任务为例。最初，课程学习在“简单”图像的一个小子集上训练模型，即苹果和橘子的图像是清晰、典型和易于识别的。随着模型训练的进展，课程学习在现有子集中添加了更多“更难”的图像(即更难识别)，这类似于人类课程中学习材料的难度增加。最后，课程学习利用整个训练数据集进行训练。

按照条件的严格程度，课程学习有几个不同程度的概念划分[89]:

Original Curriculum Learning[18]

一个课程是在 T 步机器学习训练中一系列训练标准 C=<Q1,...,Qt,...,QT>,每一个标准 Qt 都是目标训练分布 P(Z)的一个重新加权 Qt(z)=Wt(z)P(z), z 可以是任意一个训练集中的样本。并且满足以下三个条件

a) 每一步训练集的熵不断增加。H(Qt)<H(Qt+1)

b) 每个样本权重不断增加。Wt(z)<=Wt+1(z)

c) 第 T 步标准等于训练集。QT(z)=P(z)

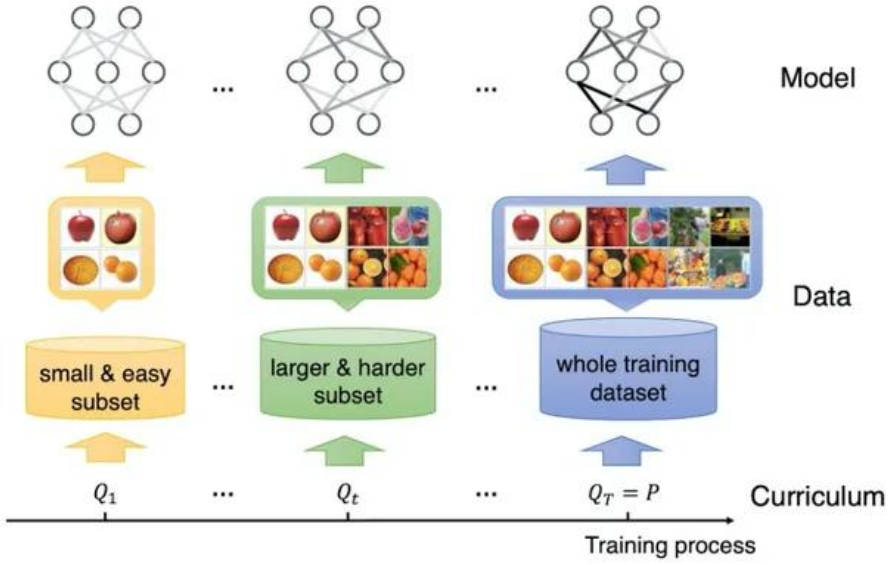


图 2.8 课程学习概念示意[88]

第一个条件表示训练集的多样性跟信息量应该慢慢增加，从而导致每一步的训练集的大小跟难度在整个训练过程中会逐渐增加，第二个条件表示应该逐渐增加训练样本，导致训练集大小不断增加，第三个条件表示应该最终步的样本重新加权结果应该等于目标训练集。这个课程学习最严格版本的定义，拥有较多的约束，在不少场景下都难以满足，例如在多任务训练中，条件 2 跟条件 3 就不满足，但是通过合理安排各个任务的训练顺序，也实现了训练难度从容易到困难的过渡。于是就有一个更加抽象的定义。

Data-level Generalized Curriculum Learning

移除了上一个概念中的三个附加条件，一个课程是在 T 步机器学习训练中一系列目标训练分布的重新加权。课程学习就是一个按照课程训练模型的策略。

后续为了扩宽课程学习的范围，将课程学习的定义进一步做了调整，将定义从数据层面转移到更佳抽象的概念。

Generalized Curriculum Learning

一个课程是在 T 步机器学习训练中一系列标准，每个标准都包括机器学习模型训练相关元素的设计，包括但不限于任数据，任务，模型容量，学习目标等。

2.5.1 课程学习的有效性分析

关于为何课程学习可以给机器学习模型带来性能上的提升，以及实现训练加速，目前的分析主要集中在两个角度，模型优化角度跟数据分析角度。

2.5.1.1 模型优化角度

课程学习可以被看作是一个特别的连续法（continuation method），是一种针对非凸问题的优化策略。如下图 2.9 所示，课程学习先在一个更平滑版本的问题上优化，然后再逐渐降低平滑性，在更加复杂的问题上优化，直到在目标版本上完成优化。类似于模拟退火算法，通过一系列优化目标，先在更加平滑的问题上找到局部最优解，再慢慢往全局最优解的方向移动。而从课程学习早期的优化目标找到的局部最优解具有更强大的泛化能力，更有可能接近于全局最优解。换个角度想，可以把课程学习早期的学习目标看作一个预训练，为后续目标的优化提供必要的帮助。

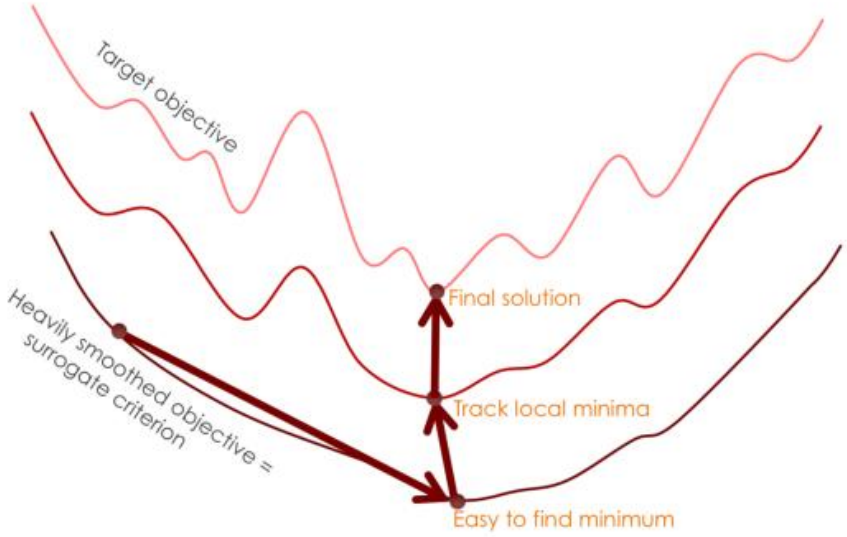


图 2.9 课程学习视作连续法[88]

2.5.1.2 数据分析角度

深度学习的训练数据来源广泛，其中不乏被错误标注的噪声数据或者复杂样本。课程学习策略会在简单数据上花费更多时间，从而避免在困难跟噪声数据上浪费过多时间，从而实现训练加速。由于错误的标注数据或者噪声，会导致训练集分布跟测试集分布之间存在偏差。

如图 2.10 示，如果将那些具有更高置信度的样本视作简单样本，而把那些低置信度的数据视作困难样本，那么课程学习就是先从高置信度样本中开始学习（目标分布），并慢慢转向低置信度样本中去（训练分布）。整个训练过程可以视作训练集分布的一个加权采样序列，如下图所示，生成一系列采样分布。波峰附近的数据代表高置信度的数据即干净的数据，两边（尾部）代表的是低置信度的数据即噪声数据。一开始先赋予低置信度数据（尾部数据）较低的权重，赋予高置信度数据（中间部分数据）较高的权重，也就是一开始的分布是接近于目标分布，然后在不断调整权重，直到全部数据具有相同权重，也就等同于训练集分布。通过一系列采样分布的逐渐调整，课程学习可以减小来自负样本的影响。

此外，课程学习的本质是将目标分布下的预期风险上界最小化，如下图右所示，这个上界表明，我们可以通过课程学习的核心思想来处理将 $P_{target}(x)$ 上的预期风险最

小化的任务：根据课程设置逐步抽取相对容易的样本，达到最小化训练数据的经验损失的目的。

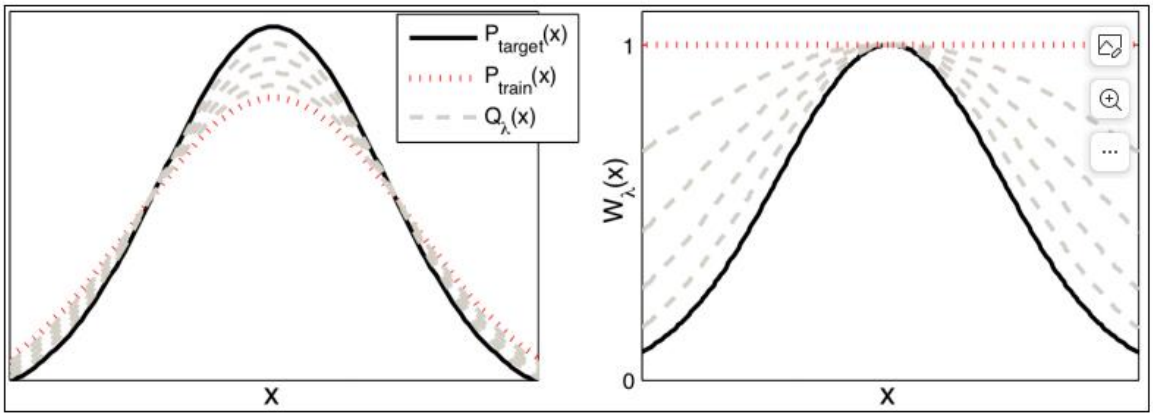


图 2.10 从数据分布角度来看课程学习[88]

2.5.2 课程学习的应用

基于 2.5.1 中有关课程学习有效性的分析，我们可以将课程学习的应用目的分为两类：基于模型优化角度，课程学习可以引导训练在参数空间中向着更加合理的区域进行移动；基于数据据分析角度，课程学习更多聚焦于高置信度数据区域，从而缓解噪声数据的干扰。课程学习的适用场景可以包含计算机视觉、自然语言处理、强化学习、多任务学习等广泛的任务。从监督信息的视角而言，课程学习被广泛应用于监督学习、弱监督（或半监督）学习和无监督学习任务。课程学习的主要作用及其对应的常见适用场景如下表 2.7 所示。

表 2.7 课程学习的主要作用及其对应的常见适用场景

目的	效果	场景	例子
引导	使训练得以完成\更好并更快	目标任务很困难或有不同的分布	稀疏奖励 RL、多任务学习、GAN 训练；领域自适应、不平衡分类任务等
去噪	使训练更快、更鲁棒和更可通用化	具有噪声、质量不均匀、异构数据的任务	弱监督或无监督学习、自然语言处理任务（神经机器翻译、自然语言理解等)

2.5.2.1 对训练进行引导

对于困难级别的目标任务而言，直接在上面训练往往会导致糟糕的性能表现或者模型难以收敛。课程学习策略可以通过引导模型从简单任务到困难任务过渡，从而避免在困难任务上直接训练的窘局。例如多任务学习如果随机选择训练任务的顺序往往会得到令人不满的效果，但是通过课程学习先选择较简单并跟上个任务有联系的任务的策略往往能产生效果。而且，对于那些目标分布跟训练分布存在较大差异的任务，课程学习也可以慢慢引导模型去适应目标分布。一个有代表性的场景任务是领域迁移，先在有充足标注数据的 A 领域（跟目标领域有一定相似）上训练，然后再在只有少量标注数据的目标领域上做进一步训练，这样更多的去训练在域内的数据，较少训练域外的数据。

2.5.2.2 去噪

对于包含噪声或者多种类型的训练数据，课程学习策略可以帮助去噪，从而加速训练，让模型更佳鲁棒，泛化能力更强。基于这种动机的课程学习的一个流行应用是神经机器翻译(NMT)，其数据集在质量、难度和噪声方面具有高度异质性。这是因为一个句子的翻译可能有长有短，有不同的词汇和语法结构，而且不同的注释者总是提供不同质量的翻译。此外，NMT 模型(如 rnn)的训练往往是耗时的。因此，课程学习自然适合于 NMT 任务在训练过程中去噪，既能提高性能又能更快地收敛。同样，课程学习

也被用于其他带有噪声或异构数据的自然语言处理任务，包括自然语言理解、关系提取、阅读理解、弱监督 CV 任务等。

2.6 本章小结

本章从以下几个方面对本文的相关工作进行了介绍：国标麻将介绍与基本规则；游戏 AI 的研究现状，包括基于人类经验的方法、基于规划方法和基于学习的方法；麻将 AI 的研究现状，包括基于专家经验及数据统计、监督学习和神经网络、深度强化学习等方法，并已经在基于深度学习的方法上取得了不错的效果；最后引入了课程学习的概念，并对课程学习的有效性进行了分析。接下来的章节中，本文将展开描述本文是如何通过课程学习的训练方式的设计以达到对国标麻将强化学习 AI 良好的训练效果的。

第三章　国标麻将 AI 强化学习的课程学习训练方式设计

本章将介绍针对国标麻将 AI 强化学习的课程学习训练方式的设计。首先通过分析国标麻将 AI 强化学习训练的难点所在来确认我们的设计需要着重解决的问题；之后阐述了课程学习的训练方案和解决思路；最后分析了课程学习用于提高国标麻将黑箱可解释性的可行性。

3.1　国标麻将 AI 强化学习训练的难点所在

本章将通过分析 IJCAI 国标麻将 AI 比赛中强化学习 AI 的表现、强化学习课程与 AI 基础课程中强化学习 AI 的表现以及本实验室国标麻将 AI 强化学习训练过程中的表现分析国标麻将 AI 强化学习训练难度大的原因。

3.1.1　Botzone 在线 AI 对战平台

Botzone 在线多智能体游戏 AI 对战平台[12]是由北京大学人工智能实验室开发的在线程序对战平台。该平台旨在提供一个通用界面，以评估不同编程语言的各种人工智能智能体在各种游戏中的表现，其中包括传统的棋类游戏，如五子棋（Gomoku）、同化棋（Ataxx）、象棋（Chess）和围棋（Go）；纸牌游戏，如斗地主和包括日麻、国标麻将在内的麻将，以及经过修改的雅达利游戏（Atari），如吃豆人和坦克大战。用户可以将他们的程序上传到平台上，程序被称为 Bot。只要他们遵循平台指定的输入-输出协议，Bot 就可以与同一款游戏中的任何其他 Bot 进行游戏。该平台还支持人类玩家和 Bot 之间进行对战，也支持人类玩家之间的相互对战。此外，Botzone 还具备建立小组，举办独立的多轮锦标赛来测试不同规格智能体能力的功能。

作为我们研究的重点，国标麻将游戏上线已有五年，在这期间，也积累了大量不同种类的麻将 AI 程序以及对局数据。用户提供的国标麻将 AI 种类大致也可分为基于专家经验、概率统计、监督学习和机器学习这几大类。为我们的研究和实验提供了大量的宝贵资料，为我们对国标麻将问题的研究打下了坚实的基础。

3.1.2 IJCAI 国标麻将 AI 比赛中强化学习 AI 表现

由于国标麻将具有的重要研究意义，为了起到推广国标麻将游戏作为人工智能研究的角斗场，并探索现代游戏人工智能算法的潜力的作用，本实验室于 2020 年、2022 年和 2023 年在 IJCAI 举办了 3 场国标麻将人工智能比赛[87]。为了减小单局游戏的随机性并获得对参赛队伍智能体更准确的排名评价，我们采取了瑞士轮加复式赛制相结合的形式。复式赛制通过四名选手智能体每局结束交换他们的座位，而每个座位总是得到相同的初始手牌，一组四个智能体一共进行 24 场比赛作为一组来进行。这种方式可以有效规避运气对比赛测评带来的影响。瑞士轮赛制作为更高层次的调度逻辑，负责决定哪四个选手智能体被分到一个复式小组中。而为了进一步确保比赛排名的准确度，瑞士轮赛制的轮次每年都在增加。在三次麻将人工智能比赛中排名前 16 名的智能体所使用的算法的类别如下图 3.1 所示。2020 年的冠军团队使用纯强化学习进行训练，达到了相当优异的效果，达到了远超一般人类玩家的水平，但这是建立在他们使用了相当大的算力规模的基础上，这点我们会在第五章的实验中予以展示。而除了 2020 年的冠军团队以外，就没有任何使用纯强化学习的队伍在比赛中进入了十六强。而后两年的比赛里更是完全没有使用纯强化学习的队伍进入前 16 名。由此可见，使用强化学习解决国标麻将问题的难度之高。

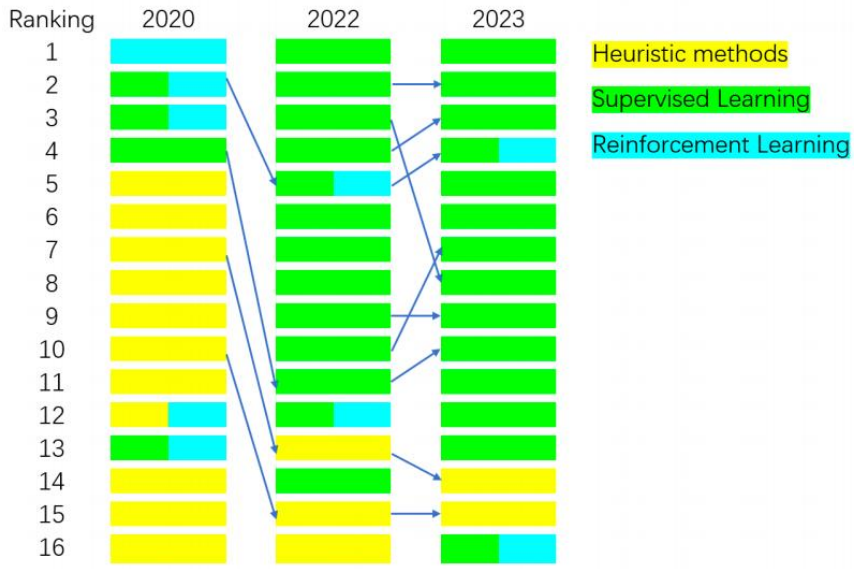


图 3.1 在三次麻将人工智能比赛中排名前 16 名的智能体使用的算法类别

3.1.3 强化学习课程与 AI 基础课程中强化学习 AI 表现

在开设的强化学习和 AI 基础课程中，学生也对国标麻将强化学习进行了尝试，但基本没有取得成效。学生们的改动主要集中于网络结构、奖励函数 Reward 的设计以及超参数的调整，但基本没有取得预想中的积极成效，最终的训练效果都基本训不出能够掌握清晰打牌思路的智能体，更无法超过同样由学生自己训练的监督学习版本。学生们对训练效果的问题反馈主要集中于训练容易原地震荡而不提升、智能体容易倾向于吃碰杠以及训练进行一段时间后会迅速崩坏。产生以上现象的具体原因笔者将在下一小节中进行分析。

3.1.4 国标麻将 AI 强化学习训练过程中表现

在训练国标麻将 AI 的过程中，笔者也观察到了一些导致训练效果不佳的现象：首先，尝试使用纯强化学习模型进行训练，将奖励函数 Reward 设置为总上听数和基本和型上听数，出现学习率低时奖励函数 Reward 曲线原地震荡且不上升的情况，这可能是因为环境过于复杂，探索不充分导致陷入局部最优或者甚至无法掌握基本的和牌能力，故只有偶尔出现恰好掌握的和型或者凑够番种才能和牌，显示在奖励函数 Reward 曲线上则表现为原地震荡不上升；故我们选择调高学习率（Learning Rate），学习率高时奖励函数Reward曲线能上升但却会突然剧烈波动变坏,这可能是因为学习率过大导致ppo的梯度在被 clip 之前就已经回传，从而导致训练崩盘。而通过观察训练后的麻将 AI 的表现时我们发现：用上听数做奖励函数 Reward 会倾向于一定吃碰杠，因为吃碰一定会减小上听数，在训练初期难以和牌时，会朝着尽可能吃碰杠的方向学习，等能和牌的时候已经陷入该局部最优中。之后笔者还曾做过其他尝试，包括使用监督学习单独训练策略网络和值网络之后接续训练强化学习和将同步训练改为半异步乃至异步训练，但都收效甚微。

3.1.5 国标麻将 AI 强化学习难度大的原因分析

根据前文对国标麻将 AI 强化学习训练的尝试与观察，笔者发现影响国标麻将 AI 表现的因素主要是以下几个方面。

第一是获胜目标种类繁多且相互干扰导致的决策目标不明确。根据游戏 AI 测评网站 botzone[16]以及人类麻将对局网站 MahjongSoft[17]的对局数据统计，在所有大于 4 番

的主番番种里，排名第一的番种是“三色三步高”（三种花色序数依次递增 1 的三副顺子，计 6 番）（图 3.2），占据了 68 万局人类对局数据中的 17.51%以及 10 万局强 AI(IJCAI2020 国标麻将比赛第一名,水平接近国标麻将职业选手)对局数据中的 22.61%的和牌概率；排名第三的番种是“混一色”（由一种花色的序数牌以及字牌组成的和牌，计 6 番）（图 3.3），占据了 68 万局人类对局数据中的 12.32%以及 10 万局强 AI 对局数据中的 9.02%的和牌概率。而这两种番种在本质上是相互排斥的：“三色三步高”需要持有三种数字牌的三个顺子，而“混一色”则只能持有三种数字牌中的一种。两种如此重要的番种区别已经如此之大，而国标麻将共有超过 80 个番种，常见主番也超过了 20 个，其相互之间也存在不同程度上的相互影响和排斥，导致在学习过程中的麻将 AI 经常会面临相当复杂的目标选择问题。其他的牌类游戏一般都只存在的单目标和简单的多目标，例如斗地主中地主玩家就只有跑得快一种目标，而农民玩家行牌就需要兼顾到自己走完和辅助队友，但两个目标兼顾起来就相对容易。而在国标麻将中，如前文所述，一旦你确定了一个主番并为其进行了一些吃碰杠操作之后，你的牌型就很大程度被固定住了，所以游戏早期的决策的合理性很大程度决定了该局游戏的获胜与否，这对 AI 的判断能力提出了很高的要求。

图 3.2 混一色番种示意（一种花色的序数牌以及字牌组成的和牌）

图 3.3 三色三步高番种示意（三种花色序数依次递增 1 的三副顺子）

第二是由于国标麻将长时决策的性质。根据对 10 万局强 AI 对局数据进行分析，我们可以看到，近 95%的强 AI 对局需要至少 7 回合（一回合至多四名玩家进行操作）及以上才能达到和牌状态（图 3.4）。而麻将中一位玩家打出牌张时，其他三位玩家也都有可能进行响应决策，这使得一局国标麻将内可能的决策数目达到了上百个。相比起其他牌类运动，例如斗地主一局的决策数目大概只有 20 个，国标麻将在单局的决策复杂度上无疑是更复杂的。而在和牌过程中，AI 还受到进张、对手打出牌张、对手附露信息和对手吃碰杠信息的影响。如何在信息繁多的长时决策中选择最优，也是影响

AI 表现最重要的问题之一。

此外，国标麻将还存在许多牌类游戏共有的性质：如初始局面复杂且多样，导致训练对齐度不一；又比如行牌过程中随机性占比很大，有时正确的出牌可能导致点炮，其他玩家的决策失误可能导致该局正确的决策付诸东流等。这些性质与国标麻将特有的多目标长时决策性质一起，导致了国标麻将 AI 的强化学习训练困难、不稳定且难以收敛。

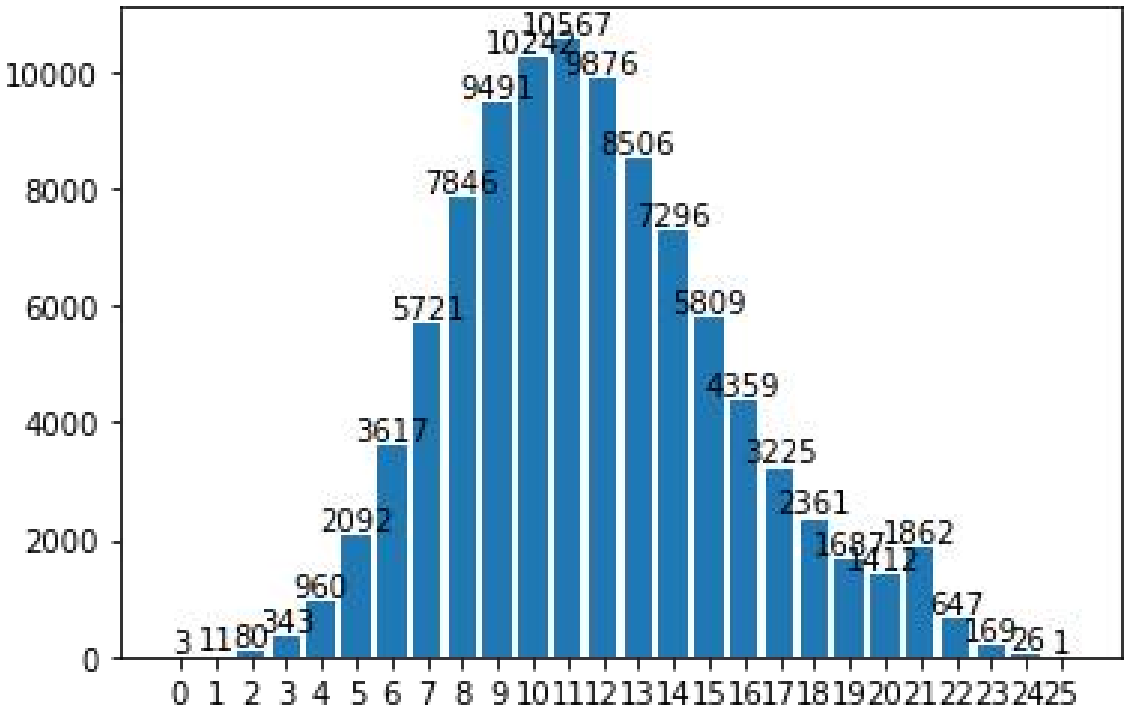


图 3.4 10 万局对局中获胜需要的回合数计数

综上所述，为了解决国标麻将 AI 训练效果不佳的问题，国标麻将多目标长时决策以及初始局面复杂、随机性强等特性无疑是需要笔者进行考虑的。因此，笔者试图通过课程学习的训练方式规避以上性质对国标麻将强化学习训练带来的挑战。

3.2 针对国标麻将 AI 的课程学习训练方案

3.2.1 方案目标

如 3.1.5 所述，国标麻将具有多目标长时决策以及初始局面复杂、随机性强的特性。所以，我们选择采用课程学习的训练方案，通过引导模型从简单任务到困难任务过渡，从而避免在困难任务上直接训练的窘局。以此，我们可以达到提升训练效果、提升训练过程可控性和加快训练收敛速度的作用。

3.2.2 解决思路

课程学习的核心问题是该如何针对特定任务设计对应的课程学习策略[12]，从而满足“从易到难”的训练逻辑。对应到国标麻将训练中，我们有一个很直观的指标来区分训练任务的困难程度：训练开始时智能体手牌的上听数。上听数指的是当前局面下麻将手牌距离最接近的满足八番起和条件的番种的听牌状态所缺少的张数。通过对 10 万局强国标麻将 AI 对局进行的数据分析，我们归纳出了每个初始手牌上听数所对应的获胜概率，如图 3.5 所示。我们可以发现，除去数量极为稀少（75 局）的六上听情况，上听数每减少一，胜率都会有显著的提升，初始手牌即为听牌局面（初始上听数为 0）的胜率更是高达 85%。显然，从低上听数手牌的初始局面开始训练即为更容易的训练任务。

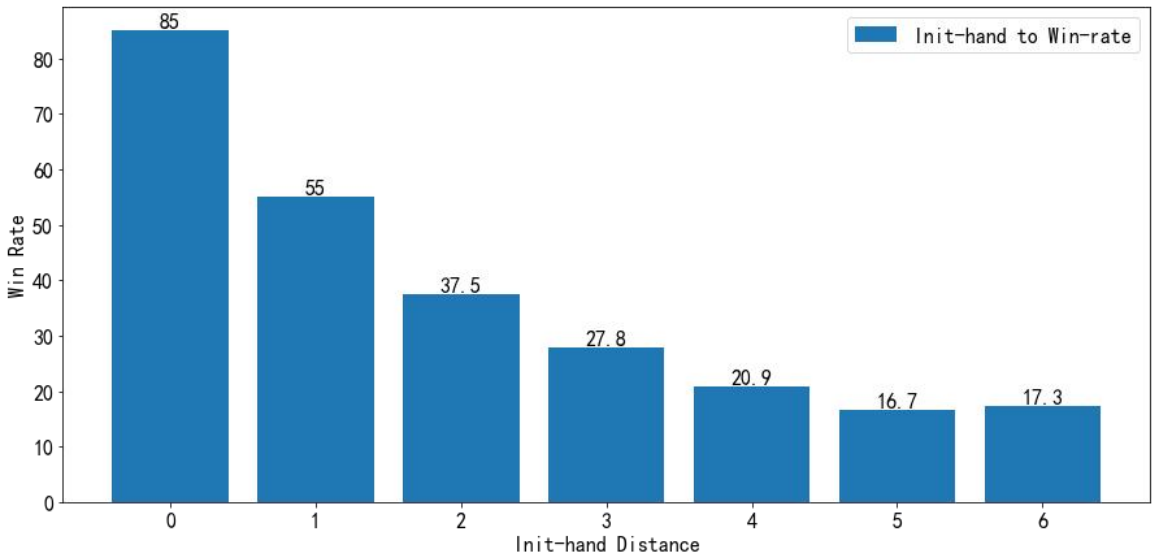


图 3.5 初始手牌上听数对应胜率

在我们的国标麻将课程学习训练场景中，课程中的每个节点代表一个训练任务，它对应了一个训练用的初始状态数据集合。我们以 2020 年 IJCAI 国标麻将比赛第一名的强国标麻将智能体的和牌对局为参照对训练初始局面进行构筑，取对局数据中和牌位置的智能体牌谱从听牌状态到其初始手牌的局面作为状态，用环境加载状态，将其作为初始状态进行强化学习训练。首先使用听牌局面作为初始状态进行训练，经过一段时间的训练之后，使用零至一上听局面作为初始状态进行训练。以此类推，最后在随机的初始状态上进行训练。如下图 3.6 所示，这是一个线性的学习流程。

如此的训练流程安排对应了前文提到的课程学习的两类应用目的：由于国标麻将中想要和牌，必然要经历从初始局面到听牌状态局面的移动，故从听牌状态作为初始状态开始训练则加强了智能体在这个场景下的处理能力，即引导训练发生在参数空间中向着更加合理的区域；而依靠上听数作为初始状态对训练流程进行划分并逐步训练，则是在一开始智能体不具备基本的和牌能力时，通过缩小状态空间以减少训练复杂度和决策需要估计的回合数，以稳定训练效果的一种尝试。同时，固定初始手牌上听数也对齐了训练的初始状态，减弱了训练的随机性。

图 3.6 线性课程学习流程

至此，我们提出了针对国标麻将多目标长时决策特性的课程学习训练方式，以期通过其解决前文提到的训练过程中遇到的掌握和牌能力困难和训练容易陷入局部最优的问题。

3.3 课程学习对国标麻将的可解释性分析

AI 可解释性是帮助人类关联 AI 算法的特性和输出的性质，本章我们将通过分析国标麻将课程学习的可解释性如何能帮助我们更好的理解与解释国标麻将这一黑箱。

3.3.1 国标麻将课程学习的可解释性分析

3.2 中提出的课程学习范式实际上是只改变了训练所处的初始局面，其待解决的马尔可夫过程是相同的。而如果我们想通过构造新的马尔可夫过程来帮助控制训练进程从而能得到更好的训练效果呢?

在国标麻将强化学习的训练过程中，我们时常可以观察到一些在我们的人类经验看来不好的行为：例如盲目吃碰杠导致失去很多潜在的番种可能：如下图 3.7 所示，当前玩家手牌为[“W3”，“W4”，“W4”，“W5”，“T5”，“T6”，“T7”，“B3”，“B4”，“B9”，“B9”，“F3”，“J2”]，潜在的主要番种为“三色三步高”，形如[“W4”，“W5”，“W6”，“T5”，“T6”，“T7”，“B3”，“B4”，“B5”]，当前局面下应该等待牌张“W6”或“B5”，距离“三色三步高”为主番种的和型为二上听。但如果此时有人打出一张“W4”，智能体可能选择碰牌，这样就完全破坏了主番种，变成了距离和牌局面一上听，但失去足够八番和牌的番种的情况；或能够优化手牌结构时却无法识别，并将摸入牌张打出：如下图 3.8 所示，当前玩家手牌为[“W9”，“W9”，“W9”，“W2”，“W2”，“W7”，“W8”，“W9”，“T8”，“T8”，“J2”，“J2”，“J2”]，其中[“W9”，“W9”，“W9”]为碰牌得到的附露。该牌型貌似已经听牌，听的牌张为“T8”或“W2”，但在国标麻将的八番起和规则下，该牌型的番种只有“箭刻（由箭牌中、发、白的刻子（杠），计 2 番）”+“四归一（四张相同且未开杠的牌，计 2 番）”+“幺九刻（由幺九牌组成的刻子（杠），每副计 1 番）”+“缺一门（和牌中只包含两种花色序数牌，计 1 番）”，合计番数为 2 + 2 + 1 + 1 = 6 番，不满足八番起和的要求。但此时摸上了一张“T9”，此时如果打出“T8”，将和的牌张调整为“T7”，虽然听牌的牌张数减少了，但最终的番种可以多出“喜相逢（两种花色序数相同的两副顺子，计 1 番）”和“边张（听顺子 123 的 3 或者 789 的 7 成基本和牌型，且整手牌只听这一种牌，计 1 番）”，合计番数达到了 2 + 2 + 1 + 1 + 1 + 1 = 8 番，满足了八番起和的要求。但智能体很可能因为不减少上听数且听的牌张数减少，从而放弃这种打法，将“T9”打出。如何针对性这些局面进行改进，是一个相当重要的问题。

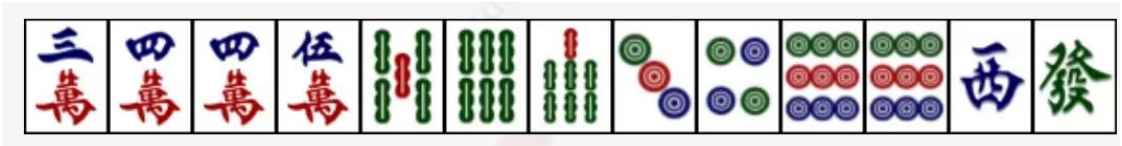


图 3.7 盲目吃碰

图 3.8 转换和型

而在试图解决该问题的过程中我们也遇到了一些问题：由于国标麻将环境是一个相当复杂的黑箱，对局面产生重要影响的决策时间点跨度可以较大，隐藏信息丰富，各超参对环境性质的影响是复杂且非独立的，故我们很难直接通过更改超参解决观察到的问题；而对奖励函数直接进行控制则是如前文所述，可能导致训练容易陷入局部最优的状态。

而在课程学习的框架中，该问题则是天然有望得到解决。我们首先引入的一上听、二上听等更接近听牌的局面，相当于固定了麻将 14 张牌和型中的至少 12 张，也基本锁定了可做的和牌番种，等于缩小了原本复杂的局面，同时也排除了决策时间长和面对局面复杂等不利于超参对局面实施直观影响的因素。理想的强化学习算法是一个映射，它是将马尔可夫决策过程的参数映射到该马尔可夫决策过程的最优策略。在上听数小的初始局面的课程学习流程下，通过对超参数和以奖励函数 Reward 为代表的问题参数的调整，我们可以创建出与原马尔可夫过程共享状态 State、观测视角 Observation 和动作空间 Action Space 以及转移机制 Transition Dynamics 的辅助式课程 MDPS，相当于在原本的如图 3.6 所示的线性课程学习流程中加入分支课程（图 3.9），可以更容易地帮助将智能体引导至我们更倾向的行为。同时经过听牌状态和一上听状态训练过的智能体具备基本的对牌型的掌握，也更容易直观的表现出我们能够观测到的行为，以帮助我们确认该类课程的效果。

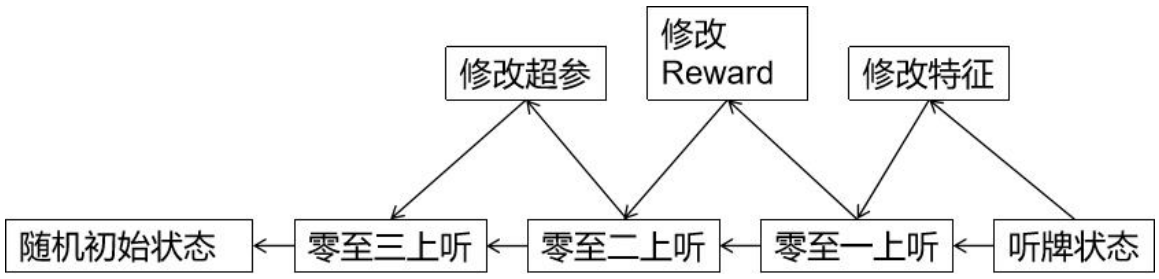


图 3.9 分支课程示意图

然而，由于前文提到的国标麻将黑箱性质的问题，我们暂时还不具备将所有坏现象和超参以及问题参数做映射的完备先验，这时，课程学习由于其可变范围小且易观测的特点，可以发挥“可解释性（explainability）”的作用。可解释性 AI 的概念[90]很难被直接定义，可以被泛泛的定义为一切有助于人们理解 AI 算法的工作，其角度可以是从输入特征依赖，模型的特性，工作原理，安全性，和公平性等等不同的角度；而可解释性 AI 领域最终的目标则是帮助人们全面的，从多个角度完整的理解所有 AI 算法的特性和原理。本文中提到的“可解释性”也在不少文章中被称为“阐述性（interpretability）”。“阐述性”更多与人类能否对 AI 算法的输出进行理解与关联，如：一个更可阐述的 AI 系统做出的决策，人类能更清晰的分析出是哪些因素所导致的这个决策，即有更清晰的因果逻辑。但在本文中我们还是统称其为“可解释性”。对应到本文的工作中，“可解释性”具体体现在帮助我们具体理解国标麻将这一特定场景中超参数和 reward shaping 等的更改对局面所产生的具体影响，以作为后续丰富课程学习课程的先验知识储备。

3.3.2 奖励函数 Reward 对国标麻将训练的具体影响

对奖励函数 Reward 的调整被视为能够最直观的改变 AI 行为的方法。在课程学习训练的过程中，由于我们是从听牌状态开始训练，和牌是相当容易的，待训练 AI 往往会产生喜欢吃、碰、杠的倾向，从而影响后续的训练表现，因为在国标麻将中，门清也是有分数的，而过多过早的吃碰杠也会影响潜在的其他番种的可能性。更重要的是，很多时候 AI 的吃、碰、杠选择是无谓的，并不能减少上听数，能达到胡牌状态也只是因为初始状态离和牌足够接近。故尝试通过在奖励函数 Reward 中加入上听数计数来限制无谓的吃、碰、杠行为。如果 AI 进行了一次吃、碰、杠操作但并没有使当前上听数减小的话，则给予奖励函数 Reward 一次-1 的惩罚（相当严厉，毕竟和牌一局也才+5）。算法伪代码如下算法 1 所示：

```
Function: shantenNumberCheck
1: while Chi, Peng, Gang, AnGang, BuGang do
2:     if shantenNumber remains then
3:         reward = reward - 1;
4:     else
5: end
```

算法 1 避免无意义吃、碰、杠伪代码

我们在一上听局面下使用掌握了基本和牌能力的 AI 进行效果实验，并与不添加吃、碰、杠限制的 AI 进行对比。实验结果于第五章进行展示。

3.3.3 折扣因子 γ 对国标麻将训练的具体影响

γ 是一个在强化学习中用于衡量未来奖励的折扣因子。它控制了对未来奖励的重视程度。在强化学习问题中，一个 AI 可能会在当前时刻做出决策，但这个决策可能会影响未来的奖励。γ 的值在 0 和 1 之间，越接近 1 表示越重视未来奖励。折扣因子的引入有助于 AI 在决策时考虑未来奖励，而不仅仅是眼前的即时奖励，即让 AI 变得更加远视。我们在二上听局面下使用训练至一上听局面且具备基本胡牌能力的 AI 进行实验。实验结果于第五章进行展示。

3.3.4 特征工程对国标麻将训练的具体影响

特征工程一定程度上代表了 AI 的理解能力。一般来说，在国标麻将中，越复杂的特征能够越好的帮助传达局面上牌张的信息给待训练 AI。先前的训练中，我们定义的特征为 145 * 4 * 9 维，145 维中包括了自己手牌 * 4 + 每个人动作(吃 * 4 + 碰 * 1 + 杠 * 1) * 4 个玩家 + 自己暗杠 * 1 + 每个人弃牌历史 28 * 4 + 剩余牌 * 4，这样的特征花了很多位置构筑其他玩家的动作、出牌历史以及弃牌历史，无疑是很全面的。但在课程学习中，听牌状态下开始训练时，由于当前待训练 AI 已经很接近和牌，AI 或许可以更专注于自己手牌的牌型优化，而不是过多关注其他玩家位置打出的牌张。因此，在听牌局面的训练中，我们尝试使用简化版特征，表示为 38 * 4 * 9 维，其中 38 通道 = 门风 * 1 + 圈风 * 1 + 自己手牌 4 + 每个人弃牌历史 4 * 4 + 每个人副露情况 4 * 4。这样的特征安排减少了其他玩家动作与弃牌占据的表达空间，更适合简单局面下 AI 的训练。分别使用这两种特征进行听牌局面训练，实验结果于第五章进行展示。

3.4 本章小结

本章介绍了本实验室研发运行的 Botzone 在线多 AI 游戏 AI 对战平台以及本实验室于 IJCAI 举办的国标麻将人工智能比赛，并通过比赛、课程以及实验室此前的国标麻将强化学习训练中遇到的问题中总结出了国标麻将 AI 强化学习训练的难点及产生这些难点的原因。针对国标麻将这一特定训练场景，我们提出了基于课程学习的训练方案，

通过划分不同的初始手牌上听数作为训练的初始状态，以满足课程学习“从易到难”的基本思想。此外我们还对构造新的马尔可夫过程来帮助控制训练进程从而能得到更好的训练效果这一想法做出了分析，希望对超参以及问题参数的调整能帮助我们更好的理解与解释国标麻将这一黑箱。

第四章　国标麻将强化学习的课程学习训练框架实现

本章将描述国标麻将强化学习的课程学习训练框架的实现。首先介绍了国标麻将强化学习框架的实现，并于其实现了按照上听数由易到难排列的课程学习流程的部署；之后对该框架的实现难点、创新性和可复用性进行了性质上的分析。

4.1　强化学习框架

首先介绍本文实现的针对国标麻将的强化学习训练框架，框架的结构示意图如下图 4.1 所示：

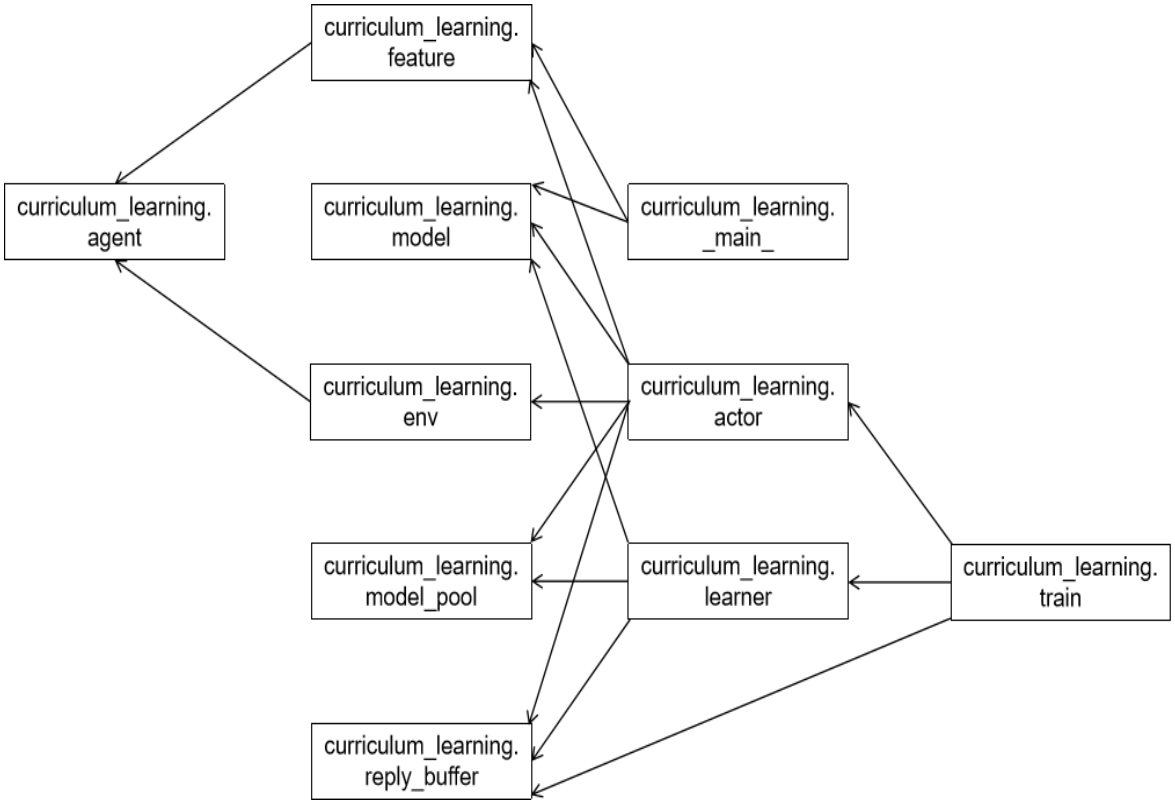


图 4.1 国标麻将强化学习训练框架

由上图可以看到，本文使用的国标麻将强化学习训练框架是由国标麻将环境组件 env.py；强化学习组件，其中包括 Actor 组件 actor.py，Learner 组件 learner.py；特征处理 agent.py、feature.py，神经网络 model.py；分布式 RL 训练所依赖的组件，其中包括

模型池 model_pool.py，样本池 replay_buffer.py，以及训练入口 train.py 和 botzone 交互入口__main__.py 组成。下面对每一部分的实现进行分析展示。

4.1.1 国标麻将环境组件模块

国标麻将环境组件模块由 env.py 组成，env.py 中包含 MahjongGBEnv 类。在 MahjongGBEnv 类中，本文实现了对国标麻将环境的包装。首先，国标麻将中，每个回合可能有多个玩家决策，因此环境返回的状态是一个字典（state_dict），字典中的 key 是当前回合需要决策的玩家，而字典中的 value 是该玩家的视野 observation。env.reset 函数负责返回第一回合状态的 state_dict，默认为随机生成的牌墙。环境中定义了所有麻将的可行动作：env.deal 表示派发初始手牌，env.drawTile 表示从牌堆中摸取初始手

Function: step

```
try:
    if state == 0:
        response = split_response(action2response(action_dict))
        if response[0] == 'Play':
            discard_tile(curPlayer, response[1])
        else:
            raise InvalidAction(curPlayer)
        isAboutKong = False
    else if state == 1:
        response = split_response(action2response(action_dict))
        if response[0] == 'Hu':
            show_tile(curTile)
            check_mahjong(curPlayer, isSelfDrawn = True, isAboutKong =
isAboutKong)
        else if response[0] == 'Play':
            add_tile_to_hand(curPlayer, curTile)
            discard_tile(curPlayer, response[1])
        else if response[0] == 'Gang' and not myWallLast and not wallLast:
            concealedKong(curPlayer, response[1])
        else if response[0] == 'BuGang' and not myWallLast and not
wallLast:
            promoteKong(curPlayer, response[1])
        else:
            raise InvalidAction(curPlayer)
    else if state == 2:
        responses = get_other_responses(action_dict, curPlayer)
        actions = split_responses(responses)
        handle_actions(actions)
    else if state == 3:
        responses = get_other_responses(action_dict, curPlayer)
        handle_bu_gang_responses(responses)
except InvalidAction as e:
    handle_invalid_action(e)
return get_observation(), get_reward(), is_done()
```

算法 4.1 env.step 函数逻辑

牌，env.draw 表示牌局中摸牌，env.discard 表示打出牌张，env.chow 表示吃牌，env.pung 表示碰牌，env.kong 表示杠牌，env.promoteKong 表示补杠，env.concealedKong 表示被抢杠后取消杠牌动作。env.checkMahjong 负责调用算番库，来判断当前局面是否满足八番起和的要求。而 env.step 函数则定义了国标麻将牌局中所有的可行动作组合，即吃、碰、杠后打出牌张，打出牌张后吃、碰、杠、过牌、和牌，摸牌后和牌、杠、补杠、打出牌张，补杠后和牌、打出牌张，以及出牌错误。 env.step 函数同时还返回局末状态 state_dict，局末奖励 reward_dict 以及完成牌局信号 done。env.step 函数的具体逻辑如算法 4.1 伪代码所示。

4.1.2 国标麻将强化学习框架组件模块

国标麻将强化学习框架组件模块由 Actor 组件 actor.py，Learner 组件 learner.py；特征处理 agent.py、feature.py 和神经网络 model.py 组成。在该模块中，本文实现了国标麻将环境的强化学习功能。

4.1.2.1 PPO 算法

本文使用的强化学习算法为 PPO（Proximal Policy Optimization）[54]，即近端策略优化算法，是一种基于策略（policy-based）的强化学习算法，是一种 off-policy 算法。

依据 Actor 网络的更新方式，PPO 算法可被被分为两种主要的变体：PPO-Penalty 和 PPO-Clip。PPO-Penalty 类似于 TRPO 算法，它使用 KL 散度作为一个约束条件，但是将 KL 散度作为目标函数的一个惩罚项，而不是一个硬性约束，并且自动调整惩罚系数，使其适应数据的规模。PPO-Clip 则没有 KL 散度项，也没有约束条件，而是使用一种特殊的裁剪技术，在目标函数中消除了新策略远离旧策略的动机。本文选择采用的也是 PPO-Clip，算法伪代码如下所示。

Algorithm 1 PPO, Actor-Critic Style

for iteration=$1, 2, \ldots$ **do**
  for actor=$1, 2, \ldots, N$ **do**
    Run policy $\pi_{\theta_{\text{old}}}$ in environment for T timesteps
    Compute advantage estimates $\hat{A}_1, \ldots, \hat{A}_T$
  end for
  Optimize surrogate L wrt θ, with K epochs and minibatch size $M \leq NT$
  $\theta_{\text{old}} \leftarrow \theta$
end for

算法 4.2 PPO-Clip 伪代码

PPO 算法是在 Policy Gradient 算法的基础上由来的，Policy Gradient 是一种 on-policy 的方法，他首先要利用现有策略和环境互动，产生学习资料，然后利用产生的资料，按照 Policy Gradient 的方法更新策略参数。然后再用新的策略去交互、更新、交互、更新，如此重复。这其中有很多的时间都浪费在了产生资料的过程中，所以我们应该让 PPO 算法转化为 Off-Policy。Off-Policy 的目的就是更加充分的利用 actor 产生的交互资料，增加学习效率。

PPO 算法的目标是在与环境交互采样数据后，使用随机梯度上升优化一个“替代”目标函数，从而改进策略。PPO 算法的特点是可以进行多次的小批量更新，而不是像标准的策略梯度方法那样每个数据样本只进行一次梯度更新。它主要包括以下几个关键的步骤：

1、收集数据：通过在环境中执行当前策略（policy）来收集一组交互数据。这些数据包括状态（state）、动作（action）、奖励（reward）以及可能的下一个状态。

2、计算优势估计：为了评价一个动作相对于平均水平的好坏，需要计算优势函数（advantage function）。这通常是通过某种形式的时间差分（TD）估计或者广义优势估计（GAE）来完成的。

3、优化目标函数：PPO 算法使用一个特殊设计的目标函数（公式 4.1）

$$r_t(\theta) = \frac{\pi_\theta(a_t \mid s_t)}{\pi_{\theta old}(a_t \mid s_t)} \tag{4.1}$$

这个函数涉及到概率比率，表示旧策略。目标函数的形式通常为（公式 4.2）：

$$L(\theta) = E(min(r_t(\theta)\hat{A}, clip(r_t(\theta), 1-\epsilon, 1\epsilon)\hat{A})) \tag{4.2}$$

其中，$\hat{A}$ 是优势函数的估计 ε 是一个小的正数（如 0.1 或 0.2），clip 函数限制了概率比率 $R_t(\theta)$ 的变化范围，防止更新步骤过大。

4、更新策略：使用梯度上升方法来更新策略参数θ，即$\theta \leftarrow \theta + \alpha \nabla_\theta L(\theta)$，其中$\alpha$是学习率。

5、重复步骤：使用新的策略参数重复以上步骤，直到满足某些停止准则，比如策略性能不再提升或者已经达到了一定的迭代次数。

PPO 算法的关键之处在于它通过限制策略更新的幅度，使得学习过程更加稳定。在每次更新时，概率比率 $R_t(\theta)$ 被限制在[$1-\varepsilon$ ， $1+\varepsilon$]范围内，防止由于单个数据点导致的极端策略更新，这有助于避免策略性能的急剧下降。同时，PPO 允许在每次迭代中使用相同的数据多次进行策略更新，这提高了数据效率。

4.1.2.2 actor 组件

Acotr 组件 actor.py 中包含 actor 类，目的是使用模型决策与环境交互收集样本。首先实例化网络模型，将其连接到模型池。之后从模型池中加载初始模型（如果没有就等待初始模型进入），并循环运行对局采样数据：首先从模型池中选取模型作为本局参与者；接着与环境 env.py 进行交互，对每个玩家的每个 step 过一次网络模型，得到每个玩家分别的一整局的状态序列 state，动作序列 action，局末奖励 reward（和牌 +5，点炮 -3）和 value 数据序列；然后通过广义优势估计 GAE 计算优势函数，计算出所需的 adv, target；最后将最新模型玩家的 state, action, adv, target 数据放入样本池。

GAE （Generalized Advantage Estimation)[91]是一种用于估计优势函数的方法。优势函数是指在某个状态下，采取某个动作比按照当前策略采取动作所能获得的期望回报的差值。优势函数可以用来减少策略梯度估计的方差，提高学习效率。GAE 的思想是利用值函数来对优势函数进行多步估计，同时使用一个衰减因子来平滑不同步长的估计，从而得到一个既有较小偏差又有较小方差的优势函数估计。具体来说，GAE 使用以下公式来计算优势函数（公式 4.3）估计：

$$\begin{aligned} A_t^{GAE} &= \delta_t + \gamma\lambda\delta_{t+1} + \gamma^2\lambda^2\delta_{t+2} + \cdots \\ &= \sum_{k=0}^{\infty}(\gamma\lambda)^k\delta_{t+k} \end{aligned} \tag{4.3}$$

计算 GAE 的伪代码如下算法 4.3：

```
Algorithm: GAE
1: td_target = rewards + gamma * values[1:]
2: td_delta = td_target - values
3: adv[i] = td_delta[i] * gamma * lambda + adv[i + 1]
```

算法 4.3 GAE 计算伪代码

4.1.2.3 learner 组件

learner 组件 learner.py 中包含 learner 类，目的是从样本池中采样训练网络。首先创建模型池。之后实例化网络模型，将初始参数放入模型池，并开始进行迭代训练：等待样本池的容量达到一定数值后训练开始，首先从样本池中采样；之后通过算法 4.4 伪代码逻辑，计算 PPO loss 以更新网络；接着将新模型参数放入模型池并定时保存新模型。

```
Algorithm: PPO loss
old_probs, values = model(states)
for _ in range(epoch):
    new_probs, _ = model(states)
    ratio = exp(log_new_probs - log_old_probs)
    policy_loss = -min(ratio * advs, clip(ratio, 1 - ε, 1 + ε) * advs)
    value_loss = MSE(values, targets)
    entropy_loss = new_probs.entropy()
```

算法 4.4 PPO loss 计算伪代码

4.1.2.4 特征组件

特征组件包含 agent.py 和 feature.py。agent.py 中包含 agent 基类，主要完成的任务是按顺序接收一个玩家在对局中观察到的所有事件，并在每个决策进行的节点整理出相对应的状态特征；同时将网络输出的动作转换为事件，并也进行如上的流程。Feature.py 需要将 agent 类传入 MahjongGBEnv 类中进行特征处理。中包含 FeatureAgent 基类。FeatureAgent 类继承自 agent 类，按照国标麻将的规则处理出每个决策点的所有的可行动作以及特征的表示。FeatureAgent 中定义的动作空间为 235 维，如表 4.1 所示，具体表示为过牌 * 1 + 和牌 * 1 + 弃牌 * 34 + 明杠 * 34 + 暗杠 * 34 + 补杠 * 34 + 碰牌 * 34 + 吃牌 * 63，其中吃牌的 63 维为 3 种花色 * 7 张顺子中间牌张可能性（2 到 8） * 3（吃的是顺子中的哪一张牌张）。FeatureAgent 中定义的特征为 145 * 4 * 9，

其中 4 * 9 表示所有的可能牌张, 145 维则由自己手牌 * 4 + 每个人动作(吃 * 4 + 碰 * 1 + 杠 * 1) * 4 个玩家 + 自己暗杠 * 1 + 每个人弃牌历史 28 * 4 + 剩余牌 * 4 组成。

表 4.1 235 维动作空间表示

动作	数量	指令码
过牌	1	1
和牌	1	2
弃牌	34	万牌 (2-10) 条牌 (11-19) 筒牌 (20-28) 风牌 (29-32) 箭牌 (33-35)
明杠	34	万牌 (36-44) 条牌 (45-53) 筒牌 (54-62) 风牌 (63-66) 箭牌 (67-69)
暗杠	34	万牌 (70-78) 条牌 (79-87) 筒牌 (88-96) 风牌 (97-100) 箭牌 (101-103)
补杠	34	万牌 (104-112) 条牌 (113-121) 筒牌 (122-130) 风牌 (131-134) 箭牌 (135-137)
碰牌	34	万牌 (138-146) 条牌 (147-155) 筒牌 (156-164) 风牌 (165-168) 箭牌 (169-171)
吃牌	63	万牌 (172-192) 条牌 (193-213) 筒牌 (214-234)

4.1.2.6 网络模型结构

model.py 中包含网络模型，本文使用 ResNet18，即 18 层 ResNet。输入为 145 * 4 * 9 的特征；ResBlock 由两层输入输出均为 256、卷积核大小为 3 * 3、最大步长为 1 的卷积组成，激活函数为 ReLu，层数为 18；摊平后分别输出动作输出 235 维和价值输出 1 维。

4.1.3 国标麻将分布式强化学习组件模块

国标麻将分布式强化学习组件模块包括模型池 model_pool.py，样本池 replay_buffer.py 两部分。分布式强化学习的目标是多个 Actor、一个 Learner 可以运行在不同进程。

4.1.3.1 样本池

样本池 reply_buffer.py 端，支持多个 actor 同时放入数据，一个 learner 乱序取出数据的功能。样本池容量有限，所以当样本数量超过样本池容量时，可以支持一些替换策略（如 FIFO）。

4.1.3.2 模型池

模型池端 model_pool.py 中包含了 ModelPoolServer 和 ModelPoolClient 两个类。Learner 在放入模型信息和参数后，需要能够支持多个 Actor 同时读取模型信息和参数，并且在每局对局前选择某个或某些模型作为本局参与者；与样本池端相同，模型池容量有限，故而当模型数量超过模型池容量时，可以支持一些替换策略（如 FIFO 也可以支持其他类型的替换策略）。

4.1.4 其他模块

其他模块包括训练入口 train.py 和 botzone 交互入口__main__.py。

train.py 是训练入口，负责创建样本池、Actor 和 Learner 对象并启动所有组件。train.py 还指定了训练中用到的所有超参数，如下表 4.2 所示：

表 4.2 训练使用的超参数

超参数名称	超参数意义	取值
replay_buffer_size	样本池容量（样本个数）	50000
replay_buffer_episode	样本池对局排队个数	400
model_pool_name	模型池容量（模型个数）	20
num_actors	Actor 数量	4
episodes_per_actor	每个 Actor 采样的对局数	10000
gamma	PPO 参数 γ	0.99
lambda	PPO 参数 λ	0.95
min_sample	learner 启动训练的最小样本数	200
batch_size	batch 大小	1024
epochs	每个 batch 训练次数	3
clip	PPO 裁剪系数	0.2
lr	学习率	2e-5
value_coeff	value loss 系数	1
entropy_coeff	entropy loss 系数	0.01
device	learner 使用 CPU/GPU	CPU
learner_interval	learner 保存模型的时间间隔	60
ckpt_save_interval	checkpoint 保存时间间隔	300

__main__.py 是负责与 botzone 交互 Bot 代码。将 Botzone 的输入整理后交给 Agent 类处理，得到状态特征作为网络输入，网络输出的动作再转成字符串，进一步转成 Botzone 输出格式。

4.2 课程学习部署

首先以过去三届 IJCAI 国标麻将比赛中最强的的国标麻将 AI（以下简称冠军 Bot）的和牌对局为参照对训练初始局面进行构筑。冠军 Bot 的自对弈对局数据以事件序列的形式记录每一局的过程，我们需要在整局的事件序列中拆分每个位置玩家的手牌信息，并使用工具 fanCalcLibPy38.so 对冠军 Bot 和牌位置玩家的手牌进行处理。该工具基于搜索，对其输入手牌、附露、场面上已知的牌墙的信息（牌河、附露数量）、门风、圈风、搜多少个结果、所需上听数等信息。对于国标综合牌型，采取自动搜索 + 算番库确定是否满足和牌的方式；对于国标特殊牌型(8 番以上)，采取特例搜索。最后返回该局面下的上听数、手牌中有用的牌张、有效进张、需要的牌张的剩余张数、目标牌型以及最后的番种，而我们需要的只是输出的上听数。我们只需要对该工具将冠军 Bot 和牌位置玩家的手牌历史，便可以将手牌历史按照输出的上听数进行分类，构成按照上听数分类的初始局面集合，并以此为起点进行训练。

接下来，我们将这些初始局面作为课程，使用 4.1.1 中提到的国标麻将环境组件进行加载。env.py 中，env.reset 函数负责返回第一回合状态的 state_dict，默认为随机生成的牌墙。在课程学习训练的一个课程中，我们将该局赢家风位玩家初始手牌固定为按照上听数分类的初始局面集合中的一个随机局面，并载入该局对应的圈风，确保其获得的初始手牌为课程对应的上听数。此后从全部的牌张中减去该玩家的初始手牌牌张，并将剩下的牌张用于组成其余玩家的初始手牌以及全部玩家的牌墙。训练只针对被分发了赢家手牌的玩家位置。训练从听牌状态开始，我们时刻观察训练的进展，并在 AI 于简单课程中获得好的和牌效果之后更换更高上听数的初始局面集合进行新的课程训练。

此外，针对 3.3 中提到的国标麻将课程学习的可解释性质，我们还进行了针对超参数 γ、奖励函数 Reward 以及特征工程进行改动的课程流程尝试，结果将在第 5 章中予以展示。

4.3 国标麻将课程学习训练框架性质分析

本章将从该国标麻将课程学习训练框架的难点、创新性以及可复用性等几个方面对该框架进行性质分析。

4.3.1 框架难点分析

以上的国标麻将课程学习训练框架在实现过程中，遭遇了如下困难：

4.3.1.1 样本池跨进程通信

为了实现样本池端的功能，需要实现 actor 端和 learner 端的跨进程异步样本传递。该难点可以通过 Python 提供的跨进程通信功能 Queue 来实现。在多线程编程中，当多个线程需要访问共享数据时，很容易出现竞争条件，即多个线程尝试同时访问和修改相同的数据，导致数据不一致或丢失。Queue，即队列，是一种用于解决这种问题的数据结构，它提供了一个适用于多线程编程的先进先出的数据结构，确保多个线程可以安全地访问和修改它，用来在生产者和消费者线程之间的信息传递。reply_buffer.py 中包含 ReplayBuffer 类。我们的方案是，在 actor 端进行数据收集后，只利用 Queue 进行跨进程数据传递，最后在端接收取出所有数据放进另一个 buffer 中重新维护 (超出容量替换、替换策略、乱序采样等机制)，以此支持支持跨进程的异步样本传递。ReplayBuffer 类实现方案逻辑如下图 4.2 所示。

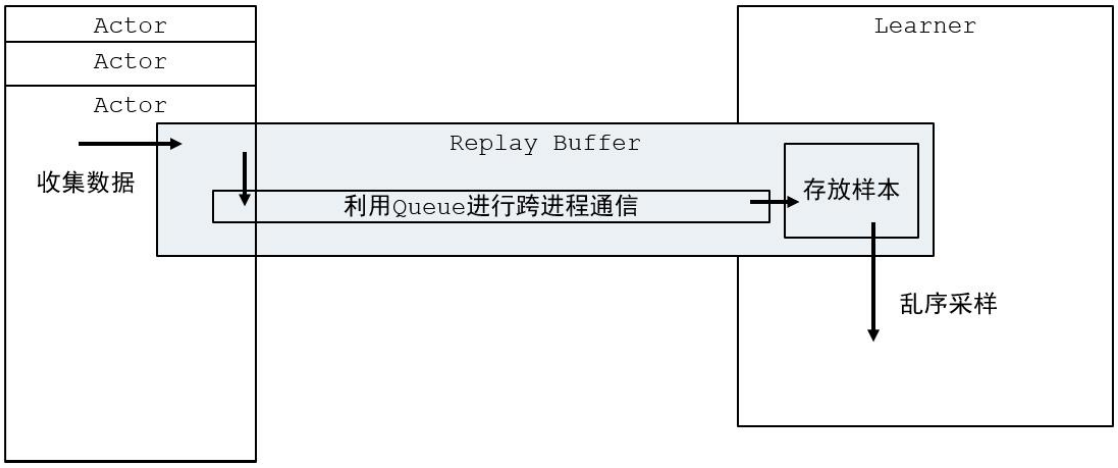


图 4.2 跨进程的异步样本传递

4.3.1.2 模型池跨进程通信

模型池端需要实现的功能则是跨进程通信功能 Queue 无法实现的。由于在模型池端，当 Learner 放入模型的信息和参数时，多个 Actor 都可能需要读取同一个模型的信息和参数。而如果使用 Queue，每当 Learner 放入新模型，任意 Actor 需要读新模型时，都必须从 Queue 中将新模型取出，从而导致其他 Actor 无法得到该新模型的信息与参数。

对此，我们选择使用共享内存 multiprocessing.shared_memory 模块，使用这个模块可从进程直接访问共享内存，该模块提供了一个 SharedMemory 类，用于分配和管理多核或对称多处理器（SMP）机器上进程间的共享内存。创建一个新的共享内存块或者连接到一片已经存在的共享内存块。每个共享内存块都被指定了一个全局唯一的名称。通过这种方式，进程可以使用一个特定的名字创建共享内存区块，然后其他进程使用同样的名字连接到这个共享内存块，并进行读取/写入操作。该方案的缺点是，一旦创建一个新的共享内存块后，内存块的大小固定，所以不适合用于样本池。而优点则是由于不存在跨进程数据传递，故开销相对较小。我们的方案是，Learner 端每放入一次新模型，都创建一块新的共享内存，即使模型参数较大，多个 Actor 进程也不需要缓存参数，可以直接频繁读取参数所在的内存。

但在实现过程中，我们发现了一个问题，共享内存的名称无法传递给 Actor，故也无法通过名称直接检索。基于此情况，我们选择使用共享内存 multiprocessing.shared_memory.ShareableList 模块。提供一个可修改的类 list 对象，其中所有值都存放在共享内存块中。这限制了可被存储在其中的值只能是 int, float, bool, str （每条数据小于 10M）, bytes （每条数据小于 10M）以及 None 这些内置类型。它另一个显著区别于内置 list 类型的地方在于它的长度无法修改（比如，没有 append, insert 等操作）且不支持通过切片操作动态创建新的 ShareableList 实例。对此，我们的方案修改为，仍然基于共享内存，但可以维护一个定长的列表。该列表用于存放每个模型参数的 metadata，如序号、内存名称，以及时间、大小等 Learner 存入的信息。将 metadata 序列化成二进制数据，然后放在列表中，列表长度是模型池的最大容量。目前该方案实现为一个循环队列，在模型池达到容量限制时，替换掉旧模型（同时释放掉对应的共享内存），但也可以支持其他类型的替换策略。实现方案逻辑如下图 4.3 所示。

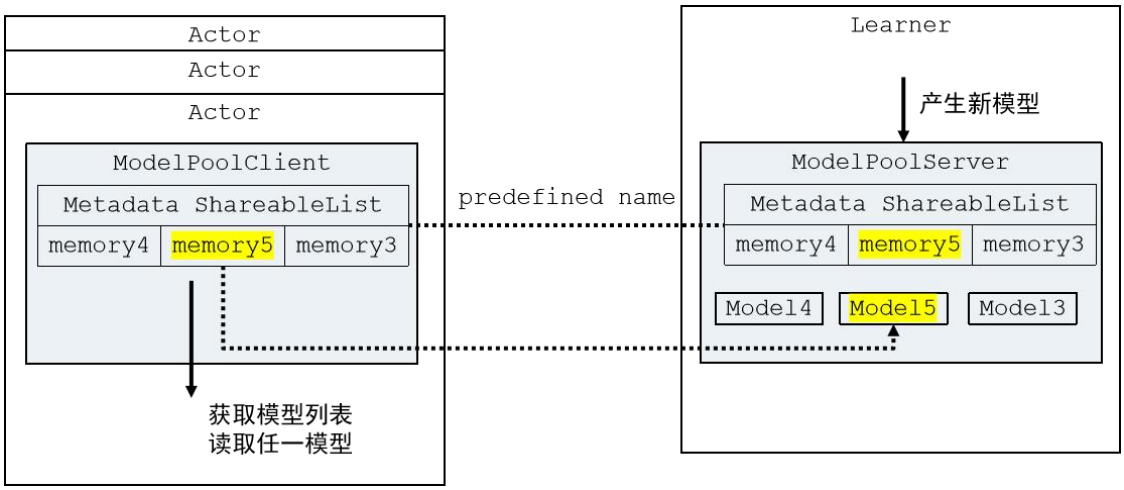


图 4.3 跨进程的异步模型传递

4.3.1.3 游戏逻辑调试

国标麻将作为麻将游戏中规则最复杂的存在，其游戏逻辑的处理在该框架的实现过程中的也是一处难点所在。麻将游戏中，待决策玩家每打出一张牌，环境就需要对潜在的吃、碰、杠或过牌操作做出判定，同时待决策玩家也会随着该打出牌张的归属进行变化，这本身已经是相当复杂的逻辑了。而国标麻将由于存在相当多的特殊番种，如“和绝张（牌池、桌面已亮明了 3 张牌，和所剩的最后一张牌）”番种的存在，意味着环境需要维护一个当前牌张所生张数的列表，供算番库调用以判断是否具备“和绝张”番种；又比如“抢杠和（和他家明刻加杠的那张牌，计 8 番。不计和绝张）”番种的存在，意味着在玩家明刻加杠的逻辑中，需要添加一次是否有玩家可以触发“抢杠和”的判定；还比如“海底捞月（牌墙已摸完，和本局打出的最后一张牌，计 8 番)”和“妙手回春（摸牌墙上的最后一张牌成自摸和，计 8 番。不计自摸）”番种的存在，意味着环境需要维护一个标识，判断当前摸入或打出的牌张是否为海底牌，而被打出的海底牌在规则中还不能被吃、碰、杠而只能被和，环境则还需要对这一特殊情况进行特判。国标麻将番种的数量多且彼此之间存在多种特判，意味着游戏逻辑的调试相当复杂，难度极大。

4.3.2 框架创新性与可复用性分析

该国标麻将强化学习的课程学习训练框架最主要的创新点是在国标麻将环境中提供了加载对局初始状态的功能，从而实现了对训练过程的极大程度控制。通过加载对局初始状态的功能，我们可以很灵活的为待训练 AI 提供各种初始手牌作为课程，例如本文中描述的按初始上听数划分的初始局面；又比如按番种划分的初始局面，以帮助 AI 掌握我们希望它掌握的番种和牌能力。我们还可以通过控制牌墙中牌张的排列，固定每个玩家该局的进张情况，实现对游戏进程进行进一步控制，从而完全消除牌类游戏固有的随机性问题。

该国标麻将强化学习的课程学习训练框架同时还具备很强的可复用性。与国标麻将相关的功能全部集中于 env.py 与 feature.py 中。

env.py 实现了国标麻将环境的包装，负责定义了所有国标麻将牌局中的可行动作，包括派发初始手牌、从牌堆中摸取初始手牌、牌局中摸牌、打出牌张、吃牌、碰牌、杠牌、补杠和被抢杠后取消杠牌动作。算番库的调用也于其中实现，来判断当前局面是否满足八番起和的要求。环境中调用算番库，用于判断当前局面是否满足八番起和的要求。环境中的 env.step 函数定义了国标麻将牌局中所有的可行动作组合，即吃、碰、

杠后打出牌张，打出牌张后吃、碰、杠、过牌、和牌，摸牌后和牌、杠、补杠、打出牌张，补杠后和牌、打出牌张，以及出牌错误。一局结束后，环境返回局末状态，局末奖励以及完成牌局信号。

而 feature.py 则是针对国标麻将性质的特征工程设计，包括其动作空间的维度于麻将的牌张数量相关，而特征的也设计为 N * 4 * 9 用于表示所有的可能牌张。

通过编写其他游戏对应的环境以及适配的特征工程，搭配其余的强化学习组件，同样可以正常进行强化学习训练。

4.4 本章小结

本章介绍了国标麻将的课程学习训练框架的实现。首先，4.1 介绍了本文实现的国标麻将强化学习框架，包括了使用的 PPO 算法以及国标麻将环境组件 env.py；强化学习组件，其中包括 Actor 组件 actor.py，Learner 组件 learner.py；特征处理 agent.py、feature.py，神经网络 model.py；分布式 RL 训练所依赖的组件，其中包括模型池 model_pool.py，样本池 replay_buffer.py，以及训练入口 train.py 和 botzone 交互入口 __main__.py 等组件的实现。4.2 则介绍了如何在强化学习训练框架中部署按照上听数由易到难排列的课程学习流程，包括如何将冠军 Bot 自对局获胜位置的牌谱按照上听数进行划分并加载进国标麻将强化学习框架的环境中构筑初始状态。最后，4.3 对该框架的实现难点、创新性和可复用性进行了性质上的分析。

第五章　国标麻将的课程学习训练结果与分析

本章对前文提出的国标麻将强化学习的课程学习训练方式进行了实验与展示。实验一通过将经过线性课程学习流程的 AI 放入 botzone 天梯进行对战，并与直接强化学习 AI 以及 2020 年 IJCAI 国标麻将 AI 比赛中的第一名、第四名、第五名和第十二名进行对战和对比，来验证本文提出的国标麻将强化学习的课程学习方法的效果。实验二则通过尝试修改奖励函数 Reward、折扣因子 γ 和特征维度来观察 AI 行为的改变，以验证 3.3 中提出的国标麻将课程学习的可解释性分析。

5.1　实验一：线性课程学习流程有效性验证

该实验旨在验证线性的课程学习训练的有效性。

5.1.1　实验目的

验证通过划分不同的初始手牌上听数作为训练的初始状态的线性课程学习流程的有效性，有效性具体体现在对 AI 训练效果进行的直接对局比较以及 AI 训练所消耗的时间与资源的相互对比。

5.1.2　实验设计

1）将使用 NVIDIA GeForce RTX 3080 10GB GPU，4 核 Intel(R) Xeon(R) Gold 6348 CPU @ 2.60GHz CPU 的实验配置进行 36 小时训练，并经过按上听数划分不同初始手牌来进行线性课程学习流程的国标麻将强化学习 AI 传入 Botzone 平台进行天梯对局。

2）将 1）中的国标麻将 AI 与使用相同实验配置和相同强化学习框架进行训练，但未经过课程学习训练流程训练的国标麻将强化学习 AI 进行 128 轮，每一轮使用相同牌墙 2v2 全排列打 6 局的对战。

3）将 1）中的国标麻将 AI 与 2020 年 IJCAI 国标麻将 AI 比赛的第一名（强化学习）、第四名（监督学习）、第五名（启发式）和第十二名（启发式+强化学习）的 AI 进行进行 128 轮，每一轮使用相同牌墙 2v2 全排列打 6 局的对战。

5.1.3 实验结果及分析:课程学习 AI 在 Botzone 天梯中的水平

实验 1）结果，天梯排名截图以及天梯排名趋势示意图如下图 5.1、图 5.2 所示。目前课程学习 AI——CurriculumBot 在 Botzone 国标麻将天梯共 551 个 AI 中排名第 57，达到了天梯前 10%的水平。

40	biubiuibiu	懂得都懂1	1222.73	123	13	.py36 ID
41	fake_env	vivian999	1220.97	1	0	.py36 ID
42	test2	Louison	1219.76	1ddens	2	.py36 ID
43	reslCM	lyingCS	1219.70	13张牌你能秒我？	0	.py36 ID
44	把他们也算上	武器大师	1215.62	开打开打	9	.py36 ID
45	show_me_the_hand	有手就行	1215.35	show me the hand	3	.py36 ID
46	c_bot	武艺_2101213229	1214.96	2021/1/2	16	.cpp17a ID
47	监督学习	melongena	1213.83	真正的监督学习-ResNet38	3	.py36 ID
48	地锅鸡	luyd_cpp	1212.76	res网络	6	.py36 ID
49	爆爆不是拖油瓶	金克斯	1211.82	初次见面	0	.py36 ID
50	陈大刀	zhan8855	1211.34	苟	7	.cpp17a ID
51	从零单排V2	工行卡十六号嗯	1210.91	1	3	.py36 ID
52	花三匁	ZCX	1210.90	=◦=	0	.cpp17a ID
53	testcppbest	breakthrough	1210.21	test	1	.cpp17a ID
54	test1	J_Alpha7	1207.23	test1	0	.py36 ID
55	cpp	ehh	1206.89	搜索	40	.cpp17 ID
56	test	b4thesunrise	1205.64	daf	2	.cpp17a ID
57	CurriculumBot	QQQQQFFF	1204.77	课程学习v1	0	.py36 ID
58	tttttest	xxxforver	1203.51	no	3	.py36 ID
59	shallow_white_m1	丰富的安静	1203.38	/**/	11	.py36 ID
60	qwqwqwq	yaphets	1202.65	.	0	.cpp17a ID

图 5.1 CurriculumBot 天梯排名截图

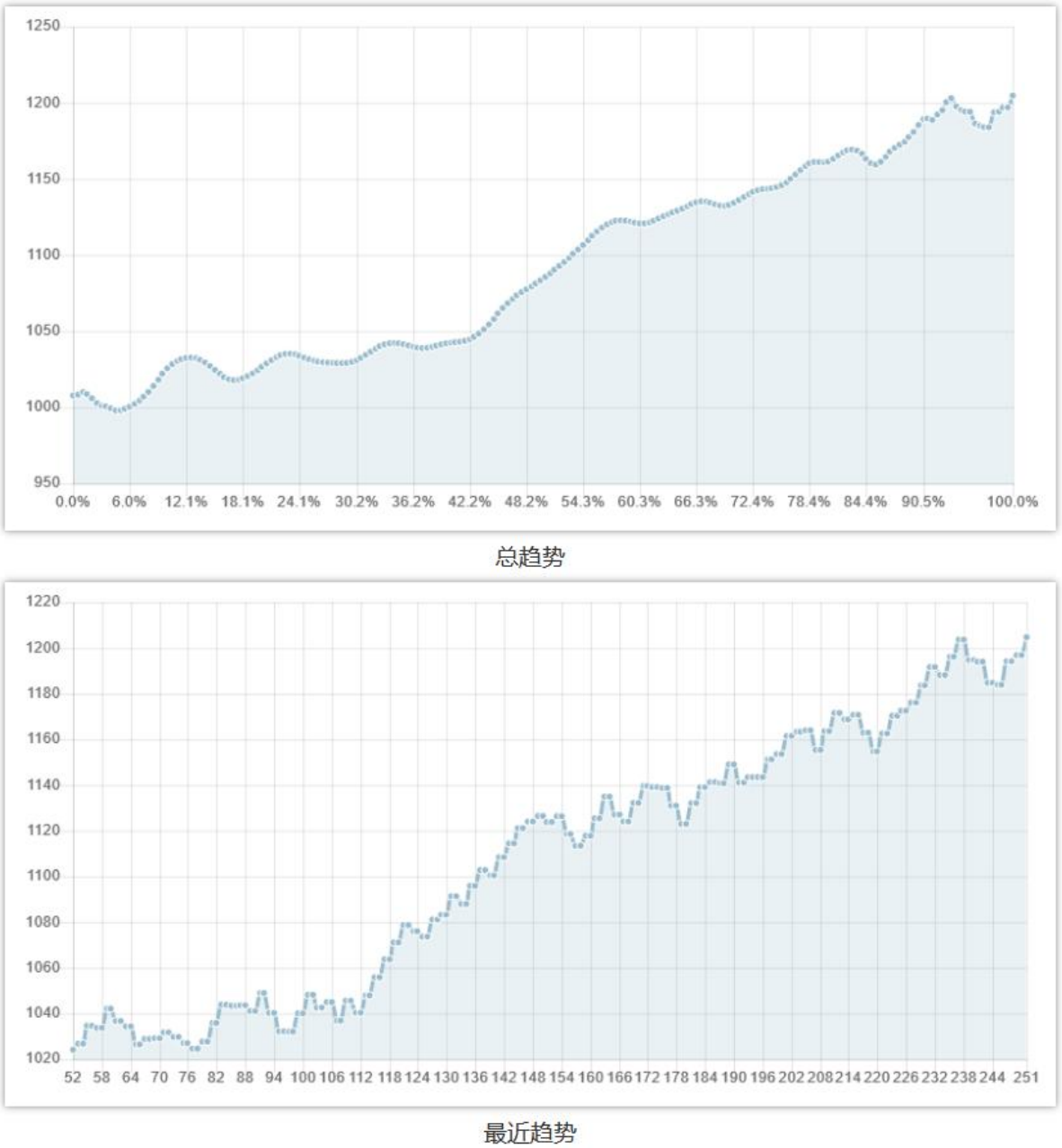


图 5.2 CurriculumBot 天梯排名趋势示意图

5.1.4 实验结果及分析:课程学习 AI 与直接强化学习 AI 对比

经过线性课程学习流程的国标麻将强化学习 AI 与使用相同实验配置和相同强化学习框架进行训练,但未经过课程学习训练流程训练的国标麻将强化学习 AI 进行 128 轮,每一轮使用相同牌墙 2v2 全排列打 6 局对战的结果如下表 5.1 所示。

表 5.1 课程学习 AI 与相同训练配置直接强化学习 AI 2v2 结果

Bot 名	分数
CurriculumBot	19969
rlfromzere	-19969

可以看到,使用课程学习流程的 AI 占据了绝对优势,累计小分为 19969 分,证明使用课程学习训练方式进行训练对国标麻将问题是极为有效的。同时,相比直接进行训练,课程学习还展现了很强的训练稳定性。由于初始课程局面较为简单且课程间的初始状态差异不大,课程学习训练基本都能稳定在低上听数阶段就掌握基本的和牌能力,在向更高上听数的课程进行训练的过程中也不会出现训练崩坏的情况。

5.1.5 实验结果及分析:课程学习 AI 与 2020 年 IJCAI 国标麻将 AI 比赛 AI 对比

经过线性课程学习流程的国标麻将强化学习 AI 与 2020 年 IJCAI 国标麻将 AI 比赛中的第一名、第四名、第五名和第十二名的 AI 分别进行了 128 轮,每一轮使用相同牌墙 2v2 全排列打 6 局对战的结果和算力资源、训练时间消耗对比如表 5.2、表 5.3、表 5.4、表 5.5 所示。

表 5.2 课程学习 AI 与 2020 年 IJCAI 国标麻将 AI 比赛第一名 AI 2v2 结果

Bot 名	分数	Cpu 数量	Gpu 数量	训练时间
CurriculumBot	-1245	4	1	36 小时
SuperJong	1245	100	2	48 小时

表 5.3 课程学习 AI 与 2020 年 IJCAI 国标麻将 AI 比赛第四名 AI 2v2 结果

Bot 名	分数	Cpu 数量	Gpu 数量	训练时间
CurriculumBot	-1354	4	1	36 小时
地锅鸡	1354	10	1	72 小时

表 5.4 课程学习 AI 与 2020 年 IJCAI 国标麻将 AI 比赛第五名 AI 2v2 结果

Bot 名	分数	Cpu 数量	Gpu 数量	训练时间
CurriculumBot	1803	4	1	36 小时
TheWitness 碰吃 debug	-1803	\	\	\

表 5.5 课程学习 AI 与 2020 年 IJCAI 国标麻将 AI 比赛第十二名 AI 2v2 结果

Bot 名	分数	Cpu 数量	Gpu 数量	训练时间
CurriculumBot	3615	4	1	36 小时
Test	-3615	256	8	14 小时

之所以选择 2020 年比赛中的 AI 是因为之后两年的比赛中训练出的 AI 水平均未达到 2020 年比赛的前三名。比赛的第一名的 AI 采用强化学习训练，但具体代码细节未公开。比赛的第四名采用监督学习进行训练，特征与动作空间定义与本文类似，监督学习的数据来自人类麻将网站大众麻将。该 AI 在训练过程中还进行了数据增强，具体为将万条筒互换和将万条筒的一九，二八，三七，四六进行互换，在绝大多数情况下不影响决策，从而显著增加数据量，提高了网络模型的泛化能力。比赛的第五名采用搜索+规则的方法，通过当前手牌搜索最接近的听牌状态缺少哪些牌张，并辅以诸如“五个对子可以做七对”或“全不靠有九张可以做”之类的经验口诀作为规则。比赛第十二名通过强化学习学习到的模型整合到行为树中，作为基本和型的策略，而如“七对（七个对子组成的特殊和牌型，计 24 番。不计不求人、门前清、单钓将）”、“推不倒（由牌面图形没有上下区别的牌组成的和牌）”、“绿一色（由【234568 条】及【发】之中的任何牌自称的和牌）”等特殊和型则使用启发式策略。可以看到，课程学习 AI 的能力不如比赛的第一名和第四名，但优于比赛的第五名和第十二名。由上表我们还可以看到，于所有基于学习的国标麻将 AI 相比，我们均消耗了更小的算力，并在训练时间上也存在一定优势。

5.2 实验二：课程学习可解释性验证

该实验旨在验证课程学习训练的可解释性。

5.2.1 实验目的

探究通过修改奖励函数 Reward、超参数以及特征构造新的马尔可夫过程来帮助控制原马尔可夫过程的训练进程，从而能得到更好的训练效果是否可行。

5.2.2 实验设计

1）在一上听局面下，对无谓的吃、碰、杠行为给予奖励函数 Reward 一次-1 的惩罚。

2）在二上听局面下分别使 $\gamma = 0.5$ 和 $\gamma = 0.99$ 来进行训练。

3）在听牌局面的训练中，尝试使用简化版特征。

5.2.3 实验结果及分析:修改奖励函数 Reward 对吃、碰、杠进行限制

为了限制 AI 无谓的吃、碰、杠选择，我们在 reward 中加入了每次 -1 的不减少上听数的吃、碰、杠操作的惩罚。表 5.6 给出了对吃、碰、杠进行限制与否对 AI 行为产生的影响。

表 5.6 吃、碰、杠限制实验对比

	出现全求人局数	总局数	全求人番种占比
不限制吃、碰、杠	58492	98209	59.6%
限制吃、碰、杠	7461	98209	7.6%

由上表我们可以看到，在 98209 局中，不进行吃、碰、杠限制的 AI 的最终胡牌番种中存在全求人番种的局数达到了 58492 局，概率达到了 59.6%。而在进行吃、碰、杠限制后, AI 的最终胡牌番种中存在全求人番种的局数则只有 7461 局, 概率只有 7.6%。虽然比起理想情况仍然偏高（强人类和出全求人数据为 2.91%，强 AI 和出全求人数据为 0.82%），但可以看到，在奖励函数 Reward 中进行吃、碰、杠限制在一上听的小规模状态中对 AI 的表现干预立竿见影。

5.2.4 实验结果及分析:不同的 γ 对训练的影响

为了测试 γ 代表的重视长期收益的属性在国标麻将中的具体表现形式，我们在二上听局面下分别使 $\gamma = 0.5$ 和 $\gamma = 0.99$ 来进行训练，结果如表 5.7 所示：

表 5.7 调整折扣因子 γ 对训练的影响

	第一次吃、碰、杠发生的进张数
γ = 0.5	1.74
γ = 0.99	3.18

在二上听局面下分别使 γ = 0.5 和 γ = 0.99 来进行训练，展现出的效果表现为第一次吃、碰、杠平均出现时间变晚。， γ = 0.5 时，AI 平均第 1.74 次进张就会出现第一次吃、碰、杠；而 γ = 0.99 时，AI 平均第 3.18 次进张才会第一次吃、碰、杠。看似差距不是很大，但每次通过摸牌产生进张之间都有三次吃、碰、杠的机会，而且由于局面为二上听，此时能够获进张的张数是不少的，故调整折扣因子 γ 的效果也可以说是比较明显的。

5.2.5 实验结果及分析:不同的特征表示对训练的影响

如 3.3.4 所述，原先的 145 维中包括了自己手牌 *4+ 每个人动作(吃 *4+ 碰 *1+ 杠 *1)*4 个玩家 + 自己暗杠 *1+ 每个人弃牌历史 28*4+ 剩余牌 *4，花了很多位置构筑其他玩家的动作、出牌历史以及弃牌历史，在听牌状态下或许是不合理的。而简化版特征，表示为 38*4*9 维，其中 38 通道 = 门风 *1+ 圈风 *1+ 自己手牌 4+ 每个人弃牌历史 4*4+ 每个人副露情况 4*4 则可能更适合听牌状态下的训练。为了验证该猜想，分别使用 145*4*9 维和 38*4*9 维的特征进行听牌局面的训练，结果如表 5.8 所示。

表 5.8 特征维度对训练的影响

	听牌状态起点胜率达到 90%
145*4*9 维	30 分钟
38*4*9 维	15 分钟

如上表所示，我们可以看到，小特征训练达到基本掌握和牌能力需要的时间基本都在 15 分钟上下，而完整特征则一般需要 30 分钟上下。说明在子问题情境下，简单课程可以使用更小、更贴合该子问题情景的特征进行训练，以节省训练时间和资源。后续则可以通过扩展特征维度来适配更复杂、难度更高的训练。

5.3 本章小结

本章展示了经过线性课程学习流程的国标麻将强化学习 AI 达到的良好效果，包括训出的 AI 在天梯上达到了 57 名，排名天梯前 10%，并大幅优于未使用课程学习训练方式的直接强化学习 AI。在与 2020 年 IJCAI 国标麻将 AI 比赛的代表性 AI 进行对比的中展现的水平也在当年比赛的 4 到 5 名之间。在算力资源消耗规模方面，该 AI 显著小于其余所有使用神经网络的 AI。训练所需时间上，该 AI 训练所需用时短于效果较好或接近的其他 AI，仅多于一个效果远逊该 AI 的 AI。此外，我们还尝试修改奖励函数 Reward、折扣因子 γ 和特征维度来观察 AI 行为的改变，获得了一些较为直观的行为差异，为后续引入更细致的课程来帮助提升课程学习训练效果打下了良好的基础。

第六章　总结与展望

本章节主要总结了本文对国标麻将强化学习的课程学习训练做出的工作，包括问题的提出与分析、课程学习方法的设计与可解释性分析、强化学习训练框架的设计与实现、课程学习训练于强化学习框架的部署与实现以及对国标麻将强化学习的课程学习训练的效果验证。本文还展望了课程学习下一步可以拓展的工作方向。

6.1　本文总结

本文研究的是如何提升国标麻将 AI 强化学习训练的效率和稳定性的问题。针对国标麻将强化学习 AI 及其在训练中遇到的问题，分析了其多目标长时决策的性质，并针对此性质设计了基于国标麻将场景的课程学习训练方案，在实际训练中达到了良好的效果。

游戏在人工智能研究中一直起到着很重要的作用，而国标麻将 AI 则由于其八番起和的复杂和牌条件，使得对局的随机性相对其他麻将有所减弱，从而比起其他游戏更具研究价值。然而国标麻将的多目标长时决策的性质会导致对其进行的强化学习训练容易无法掌握基本和牌能力或容易达到局部最优，对此，我们提出了将课程学习应用于国标麻将训练的方案。课程学习本质上是一种“由易到难” 的训练策略。在国标麻将中，我们通过划分不同的初始手牌上听数作为训练的初始状态，以满足课程学习“从易到难”的基本思想。此外我们还对构造新的马尔可夫过程来帮助控制训练进程从而能得到更好的训练效果这一想法做出了分析，希望对超参以及问题参数的调整能帮助我们更好的理解与解释国标麻将这一黑箱。

本文对已经应用于游戏 AI 领域的各种算法进行了综述，并重点总结了国标麻将 AI 的研究现状，分析了国标麻将 AI 强化学习训练面临的难点及其原因。提出了一种基于课程学习的国标麻将 AI 训练方法。用上听数对初始局面进行划分，并将不同的初始局面组织成了四级由易到难的课程。自行设计开发了一套国标麻将强化学习训练框架。该框架带有初始局面载入功能，可以方便地设置初始局面，进而通过不同的初始局面设定控制训练的过程。通过关联超参数和问题参数与训练获得 AI 的行为，达到了一定程度解释 AI 通过训练而成长的过程。

通过该功能进行了课程学习流程的部署，取得了接近现有最好的国标麻将强化学习训练方法的效果。经过课程学习的强化学习 AI 在 botzone 天梯排名前 10%，并大幅优于使用相同实验配置和相同强化学习框架进行训练，但未使用课程学习训练方式的直接强化学习 AI。在与 2020 年 IJCAI 国标麻将 AI 比赛的代表性 AI 进行对比的中展现的水平也在当年比赛的 4 到 5 名之间，并比起其他使用神经网络的 AI 实现了所需算力

规模的显著缩小和同等效果下的训练所需时间的缩短。本文还通过在训练流程中修改奖励函数 Reward、折扣因子 γ 和特征维度等参数完成了对 AI 行为差异的分析与总结，让我们能够更好的理解国标麻将这一黑箱，也对如何解释训练参数和所训 AI 行为关联性问题作出了有益的尝试。

6.2 本文研究展望

本文提出的基于国标麻将的课程学习训练方式实际上还有很多大的方面和小的细节可以进行拓展:

首先，“从易到难”的大框架下，关于“易”和“难”的定义除了上听数的递增，还可以有更精确的定义。事实上，两个同为二上听的局面，由于进张方式的差别，难度差异可能是很大的。比如“全不靠”二上听想要听牌，进张只能全靠自己抓，但“碰碰和”二上听则可以通过碰牌这一高效的手段进张。其实本人之前参与的工作中有能够帮助解决这一问题的方式。在本实验室一篇在投的关于 AI 可解释性工作的文章中，构建了一个基于搜索的模型，并提出了一个胜率计算器的概念。通过搜索当前局面下可能达到的番种，并通过胜率计算器将搜索到番种的每个缺失的牌张的概率作为其估计的胜率进行相乘。打出牌张的概率则是每个番种对其不需要的程度之和。通过这个工作，我们可以将每个局面进行搜索，并按照建议的胜率对而不是单纯的上听数进行区分；而此方法在一个局面下可以生成多个建议获胜局面，也大大扩充了训练的初始局面数据集。进一步说，我们完全也可以构筑一些特定高难度局面来对 AI 进行考验，以测试其能力上限。

其次，可以给待训练 AI 提出更精确的要求，以帮助其更好的获得我们期待的能力。当前我们采取的训练方案中，奖励函数主要聚焦在局末胜负，即和牌 +5，点炮 -3。而我们在对冠军 Bot 和牌位置玩家的手牌进行处理时，是可以获取其和的番种的。我们完全可以在课程学习训练时对 AI 和的番种作出限制，强制其和原本冠军 Bot 和的番种，以更好地获取冠军 Bot 的行牌思路。此外，我们还可以通过局末和的番种对课程进行特化分类，在想要 AI 更好的掌握某一番种的和牌能力时设置特定的番种课程。

此外，目前的课程学习训练流程还是一个纯手动的流程，后续可以添加自动转换课程的模块。

总的来说，只要在遵循“从易到难”原则下进行训练设置，你可以将任何你对国标麻将训练的想法作为课程进行输入。故课程学习具有很强的可拓展性。期待能在未来的训练中，在国标麻将强化学习的训练中再做突破。

参考文献

[1] Silver, D.; Huang, A.; Maddison, C.J.; Guez, A.; Sifre, L.; Van Den Driessche, G.; Schrittwieser, J.; Antonoglou, I.; Panneershelvam, V.; Lanctot, M.; et al. Mastering the game of Go with deep neural networks and tree search. Nature 2016, 529, 484 – 489.

[2] Silver, D.; Schrittwieser, J.; Simonyan, K.; Antonoglou, I.; Huang, A.; Guez, A.; Hubert, T.; Baker, L.; Lai, M.; Bolton, A.; et al. Mastering the game of go without human knowledge. Nature 2017, 550, 354 – 359.

[3] Silver, D.; Hubert, T.; Schrittwieser, J.; Antonoglou, I.; Lai, M.; Guez, A.; Lanctot, M.; Sifre, L.; Kumaran, D.; Graepel, T.; et al. A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play. Science 2018, 362, 1140 – 1144.

[4] Morav ˇ cík, M.; Schmid, M.; Burch, N.; Lisy, V.; Morrill, D.; Bard, N.; Davis, T.; Waugh, K.; Johanson, M.; Bowling, M. Deepstack: Expert-level artificial intelligence in heads-up no-limit poker. Science 2017, 356, 508 – 513.

[5] Brown, N.; Sandholm, T. Superhuman AI for heads-up no-limit poker: Libratus beats top professionals. Science 2018, 359, 418 – 424.

[6] Brown, N.; Sandholm, T. Superhuman AI for multiplayer poker. Science 2019, 365, 885 – 890

[7] Vinyals, O.; Babuschkin, I.; Chung, J.; Mathieu, M.; Jaderberg, M.; Czarnecki, W.M.; Dudzik, A.; Huang, A.; Georgiev, P.; Powell, R.; et al. Alphastar: Mastering the realtime strategy game starcraft ii. DeepMind Blog 2019, 2. Available online: https://www.deepmind.com/blog/alphastar-mastering-the-real-time-strategy-gamestarcraft-ii

[8] Berner, C.; Brockman, G.; Chan, B.; Cheung, V.; Debiak, P.; Dennison, C.; Farhi, D.; Fischer, Q.; Hashme, S.; Hesse, C.; et al. Dota 2 with large scale deep reinforcement learning. arXiv 2019, arXiv:1912.06680.

[9] Ye, D.; Chen, G.; Zhang, W.; Chen, S.; Yuan, B.; Liu, B.; Chen, J.; Liu, Z.; Qiu, F.; Yu, H.; et al. Towards playing full moba games with deep reinforcement learning. Adv. Neural Inf. Process. Syst. 2020, 33, 621 – 632.

[10] Yannakakis G N, Togelius J. Artificial intelligence and games[M]. New York: Springer, 2018.

[11] Copeland, B.J. The Modern History of Computing. Available online: https://plato.stanford.edu/entries/computing-history/

[12] Tesauro, G. Temporal difference learning and TD-Gammon. Commun. ACM 1995, 38, 58 – 68.

[13] Schaeffer, J.; Lake, R.; Lu, P.; Bryant, M. Chinook the world man-machine checkers champion. AI Mag. 1996, 17, 21 – 21

[14] Campbell, M.; Hoane, A.J., Jr.; Hsu, F.h. Deep blue. Artif. Intell. 2002, 134, 57 – 83.

[15] Bowling, M.; Burch, N.; Johanson, M.; Tammelin, O. Heads-up limit hold'em poker is solved. Science 2015, 347, 145–149.

[16] Zhou H, Zhou Y, Zhang H, et al. Botzone: A competitive and interactive platform for game AI education[C]//Proceedings of the ACM Turing 50th Celebration ConferenceChina. 2017: 1-5.

[17] Mahjong Soft. https://mahjongsoft.com

[18] Bengio, Y, et al. Curriculum learning. In ICML, 41 – 48, 2009

[19] Bengio, Y. Evolving culture versus local minima. In Growing Adaptive Machines, 109 – 138. Springer, 2014.

[20] Mahjong Competition Rules. (2019, December 27). Retrieved May 1, 2020, from https://en.wikipedia.org/wiki/Mahjong_Competition_Rules

[21] 国标麻将-教学-新手指南-国标麻将规则详解. (n.d.). Retrieved April 10, 2020, from http://mj.lianzhong.com/gbmj/home/teaching_new_rule1

[22] Michael Buro. The evolution of strong othello programs. Entertainment Computing, pages 81 – 88, 2003.

[23] Kai Li, Hang Xu, Meng Zhang, Enmin Zhao, Zhe Wu, Junliang Xing, and Kaiqi Huang. Openholdem: An open toolkit for large-scale imperfect-information game research. arXiv preprint arXiv:2012.06168, 2020.

[24] Michele Colledanchise and Petter Ogren. ¨ Behavior trees in robotics and AI: An introduction. CRC Press, 2018.

[25] Hart, P.; Nilsson, N.; Raphael, B. A formal basis for the heuristic determination of minimum cost paths. IEEE Trans. Syst. Sci. Cybern. 1968, 4, 100 – 107.

[26] Stockman, G. A minimax algorithm better than alpha-beta? Artif. Intell. 1979, 12, 179 – 196.

[27] Sutton, R.S.; Barto, A.G. Reinforcement Learning: An Introduction; MIT Press: Cambridge, MA, USA, 2018.

[28] Kocsis, L.; Szepesvári, C. Bandit based monte-carlo planning. In Proceedings of the European Conference on Machine Learning, Berlin, Germany, 18 – 22 September 2006; pp. 282 – 293.

[29] Ginsberg, M.L. GIB: Imperfect information in a computationally challenging game. J. Artif. Intell. Res. 2001, 14, 303 – 358.

[30] Bjarnason, R.; Fern, A.; Tadepalli, P. Lower bounding Klondike solitaire with MonteCarlo planning. In Proceedings of the Nineteenth International Conference on Automated Planning and Scheduling, Thessaloniki, Greece, 19 – 23 September 2009.

[31] Frank, I.; Basin, D. Search in games with incomplete information: A case study using bridge card play. Artif. Intell. 1998, 100, 87 – 123.

[32] Cowling, P.I.; Powley, E.J.; Whitehouse, D. Information set monte carlo tree search. IEEE Trans. Comput. Intell. Games 2012, 4, 120 – 143.

[33] Whitehouse, D.; Powley, E.J.; Cowling, P.I. Determinization and information set

Monte Carlo tree search for the card game Dou Di Zhu. In Proceedings of the 2011 IEEE Conference on Computational Intelligence and Games (CIG’ 11), Seoul, Korea, 31 August 2011 – 3 September 2011; pp. 87 – 94.

[34] Burch, N. Time and Space: Why Imperfect Information Games Are Hard. Available online: https://era.library.ualberta.ca/items/db44409f-b373-427d-be83-cace67d33c41

[35] Eiben, A.E.; Smith, J.E. Introduction to Evolutionary Computing; Springer: Berlin/Heidelberg, Germany, 2003; Volume 53.

[36] Rechenberg, I. Evolutionsstrategien. In Simulationsmethoden in der Medizin und Biologie; Springer: Berlin/Heidelberg, Germany, 1978; pp. 83 – 114.

[37] Dawkins, R.; Krebs, J.R. Arms races between and within species. Proc. R. Soc. Lond. Ser. B Biol. Sci. 1979, 205, 489 – 511.

[38] Angeline, P.; Pollack J. Competitive Environments Evolve Better Solutions for Complex Tasks. In Proceedings of the 5th International Conference on Genetic Algorithms, San Francisco, CA, USA, 1 June 1993; pp. 264 – 270.

[39] Reynolds, C.W. Competition, coevolution and the game of tag. In Proceedings of the Fourth International Workshop on the Synthesis and Simulation of Living Systems, Boston, Massachusetts, USA, 6 – 8. July 1994; pp. 59 – 69.

[40] Sims, K. Evolving 3D morphology and behavior by competition. Artif. Life 1994, 1, 353 – 372.

[41] Smith, G.; Avery, P.; Houmanfar, R.; Louis, S. Using co-evolved rts opponents to teach spatial tactics. In Proceedings of the 2010 IEEE Conference on Computational Intelligence and Games, Copenhagen, Denmark, 18 – 21 August 2010; pp. 146 – 153.

[42] Fernández-Ares, A.; García-Sánchez, P.; Mora, A.M.; Castillo, P.A.; Merelo, J. There can be only one: Evolving RTS bots via joust selection. In Proceedings of the European Conference on the Applications of Evolutionary Computation, Porto, Portugal, 30 March – 1 April 2016; pp. 541 – 557.

[43] García-Sánchez, P.; Tonda, A.; Fernández-Leiva, A.J.; Cotta, C. Optimizing hearthstone agents using an evolutionary algorithm. Knowl.-Based Syst. 2020, 188, 105032.

[44] Hornik, K.; Stinchcombe, M.; White, H. Multilayer feedforward networks are universal approximators. Neural Netw. 1989, 2, 359 – 366.

[45] Li, J.; Koyamada, S.; Ye, Q.; Liu, G.; Wang, C.; Yang, R.; Zhao, L.; Qin, T.; Liu, T.Y.; Hon, H.W. Suphx: Mastering mahjong with deep reinforcement learning. arXiv 2020, arXiv:2003.13590.

[46] Hinton, G.; Vinyals, O.; Dean, J. Distilling the knowledge in a neural network. arXiv 2015, arXiv:1503.02531.

[47] Mnih, V.; Kavukcuoglu, K.; Silver, D.; Graves, A.; Antonoglou, I.; Wierstra, D.; Riedmiller, M. Playing atari with deep reinforcement learning. arXiv 2013, arXiv:1312.5602.

[48] Williams, R.J. Simple statistical gradient-following algorithms for connectionist

reinforcement learning. Mach. Learn. 1992, 8, 229 – 256.

[49] Konda, V.; Tsitsiklis, J. Actor-critic algorithms. In Proceedings of the Advances in Neural Information Processing Systems 12 (NIPS 1999), Denver, CO, USA, 29 November – 4 December 1999; Volume 12.

[50] Lillicrap, T.P.; Hunt, J.J.; Pritzel, A.; Heess, N.; Erez, T.; Tassa, Y.; Silver, D.; Wierstra, D. Continuous control with deep reinforcement learning. arXiv 2015, arXiv:1509.02971.

[51] Mnih, V.; Badia, A.P.; Mirza, M.; Graves, A.; Lillicrap, T.; Harley, T.; Silver, D.; Kavukcuoglu, K. Asynchronous methods for deep reinforcement learning. In Proceedings of the International Conference on Machine Learning, PMLR, New York, NY, USA, 20 – 22 June 2016; pp. 1928 – 1937.

[52] Espeholt, L.; Soyer, H.; Munos, R.; Simonyan, K.; Mnih, V.; Ward, T.; Doron, Y.; Firoiu, V.; Harley, T.; Dunning, I.; et al. Impala: Scalable distributed deep-rl with importance weighted actor-learner architectures. In Proceedings of the International Conference on Machine Learning. PMLR, Stockholm, Sweden, 10 – 15 July 2018; pp. 1407 – 1416.

[53] Schulman, J.; Levine, S.; Abbeel, P.; Jordan, M.; Moritz, P. Trust region policy optimization. In Proceedings of the International Conference on Machine Learning, PMLR, Lille, France, 7 – 9 July 2015; pp. 1889 – 1897.

[54] Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; Klimov, O. Proximal policy optimization algorithms. arXiv 2017, arXiv:1707.06347.

[55] Osborne, M.J.; Rubinstein, A. A Course in Game Theory; MIT Press: Cambridge, MA, USA, 1994.

[56] Hart, S.; Mas-Colell, A. A simple adaptive procedure leading to correlated equilibrium. Econometrica 2000, 68, 1127 – 1150.

[57] Zinkevich, M.; Johanson, M.; Bowling, M.; Piccione, C. Regret minimization in games with incomplete information. Adv. Neural Inf. Process. Syst. 2007, 20, 1729 – 1736.

[58] Tammelin, O. Solving large imperfect information games using CFR+. arXiv 2014, arXiv:1407.5042.

[59] Brown, N.; Sandholm, T. Solving imperfect-information games via discounted regret minimization. In Proceedings of the Proceedings of the AAAI Conference on Artificial Intelligence, Honolulu, HI, USA, 27 January – 1 February 2019; Volume 33, pp. 1829 – 1836.

[60] Lanctot, M.; Waugh, K.; Zinkevich, M.; Bowling, M.H. Monte Carlo Sampling for Regret Minimization in Extensive Games. In Proceedings of the NIPS, Vancouver, BC, Canada, 6 – 11 December 2009; pp. 1078 – 1086.

[61] Schmid, M.; Burch, N.; Lanctot, M.; Moravcik, M.; Kadlec, R.; Bowling, M. Variance reduction in monte carlo counterfactual regret minimization (VR-MCCFR) for extensive form games using baselines. In Proceedings of the AAAI Conference on

Artificial Intelligence, Honolulu, HI, USA, 27 January – 1 February 2019; Volume 33, pp. 2157 – 2164.

[62] Waugh, K.; Schnizlein, D.; Bowling, M.H.; Szafron, D. Abstraction pathologies in extensive games. In Proceedings of the AAMAS, Budapest, Hungary, 10 – 15 May 2009; pp. 781 – 788.

[63] Waugh, K.; Morrill, D.; Bagnell, J.A.; Bowling, M. Solving games with functional regret estimation. In Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, Austin, TX, USA, 25 – 30 January 2015.

[64] Brown, N.; Lerer, A.; Gross, S.; Sandholm, T. Deep counterfactual regret minimization. In Proceedings of the International Conference on Machine Learning, PMLR, Long Beach, CA, USA, 9 – 15 June 2019; pp. 793 – 802.

[65] Li, H.; Hu, K.; Ge, Z.; Jiang, T.; Qi, Y.; Song, L. Double neural counterfactual regret minimization. arXiv 2018, arXiv:1812.10607.

[66] Steinberger, E. Single deep counterfactual regret minimization. arXiv 2019, arXiv:1901.07621.

[67] Steinberger, E.; Lerer, A.; Brown, N. DREAM: Deep regret minimization with advantage baselines and model-free learning. arXiv 2020, arXiv:2006.10410.

[68] Brown, G.W. Iterative solution of games by fictitious play. Act. Anal. Prod. Alloc. 1951, 13, 374 – 376.

[69] Heinrich, J.; Lanctot, M.; Silver, D. Fictitious self-play in extensive-form games. In Proceedings of the International Conference on Machine Learning, PMLR, Lille, France, 7 – 9 July 2015; pp. 805 – 813.

[70] Heinrich, J.; Silver, D. Deep reinforcement learning from self-play in imperfectinformation games. arXiv 2016, arXiv:1603.01121.

[71] McMahan, H.B.; Gordon, G.J.; Blum, A. Planning in the presence of cost functions controlled by an adversary. In Proceedings of the 20th International Conference on Machine Learning (ICML-03), Washington, DC, USA, 21 – 24 August 2003; pp. 536 – 543.

[72] Lanctot, M.; Zambaldi, V.; Gruslys, A.; Lazaridou, A.; Tuyls, K.; Pérolat, J.; Silver, D.; Graepel, T. A unified game-theoretic approach to multiagent reinforcement learning. Adv. Neural Inf. Process. Syst. 2017, 30, 4193 – 4206.

[73] Bansal, T.; Pachocki, J.; Sidor, S.; Sutskever, I.; Mordatch, I. Emergent complexity via multi-agent competition. arXiv 2017, arXiv:1710.03748.

[74] Jaderberg, M.; Dalibard, V.; Osindero, S.; Czarnecki, W.M.; Donahue, J.; Razavi, A.; Vinyals, O.; Green, T.; Dunning, I.; Simonyan, K.; et al. Population based training of neural networks. arXiv 2017, arXiv:1711.09846.

[75] Sunehag, P.; Lever, G.; Gruslys, A.; Czarnecki, W.M.; Zambaldi, V.; Jaderberg, M.; Lanctot, M.; Sonnerat, N.; Leibo, J.Z.; Tuyls, K.; et al. Value-decomposition networks for cooperative multi-agent learning. arXiv 2017, arXiv:1706.05296.

[76] Rashid, T.; Samvelyan, M.; Schroeder, C.; Farquhar, G.; Foerster, J.; Whiteson, S.

Qmix: Monotonic value function factorisation for deep multi-agent reinforcement learning. In Proceedings of the International Conference on Machine Learning, PMLR, Stockholm, Sweden, 10 – 15 July 2018; pp. 4295 – 4304.

[77] Lowe, R.; Wu, Y.I.; Tamar, A.; Harb, J.; Pieter Abbeel, O.; Mordatch, I. Multi-agent actor-critic for mixed cooperative-competitive environments. Adv. Neural Inf. Process. Syst. 2017, 30, 6382 – 6393.

[78] Foerster, J.; Farquhar, G.; Afouras, T.; Nardelli, N.; Whiteson, S. Counterfactual multi-agent policy gradients. In Proceedings of the AAAI Conference on Artificial Intelligence, New Orleans, LA, USA, 2 – 7 February 2018; Volume 32.

[79] Sato, H., Shirakawa, T., Hagihara, A., & Maeda, K. (2017). An analysis of play style of advanced mahjong players toward the implementation of strong AI player. International Journal of Parallel, Emergent and Distributed Systems, 32(2), 195-205.

[80] Mizukami, N., & Tsuruoka, Y. (2015, August). Building a computer mahjong player based on monte carlo simulation and opponent models. In 2015 IEEE Conference on Computational Intelligence and Games (CIG) (pp. 275-283). IEEE.

[81] Mizukami, N.; Nakahari, R.; Ura, A.; Miwa, M.; Tsuruoka, Y.;and Chikayama, T.; ―Realizing a four-player computer mahjong program by supervised learning with isolated multi-player aspects, ‖ Transactions of Information Processing Society of Japan, vol. 55, no. 11, pp. 1 – 11, 2014, (in Japanese).

[82] Gao, S, et al. "Supervised Learning of Imperfect Information Data in the Game of Mahjong via Deep Convolutional Neural Networks." Information Processing Society of Japan (2018).

[83] Tenhou: https://tenhou.net/. [Online; accessed 08-March- 2020].

[84] Gao, S., Okuya, F., Kawahara, Y., & Tsuruoka, Y. . Building a Computer Mahjong Player via Deep Convolutional Neural Networks. arXiv 2019, arXiv:1906.02146.

[85] Mori, S., Richardson, J., Ushiku, A., Sasada, T., Kameko, H., & Tsuruoka, Y. (2016, May). A Japanese chess commentary corpus. In Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16) (pp. 1415-1420).

[86] Fu, H.; Liu, W.; Wu, S.; Wang, Y.; Yang, T.; Li, K.; Xing, J.; Li, B.; Ma, B.; Fu, Q.; et al. Actor-Critic Policy Optimization in a Large-Scale Imperfect-Information Game. In Proceedings of the International Conference on Learning Representations, Virtual Event, 3 – 7 May 2021.

[87] Lu, Y.; Li, W. Mahjong AI Competition: Exploring AI Application in Complex Real-World Games, International Joint Conference on Artificial Intelligence (2024).

[88] Wang, X.; Chen, Y.; Zhu, W. A Survey on Curriculum Learning. arXiv 2021, arXiv:2010.13166

[89] Petr, S.; Radu, T.; Paolo, R.; Nicu, S. Curriculum Learning: A Survey. arXiv 2022, arXiv:2101.10382v3

[90] F. Doshi-Velez, B. Kim. Towards a rigorous science of interpretable machine learning. arXiv 2015, arXiv:1702.08608.

[91] Schulman, J, et al. High-dimensional continuous control using generalized advantage estimation. arXiv 2015, arXiv:1506.02438.

附录 A　在学期间获得的奖励

2021-2022 学年北京大学硕士专项学业奖学金

2022-2023 学年北京大学硕士专项学业奖学金

2022-2023 学年北京大学社会工作奖

致谢

行文至此，三年的北大时光也在一晃间近了尾声。这三年里，我终于初窥了学术的门径，也终于在身心上都习惯了北京，而分别也总在这最好的时候如期而至。我能够健康而顺利的走到尽头，离不开这段生命旅程中各位亲朋好友给予我的帮助和爱护。以下寥寥数语，难以言尽我的感激之情。

首先，我由衷的感谢我的导师，李文新教授。李老师三年前对我的认可，让我有机会开启在北大的这段令人难忘的旅程。李老师在科研上的态度是严谨但不拘泥的，您常常鼓励我们要勇于思考和探索，而您则作为我们最坚实的后盾，让我们都能够聚焦于自己最感兴趣的方向。完成毕业论文期间，由于我的拖沓和马虎，没少让您费心，但老师还是很细致的对我的文章提出了很多一针见血的意见。科研之余，李老师也是值得我们学习的人生榜样。李老师是一个真正智慧的人，您对事物的洞察与分析往往能令我们叹服，而您豁达的人生观也在潜移默化中改变着我。学业有尽头，但在之后的人生中，希望我还能继续向您学习。

感谢人工智能实验室的师兄弟们，因为有你们，我总能快乐的走进 1328 和 223。感谢鲁云龙师兄对我的科研工作的鼎力相助，在各种阴间时间（可能对我们来说还挺早）向你提出各种奇怪问题，总能得到你情绪稳定的秒回，这种安心感胜过苍天大树。感谢我实验室里最好的两个朋友：汪永毅同学和李凌峰同学。感谢汪永毅同学，我们常常在一起学习、健身和扯闲篇中见证凌晨四点的昌平，我从你身上真正见识到了北大人的优秀。感谢李凌峰同学，最怕麻烦的你给我工具包时写的使用说明甚至超过了你给自己代码写的注释，你大方而乐天的性格总能让我会心一笑。感谢李昂师兄对我们这些师弟方方面面的照顾和关心。感谢王星博同学，和你相处总是令人如沐春风，也祝你毕业快乐。感谢陈泊舟同学和刘涵宇同学，你们的到来让 223 更有了家的感觉。感谢洪星星师兄、林慎吾同学、王楚才同学和杨雄辉同学，与你们的相遇是我人生的一笔宝贵财富。祝大家都能科研进步，万事胜意。

此外，我还要感谢我的家人和朋友们，是你们一贯的爱与关怀，让我成为了一个更好的人。感谢我的爸爸妈妈，在我成长的道路上，你们一直信任我、爱护我、尊重我、关心我。你们对我的爱可能时而太过啰嗦，时而则过分盲目，但也永远会是我最宝贵的精神支柱。希望我有成长为一个令你们感到骄傲和欣赏的人。我爱你们。感谢我的弟弟郑启睿同学，我们虽然不一定是最亲密的兄弟，但我一直为你的点滴成长而感到欣喜。祝你接下来在北大的三年也能和我一样快乐而收获颇丰。感谢我的朋友们，不论是新的、老的，从近在咫尺的到相隔万里的，或许我不是一个最完美的朋友，但你们的存在真的有让我感受到温暖和力量。

感谢我在北大遇到的所有人，是你们和我一起完整了这段美好的时光。青山不改,绿水长流,来日方长,我们后会有期。