北京大学

硕士研究生学位论文

题目：IME：基于神经网络的游戏 AI 模仿及评价方法

姓　　名：	周昱杉
学　　号：	1801213727
院　　系：	信息科学技术学院
专　　业：	计算机软件与理论
研究方向：	人工智能
导师姓名：	李文新　教授

二〇二一 年 六 月

版权声明

摘要

游戏是人工智能算法的高效测试平台和重要应用环境。在游戏环境中，除了可以研究游戏智能体 AI 完成对抗性或者合作性任务的能力以外，还可以研究智能体 AI 模仿学习另一个智能体的能力。模仿学习通过对示范行为数据进行模仿，从而获得与被模仿者相似的策略模型，并在相同状态下做出尽量一致的决策。模仿学习广泛应用于显式对手建模、比赛人数补缺、玩家托管等场景。现有的模仿学习研究主要集中在模仿人类玩家的决策模式，缺少对游戏智能体 AI 行为的模仿研究。本文工作聚焦于模仿单个智能体 AI 的决策行为的方法和评价模型，并试图回答游戏智能体 AI 是否和人类玩家一样具有个性化特征的问题。

本文的主要工作包含四个部分：

1. 提出了一种基于神经网络的模仿单个游戏智能体 AI 的决策方式的方法；

2. 定义了一个评价智能体相似度的模型。该评价模型从两个不同层次对模仿相像程度进行量化评估，一是相同状态下的单步决策一致性，另一个是相同状态下的序贯决策结果一致性；

3. 模仿和被模仿智能体 AI 的相似程度越高，说明被模仿 AI 越容易被模仿。基于这一假设，提出了一种根据被模仿的难易程度对被模仿 AI 进行聚类，并结合代码静态分析，赋予聚类结果类别以语义信息的方法。该语义信息可以理解为被模仿 AI 的个性化特征。例如，短视且贪心，或者决策模型较僵化等等；

4. 在黑白棋和斗地主两款游戏上，分别搭建了神经网络，完成了智能体 AI 模仿学习；并应用本文提出的智能体相似度评价模型进行了模仿学习效果评价和聚类分析实验。实验中，还对比了不同数据集大小、不同神经网络结构对相似度的影响。实验结果表明精心设计和训练的神经网络可以较好地模仿游戏智能体 AI 的决策方式。聚类分析表明游戏智能体 AI 也可以像人类玩家一样具有个性化特征。

该模仿方法和评估标准的提出，使得研究者对模仿学习中影响模仿相似度的因素及其影响程度更加了解，使得模仿过程能够迭代进行，为玩家建模挖掘玩家类别特性、陪玩 AI 提升玩家游戏体验等等研究奠定了基础。

关键词：神经网络，模仿，游戏 AI 个性，相似度评估

IME: A Neural Network-based Method of Imitating Game AI and Evaluation

Zhou, Yushan (Computer Software and Theory)
Directed by Prof. Li, Wenxin

ABSTRACT

Game is an efficient testing platform and an important application environment for artificial intelligence algorithms. In the game environment, in addition to the ability of the game agent AI to complete competitive or cooperative tasks, the ability of the AI to imitate and learn from another agent can also be studied. Imitation learning obtains a strategy model similar to that of the modeled by imitating the data of the demonstration behavior, and makes as consistent a decision as possible under the same state. Imitation learning is widely used in explicit opponent modeling, matchmaking, player hosting and other scenarios. Existing research on imitation learning mainly focuses on the imitation of human players' decision-making patterns, and there is a lack of imitation research on the behavior of game agents AI. This paper focuses on methods and evaluation models that mimic the decision-making behavior of individual AI, and attempts to answer the question of whether game AI has the same personalized characteristics as human players.

The main work of this paper includes four parts:

1. Proposed a method based on neural network to imitate the decision-making mode of a single game agent AI.

2. Defines a model to evaluate the similarity of agents. This evaluation model evaluates the similarity degree quantitatively from two different levels, one is the consistency of one-step decision under the same state, and the other is the consistency of sequential decision results under the same state.

3. The higher the similarity between the imitated and the imitated agent AI, the easier the imitated AI is to be imitated. Based on this assumption, we proposed a method to cluster the simulated AI according to the degree of difficulty to be imitated, and combine with the static analysis of code to give semantic information to the clustering result category. This semantic information can be seen as the personalization of the AI being mimicked. For example, they are short-sighted and greedy, or their decision models are rigid, etc.

4. In the two games of Reversi and Doudizhu, we built the neural network respectively to complete the imitation learning of intelligent agent AI; The simulation learning effect

evaluation and cluster analysis experiments are carried out by using the agent similarity evaluation model proposed in this paper. In the experiment, the influence of different data set size and different neural network structure on similarity is also compared. The experimental results show that the well-designed and trained neural network can better imitate the decision-making mode of game agent AI. Cluster analysis shows that game AI can be as personalized as human players.

The proposed imitation method and evaluation criteria enable researchers to have a better understanding of the factors affecting imitation similarity in imitation learning and the degree of their influence. The imitation process can be carried out iteratively, which lays a foundation for the research of player modeling, mining the characteristics of player categories, and improving the game experience of players by playing with AI.

KEY WORDS: Neural Network, Imitation, Game AI Character, Similarity Evaluation

目录

图目录

表目录

第一章　引言

1.1　游戏在人工智能研究中的重要地位

人工智能的兴起，起源于上世纪五十年代，在美国举办的一场学术研讨会。会议上首次提出的人工智能（AI，Artificial Intelligence）术语以及断言，成为划时代的象征。这句断言同时也成为后世众多人工智能研究员的信念基础——“人类应当能精确描述智能特性的方方面面，因此，机器可以进行模仿从而拥有这些智能”[1]。这之后，人工智能领域面临寒冬又再次焕发生机，到如今，人工智能正处于以深度学习为代表技术的第三次浪潮。2021 年新华社公布的“十四五”规划及纲要中有多达五十多处表述中含有“智能”、“智慧”，人工智能在社会中发挥的作用逐日变大地位渐升。

在人工智能算法的研究中，游戏被认为是人工智能高效、最合适的试验田之一[2]。游戏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。同时，游戏 AI 作为评估人工智能进展的标准之一，其发展历程也反映了 AI 智能水平的进步。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 算法被应用于其他领域，展现出了很好的延展性和泛化性。游戏 AI 的一些行为表现也很好的解释了在社会中人类的行为模式，也有研究将其用于预测未来社会的资源分配、发展走向。交互式游戏在这一层面上，确实是人工智能研究中的“堪称杀手级的应用”[3]。

1.2　游戏 AI 的发展历程

游戏 AI 研究与游戏性质息息相关。从大类看，研究使用的游戏主要有两种，一种是“电子游戏”，一种是“桌面游戏”。前者因电子设备普及进入人们视野，比后者更晚出现，更注重玩家视觉体验，要求玩家反应迅速、操作敏捷，出现后一直受到广泛关注。后者通常为回合制游戏，各种棋牌游戏作为经典桌面游戏，拥有广大受众。随着游戏 AI 研究的在更多的应用场景下找到新的研究意义，技术持续更新，游戏 AI 从以玩家身份解决游戏难题，到玩家建模，再到能创建游戏关卡，游戏 AI 的含义越来越丰富。下文将重点介绍桌面游戏中，玩家 AI 的发展史，在下一小节中则介绍玩家建模——游戏 AI 模仿人类玩家的研究现状。

游戏玩家 AI 最早出现在上世纪五十年代，而受到广泛关注却是在四十多年后，国际象棋 AI“深蓝”与当年世界冠军的对局。1997 年 IBM “深蓝”对战人类高手加里·卡斯帕罗夫[4]，2016 年围棋 AI“阿尔法狗”（AlphaGo）对战韩国职业九段李世石[5]，2017 年围棋 AI “阿尔法狗·大师”（AlphaGo Master）对战当时世界第一柯洁，这三次世界级水平的对局中，人类顶尖高手均败阵而归。围棋被认为是已知棋类中最困难的游戏，具有多达10^{172}之多的棋盘局面变化，AlphaGo 在围棋上的成功是 AI 破解双人完全信息游戏的一个里程碑事件。人类不得不承认，AI 在这类问题上已经超越了人类。

图 1.1　AlphaGo 与李世石（右）对弈①

游戏玩家 AI 的下一个冲锋点，是多人非完全信息博弈游戏。2017 年 1 月，图马斯·桑德霍尔姆团队研发的德州扑克 AI，Libratus，在美国与四位人类顶级德扑高手连续 20 天总计 12 万局对战中，赢得了游戏货币系统的 176 万美元[6]。2019 年，在 WAIC（世界人工智能大会）上，微软亚洲研究院宣布，由其团队自主研发的基于麻将平台“天凤”的日本麻将 AI，Suphx，成为平台上所有 AI 中第一个荣升十段的智能体，AI 的实力高于平台上的顶尖人类玩家水平[7]。同年在机器学习顶会之一的 ICLR 2019 的盲选阶段一篇关于中国传统游戏斗地主的论文[8]引起人们关注。

① 图片来源：https://www.sohu.com/a/72529832_377096

图 1.2　Suphx（南风位）在天凤平台上与其他玩家对抗[7]

目前游戏玩家 AI 研究大多追求一个“最强”甚至是“超人”AI，在于人类高手对局的过程，研究者发现，AI 的某些决策并不像人类，仿佛它们形成了自己的风格，而人类需要一段时间的学习理解才能接受这种决策风格。这使得游戏 AI 另一个子领域研究逐渐受到人们关注和重视，那就是像人类一样的游戏 AI。游戏 AI 模仿人类玩家，目前比较明朗的应用场景包括游戏运营中的新游戏冷启动、玩家掉线 AI 代打、陪玩 AI 等等。下一小节中本论文将介绍这方面的研究现状。

1.3　游戏 AI 模仿人类玩家的研究现状

在游戏 AI 领域里，玩家 AI 研究占据主体，这个方向对研究人员提出的要求，是获得一个高水平的 AI，甚至于超过人类。随着游戏行业整体技术的提升，人们对 AI 在游戏中发挥的作用产生了更多的期待，模仿智能体 AI 研究也从新问题新要求中被赋予了新的研究意义。

游戏 AI 模仿早期研究主要集中在如何使游戏玩家 AI 表现得像人类一样，或者在游戏中，以 NPC 的身份（NPC，非玩家参与，游戏中的角色）担任人类玩家的对手或

同伴。研究主要将群体人类玩家作为被模仿的主体，通过学习人类玩家的对局数据，使得游戏 AI 的行为决策足够像人类，整体逻辑符合人类认知。与“超人”AI 不同，人类玩家目标具有多样性，并非单一追求最优解，各有各的偏好。

模仿人类玩家 AI 的研究中，根据模仿的层次不同，分为低层次的动作模仿、高层次的战略模仿，不同目标设定使得最后的评价标准也不同。实际上对于这个问题，游戏 AI 是否足够像人，十分接近在游戏环境中进行图灵测试。低层次的动作模仿更关注游戏轨迹的相像，比如平台游戏“超级马里奥兄弟”AI 比赛[9]的“图灵测试”赛道，将 AI 与人类玩家的行进轨迹进行对比，在决赛中邀请专业玩家对 AI 的游玩视频进行打分评估。战略模仿则带有“风格”意味，更强调整体或注重游戏结果的相像，van Hoorn 等人[10]在赛车游戏中构建了具有和人类一样的流畅驾驶风格，同时表现良好的 AI。

事实上，对于游戏 AI 是否可能具有自己的风格，或许和人类相似，或许是难以被当下理解的特殊风格这个问题，不仅仅只出现在模仿人类玩家 AI 的讨论中。本论文在追溯社会对于“超人”AI 游戏风格的时候，看到了有不少公开的评论传达了肯定的意见。对于 2016 年横空出世的 AlphaGo，不少人类棋手甚至是顶尖高手评价其具有自己的棋风，“下棋中该弃、该退出的地方，AlphaGo 会像一个真正的人一样弃掉、退出”。曾获得欧洲冠军的樊麾则说，他无法想象这是一个 AI，因为行棋模式很像人类棋手。AlphaGo 在前期训练中对顶尖高手的对局数据进行模仿学习，之后才进行自我博弈提升水平。而之后“从零开始”，自己与自己进行对局博弈的 AlphaZero[11]，在对局中或多或少也体现出了一些特定的棋风。AlphaZero 在国际象棋游戏中，展现出的独特走棋思维，被专家反复仔细琢磨。不像现代国际象棋中，人类基本走棋思维将“子力”看得非常重要，AlphaZero 并没有苛刻追求这一点，相反，它有时甚至会为了更长远的收益，在早期牺牲子力。国际象棋大师玛修·撒德勒指出，AlphaZero 在整个过程中都很明确地表现出了这一点，“风格非常明显”。可以说 AlphaZero 在训练过程中形成了自己的走棋思维，风格独特而鲜明。

目前还没有超过人类顶尖水平的 AI，是否具有自己的风格，对于这个问题，本论文也找到了公开的评论意见。国际象棋大师卡斯帕罗夫——曾在与“深蓝”对战中落败的当年的世界冠军——在对 AlphaZero 的评论中提到，“计算机程序一般会体现出代码编写者的偏好与侧重”，游戏 AI 本身应当蕴含着编码者的信念，与传统算法结合之后，将在与其他智能体的对战中，体现出自己的风格个性。

本文工作通过让一个 AI 去模仿另一个 AI 的方法，来探究被模仿的 AI 是否具有个性化特征。

1.4　本文提出的问题 - AI 模仿另一个 AI

本论文进行了关于游戏 AI 模仿智能体的调研，发现当前游戏 AI 模仿的研究主要集中于模仿人类群体，常用的评价指标有拟人可信度（believability）[12]。模仿智能体个体行为个性的研究较为缺乏，要进行这方面研究，建立统一的模仿个体是否相像的评价标准是非常有必要的。

本文欲探究，在构建一个新游戏 AI 模仿另一个游戏 AI 时，要如何进行个体行为的模仿，怎么评估模仿的相像程度。在模仿的过程中，是否能了解被模仿的游戏 AI 的风格，而这种风格个性，是否能通过模仿准确地复刻。

为了回答以上问题，需要完成以下工作：

- 确定模仿游戏 AI 的方法，如何在不同的游戏上应用。
- 给出模仿相像的具体定义，并给出量化评估相似度的指标计算公式。
- 基于相似度评估，探寻游戏 AI 的风格特征。

本文将选用 Botzone 平台[13]上的三款回合制游戏，贪吃蛇、黑白棋和斗地主作为实验用游戏[①]。贪吃蛇游戏是一个双人同时决策游戏，本文将重点探究模仿方法的细节设置。黑白棋是双人完全信息确定性游戏，斗地主是三人非完全信息随机性游戏，本文在这两个游戏上将重点讨论模仿方法及其评估模型的应用。在以上三个游戏中，本文将根据模仿的实际效果，对被模仿的 AI 风格特征进行讨论分析，用以验证本文提出的方法的有效性。

在下面的叙述中，本文也将 AI 具体实现的程序实例称为 Bot。

1.5　本文主要工作与创新点

本文提出游戏 AI 模仿另一个 AI 个性这一问题，描述了解决问题的意义价值，并提供了实际的应用场景。

对于以上问题，本文提出了基于神经网络的模仿游戏 AI 个性的方法及其评价模型，

[①] Botzone 平台：www.botzone.org.cn

借助通用性学习框架——神经网络——进行模仿学习。神经网络能在仅提供局面特征的情形下，自动发现并学习 AI 的行为模式，具备良好的模仿表达能力。本文创新性地引入两种相似度评估标准，一种是在相同局面状态下单步决策动作一致，一种是相同局面状态完成游戏的序贯决策的胜负结果一致。

本文基于 Botzone 平台上的贪吃蛇、黑白棋、斗地主这三款游戏，分别做了模仿实验，在黑白棋、斗地主实验中使用评估模型对模仿 AI 进行评价，具体描述了工作流程，并分析了实验结果。

本文工作是对研究生期间模仿学习研究工作的承接与延伸，作者曾发表关于基于模仿方法的 AI 聚类的论文，具体请参见附录。

1.6 本章小结及后续章节安排

本章叙述了游戏在 AI 研究中作为测试环境的重要地位，简述了游戏 AI 的发展史，介绍了游戏 AI 作为玩家和模仿人类这两个方面的相关研究。本文进行了游戏 AI 模仿现状更为详细的调研，提出了本文欲探究的问题：游戏 AI 模仿另一个 AI，使用什么样的模仿方法，怎么评估模仿的相像程度。

本文将在第二章介绍游戏 AI 模仿方法及其评价模型的现有工作基础，在第三章中提出本文重点 IME（Imitation and Evaluation）模型，并在第四、第五这两章里分别介绍在黑白棋、斗地主上应用 IME 模型的实验结果及分析。最后，第六章总结本文内容并展望模仿 AI 的未来。

第二章　游戏 AI 模仿及评价方法的研究进展

目前游戏 AI 模仿研究以模仿人类玩家为主。这一章将介绍模仿人类玩家的游戏 AI 模仿及评价方法的研究进展。

2.1　游戏 AI 模仿方法

游戏 AI 模仿指对给定的决策数据进行学习，决策数据由这样一组状态和动作构成：用s_i，a_i表示第i回合的状态和动作，下一回合的状态由当前状态及采取的动作决定，也即$s_{i+1} = \text{execute}(s_i, a_i)$。因此，给定的决策数据序列$\{s_1, a_1, s_2, a_2, \ldots, s_{end}, a_{end}\}$。

在基于监督学习的模仿方法中，往往将其中的状态动作对提取出来，将状态s_i作为特征输入，将动作a_i作为标签输出，进行分类（动作空间是离散的的）或回归（动作空间是连续的）。基于监督学习的模仿方法强调测试其泛化性，也即在没有见过的状态上能预测到准确的动作，或者在一段动作决策之后能达到相同的效果。

基于强化学习的模仿方法往往使用逆强化学习手段。强化学习方法大多应用在做决策的智能体需要与环境进行积极探索的问题，这类决策问题一般被建模成以五元组$<S, A, P, R, \pi>$为代表的马尔科夫问题，其中S表示包括环境及智能体在环境中的信息的状态集，A表示智能体可以采取的合理且可能的动作集，P表示在某状态下采取某动作转移到特定状态的概率，R表示在某状态下采取某动作能得到的奖励回报，π为策略，指在某状态下有多少概率采取某动作。正向强化学习是给定回报函数，学习最佳策略。逆向强化学习则相反，一般用在回报函数无法清晰定义的问题中，在给定策略的前提下，学习合理的回报函数。在模仿学习的场景里，策略为状态动作序列，通过比较智能体策略与给定的策略数据的距离学习回报函数，使得智能体的策略函数与目标策略函数相似，从而达到模仿的目的。

接下来的 2.1.1　和 2.1.2　两小节将给出这两种模仿方法的实证研究进展，2.1.3 小节介绍本文作者研究生在读早期模仿贪吃蛇游戏 AI 个性的工作。

2.1.1　监督学习模仿

监督学习中常用的算法包括贝叶斯、决策树、线性回归、神经网络等等。将其用于游戏 AI 模仿任务中，需要结合游戏 AI 的工作流程进行。使用搜索的游戏 AI 往往由搜索算法框架、局面估值函数组成，监督学习算法可以用于拟合局面估值函数，也可以直

接代替搜索算法，接受当前局面直接输出动作。

Gao 等人在六贯棋游戏[①]中，构建 AI 对人类对局数据进行学习，设计神经网络接受状态直接输出动作[14]。他们在这个工作中将神经网络与蒙特卡洛树搜索结合，将神经网络用于搜索的子结点选择部分，在新结点的探索和能带来更多胜利的结点的利用之外，加入神经网络类似于给结点赋予人类经验偏好。结合了训练网络的蒙特卡洛树搜索比不结合的搜索 Bot 胜率更高。

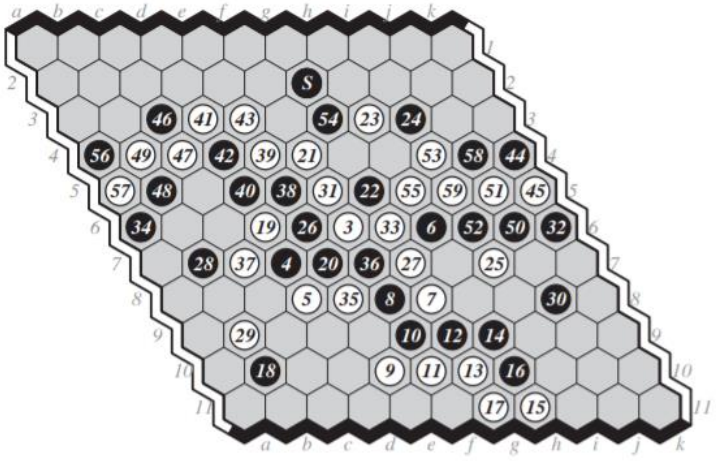

图 2.1　六贯棋棋盘[14]

Runarsson 等人基于黑白棋游戏，使用偏好学习方法来学习专家策略并近似得到评估函数。偏好学习中，AI 根据给定局面获取动作偏序关系，在决策时选择排名最高的动作，以此近似得到游戏局面评估函数[15]。与一般局面估值函数计算不同，他们不根据从当前状态开始，随机采样动作直到结局的胜率来为状态进行估值，而是通过类似神经网络的方式进行估值。但与神经网络输出所有动作概率不同，偏好学习侧重于获取偏序关系，对于动作的概率值多少并不关心。

Devlin 等人基于“黑桃”（Spades）纸牌游戏，希望在一个水平够好的 Bot 的基础上，使其足够像人[16]。他们注意到非人类玩家的策略与人类玩家有较大的风格差别，而单纯的模仿人类玩家的 Bot 水平较差，故他们在 MCTS 框架基础上，修改了叶结点回传公式，使用人类玩家的对局数据学习回传公式中的参数，从而使得 Bot 更像人类。

[①] 六贯棋，双人零和游戏，六边形格子的棋盘，白子和黑子最快达到对面边缘的获胜。

图 2.2　黑桃纸牌游戏①

Bindewald 等人的单体模仿研究是少有的模仿单个游戏玩家的工作[17]。文章基于导航游戏 Space Navigator，提出一种聚类和局部加权回归的方法，来建模和模仿单个玩家。算法先在所有人类玩家数据上，学出一个通用的玩家集群模型，以此为基础，收集个人玩家的游戏数据来更新个人模型。

Renman 等人基于寻路 3D 电子游戏构建像人类一样的 AI，使用带有 KD 树的最近邻算法，将状态映射到动作[18]。不同于本文将模仿 AI 问题视作分类问题，他们将其视作聚类问题，并采用了聚类算法确定 AI 动作策略。

以上工作的共性是模仿人类玩家，模仿学习的数据来自人类玩家对局数据。本文的模仿对象是游戏 AI，模仿人类玩家的方法也可以运用到模仿游戏 AI 上，收集 AI 的对局数据用以学习训练，也可以达到模仿学习的目的。最大的不同在于，AI 能在短期内产生大量对局数据，所以本文能在训练时使用足够多的 AI 对局对单个 AI 进行模仿，而人类玩家对局数据产生的效率远低于 AI，在上面的相关工作也可以看到，大多数都以模仿群体人类玩家为目标，模仿单个玩家决策能使用的对局数据受到玩家投入时间、精力的限制。

2.1.2　强化学习模仿

Spronck 等人使用一种叫动态脚本（Dynamic Scripting）的强化学习技术，使得 AI 能在线学习玩家的策略，并根据玩家的具体水平，来调整自己的表现[19]。

① 图片来源：https://www.trickstercards.com/home/spades/

Tang 等人在双人格斗视频游戏中，提出一种新颖的对手建模方法，收集对手的历史数据，使用基于交叉熵监督学习和基于 Q 学习、策略梯度的强化学习方法，对其进行优化[20]。借助这样的对手模型，预测对手的可能动作，并制定针对这些动作的有力回击。作者团队研发的 Bot 击败了 FTGAIC 比赛在 2018 年的所有参赛者，并在 2019 年的比赛中获得了第二名。

2.1.3 本文作者早期工作

本文作者研究生在读早期基于双人贪吃蛇游戏，使用监督学习方法，构建神经网络模仿 AI 个性，并根据神经网络训练效果及对被模仿 AI 代码静态分析对 AI 进行聚类分析。

与传统贪吃蛇不同，本实验中的贪吃蛇是双人同时决策回合制游戏，每回合双方玩家同时做决策，让己方的蛇在不被围困的前提下，尽量迫使对方的蛇无路可走。玩家在$N * M$的有障碍物的网格中操纵自己的蛇，玩家控制蛇头朝向往东南西北四个方向行进，每回合行进一格。双方蛇的蛇身长度不因吃豆子增加，每次最多加 1。双方的蛇在对局初始分别位于地图的左上角与右下角。当蛇头超出了网格地图、与障碍物或者双方蛇身重叠，或玩家做出了非法操作时，会被判输，对局结束。

Botzone 平台的游戏中，使用长 11 宽 10 的地图，地图上将随机产生位置中心对称的障碍物，如图 2.3 所示，环绕的是墙壁，灰格是障碍物，在网格地图中，两条连接的蛇分别为玩家所控制，蛇头用两个小白点特别标出。其余网格都为空格。

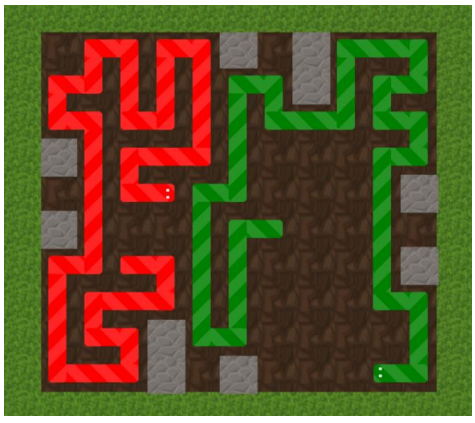

图 2.3 Botzone 平台贪吃蛇游戏截图

表 2.1　贪吃蛇及常见棋类游戏复杂度

游戏	状态空间复杂度	博弈树复杂度
国际跳棋	10^{21}	10^{31}
黑白棋	10^{28}	10^{58}
国际象棋	10^{46}	10^{123}
中国象棋	10^{48}	10^{150}
贪吃蛇	10^{54}	10^{122}
六角棋	10^{57}	10^{98}
将棋	10^{71}	10^{226}
围棋（19x19）	10^{172}	10^{360}

贪吃蛇游戏与其他游戏的复杂度列在表 2.1 中。

实验选取了 Botzone 天梯排行榜上 22 个 Bot，这 22 个 Bot 都使用了搜索算法框架。在搜索算法中，Bot 需要在搜索结点展开时，确定子结点的搜索顺序，并对局面进行评估。了解 Bot 的搜索顺序和估值函数中的预定义权重是一种分析 Bot 偏好特征的启发式方法。由于搜索时限，Bot 无法搜索全部的可能分支，从而会在搜索顺序上有强烈的偏好。此外，估值函数中的预定义权重也会影响最终决策。

这 22 个 Bot 使用的算法框架有蒙特卡洛树搜索（MCTS）、基于纳什均衡的蒙特卡洛方法（NE）、蒙特卡洛方法（MC）、Alpha-Beta 方法、深度优先搜索（DFS）以及人类专家经验（ES）。此外，图 2.4 展示了 Bot 使用的搜索算法及搜索技巧，第一列为 Bot 在实验中的序号。第二列为 Bot 在天梯上的排名。第三列为 Bot 使用的算法框架，蒙特卡洛树搜索（MCTS, Monte Carlo Tree Search）、基于纳什均衡的蒙特卡洛树搜索方法（NE, Nash Equilibrium）、快速走子的蒙特卡洛方法（MC, Monte Carlo）、Alpha-Beta 方法、深度优先搜索（DFS）以及人类专家经验（ES, Expert System）。第四列为搜索深度（SD, Search Depth），不限制搜索深度（inf），迭代加深（ID, Iterative Deepening），没有使用（N/A），有具体数值的即为搜索深度。第五列为是否有剪枝，前 14 个 Bot 和第 19 个 Bot 都使用了剪枝（Yes）。第六列为是否有分支限时（Yes 表示使用了），平台 Bot 运行有总的时间限制，个别 Bot 在搜索时会对某些分支进行搜索限时，从而能有更多机会搜索其他分支。

BID	Rank	Algo	SD	Prune	Timelimit
1	1	MCTS	inf	Yes	Yes
2	2	NE	ID	Yes	No
3	3	MCTS	inf	Yes	Yes
4	6	Alpha-Beta	ID	Yes	No
5	9	NE	ID	Yes	No
6	13	MCTS	ID	Yes	No
7	15	Alpha-Beta	ID	Yes	No
8	17	Alpha-Beta	ID	Yes	No
9	19	Alpha-Beta	11	Yes	No
10	20	MC	inf	Yes	Yes
11	31	DFS	8	Yes	No
12	32	DFS	8	Yes	No
13	34	Alpha-Beta	10	Yes	No
14	49	DFS	inf	Yes	No
15	54	MC	inf	No	Yes
16	56	MC	inf	No	Yes
17	77	ES	N/A	No	No
18	87	ES	1	No	No
19	89	DFS	inf	Yes	No
20	91	MC	1	No	No
21	149	ES	1	No	No
22	156	ES	1	No	No

图 2.4　贪吃蛇 - Bot 搜索算法及搜索参数

对 Bot 代码进行静态分析时，总结了三点内容。首先，较大的搜索深度能确保 Bot 能获取到更精准的状态局面估值，从而表现出更具有“远见”的特性，反映在天梯排名上，后几名的 Bot 的搜索深度只有 1，靠前的 Bot 都具有较大的搜索深度。其次，进行剪枝的 Bot 可以加快搜索速度，有助于更高效地进行分支探索。第三，不限制搜索深度且不设置搜索分支时限的 Bot，虽然表现出“远见”的特性，但这也使得它们将花费大部分时间搜索局部分支，从而导致更少的其他分支探索，第 14 和第 19 号 Bot 是典型例子。

蒙特卡洛树搜索和蒙特卡洛都会尽可能地探索更多的分支，从算法原理看，蒙特卡洛树搜索同时也会提高对那些回报较多的分支的探索率，故与蒙特卡洛相比，蒙特卡洛树搜索的效率更高。纳什均衡的蒙特卡洛树搜索方法则改写了叶结点结果回传，更新父结点值的函数。贪吃蛇要求对局双方同时决策，假如在某状态下，任意一个 Bot 在对手的动作策略确定时，选择的动作是最优解，那么称这个状态存在纳什均衡。解纳什均衡的 Bot 考虑了对手可能的最优策略，输出己方的最优动作。在展开搜索结点的时候，Alpha-Beta 能赋予子结点优先级，更“聪明”地向下搜索，在有回报价值的分支上花费更多的探索时间，而深度优先搜索的 Bot 则没有这样做，故而 Alpha-Beta 的 Bot 表现出水平更高，在对局中的决策更优、更有远见。使用专家系统的 Bot 则表现不佳，这与黑白棋、五子棋 Bot 中不一致，这或许与黑白棋、五子棋有更多更丰富的开局定式有关，人类经验已经能在这两个棋类游戏上玩得很好，而贪吃蛇只是经典游戏改编，人

类经验不足以在短时间内追上使用搜索算法的水平。此外，在上一小节中对比了贪吃蛇与其他棋类游戏的游戏复杂度，可以看到贪吃蛇的搜索空间大小比中国象棋还要大，博弈树复杂度与国际象棋相近，这可能也能佐证，短期内无法将人类经验具象化成一棵决策树，从而无法在贪吃蛇游戏上获得很好的结果。

模仿实验之前，从 Botzone 平台上下载了公开的贪吃蛇对局数据，并从这些对局中筛选出这 22 个 Bot 相关的对局数据。根据对局的历史行为数组恢复局面状态，为 22 个 Bot 各自生成“状态-动作”对数据集。处理完的数据中，状态是形状为 10x11x4 的三维矩阵，动作为有 4 个元素的一维向量，分别表示 4 个方向。根据局面的合法动作可以计算随机选择的准确率，相当于使用随机决策的 AI，在以上数据集中的状态下输出动作，与被模仿的 Bot 选择相同的概率为多少。通过计算，这 22 个 Bot 的随机选择准确率在 40.0%到 47.3%之间。

图 2.5 呈现了将贪吃蛇的局面状态转换成输入形状为 10x11x4 矩阵，第三维分别包含蛇头、蛇身、敌方蛇和障碍物信息，从局面状态可以获取合法动作，将这些信息作为输入传进神经网络，输出位于图片左方的蛇的唯一可行动作是向下。

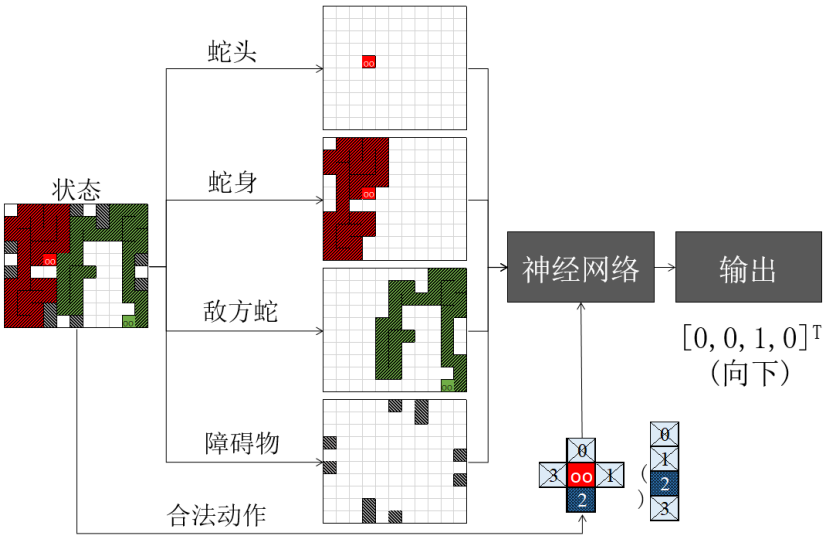

图 2.5　贪吃蛇 - 状态输入示意图

模仿实验中，作者设计了 5 种神经网络，这些网络的输入和输出格式与训练数据格式一致。这 5 种神经网络的参数量量级保持在 60K 左右，按网络层类型分为全连接网络和卷积网络，具体的网络结构见表 2.2。

表 2.2 贪吃蛇 - 网络结构及超参

网络名	参数量	层类型	网络结构
MLP-1	56964	全连接	两层全连接，中间层有 128 个隐藏单元
MLP-2	57044	全连接	三层全连接，中间层分别有 80 和 256 个隐藏单元
CONV-1	58052	卷积	卷积核使用 2x2 和 3x3，最大步长为 2
CONV-2	61724	卷积	卷积核使用 2x2 和 3x3，最大步长为 2
CONV-3	59452	卷积	卷积核使用 2x2 和 3x3，最大步长为 2

网络训练使用 Adam 优化器，批处理大小为 32，迭代最大次数限制在 300 次。在所有网络中都应用批归一化（BN，Batch Normalization），学习率开始设置为 0.02，随着训练进行递减。

训练采用 5 折交叉验证法，计算并记录验证集上的平均准确率。对于每一种网络结构，作者都将其应用在所有 Bot 上分别进行训练，一共训练 22x5=110 个“网络结构-Bot”组合。

考虑到数据集大小会对模型训练结果产生较大影响，实验随机选取了第 1、2、4、6 号 Bot，在不同的数据集大小上进行实验。数据集以对局为单位，在对局数分别为 100,400,700,100,1300,1600 时进行训练，并根据平均验证准确率绘制图 2.6。图中横轴是数据集的对局数，纵轴是平均验证准确率。可以看到，当数据集达到 1000 场对局后，对局数再增加，平均验证准确率也不再提升。由此可见，1000 场对局是比较合适的数据集大小。数据集过小时，验证集准确率较小；数据集继续增大，验证集准确率也不继续提升。故在实际实验中，所有 Bot 都使用 1000 场对局的数据集。

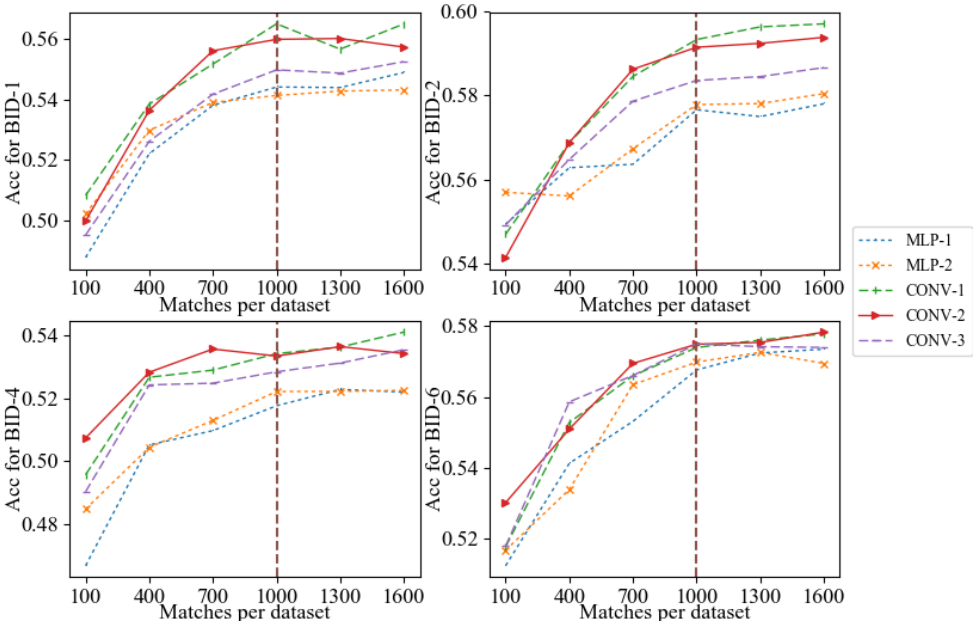

图 2.6　贪吃蛇 - 不同模型在不同大小的数据集的平均验证准确率[21]

实验在 GPU 集群上进行，进行一次模仿 22 个 Bot 的训练总共需要 80 到 100 小时。实验将随机选择准确率作为基准（Baseline），与训练后得到的所有 Bot 的平均验证准确率一起绘制到图 2.7 中。图中横轴为 Bot 的序号，纵轴为 Bot 的平均验证准确率。大致走向为排名越靠前的 Bot 平均验证准确率越低，模仿难度越高。

将这 22 个 Bot 根据平均验证准确率进行聚类，按照数量进行二分，有序号为 1、2、3、4、5、6、7、10、13、15、16 的 Bot 被聚到第一类，其余的被聚到第二类。可以看到，算法框架为蒙特卡洛树搜索和纳什均衡的 Bot 被分到第一类中，深度优先搜索和专家经验的 Bot 被分到第二类中，Alpha-Beta 和蒙特卡洛方法的 Bot 在两类中都有。这和本章第二小节中的分析是一致的，蒙特卡洛树搜索和纳什均衡的 Bot 具有框架上的相似性，只是在更新结点值的方式上有所不同，都表现出比较高的水平，在对局中是“远视”、“有大局观”的。深度优先搜索及专家经验的 Bot 在搜索效率及分支探索上都弱于其他的 Bot，表现出“短视”、“只关注局部”的特性。Alpha-Beta 和使用蒙特卡洛方法的 Bot 因为设置的搜索深度、是否剪枝以及时限相差较大，在两类中都各占一席之地。

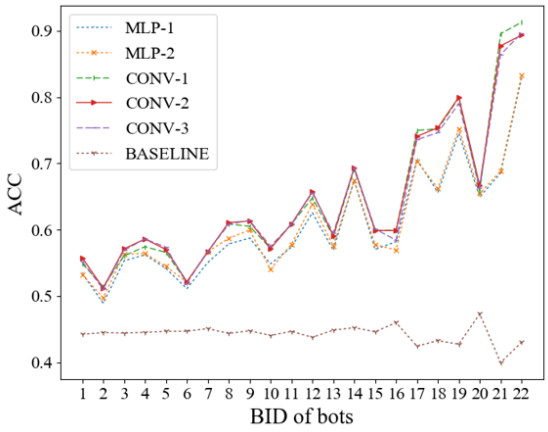

图 2.7　贪吃蛇模仿 AI 平均验证准确率[21]

2.2　人类主观评价模型

人类主观评价指人类评审对游戏 AI 模仿效果进行评价的方法，这个过程往往以黑箱测试进行，也即评审员并不知道对方是人类还是 AI，需要通过分析对方的行为数据来确定。Hernndez-Orallo 将 AI 黑箱测试分成了三大类：人类判定；基准测试；对抗比赛[22]。其中人类判定分为交互型以及非交互型，交互型判定也分为与人类沟通、与人类对抗等等，非交互型则可能将对局数据以视频的方式展现给人类评测员。人类主观评价模型中，非交互型评价指人类观察员对模仿 AI 的对局数据进行分析打分，交互型评价指在游戏中与模仿 AI 或对抗或合作，并通过判定对方是人类还是 AI 做出不同的决策。下面将从这两方面介绍当前人类主观评价模型的实证研究进展。

2.2.1　人类观察员对游戏 AI 历史数据进行推演分析

人类评审员通过观察 AI 的行动来定性或定量评估 AI 是否像人类。

Togelius 等人在 2012 年“超级马里奥”平台游戏 AI 比赛的“图灵测试”赛道评测中使用了定性评估的方法[12]。这个赛道的目标是提交一个表现像人类玩家的 AI。在“超级马里奥”这个游戏中，由于只有一个玩家角色，所以无法令人类玩家与 AI 进行交互。比赛评测的方法是让人类评审员观看两个视频，分别由 AI 和人类完成，评审员需要回答哪个视频中的玩家可能是人类玩家。

图 2.8　AI 玩“无限超级马里奥”游戏截图[①]

Bernard 使用可信度指数量化评估多智能体游戏 AI[23]。可信度指数分 5 档，指数越高代表 AI 表现越像人类玩家。评估过程由多个人类观察员独立进行。

另一个使用人类主观印象打分来进行评估的研究中，Renman 基于寻路 3D 电子游戏，构建像人类一样的 AI。文章通过用户研究调查，在以 AI 第一视角观看其行进过程之后，打分评价 AI 是否像人类[18]。文章讨论了有哪些因素会决定角色是否像人类，比如有时不出于任何目的地往地上看，视频内容向地面倾斜，这一行为会让观察员认为这个 AI 很“自然”、“像人类”，从而更可能给出高分。

2.2.2　人类玩家与游戏 AI 进行对抗

Hingston 介绍了 2008 年 IEEE CIG 研讨会主办的 2K BotPrice 比赛，比赛中使用第一人称射击电子游戏，应用了人类玩家与游戏 AI 进行对抗交互的评估方法[24]。所有参赛的 AI 将分别与多个人类玩家进入竞技场。人类玩家配有特殊的武器，需要确定其他玩家是人类还是 AI，来决定使用这一武器还是不使用。在这个测试流程中，人类玩家需要能够准确地判别出 AI，并用这个武器进行攻击，从而获得足够高的分数。而 AI 的任务则是尽可能地不被分辨出来，行为足够像人。人类玩家的分数越高，则对 AI 拟人效果的评价越低。

2.2.3　现有人类主观评价模型的局限性

对游戏 AI 模仿人类玩家的效果进行人类主观评价，其实是水到渠成、自然而然的一种方法。很多人工智能的任务都以达到人类智能或超过人类智能为终极目标，将其

[①] AI 玩超级马里奥路线示意的视频截图，视频来源：https://www.youtube.com/watch?v=DlkMs4ZHHr8

与人类智能作对比是必要的过程。然而正如模型命名的那样，人类主观评价模型需要大量观察人员，且需要保证观察人员的多样性。而人本身具有的主观能动性，使每个人对模仿 AI 是否足够像人，有不同的标准及意见。人类主观评价模型缺少统一明确的量化标准。

此外，对于发生在不同时期的模仿人类玩家研究，往往会使用不同观察人员，将这两个研究的人类主观评价结果放在一起，很难说是进行了严谨的对比。现有人类主观评价模型的应用场景里，通常因为缺少控制变量使得对比不同模仿方法的效果变得非常困难，使用人类观察员打分进行评估的方法很难进行 AI 迭代优化。

2.3 数据分析评价模型

数据分析评价模型指通过对人类玩家和 AI 玩家在游戏中的行为数据进行特征提取，并进行比较的方法。

2.3.1 针对游戏 AI 与游戏环境交互的数据分析

Khaustov 等人基于足球游戏开发了基于规则的 AI，提取 AI 在游戏中的传球长度和传球时间两个特征与人类团队进行了对比，结果显示基于规则的 AI 玩家与人类玩家相比在传球行为上表现出显著差异[25]。

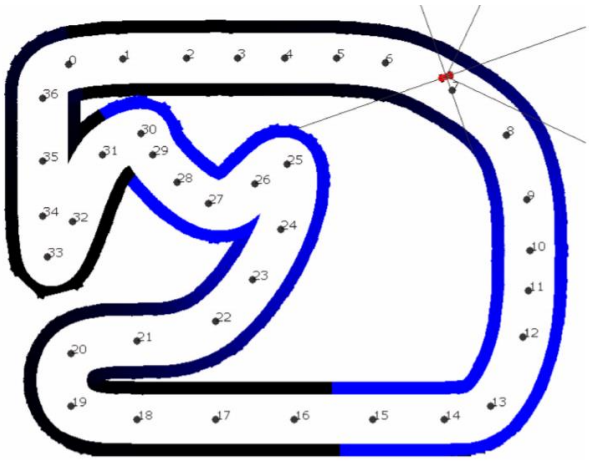

图 2.9 赛车车道设点及 AI 过点速度示意图[26]

Ortega 等人使用一个评估框架衡量游戏风格的相似性，该框架将人类玩家的游戏轨迹与 AI 玩家的标点轨迹进行比较[9]。类似的，使用游戏轨迹评估相似度的还有 Togelius 发表在 IEEE CIG 2007 的工作，他们为赛车 AI 定义了适应度函数，采用对车道设点并计算在固定时间内通过的点数，以及过点时的速度等指标（见图 2.9），将 AI

与人类玩家的特征数据做对比[26]。

Khalifa 等人提出了视频游戏中，考虑到人类玩家操作并不能像 AI 一样精细、转变迅速，可以通过三个量化指标对比人类玩家和 AI 玩家的风格，第一个是动作长度，指执行同一个动作的持续时间，第二个是空动作长度，指不执行任何动作的持续时间，最后一个是换动作频率，指动作变换次数除以帧数[27]。

2.3.2　现有数据分析评价模型的局限性

数据分析评价模型相较于人类主观评价模型更为客观，但同时也将之后的研究代入了这样的窘境：当换一个游戏进行实验，需要重新制定评价指标，之前指标的设计经验难以推广到其他游戏上，比如 Khalifa 等人的工作[27]实际上是在利用 AI 的强大计算力与人类思考时间的差异，对 AI 玩家和人类玩家做区分，而在回合制游戏中，特别是无视决策时间的游戏里，是无法用这样的指标进行区分的。此外，应用数据分析评价模型评分较高的 AI，后续缺少人类评测，无法获知这些拟人 AI 是否真的像人一样决策。

2.4　本章小结

本章介绍了游戏 AI 模仿及评价方法，由于目前游戏模仿对象主要为人类玩家，故本章主要介绍的是拟人 AI 的模仿及评价。调研中，作者发现模仿 AI 以及 AI 个性化风格的研究很少，下一章将提出 IME 模型，探究 AI 个性化风格模仿及评价方法，并在第四、第五两章分别基于黑白棋、斗地主游戏，应用 IME 模型模仿游戏 AI 个性，评价模仿效果，通过聚类方法分析 AI 的个性化特征。

第三章　IME（Imitation and Evaluation）：基于神经网络的模仿 AI 个性的方法及其评价模型

本章提出了 IME（Imitation and Evaluation），一种基于神经网络的模仿 AI 个性的方法及其评价模型。IME 采取监督学习方法，使用神经网络构造模仿 AI（下也称模仿 Bot）。由于本文选用的游戏的动作都是离散的，所以可以将模仿 AI 的任务视作分类问题。通过被模仿 AI 的对局数据生成一系列“特征-动作对”数据，使用监督学习的方法，得到一个神经网络模型及其参数——即模仿 AI。该模仿 AI 在遇到一个新的状态时，会给出类似于被模仿 AI 在该局面下的动作选择。

本文中将局面状态表示为数据特征，Bot 动作表示为标签。后文中将一条训练数据称为一个“状态动作对”。使用监督学习方法搭建基于神经网络的模仿 AI，主要包括训练数据预处理和神经网络设计两个部分。后文将在 3.1　小节中详细介绍如何提供规范化的训练数据。随后，在 3.2　小节中介绍如何计算模仿 AI 和被模仿 AI 的相似度。

3.1　生成模仿 AI 的核心算法

3.1.1　训练数据预处理

在 Botzone 平台上，游戏环境与 Bot 之间的交互基于 JSON 格式，一场对局的参与者包含裁判程序和参与游戏的 Bot。

对局记录分为两大模块，其中一个模块记录对局的初始数据，另一个模块记录 Bot 的历史行为输出和裁判程序输出，后者在对局记录中称为 Log 数组，在此数组中，Bot 的输出与裁判程序输出交错出现，Bot 输出在数组中的顺序也记录了其在对局中做决策的顺序。裁判程序的作用是确认玩家决策行为的合理性、确定下一回合做决策的 Bot 的输入以及给出对局的最终结果。以黑白棋为例，记黑白棋的两个玩家为P_0，P_1，裁判程序为J，历史行为数组的输出顺序为：

$$J(\text{包含开局信息}) - P_0 - J - P_1 - J - P_0 - \cdots - J\ (\text{包含对局结果分数})$$

根据以上信息，根据一场双人游戏对局生成目标玩家的所有“状态-动作”对数据集的伪码列在图 3.1 中。

```
FUNCTION 对局记录生成状态动作对序列(初始数据 initdata, 历史行为数组 Logs, 目标 Bot, 裁判 Judge)
    LET state = init_game(initdata)          ;根据初始数据初始化对局状态
    LET TestStates = []                      ;记录目标玩家的状态-动作对的数组
    FOR log in Logs
        IF log is Judge's output             ;如果是裁判程序输出，跳过
            CONTINUE
        IF log is Bot's output               ;如果是目标玩家输出
            LET mask = cal_mask(state)       ;计算合理动作
            ;将状态-动作及合理动作追加到数据集中
            append(TestStates, [state, log.action], mask)
        state = place(log.action)            ;执行此元素的动作，更新状态
    RETURN TestStates
```

图 3.1　对局记录生成状态动作对序列算法

模仿任务中的数据集以对局形式存储，每个对局包含一组“状态-动作”对，这些对呈现出连续性与一致性，连续性指后一个对的状态是由前一个对的状态在执行了动作之后获得的，一致性指每一个对具有相同的形状结构。

为了学习到 Bot 个性，作者对平台对局分别整理收集，每个 Bot 有一个对局数据集。下一小节中设计的神经网络将分别用这些数据集进行训练，而非在所有数据的合集上进行训练。

3.1.2　神经网络设计

模仿 AI 的神经网络设计包括输入输出、超参设置两个部分。

输入是将训练数据中的状态数据（State）喂给网络进行训练的入口，需要保证二者的形状一致，用纯数值形式表示，一般为多维二值矩阵，同时输入部分还包括当前状态下的合理动作掩膜（Mask），加速后续输出决策时，将不合理动作剔除。输出需要能唯一确定动作，一般用维度等于游戏中所有合理动作数量的一维矩阵表示，当动作相对复杂时，需要对动作进行编码，使得动作标签与实际动作一一对应。

以井字棋游戏举例，将二值化棋盘状态及动作示意图列在图 3.2 中。训练数据形状为 3x3x2，3x3 表示棋盘的宽度和长度，2 表示参与游戏的不同玩家特征，这里每一

个 3x3 都是一个玩家的落子，一共有两个玩家。图 3.2 的棋盘状态显示下一个做决策的是执黑子的玩家，合理动作（Mask）表示执黑玩家的可行落子点。将其展平得到一维向量，与网络输出前一层相同形状的张量按位相乘，使用 Softmax 算子，使输出向量所有元素加和为 1，每个元素可以表示对应动作的选择概率。如果不计算 Mask，等同于让网络学会游戏规则，这点在长期的实证经历来看是非常困难的。为了保证模仿的效果，本文根据具体游戏规则，提前计算合理动作位并喂给网络。

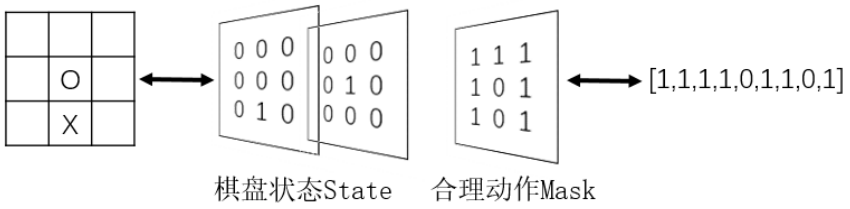

图 3.2　井字棋状态及动作二值化示意图

超参分为与网络结构相关和与模型训练相关的两部分。

网络结构部分包括网络中间层层数及类型，类型上主要使用卷积层和全连接层，下文将含有卷积层的网络称为卷积网络，只有全连接层的网络称为全连接网络。特别地，在模仿 AI 任务中不使用池化层，池化层会抽取局部最大值，使得状态维度变小，对于本身信息足够凝练的游戏状态来说，池化层会使得部分信息丢失。层与层之间的激活函数使用线性整流函数（ReLU）。

模型训练部分，使用交叉熵损失函数，Adam 优化替代传统随机梯度下降优化算法（SGD），批处理训练数据，设置早停防止过拟合，使用标准正态初始化方法。参数初始化中，不能全部初始化为 0，根据网络反向传播算法原理，这会导致在计算梯度更新参数时，同一层的不同节点进行对称式更新，不具有区分度，从而使训练无法提升。

3.2　基于相似度计算的评价模型

评价模型中的相似度指行为决策上的相似及序列决策最终结果上的相似，将 Bot 视作黑箱，只能获取 Bot 在某状态下的动作输出，这种将 AI 视作观察对象、架构内容不可知的假设更贴合现实。下面两个小节将分别介绍游戏 AI 的相似度定义及计算的工作流程，值得一提的是，基于相似度计算的评价模型不仅可以计算模仿 Bot 与被模仿对象之间的相似程度，也可以计算任意两个 Bot 之间的相似程度，这为之后的 AI 聚类提

供了强有力的工具。在此模型中，假设 Bot 中不含随机成分，从而简化计算流程。

3.2.1 游戏 AI 的相似度定义

本小节给出衡量游戏 AI 相似度的两个标准，从两个层次描述相似性。相同状态下的单步动作相似度衡量的是两个 Bot 低层次具体的行为一致度，只考虑在单状态下决策是否一样。假如需要利用对局中状态变换的连续性，单步动作相似度无法衡量这一点，因为只要在对局的某一个结点策略发生分歧，之后的状态都将不一样，从而无法比较。解决残局的胜负相似度则认为，在某些结点做出不一样的策略也无可厚非，从决策价值角度看，不同的决策可能价值相同，比如都能获胜，那么用于计算相似度的两个 Bot 在对战水平这一高层次、更抽象的层面来说，可以被认为是相似的。

在残局胜负相似度讨论中，将特定结点或特定回合开始，一直到对局结束，不同 Bot 的胜负结果不同的局面，成为关键局面。关键局面主要包含这两种的情况：

1. 不同 Bot 在关键局面这一回合的决策不同从而导致胜负结果不同；
2. 不同 Bot 在关键局面之后的决策不同从而导致胜负结果不同。

关键局面一定存在决策分歧，这种分歧是可能带来最终结果的变化。假如从某个回合开始，不管做什么决策，最后都会胜利或都会失败，那这样的回合就不是关键局面。

在残局胜负相似度中收集这样的关键局面，并在具体测试中让模仿 Bot 完成残局，统计最后的胜负情况。单步动作相似度的测试局面可以是任意局面，也可以是上述关键局面。

下面的小节将详细介绍两种标准的定义及计算公式，并叙述用于测试的状态数据集收集流程与测试过程。

3.2.1.1 相同状态下的单步动作相似度

相同状态下的单步动作相似度指，给定一组状态集合，获取 AI 在所有状态下的决策输出，计算两个 Bot 在同一状态下输出了相同动作的数量，这个数量与所有状态数的比，即为相同状态下的单步动作相似度，下简称为单步动作相似度。

计算两个 Bot，Bot_0和Bot_1的相似度，给定测试用的状态集合$\{\mathrm{S}_1, \mathrm{S}_2, \dots, \mathrm{S}_\mathrm{N}\}$，需要对比在这些状态下，$\mathrm{Bot}_0$和$\mathrm{Bot}_1$的决策输出。记在状态$\mathrm{S}_\mathrm{i}$下的动作分别为$\mathrm{A}_{\mathrm{i}0}, \mathrm{A}_{\mathrm{i}1}$，则相同状态下的单步动作相似度的计算公式为：

$$S_1 = \frac{1}{N}\sum_{i=1}^{N} \mathrm{IsSame}(A_{i0}, A_{i1}),\ \mathrm{IsSame}(A_0, A_1) = \begin{cases} 1, A_0 = A_1 \\ 0, A_0 \neq A_1 \end{cases} \qquad (3-1)$$

单步动作相似度计算的是 AI 在状态上的策略相似程度，而不是在整个对局中的相似程度，衡量的是低层次、具体的相似，割裂了对局的连续性。但当单步动作相似度高到一定程度时，在整个对局中的决策也将更多地趋于一致。

3.2.1.2　解决残局的胜负相似度

残局胜负相似度是指，给定一组残局集合，令 Bot 与指定对手完成残局，计算两个 Bot 在同一残局中胜负情况相同的数量，这个数量与所有残局数的比，即为残局胜负相似度。

计算Bot_0和Bot_1的相似度，给定一组残局集合$\{(S_1, O_1), (S_2, O_2), \dots, (S_N, O_N)\}$，元素$(S_i, O_i)$中，$S_i$表示残局状态，$O_i$表示指定对手，对比$Bot_0$和$Bot_1$在与指定对手对战的结果。记在残局$S_i$下，与对手$O_i$对战的结果分别为$W_{i0}, W_{i1}$，则解决残局的胜负相似度的计算公式为：

$$S_2 = \frac{1}{N}\sum_{i=1}^{N} \mathrm{IsSame}(W_{i0}, W_{i1}), \mathrm{IsSame}(W_0, W_1) = \begin{cases} 1, W_0 = W_1 \\ 0, W_0 \neq W_1 \end{cases} \qquad (3-2)$$

残局胜负相似度相比单步动作相似度，对单个状态的策略一致要求低，但对整体的策略要求更高，衡量的是整体的对敌水平、策略压制方面的一致性。在某个残局中，是否都赢过某个对手，是否都输给了某些对手，这种一致性是残局胜负相似度的基础。

单步动作相似度增大并不一定会使得残局胜负相似度的提升，较高的残局胜负相似度也不一定有很高的单步动作相似度。残局胜负相似度希望能解决，单步动作相似度只能衡量片面的相似度，无法衡量整体逻辑以及水平是否相像的问题。

3.2.2　评价模型的工作流程

在上一节的相似度定义中，用到了计算两种相似度的状态集合和残局集合。本小节将叙述如何收集这些测试数据，以及具体计算相似度的算法。

在评价模型工作流程中，状态集合和残局集合这两个测试数据集的收集并非必要，可以在评价流程中随机生成。但是从数据的可用性、区分性及后续对比新 Bot 相似度这三个角度来说，在评价流程中随机生成测试数据的方式存在一定的缺陷。可用性指的是在某些状态下，没有合法动作可以选择，对局或直接结束，或没有合法动作的一方

跳过回合，这无法获取 Bot 的行为输出，从而无法进行比较。区分性指的是在当前状态下只有一种合法动作，或者在当前残局下不管使用什么样的策略都不能获胜，甚至有时只有一条可行的动作序列，那么对于“遵守游戏规则”的 Bot 来说，势必会选择相同的行为策略，解决残局都是相同的结果，从而无法区分出 Bot 的不同。

后续对比新 Bot 相似度时，需要将之前的旧 Bot 与新 Bot 放在一起重新比较，这对旧 Bot 来说，是对过去历史记录的浪费及重复实验。

准备测试数据可以为实验提供更多的选择，模型工作流程开始前，可以对准备好的测试数据进行筛选，以满足可用性及区分性。当记录了旧 Bot 在未知状态下的动作以及解决残局的对局数据，整理形成状态数据集及残局数据集，再与后续新加的 Bot 进行相似度计算时，只需要让新 Bot 在以上数据集中进行测试，此时再将结果与旧 Bot 进行比较，即可计算出二者的相似度。实验过程满足轻量级、增量式要求。

3.2.2.1 状态集数据和残局集数据的采集

上一节介绍了游戏 AI 的相似度定义及计算公式，本小节的测试数据采集流程从相似度定义的两个方面分别叙述。将用于对比的目标 Bot 程序统一记为目标智能体Bot_0，在收集测试数据过程中需要借助其他智能体对数据进行筛选，这些用于参考的智能体记为$\{\mathrm{Bot}_i\}_{i=1}^{n}$。

状态数据集的数据格式与训练数据相同，都是状态动作对。数据采集的流程主要为采集目标智能体在不同状态下的动作输出。数据来源有多种，一种是来自除了目标智能体之外，其他智能体的对局，这种收集数据的方式将模仿任务视作传统监督学习任务，与目标智能体相关的对局数据都作为训练集与验证集数据投入使用，测试集要求与训练集不重叠。将这样的对局按照回合拆分成状态集，记录目标智能体在这些状态下的动作。这些状态之间可能不具有连续性，因为目标智能体并没有完成整个对局，只是给出在每一个局面状态下的动作，真正的动作执行者是其他智能体。当至少有一个参考智能体在相同状态下输出的动作，与目标智能体的动作输出不相同时，表明这个状态是能够区分开目标智能体和参考智能体的。

第二种数据来源是目标智能体的对局数据。目标智能体的对局数据更可能是模仿 Bot 在训练过程中见过的状态，同一个 Bot 在决策过程中更可能遇到相同的状态。

第三种数据来源是残局胜负相似度计算收集的关键局面。

不管是哪种数据来源，状态集数据收集具有统一的模式，都需要确定对局回合，目

标 Bot 以及参考 Bot。状态集数据收集伪码列在图 3.3 中，并在下一小节中介绍关键局面的收集。

```
FUNCTION 状态集数据收集(智能体对局 matches, 目标Bot_0)
    LET TestStates = []                          ;记录目标玩家的状态-动作对的数组
    FOR match in matches
        LET Logs = match['log']
        FOR i = 0 to length(Logs)-1
            LET state = recover(Logs[0:i])       ;恢复不同回合下的局面状态
            LET mask = cal_mask(state)           ;计算合理动作
            LET action = get_bot_action(state,mask,Bot_0)   ;获取目标Bot_0合法动作
            IF action ≠ Logs[i].action           ;如果目标Bot_0与其他智能体动作不一样
                append(TestStates, [state, action], mask)
    RETURN TestStates
```

图 3.3　状态集数据收集算法

在状态集数据收集任务中，假设目标智能体不具有随机成分，否则，随机决策的目标智能体在相同状态下的输出动作可能有不同种，无法确定选择哪一种作为目标智能体的动作决策。参考用的其他智能体在这一任务中，在不同的状态数据来源里对随机性的要求不同。

残局数据集的数据格式中，每一个数据包含残局信息和对手信息。残局数据收集任务的流程是，对于一个已有的对局，从不同的回合开始，面对相同的对手，目标智能体和参考智能体分别完成残局，假如参考智能体中有一个与目标智能体的结果不同，那么这个残局将作为“关键局面”加入残局数据集。关键局面的直观理解是目标智能体与其他智能体在某个局面状态下做的不同选择，导致最后的胜负结果不一样，目标智能体可能因为走了一步好棋获胜，也可能因为做了错误的选择导致失败。假设目标智能体和参考智能体都没有随机成分，含有随机决策的智能体在与相同对手的对局可能会有不一样的结果，无法决定使用哪一个结果作为目标智能体的残局胜负情况。同时，解决残局时，随机智能体的个性信息更难捕捉，水平相似程度需要更多的重复实验重复计算才能稳定，故在这一任务中假设智能体都不具有随机成分。残局的数据来源也可以根据是否为目标智能体对局的残局区分，本文只给出使用目标智能体对局残局作

为数据来源的筛选方法。筛选获取关键局面的残局伪码列在图 3.4 中。

```
FUNCTION 残局集数据收集：筛选关键局面(目标智能体对局 matches,目标Bot_0,其他{Bot_i}_{i=1}^n)
    LET TestMidgames = []
    FOR match in matches
        LET Logs = match['log']
        LET botiswinner = check_if_win(Logs, Bot_0.id)      ; Bot 是否获胜，存为 bool 值
    LET rounds = []                                         ; 存储发生变化的回合数
        FOR i = 0 to length(Logs)-1
            IF Logs [i] is not object's output   ;如果不是目标智能体输出，跳过
                CONTINUE
            LET partlog = Logs [0:i]                        ;截取从开始到目前为止之前的 log
            LET forkmatches = get_fork_match_log(partlog, {Bot_i}_{i=1}^n)
            ;让其他智能体完成当前残局，此时残局状态由 partlog 恢复得到
            FOR forkmatch in forkmatches
                LET forklogs = forkmatch['log']
                LET forkbotid = forkmatch['botid']
                LET w = check_if_win(forklogs, forkbotid)
                IF botiswinner ≠ w     ;其他智能体结果和目标智能体结果不一样
                    append(rounds, i)
                    BREAK
        append(TestMidgames, (match, rounddict))
    RETURN TestMidgames                                     ;返回对局中的关键局面
```

图 3.4　残局集数据收集算法

在残局数据收集任务中，假设目标智能体和参考智能体都不具有随机成分。解决残局时，随机决策的智能体个性信息更难捕捉，需要使其在相同局面下重复决策，对其动作输出以及终局输赢进行计数分析，足够多次才能获知其随机成分占多少、对终局结果影响如何，给评价模型的效率带来很大的挑战。

3.2.2.2　相似度计算的算法

上一小节说明了相同状态下的单步状态动作相似度计算所需要的状态集的收集方法，以及解决残局的胜负相似度计算所需要的残局集的收集方法。本小节在这些数据集的基础上，按照在 3.2.1　小节中给出的两种相似度计算公式，分别给出计算具体算法流程。将相同状态下的单步动作相似度计算算法伪码列在图 3.5 中，将解决残局的胜负相似度计算伪码列在图 3.6 中。

FUNCTION 相同状态下的单步动作相似度计算算法(状态集 TestStates, 目标Bot_0, 其他$\{\mathrm{Bot}_i\}_{i=1}^{n}$)

　　LET SimiList = []

　　FOR Bot_i in $\{\mathrm{Bot}_i\}_{i=1}^{n}$

　　　　LET SameCnt = 0

　　　　FOR ([state, action], mask) in TestStates

　　　　　　LET action_i = **get_bot_action**(state,mask,Bot_i) ;*获取Bot_i的合法动作输出*

　　　　　　SameCnt += 1 **IF** action_i ≠ action **ELSE** 0

　　　　LET Similarity = SameCnt / **length**(TestStates)

　　　　append(SimiList, Similarity)

　　RETURN $\{\mathrm{Bot}_i\}_{i=1}^{n}$, SimiList

图 3.5　相同状态下的单步动作相似度计算算法

```
FUNCTION 解决残局的胜负相似度计算算法(残局集 TestMidgames,目标Bot_0,其他{Bot_i}_{i=1}^n)
    LET SimiList = []
    LET TestCnt = size(TestMidgames)
    FOR Bot_i in {Bot_i}_{i=1}^n
        LET SameCnt = 0
        FOR (match, rounds) in TestMidgames
            LET Logs = match['log']
            LET botiswinner = check_if_win(Logs, Bot_0.id)      ; Bot 是否获胜，存为 bool 值
            FOR round in rounds
                LET partlog = Logs[0:i]
                LET forkmatches = get_fork_match_log(partlog, Bot_i)
                ;让Bot_i完成当前残局，此时残局状态由 partlog 恢复得到
                FOR forkmatch in forkmatches
                    LET forklogs = forkmatch['log']
                    LET w = check_if_win(forklogs,Bot_i)
                IF botiswinner ≠ w     ;Bot_i对局结果和目标智能体结果不一样
                    CONTINUE
                SameCnt += 1
        LET Similarity = SameCnt / TestCnt
        append(SimiList, Similarity)
    RETURN {Bot_i}_{i=1}^n, SimiList
```

图 3.6　解决残局的胜负相似度计算算法

3.3　小结

本章介绍了 IME 模型中基于神经网络的模仿 AI 个性的方法及评价模型。

生成模仿 AI 的关键步骤将与平台相关的对局数据规范化为格式统一的矩阵数据，处理成“状态-动作”对作为神经网络的训练数据。神经网络结构设计思路与一般的深度学习任务中的一致。

基于相似度计算的评价模型根据相似度定义中关于状态和残局定义，采集筛选整理

状态集和残局集数据。根据相似度的计算公式，本文给出了评价模型两种相似度计算的具体算法流程。

值得一提的是，本文将这一评价模型用于评估模仿 AI 与被模仿 AI 之间的相似度，但实际上这种基于相似度计算的评价模型也适用于评估任意 AI 之间的相似度。以计算得到的相似度作为 AI 之间的距离定义，可以使用这一评价模型来给一些 AI 进行聚类，一是探索 AI 之间的共性，二是更能挖掘 AI 的个性。

之后的章节将在黑白棋和斗地主游戏上分别应用 IME 模型，分析并对比模型的应用效果。

第四章　IME 在黑白棋游戏中的应用与分析

4.1　黑白棋游戏规则及性质分析

黑白棋（Reversi）是一个双人回合制棋盘游戏，棋盘一般采用 8x8 大小，共 64 格正方格。黑白棋的胜负根据终局双方棋子个数决定，个数较多的一方获胜，若个数一样则平局。下图为对局初始状态，黑子和白子各有两颗摆在棋盘中央，对局双方执黑者先行。合理的落子点有如下要求：

1. 落在棋盘空格上。
2. 能翻转敌方一颗或多颗棋子，成为己方棋子。能翻转棋子的要求是落子点在八个直线方向上有一颗己方棋子，与这颗棋子的连线上都是敌方的棋子（不能有空格，且至少有一颗）。这八个直线方向上满足要求的敌方棋子都必须要全部翻过来。

如果一方没有合理落子点，那他将跳过这一回合，对手将继续落子，直到他有合理落子点或比赛结束为止。如果一方至少有一个合理落子点，他就不能跳过这一回合，必须落子。当棋盘被填满或双方都没有合理落子点时，游戏结束，棋盘上棋子个数较多的一方胜利，如果双方棋子一样则宣告平局。

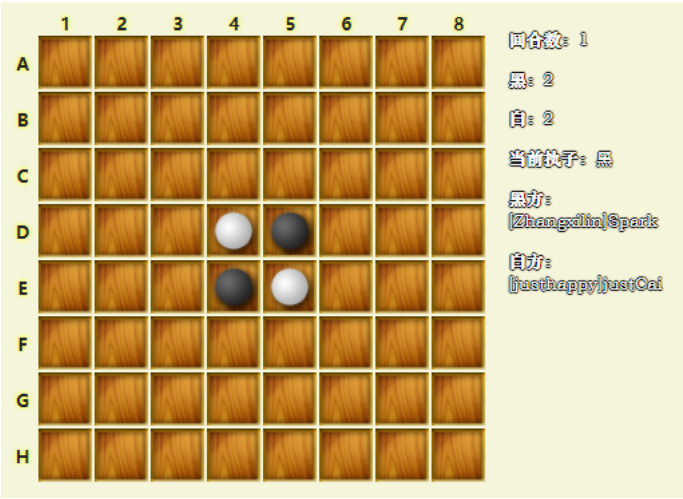

图 4.1　Botzone 平台黑白棋游戏截图

游戏过程中随时都显示黑棋数和白棋数，每一回合的棋子数都可能会急剧变化。黑白棋游戏中，位置的重要性超过棋子个数占优，在上图中，一般称 A1、A8、H1、H8 为“角”，A 行、H 行、1 列、8 列为“边”。玩家会更加关注对角及边的占据，因为在角上的棋子不会被翻转，属于“稳定子”，这也是大多数黑白棋 AI 实现的要点之一。

4.2 被模仿 AI 代码静态分析

截至 2021 年 3 月， Botzone 平台上的黑白棋 Bot 共有 434 个，使用算法包括蒙特卡洛树搜索（MCTS，Monte Carlo Tree Search）、极大极小搜索（MiniMax Search）、Alpha-Beta 剪枝、人类专家经验等等，其中技巧包括使用神经网络代替估值函数、自我对弈训练参数等等。

本实验选取的被模仿 AI 是天梯榜上公开的 5 个使用搜索算法的 Bot，在表 4.1 中整理了 Bot 的信息及其使用的算法框架以及搜索层数，其中 Bot 序号按照天梯排名给出，1、2、3 号 Bot 实力相近、排名靠前，4 号 Bot 较弱、排名中等，5 号 Bot 最弱、排名靠后。由于所有 Bot 都使用了剪枝，且用尽了搜索时间，并未对个别分支进行搜索限时，故不在表格中列出。

表 4.1 黑白棋 - Bot 搜索算法及搜索参数

序号	排名	Bot 名	用户名	算法框架	搜索层数
1	17	Minimaximin	Carl_xiao	NegaMax AlphaBeta 剪枝	不限制搜索层数
2	28	justCai	justhappy	Negamax AlphaBeta 剪枝	迭代加深 最多 20 层
3	36	mctsV5	rayeren	MCTS MiniMax AlphaBeta 剪枝	迭代加深 最多 15 层
4	94	test	Troye_Fun	MiniMax AlphaBeta 剪枝	迭代加深 最多 8 层
5	206	Altair	dataisland99	MiniMax AlphaBeta 剪枝	不限制搜索层数

黑白棋游戏从诞生到如今已有两百余年，专家经验在黑白棋 AI 的编写中发挥着重大作用，定式开局库使得搜索需要的时间降低，优化了 AI 的性能。在以上的 Bot 中也加入了人类经验，主要体现在对局面进行估值时，除了常用的快速走子获得终局胜负反推估计当前不同策略的优劣，还有直接赋予局面不同位置的权重预估，计算与游戏规则相关的特征数据，一并用于局面估值。其中，1 到 4 号的 Bot 的局面权重数组都相同，5 号 Bot 的权重数组在棋盘中央与其他 Bot 的权重数组相似，角落则不同。

4.3　生成模仿 AI 的关键步骤

4.3.1　训练数据预处理

在 Botzone 平台上每个 Bot 选取 5800 个对局，每个对局平均包含 30 个状态动作对，约为 174000 个状态动作对。

为 5 个 Bot 分别生成“状态-动作”对数据集，状态是形状为 8x8x2 的三维矩阵，第一个 8x8 是己方的落子点，第二个 8x8 是敌方的落子点，动作为含有 64 个元素的一维向量，分别表示棋盘上的 64 个格子，合法动作掩膜和动作具有相同形状。数据集中所有数据都是 01 矩阵，“状态-动作”对以对局为单位存储。根据状态计算随机选择准确率作为基线（Baseline），对于状态S_i有A_i个合法动作，随机选择一个的准确率是$\frac{1}{A_i}$，数据集中一共有N个状态，则随机选择准确率如式子（4-1）。

$$baseline = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{A_i} \tag{4-1}$$

将收集到的 5 个 Bot 各自的数据集及随机选择准确率列在表 4.2 中。下一小节中设计的神经网络将分别用这些数据集进行训练。

表 4.2　黑白棋 - Bot 数据集及随机选择准确率

Bot 序号	1	2	3	4	5
状态动作对个数	175283	174694	175504	173391	167757
随机选择准确率（Baseline）	17.44%	21.24%	23.82%	18.65%	24.30%

4.3.2　神经网络搭建与训练

本实验中设计了三种不同的神经网络，分别记为 MLP-1、MLP-2、CONV，表示两种全连接网络和一种卷积网络。网络的结构与参数量细节参数列在表 4.3 中。

表 4.3　黑白棋 - 网络结构及超参

网络名	参数量	网络层类型	网络结构
MLP-1	33472	全连接	三层全连接，中间层有 64 和 64 个隐藏单元
MLP-2	79104	全连接	三层全连接，中间层分别有 128 和 64 个隐藏单元
CONV	75088	卷积	两层卷积加一层全连接，卷积核大小为 3x3，最大步长为 1，卷积后用 0 填充使得相邻两层形状相同

表 4.2 中，全连接网络 MLP-1 和 MLP-2 网络结构高度相似，只有神经元个数不同，

这使得二者参数量不同。MLP-2 和 CONV 的参数量相近，处于同一量级，后者相较前者用了卷积层。

网络输入除了对局状态特征矩阵，还包括合法动作掩膜矩阵，与网络倒数第二层的输出相乘，故网络倒数第二层的输出与最后的输出维度相同，都是 64。

网络训练部分，所有模型都使用了 Adam 优化器、ReLU 激活函数，最后一层用 softmax 算子输出，CONV 中全连接与卷积的层连接用批归一化层过渡，学习率范围为 $[2e^{-5}, 2e^{-3}]$，设置早停为 3 的代际次数，批处理样本大小为 64，最大迭代次数为 30。

4.3.3 训练结果与分析

将上一小节中设计好的 3 个模型分别在 5 个 Bot 的数据集上进行训练。每个 Bot 的数据集都有 5800 个对局，训练时以 9:1 划分训练集和验证集。将三种网络的训练准确率（train_acc）和验证准确率（val_acc）列在表 4.4 中。

表 4.4 黑白棋 - 网络训练结果

Bot 序号	Baseline	ACC	MLP-1	MLP-2	CONV
1	17.44%	train_acc	98.25%	98.69%	98.74%
		val_acc	89.64%	89.67%	90.24%
2	21.24%	train_acc	98.28%	98.49%	98.55%
		val_acc	88.99%	88.77%	89.50%
3	23.82%	train_acc	98.22%	98.48%	98.55%
		val_acc	88.84%	88.92%	89.53%
4	18.65%	train_acc	98.44%	99.88%	99.98%
		val_acc	83.50%	84.79%	85.81%
5	24.30%	train_acc	99.48%	99.83%	99.88%
		val_acc	86.66%	89.91%	90.21%

表中，所有网络在不同 Bot 的数据集上都达到了 98%以上的训练准确率。CONV 卷积网络准确率最优，整体高于 MLP-2 和 MLP-1 这两种全连接网络。但由于模型在所有 Bot 数据集上的训练准确率都在 98%以上，故这三种网络在拟合能力上相差较小。

所有模型都达到了 75%以上的验证准确率，可以印证网络具有一定的学习能力，在验证集上的准确率都高于随机选择准确率（Baseline）。网络随着 Bot 数据集增大，整体验证准确率呈现出上升趋势。验证准确率上，CONV 卷积网络整体表现优于其他两个

网络，MLP-1 和 MLP-2 的差别不明显。除了第四个 Bot，其他 Bot 的最优表现网络 CONV 都达到了 90%左右的验证准确率。

根据以上结果，选出 CONV 卷积网络进行下一小节的相似度评估。

4.4　模仿 AI 相似度评价工作流程

4.4.1　测试数据采集与分析

本小节介绍采集黑白棋模仿 AI 任务中，用于计算相似度的状态集和残局集的流程。

状态集和残局集都使用关键局面作为测试数据。从 5 个 Bot 的历史对局中选出 1000 个残局，要求被模仿 Bot 和至少一个参考 Bot 解决残局的胜负结果不一样。参考 Bot 由除了当前生成残局集的被模仿 Bot 以外的 4 个 Bot，也即生成第 1 号 Bot 的残局集时，使用 2、3、4、5 号 Bot 作为参考 Bot。

4.4.2　模仿 AI 相似度评价计算与分析

使用上一小节收集的测试数据，应用 IME 中的评价模型，计算模仿 Bot 与原 Bot 的单步动作相似度和残局胜负相似度并列在表 4.5 中。

表 4.5　黑白棋 - 模仿 Bot 相似度评估

被模仿 Bot 天梯排名	被模仿 Bot 序号	单步动作相似度	残局胜负相似度	算法
17	1	74.80%	69.90%	NegaMax+AlphaBeta+局面估值+不限制搜索层数
28	2	80.40%	58.90%	Negamax+AlphaBeta+局面估值+迭代加深+最多搜 20 层
36	3	59.90%	36.60%	MCTS+MiniMax+AlphaBeta+局面估值+迭代加深+最多搜 15 层
94	4	69.70%	87.20%	MiniMax+AlphaBeta+局面估值+迭代加深+最多搜 8 层
206	5	79.90%	73.90%	MiniMax+AlphaBeta+局面估值+不限制搜索层数

残局胜负相似度与单步动作相似度没有线性相关性。

虽然 2 号 Bot 的单步动作相似度最高，但是残局胜负相似度仅次于最小的 3 号 Bot。

从残局胜负相似度来看，排名靠后的 Bot（4 号和 5 号）相比其他 Bot 更容易模仿，而 3 号 Bot 模仿效果最差，这和 3 号 Bot 使用的算法框架包含 MCTS 有关，MCTS 是一类启发式随机搜索算法，算法中的随机性使得 Bot 更难模仿。

实际搜索层数和局面估值准确性会影响 Bot 能力。从表中最后一列算法的搜索层数来看，迭代加深会使得实际的搜索层数低于限定的搜索层数上限，1、2、3、4 搜索层数依次递减，天梯排名依次降低。天梯排名最靠后、能力最弱的 5 号 Bot 虽然不限制搜索层数，但其局面估值中关于局面的权重估值与其余 Bot 大不相同，更突出了边角的重要性，从图 4.2 中可以看出，5 号 Bot 在决策时对某些特定位置的抢占比例明显更多。实际证明，5 号 Bot 的局面估值相对不合理，排名大大落后于其他 Bot。

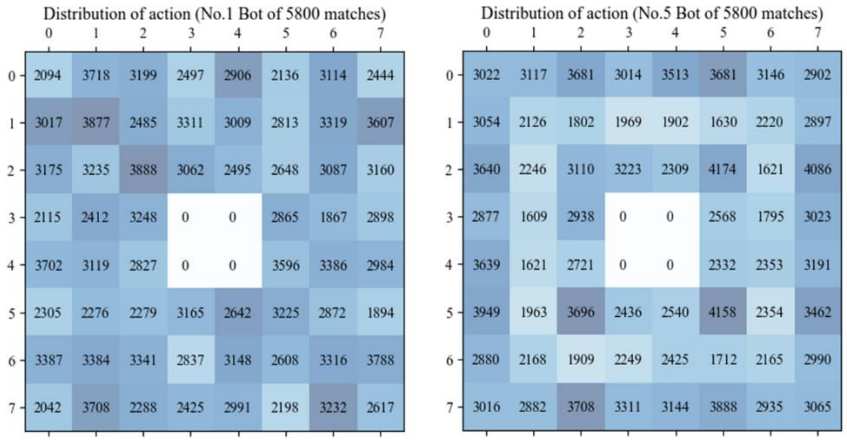

图 4.2　黑白棋 - 1 号 Bot 和 5 号 Bot 的动作决策分布

此外，根据以上分析可以提出两个猜想。首先，实际搜索层数越浅，逻辑偏于浅层，越容易模仿。其次，模仿 Bot 的残局胜负相似度都显著高于单步动作相似度，这可能是因为存在对称局面。

4.5　评价模型在游戏 AI 聚类中的应用

这一节将基于相似度的评价模型应用于 5 个黑白棋被模仿 Bot。

4.5.1　基于模仿效果的游戏 AI 聚类及个性分析

这一小节根据模仿任务中训练完的网络的与被模仿 Bot 之间的相似度，对被模仿 Bot 进行聚类。

根据模仿相似度评价结果，可以将这 5 个 Bot 分为两类，一类是 1、2、4、5 号 Bot，

一类是 3 号 Bot，前一类的 Bot 的模仿相似度要远高于后一类。结合 Bot 使用的算法框架分析，可以发现前一类 Bot 都使用的确定性搜索，在相同场景下“行为一致”，而后者具有一定的随机性，表现出“不可捉摸”的个性特征，也更难模仿。

观察上一小节模仿相似度评估表中对 5 个 Bot 的算法罗列，4 号 Bot 的搜索层数比 1、2 号更浅，表现出“短视”的特征，更容易模仿。5 号则表现出局部性，偏好边角。靠近中心的动作较多，是因为黑白棋游戏开局一般会下在这些格点，这也更加说明了 5 号 Bot 的决策是有局部性偏好的。

模仿 1、2 号的 Bot 在残局胜负相似任务中，表现不佳，模仿 Bot 只学到了表面的决策逻辑，无法学到 1、2 号 Bot“远视”特性。

3 号 Bot 使用的 MCTS 算法因为具有随机成分，模仿 Bot 在学习对局数据时，很难有效提取行为模式以及学到探索能力，残局胜负相似度在所有模仿 1、2、4、5 号的 Bot 中垫底。

4.5.2　基于相似度模型的被模仿 AI 聚类结果及分析

这一小节使用相似度评价模型，度量无模仿关系的 Bot 的相似性。

评价模型除了度量被模仿 AI 与模仿 AI 之间的相似性，还可以用来度量非模仿关系的 AI。在第三章给出的相似度定义下，本文概括两个 Bot 相似的要点是，相同局面状态下动作策略相似，解开残局问题的结果相似，这对于同一游戏中任何两个的 AI 都是适用的。

对被模仿的 5 个 Bot 先进行单步相似度计算，使用 4.4.1　小节中收集的关键局面作为测试数据，将被模仿 Bot 之间相似度整理到表 4.6。表格最后一列（IMBOT）给出单步相似度最高的模仿 Bot 与被模仿 Bot 之间的相似度。

表 4.6　黑白棋 Bot 之间的单步动作相似度

被模仿 Bot 天梯排名	单步动作相似度	1	2	3	4	5	IMBOT
17	1		7.55%	5.85%	16.95%	10.75%	13.9%
28	2	7.55%		4.35%	18.85%	8.30%	8.7%
36	3	5.85%	4.35%		16.55%	8.30%	7%
94	4	16.95%	18.85%	16.55%		37.25%	72.1%
206	5	10.75%	8.30%	8.30%	37.25%		40.9%

从表中看出，1、2、3 号 Bot 之间互相不相似，4、5 号 Bot 之间相似度较高，但也低于与各自模仿 Bot 的相似度。

接下来对被模仿的 5 个 Bot 进行残局胜负相似度计算，并将表现最好的模仿 Bot 的残局胜负相似度列在表 4.7 的最后一列。从表中看出，1、2、3 号在残局胜负相似上非常接近，达到 52%以上，4、5 号 Bot 与 1、2、3 号 Bot 的相似度都较低。除了 3 号 Bot，1、2、4、5 号 Bot 都与其模仿 Bot 的相似度最高。

表 4.7　黑白棋 Bot 之间的残局胜负相似度

残局胜负相似度	1	2	3	4	5	IMBOT
1		52.94%	55.73%	21.97%	35.52%	69.90%
2	52.94%		58.43%	30.65%	45.09%	58.90%
3	55.73%	58.43%		29.69%	41.48%	36.60%
4	21.97%	30.65%	29.69%		38.05%	87.20%
5	35.52%	45.09%	41.48%	38.05%		73.90%

从另一个角度更直观地了解模仿 Bot 能力。将 5 个 Bot 与各自的模仿 Bot 一起进行 50 次双循环赛，一共进行 50*10*9=4500 场对局，胜者得 3 分，败者不得分，平局双方各得 1 分。

表 4.8　黑白棋 - 所有模仿 Bot 与被模仿 Bot 的双循环赛分数及排名

双循环赛排名	Bot	分数
1	1 号 Bot（天梯排名 17）	2591
2	2 号 Bot（天梯排名 28）	2009
3	3 号 Bot（天梯排名 36）	1695
4	模仿 1 号 Bot	1607
5	4 号 Bot（天梯排名 94）	1464
6	模仿 2 号 Bot	927
7	5 号 Bot（天梯排名 206）	817
7	模仿 4 号 Bot	817
8	模仿 5 号 Bot	801
9	模仿 3 号 Bot	702

1 号到 5 号 Bot 的天梯排名依次降低，表 4.8 中看出双循环赛中 1 到 5 号 Bot 的排名也依次降低。模仿 Bot 的排名和被模仿 Bot 的排名有关，模仿 Bot 的排名顺序也从 1

到 5，说明模仿有一定效果，且模仿强 Bot 比模仿弱 Bot 更厉害。对比被模仿 Bot 与其模仿 Bot，可以看到模仿 Bot 排名分数低于被模仿 Bot，说明仅靠模仿不能超越原 Bot。

4.6　本章小结

本章在黑白棋游戏中应用了 IME 模型。首先，构建模仿 Bot 步骤设计了三种神经网络，对比不同网络结构、参数量会对模仿产生什么影响。选出表现最好的网络进行相似度评价，然后与 5 个原 Bot 之间的相似度进行比较，对比原 Bot 使用的不同的算法框架会对模仿产生什么影响。

因此得到以下结论：

1. 当网络参数量量级相同时，卷积网络比全连接网络在模仿验证集上表现更好；当网络结构相同时，不同参数量的全连接网络表现差别不大；
2. 被模仿 Bot 使用的算法框架中，实际搜索层数和局面估值准确性会影响到 Bot 的能力，可能影响到模仿其的难易程度；算法中含有随机性成分的 Bot 相对于确定性算法的 Bot 更难模仿；
3. 模仿在黑白棋游戏上有效，模仿强 Bot 比模仿弱 Bot 更好，但仅靠模仿不能超越原 Bot。

第五章　IME 在斗地主游戏中的应用与分析

5.1　斗地主游戏规则及性质分析

斗地主是一个三人回合制非完全信息随机性纸牌游戏，用一副 54 张的扑克牌，包含大王小王，3 个玩家中一人为地主，另外两人为农民，农民为合作关系，地主和农民阵营互相对抗。游戏开始时，农民拿 17 张牌，哪方先出完，哪方就获胜。本实验中使用 Botzone 上斗地主游戏，没有叫地主环节，系统指定 0 号玩家为地主。

游戏分几个阶段进行：（1）发牌阶段，地主拿到 20 张牌，两个农民分别拿到 17 张牌，地主比农民多出的额外的 3 张牌为明牌，也即农民知道这 3 张牌在地主手里，但其他的牌互相不知道；（2）出牌阶段，游戏开始时地主先出牌，然后位于地主下家的农民甲出牌，然后农民甲下家的农民乙出牌，再到地主，如此循环，轮到某家出牌时，可以选择过牌或跟牌，过牌即不出，跟牌需要出比当前最后一次出牌更大的牌，直到某一方出完手中所有牌；（3）如果某方出牌后，其他两位玩家都没有出牌，那么该玩家获得一次任意出牌的机会，且不能选择不出。特别的，在跟牌时，如果不是出火箭或炸弹，牌型需要和上家出的牌型相同，不比花色，只比纸牌数值大小。

斗地主状态空间复杂度为10^{43}，博弈树复杂度为10^{125}。参考在 2.1.3　小节中常见游戏复杂度表 2.1，斗地主复杂度与国际象棋接近[29]。这两种复杂度更适合用于描述完全信息游戏，如棋类游戏，在完全信息游戏中使用搜索算法能获得较好的性能，而复杂度也对应搜索空间大小。斗地主的另一个困难之处在于，它是一个非完全信息游戏，在斗地主游戏里进行最优解搜索的效率远不如棋类游戏，斗地主更适合用信息集数目和信息集平均大小来度量。信息集指在你的视角里，无法区分的游戏状态集。比如扑克游戏中，你和其他玩家各拿一张，你只知道你自己的手牌，其他玩家的手牌是什么对你来说无法分辨。搜索算法在非完全信息游戏上的失败也正因如此，细粒度的考虑对手是什么牌导致大多数的结点展开分支都是无效的。在以上扑克游戏中，信息集平均大小会随着各玩家手里的牌数目增多而变大，量级在组合数级别。图 5.1 展示了常见的非完全信息游戏的信息集数目及平均大小。按列看，从左到右分别是游戏环境、信息集数目、平均信息集大小、合理动作空间大小。倒数第四行为斗地主游戏，信息集数目在10^{53}~10^{83}，信息集平均大小为10^{23}，合理动作空间大小为10^{4}。处于限注德州扑克和麻将游戏之间。

Environment	InfoSet Number	Avg. InfoSet Size	Action Size
Blackjack	10^3	10^1	10^0
Leduc Hold'em	10^2	10^2	10^0
Limit Texas Hold'em	10^{14}	10^3	10^0
Dou Dizhu	$10^{53} \sim 10^{83}$	10^{23}	10^4
Mahjong	10^{121}	10^{48}	10^2
No-limit Texas Hold'em	10^{162}	10^3	10^4
UNO	10^{163}	10^{10}	10^1

图 5.1 常见非完全信息游戏的信息集数目及平均大小[28]

5.2 被模仿 AI 代码静态分析

由于缺少了叫牌环节，且游戏计分与传统斗地主不同，没有翻倍只有根据牌型计算的小分，故对 Bot 的要求更加注重拆牌以及对其他两家的手牌估计。由于斗地主是农民合作与地主对抗，农民一方的 Bot 需要考虑与无法通信的农民队友合作，而地主则应尽力保持压制农民，两方阵营都应抢夺牌权获得主动出牌机会。

截至 2021 年 3 月， Botzone 平台上的斗地主 Bot 共有 348 个，Bot 以人类专家经验为主，天梯排行榜第一的是使用监督学习模仿专家的 Bot。涉及算法主要包括专家系统、Alpha-Beta 剪枝、纳什均衡、强化学习等。本实验选取了天梯榜上的 5 个 Bot，都使用人类专家经验。表 5.1 整理了 Bot 在主动出牌与被动出牌、与农民队友合作两方面的差异。

第 1 号 Bot 在主动出牌时，优先考虑多于一个的牌的类型，比如 333、666，另外也优先考虑出长顺，一来希望能获得下一圈的牌权，二来减少手牌数降低拆牌复杂度，使得 Bot 在对局中更表现出进攻性，有时会因为太早把大牌打光而防守变弱，无法抢到对局后半段的牌权，也即主动出牌的机会。第 2 号 Bot 与农民队友的合作部分较为突出，当坐在地主上位时让队友过牌并压制地主，也不在队友已经出了较大的时，盲目垫牌，以此保留实力，在做地主进攻较强，做农民表现出的合作辅助的特性。这两个 Bot 都非常重视牌权，在出牌的优先顺序上有自己的偏好。

3 号 Bot 在出牌估值时也考虑了手牌减少引起的价值变动，并给不同大小的牌和牌组合赋予了人类经验估计的价值。跟牌和主动出牌都对牌型进行了特判，在当农民时会计算队友胜利可能性决定是否让牌。

4 号 Bot 在拆牌方面体现出了自己的偏好，首先拆出火箭和大小王，然后是炸弹，接着是顺子、连对，飞机，最后剩下三张、对子和单张。接下来给三张和飞机的候选牌

型从小到大带上副牌。估值和前面的 Bot 拆牌估值流程类似，与农民队友合作，会送小牌让队友过牌，不用炸弹炸队友。

5 号 Bot 的主动出牌和跟牌有很明显的防守性。主动出牌和跟牌时都选择出最小牌型。

表 5.1　斗地主 - Bot 专家经验

序号	排名	Bot 名	用户名	主动出牌与被动跟牌	与农民队友合作
1	30	小咪咪熊	啊咪咪小熊	遍历拆牌种类牌型估值	给队友让牌；压制地主
2	69	次代版本	子曰木天	遍历拆牌种类牌型估值	与地主争夺牌权
3	117	我们有炸弹你们怕不怕	我是地球人	遍历拆牌种类牌型估值	计算队友胜率考虑让牌
4	167	BrandNewCPP	Truckey	搜索拆牌种类牌型估值	给队友送小牌，不炸队友
5	235	狗狗狗	一条狗	出最小牌	当队友胜率较高时放弃出牌

总结来看，1 号到 4 号 Bot 在估值之前都搜索拆牌种类，并对牌型进行估值。5 号 Bot 基于强规则出牌，每次都出最小牌型。

下文实验将使用这 5 个 Bot 的对局数据进行训练。从上述静态分析中可以看出，部分 Bot 的表现出了一些特定模式，比如有小牌出小牌，当农民的时候不压队友的牌；但是另一些 Bot 较难把握特性。本文希望能借助神经网络这一工具，模仿 Bot 行为策略，帮助找到其决策典型模式。Bot 的序号排序按照其天梯排名，从 1 到 5 能力依次减弱，1 号到 4 号积极进攻，5 号消极防守。

5.3　生成模仿 AI 的关键步骤

5.3.1　训练数据预处理

在 Botzone 平台上每个 Bot 选取 10600 个对局，每个对局平均包含 11 个状态动作对，约为 116600 个状态动作对。

Botzone 平台上的斗地主在天梯及以往的比赛中被当做双人游戏使用，两个农民是

同一个 Bot，但它们之间不能交流，只能各自根据当前局面状态做决策。考虑到同一个 Bot 可能在两个农民位都有策略输出，实验将整理所有被模仿 Bot 的局面状态。对上述被选中的 5 个 Bot 的历史对局整理成状态动作对序列。

实验中用统一的方式来表示一组牌：将牌组使用 4x15 的 01 矩阵表示，4 表示每一种点数的扑克牌都有 4 张，第二维维度的 15 表示一共有 15 种点数的牌。花色不计。图 5.2 展示了将地主手牌编码成 01 矩阵，图的右半边是一个 4x15 的矩阵，黑框表示 1，白框表示 0。J 有 3 张，故有 3 个格子被赋值 1。

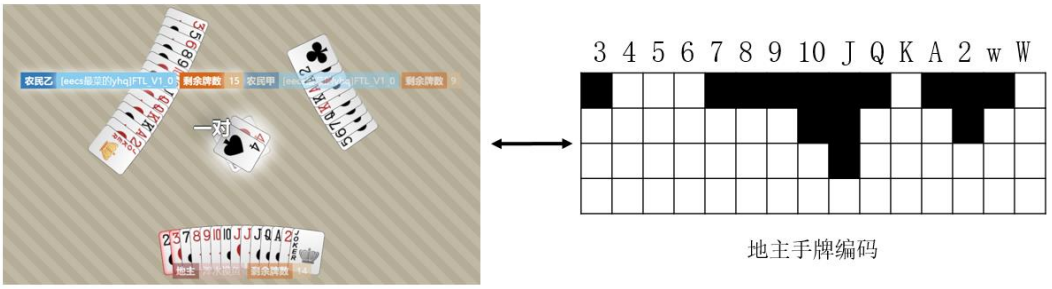

图 5.2　斗地主 - 地主手牌编码矩阵示意图

在说明输入特征数据格式之前，先讨论输出标签格式的问题。将出牌视作主牌和小牌的组合，游戏规则中规定对比出牌大小时以主牌为准。经过整理，主牌类型一共有 309 种，小牌类型有 28 种。表 5.2 为将主牌类型编码。

表 5.2　斗地主 - 主牌类型编码

编码范围	主牌类型	举例	编码范围	主牌类型	举例
0	不跟牌	**PASS**	[131,143]	三不带	**666**
1	火箭	**大王小王**	[144,188]	三顺	**888999**
[2,14]	炸弹	**4444**	[189,201]	三带一	**888** 3
[15,29]	单张	**3**	[202,239]	三带一（顺）	**777888** 45
[30,65]	单顺	**45678**	[240,252]	三带二	**777** 66
[66,78]	对子	**88**	[253,282]	三带二（顺）	**777888** 4455
[79,130]	双顺	**778899**	[283,295]	四带两只	**4444** 67
			[296,308]	四带两对	**4444** 6677

小牌类型编码相对简单。在斗地主里，小牌为跟着主牌一起出的部分，在 Botzone 上设置为与主牌不重复，有单只、一对两种类型。表 5.3 为小牌类型编码。

表 5.3　斗地主 - 小牌类型编码

编码范围	小牌类型	举例	编码范围	小牌类型	举例
[0,14]	单只	4	[15,27]	一对	66

接下来对主牌数据集和小牌数据集的状态动作对分别进行处理。主牌数据集的状态为 4x15x6 的 01 矩阵，前两个维度的 4x15 代表某一个牌组，第三维度包含 6 部分的信息，分别为己方手牌、手牌中数量超过 3 张的牌、座次、另外两家没有打出的未知牌、这一圈里下家的出牌以及上家的出牌。小牌网络的输入特征数据为 4x15x4 的 01 矩阵，第三维度包含 4 部分的信息，分别为己方手牌、主牌、单只牌需要个数、一对牌需要个数。在处理主牌跟多种小牌时，对应的动作将变成小牌序列，从小到大排序。生成小牌数据集的伪码列在图 5.3 中。

```
FUNCTION 生成小牌数据集 (手牌 hand, 主牌 major, 小牌序列 minors)
    LET pairs = []
    LET num_solo, num_pair = count(minors < 15), count(minors >= 15)
    FOR minor_solo in minors                    ; 对于小牌序列中的每种单只
        state = concatenate(hand, major, num_solo, 0) ; 连接得到当前状态
        append(pairs, (state, minor_solo))      ; 加入状态动作对序列
        remove(hand, minor_solo)                ; 从手牌中将单只小牌去除
        num_solo = num_solo – 1                 ; 单只种类数减一
    FOR minor_pair in minor                     ; 对于小牌序列中的每一对
        state = concatenate(hand, major, 0, num_pair)  ; 和上述类似处理
        append(pairs, (state, minor_pair))
        remove(hand, minor_pair)
        num_pair = num_pair – 1
    RETURN pairs
```

图 5.3　斗地主 - 小牌网络动作序列拆解为状态动作对序列算法

与黑白棋类似，本实验将“状态-动作”对按对局为单位存储，并同时提前计算在局面状态下的合法动作掩膜，加速训练过程。

将收集到的 5 个 Bot 各自的数据集及随机选择准确率列在下表中。下一小节设计的

神经网络将分别使用这些数据集进行训练。

表 5.4 斗地主 - Bot 数据集及随机选择准确率

Bot 序号	1	2	3	4	5
状态动作对个数	112924	116230	124727	125962	127717
随机选择准确率（Baseline）	52.65%	55.27%	58.43%	59.46%	54.81%

5.3.2 神经网络搭建与训练

斗地主实验中只使用卷积网络。斗地主模仿 AI 实验设计了两组卷积神经网络 CONV-1 和 CONV-2，网络组用主牌网络（Major）和小牌网络（Minor）的元组表示，CONV-1 网络组为(Major-CONV-1,Minor-CONV-1)，CONV-2 网络组为 (Major-CONV-2,Minor-CONV-2)。Major-CONV-1 和 Major-CONV-2 只有卷积核的个数不同，这使得二者参数量不同。网络组中主牌网络和小牌网络协同合作。网络的结构与参数量细节参数列在表 5.5 中。

表 5.5 斗地主 - 网络结构及超参

网络组	网络名	参数量	网络层类型	网络结构
CONV-1	Major-CONV-1	399219	卷积	两层卷积，两层全连接，卷积核大小都为 3x3，卷积核个数分别为 32 和 16，两层全连接输出都为 309。
	Minor-CONV-1	33720	卷积	两层卷积，两层全连接，卷积核大小都为 3x3，卷积核个数分别为 32 和 16，两层全连接输出都为 28。
CONV-2	Major-CONV-2	173247	卷积	两层卷积，两层全连接，卷积核大小都为 3x3，卷积核个数分别为 32 和 4，两层全连接输出都为 309。
	Minor-CONV-2	17856	卷积	两层卷积，两层全连接，卷积核大小都为 3x3，卷积核个数分别为 32 和 8，两层全连接输出都为 28。

主牌网络和小牌网络共同决定输出决策动作。特别的，对于四带两只、四带两双这样的牌型，需要两种小牌，在训练过程中，根据主牌类型确定需要什么类型的小牌，小

牌需要多少种，然后重复调用小牌网络。

网络训练时，所有模型都使用了 Adam 优化器、ReLU 激活函数，图中所示的全连接与卷积的层连接用批归一化层过渡，最后一层用 softmax 算子输出，学习率范围为 $[2e^{-5}, 2e^{-3}]$，设置早停为 3 的代际次数，批处理样本大小为 64，最大迭代次数为 40。训练时，每个网络组里的主牌网络和小牌网络一起训练。

5.3.3　训练结果与分析

将上一小节中设计的两个网络组分别在 5 个 Bot 的数据集上进行训练。每个 Bot 都有 10600 个对局，训练时以 9:1 划分训练集和验证集。网络组的训练准确率（train_acc）和验证准确率（val_acc）列在下表中。

表 5.6　斗地主 - 网络组训练结果

Bot 序号	Baseline	ACC	**CONV-1**	CONV-2
1	52.65%	train_acc	**100.00%**	60.48%
		val_acc	**88.70%**	58.57%
2	55.27%	train_acc	**99.99%**	48.58%
		val_acc	**82.47%**	48.40%
3	58.43%	train_acc	**99.99%**	45.53%
		val_acc	**87.55%**	45.12%
4	59.46%	train_acc	**98.82%**	99.96%
		val_acc	**74.24%**	86.92%
5	54.81%	train_acc	**99.86%**	38.52%
		val_acc	**99.47%**	38.71%

表中 CONV-1 网络组在验证集上的准确率高于 CONV-2 和 Baseline（随机选择准确率）。CONV-2 的准确率反而低于 Baseline，这可能是因为 CONV-2 的网络结构设置不当或参数量过少，网络训练无法收敛到较好值[①]。

根据以上结果，选出 CONV-1 网络组进行下一小节的相似度评估。

[①] 限于实验设置无法收敛。

5.4 模仿 AI 相似度评价工作流程

5.4.1 测试数据采集与分析

用于计算斗地主模仿 AI 与被模仿 AI 相似度的状态集和残局集的收集流程，与在黑白棋实验中类似。状态集和残局集都使用关键局面作为测试数据。从 5 个 Bot 的历史对局中选出 1000 个残局作为关键局面。

5.4.2 模仿 AI 相似度评价计算与分析

使用上一小节收集的测试数据，应用 IME 中的评价模型，计算斗地主模仿 Bot 与原 Bot 的单步动作相似度和残局胜负相似度，结果列在表 5.7 中。

表 5.7 斗地主 - 模仿 Bot 相似度评估

被模仿 Bot 天梯排名	被模仿 Bot 序号	单步动作相似度	残局胜负相似度	算法
30	1	85.03%	83.90%	遍历拆牌种类+牌型估值
69	2	84.00%	74.50%	遍历拆牌种类+牌型估值
117	3	77.40%	61.90%	遍历拆牌种类+牌型估值
167	4	73.20%	78.20%	搜索拆牌种类+牌型估值
235	5	81.50%	91.80%	出最小牌

从表中可以很明显看出，5 号 Bot 规则更加简单，决策确定性更高，残局胜负相似度最高，更容易模仿。

与黑白棋实验对比，斗地主的模仿 Bot 相似度评估值要普遍更高，但单步动作相似度和残局胜负相似度没有表现出线性相关性。这可能与斗地主 Bot 使用的算法框架差异更大有关。对此可以提出一个猜想，存在这样的关键局面，出牌顺序不一致不影响最后的胜负结果。

5.5 评价模型在游戏 AI 聚类中的应用

这一节将基于相似度的评价模型应用于 5 个斗地主被模仿 Bot。

5.5.1　基于模仿效果的游戏 AI 聚类及个性分析

这一小节根据训练完的网络组与被模仿 Bot 之间的相似度，对被模仿 Bot 进行聚类。

根据模仿相似度评价结果，可以将这 5 个 Bot 分为两类，一类是 1、2、3、4 号 Bot，一类 5 号 Bot，前一类的 Bot 的残局胜负相似度要低于后一类。结合斗地主原 Bot 使用的算法框架分析，可以发现前一类 Bot 在出牌时对拆牌可能的组合进行了搜索，而后一类也即 5 号 Bot 在出牌时使用的是确定性规则——出最小牌。

1、2、3、4 号 Bot 会对所有拆牌可能进行估值，其中也包含出牌后的手牌估值，在跟牌场景下，如果发现出牌后手牌价值下降太多，那么很可能会选择不跟牌，表现出“注重全局”的特性。5 号 Bot 使用强规则，不管是主动出牌还是被动出牌，都是出手中的最小牌。5 号 Bot 没有很好地规划手牌，在对局中很可能因为主动出牌出了最小牌而丢失本轮牌权，强硬跟牌而使手牌价值变低，表现出“注重局部”的特性，在天梯中的排名也大不如其他 4 个 Bot。

5.5.2　基于相似度模型的被模仿 AI 聚类结果及分析

这一小节使用相似度评价模型，度量无模仿关系的 Bot 的相似性。

和黑白棋实验类似，对被模仿的 5 个 Bot 先进行单步动作相似度计算，测试状态集使用的是 5.4.1　小节中收集的关键局面。为了对比，在表 5.8 中也列出 5.4.2　小节中模仿 Bot 的相似度值。

表 5.8　斗地主 Bot 之间的单步动作相似度

被模仿 Bot 天梯排名	被模仿 Bot 序号	1	2	3	4	5	IMBOT
30	1		68.15%	64.80%	57.35%	57.70%	85.03%
69	2	68.15%		65.25%	58.75%	58.85%	84.00%
117	3	64.80%	65.25%		58.05%	57.35%	77.40%
167	4	57.35%	58.75%	58.05%		56.05%	73.20%
235	5	57.70%	58.85%	57.35%	56.05%		81.50%

从表 5.8 中可以看出，1 号、2 号、3 号的动作相似度互相之间达到 64%以上，而 4、5 号 Bot 与其他 Bot 的相似度都较低。

接下来对斗地主 5 个 Bot 进行残局胜负相似度计算，表 5.9 中最后一列是 5.4.2

小节中模仿 Bot 的相似值。表中最后一列（IMBOT）显示，被模仿 Bot 与其模仿 Bot 的残局胜负相似度普遍高于与其他 Bot 的相似度。这说明在斗地主游戏中，使用神经网络进行模仿可能是比较合适的方法。

表 5.9　斗地主 Bot 之间的残局胜负相似度

被模仿 Bot 天梯排名	被模仿 Bot 序号	1	2	3	4	5	IMBOT
30	1		61.76%	58.78%	45.28%	47.56%	83.90%
69	2	61.76%		56.28%	46.84%	43.98%	74.50%
117	3	58.78%	56.28%		44.92%	45.24%	61.90%
167	4	45.28%	46.84%	44.92%		47.56%	78.20%
235	5	47.56%	43.98%	45.24%	47.56%		91.80%

从表 5.8 和表 5.9 中可以看到在评估单步动作相似度和残局胜负相似度时，不同 Bot 之间的相似度都不如与模仿 Bot 之间的相似度高。这也验证了关键局面的选取是有效的，在这些局面中能区分开模仿 Bot 和其余 Bot。

将 5 个 Bot 与各自的模仿 Bot 放在一起进行 200 次双循环赛，一共进行 200*10*9=18000 场对局，胜者得 3 分，败者不得分。斗地主游戏中无平局。

表 5.10　斗地主 - 所有模仿 Bot 与被模仿 Bot 的双循环赛分数及排名

双循环赛排名	Bot	分数
1	1 号 Bot（天梯排名 30）	7566
2	2 号 Bot（天梯排名 69）	7107
3	3 号 Bot（天梯排名 117）	7011
4	4 号 Bot（天梯排名 167）	5658
5	模仿 1 号 Bot	5436
6	5 号 Bot（天梯排名 235）	4890
7	模仿 2 号 Bot	4563
8	模仿 3 号 Bot	4551
9	模仿 4 号 Bot	3975
10	模仿 5 号 Bot	3243

根据以上结果，能得出与黑白棋中类似的结论。1 号到 5 号 Bot 的天梯排名依次降

低，双循环赛中 1 到 5 号 Bot 的排名也依次降低。模仿 Bot 之间的排名顺序与被模仿 Bot 之间的排名顺序一致，模仿有一定效果，且模仿强 Bot 比模仿弱 Bot 更厉害。对比被模仿 Bot 与其模仿 Bot，可以看到模仿 Bot 排名分数低于被模仿 Bot，仅靠模仿是不能超越原 Bot 的。

另外借助一场对局数据来看为什么模仿 Bot 的残局胜负相似度要高于其他 Bot。以 1 号 Bot 为例，IMBOT 的残局胜负相似度来源于 CONV-1 网络组（在大小为 10600 对局的数据集上训练），达到了 83.90%，远高于 2、3、4、5 号 Bot 与 1 号 Bot 的残局胜负相似度。

图 5.4 是一场残局截图，将残局手牌整理到表 5.11 中。残局是从原对局[①]的第 6 回合开始的，此时农民乙跟牌出了一对 10。

图 5.4　斗地主 - 残局起始手牌截图

表 5.11　斗地主 - 残局起始手牌

Bot	手牌
1 号 Bot（地主）	6 6 8 8 9 9 10 Q Q K K 2 2 大王
对手（农民甲）	4 4 4 5 7 7 7 9 J J Q A 2 2 小王
对手（农民乙）	3 3 3 4 5 6 6 7 9 10 J J Q K K

原对局中，1 号 Bot（地主）获得了对局的胜利。2、3、4、5 号 Bot 在这场残局中

[①] 原对局链接：https://www.botzone.org.cn/match/5c9088bd7857b210f901bab0

的结果与 1 号 Bot 不同，都失败了[①]。而 1 号的模仿 Bot 获得了对局胜利[②]，决策序列与 1 号 Bot 一致。

对局出现决策分叉是在第 19 回合，图 5.5 中展示了三种决策：

1. 左边子图展示的是 1 号 Bot（小咪咪熊），选择拆对 2 打单张，其模仿 Bot 和 5 号 Bot 与 1 号 Bot 决策相同；
2. 中间子图展示的是 2 号 Bot（次代版本），选择打大王，3 号 Bot 与 2 号 Bot 决策相同；
3. 右边子图展示的 4 号 Bot（BrandNewCPP），选择不跟牌。

图 5.5　斗地主 - 残局第 19 回合 Bot 决策截图

2 号 Bot 的失败节点出现在 28 回合，选择出单张 10，农民甲手中的单张 A 和小王是场上最大的牌，可以出完，2 号 Bot 因此失败（见图 5.6）。2 号和 3 号完成残局时决策序列一致，这里只展示 2 号 Bot 的决策截图。

[①] 2、3、4、5 号 Bot 完成残局链接：

2 号 Bot：https://www.botzone.org.cn/match/604dc0a72765d6060b78f5cb

3 号 Bot：https://www.botzone.org.cn/match/604dc0a72765d6060b78f5cf

4 号 Bot：https://www.botzone.org.cn/match/604dc0a72765d6060b78f5d3

5 号 Bot：https://www.botzone.org.cn/match/604dc0a72765d6060b78f5d7

[②] 模仿 Bot 完成残局链接：https://www.botzone.org.cn/match/6092f22e66ecf10ef5cf4230

图 5.6　斗地主 - 残局第 28 回合 2 号 Bot 决策截图

4 号 Bot 的失败节点出现在 37 回合，选择出对 6，农民甲对 J 刚好压过地主且出完牌，4 号 Bot 因而落败（见图 5.7）。

图 5.7　斗地主 - 残局第 37 回合 4 号 Bot 决策截图

5 号 Bot 的失败节点出现在 37 回合，从这个回合开始，5 号 Bot 和 1 号 Bot 的决策才出现分歧。图 5.8 左侧子图是 5 号 Bot 决策截图，可以看到出了单张 10，刚好被农民甲的单张 A 压过从而输掉对局。右侧子图是 1 号 Bot 决策截图，先出场上最大的单张 2，最后顺利赢得对局。

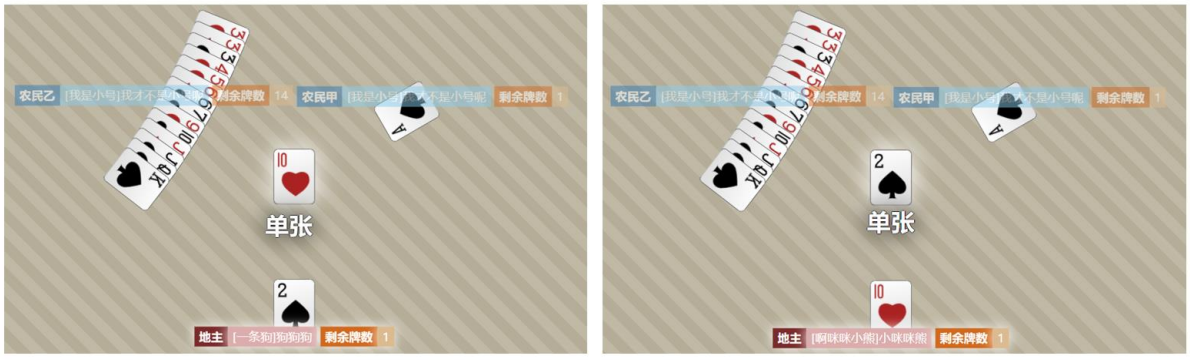

图 5.8　斗地主 - 残局第 37 回合 5 号 Bot 和 1 号 Bot 决策截图

以上对局分析说明，残局胜负相似度中，模仿 Bot 决策和被模仿 Bot 的决策结果一致度高，模仿效果好。实验中收集的残局集合，是被模仿 Bot 群体因为决策差异导致最后的胜负结果不一样的关键局面，这样的关键局面可以用于测试模仿 Bot 与被模仿 Bot 的决策一致度。

5.6 本章小结

本章在斗地主游戏中应用了 IME 模型。首先，构建模仿 Bot 步骤设计了两组神经网络，每组网络包括主牌网络和小牌网络，均使用卷积网络，两组网络只有参数量不同。选出表现最好的网络组进行相似度评价，然后与 5 个原 Bot 之间的相似度进行比较，对比原 Bot 使用的不同的算法框架会对模仿产生什么影响。

因此得到以下结论：

1. 当网络结构相同时，参数量对模仿效果有影响，过小的参数量会使得网络在训练时无法收敛到较好值，从而影响模仿效果；
2. 被模仿 Bot 使用的算法框架中，使用越简单规则的 Bot 决策越确定，越容易模仿；
3. 模仿在斗地主游戏上有效，模仿强 Bot 比模仿弱 Bot 更好，但仅靠模仿不能超越原 Bot。

第六章　总结与展望

6.1　本文工作总结

本文在“如何模仿游戏 AI”、“如何评价模仿效果”、“游戏 AI 是否具有个性化特征”这三个问题展开了讨论，提出了 IME 模型——基于神经网络的模仿游戏 AI 方法以及基于相似度计算的评价模型，并分别在黑白棋游戏和斗地主游戏上做了应用实例。

IME 模型给出了基于神经网络的模仿游戏 AI 个性的方法，以及评估模仿效果的评价模型。神经网络作为一个良好的学习表达的工具，为模仿游戏 AI 方法提供了一个统一的学习框架，神经网络的设计部分只有输入输出需要根据游戏规则改变，中间层的设计可以脱离游戏本身。基于相似度的评价模型提出两个相似指标，一个是在相同状态下的单步动作一致，一个是解决残局的胜负结果一致。单步动作一致更倾向于从底层动作上衡量相似，而残局胜负一致则从完成游戏达成目标的水平方面进行相似判定。残局胜负一致并不能简单地通过比赛胜率来衡量，要求模仿 AI 在特定的局面，与特定的对手对抗，取得特定的结果。这种相似标准可以广泛应用于以胜负为定论的游戏中。

本文在黑白棋、斗地主游戏上应用了 IME 模型，探究了不同网络结构、不同参数量对模仿 AI 产生的影响，结合 AI 算法框架分析，探究 AI 个性化特征。

网络结构方面，卷积网络在本文进行的实验中均好于纯全连接网络。两种网络的参数量处于同等量级时，卷积网络提取行为模式更具优势。相较于全连接网络，卷积网络能尽量保留重要参数，避免大量的无效连接训练，从而获得更好的性能。

不同参数量对模仿 AI 的影响在黑白棋和斗地主实验中不同。黑白棋实验中，不同参数量的全连接网络在评估相似度时差异不大，而在斗地主实验中则有比较明显的高低区分，参数量较大的网络组训练得到的模仿 AI 在衡量相似度时，要明显高于参数量小的网络组，参数量小的网络组无法收敛到较好值。

特别地，IME 模型能利用模仿 Bot 的学习数据，分析被模仿 Bot 的特性，并对其进行聚类。而基于相似度计算的评价模型，则提供了另一种聚类方法。评价模型不限于计算被模仿 AI 与模仿 AI 的相似度，也可以用于计算任意两个 Bot 之间的相似度。使用计算得到的任意两个 Bot 相似度反推差异度，并以此作为 Bot 之间的距离，可以对其进行聚类。

相似度评估实验中发现，AI 表现出个性化特征与 AI 使用的算法有关。实验发现了

三类 AI 算法，统计采样搜索、确定性搜索、基于规则，模仿的困难程度依次递减。将黑白棋和斗地主的模仿 Bot 的残局胜负相似度按照游戏、残局胜负相似度从低到高的顺序列在下表中。

表 6.1　AI 算法分类

游戏	天梯排名	序号	残局胜负相似度	分类	算法
黑白棋	36	3	36.60%	统计	MCTS+MiniMax+AlphaBeta+局面估值+迭代加深+最多搜 15 层
	28	2	58.90%	搜索	Negamax+AlphaBeta+局面估值+迭代加深+最多搜 20 层
	17	1	69.90%	搜索	NegaMax+AlphaBeta+局面估值+不限制搜索层数
	206	5	73.90%	搜索	MiniMax+AlphaBeta+局面估值+不限制搜索层数
	94	4	87.20%	搜索	MiniMax+AlphaBeta+局面估值+迭代加深+最多搜 8 层
斗地主	117	3	61.90%	搜索	遍历拆牌种类+牌型估值
	69	2	74.50%	搜索	遍历拆牌种类+牌型估值
	167	4	78.20%	搜索	搜索拆牌种类+牌型估值
	30	1	83.90%	搜索	遍历拆牌种类+牌型估值
	235	5	91.80%	规则	出最小牌

从表中可以明显看出，使用统计采样搜索的 Bot 的残局胜负相似度最低，确定性搜索处于中段，而基于强规则的 Bot 的残局胜负相似度最高。对于黑白棋 AI，赋予“确定性决策”、“随机决策”的语义特征，“确定性决策”的 Bot 在同一局面下的决策不变，能更好地找到行为模式，更容易模仿；“随机决策”的 Bot 则相反，模仿困难程度更高。对于斗地主 AI，赋予其“注重全局”和“注重局部”的语义特征，“注重局部”的语义来自于使用强规则，因为在局面下不懂得灵活变通、没有很好地规划未来，而是只关注眼前这一回合压过对手，能力值更低，也因为决策行为模式简单，故而更好模仿；“注重全局”的 Bot 在决策时也考虑未来的局面，使用搜索遍历拆牌种类并对牌型进行估值，能力值更高，也更难模仿。

6.2　本文研究展望

IME 模型仍存在一些需要改进的地方，也需要拓展更多的应用场景。

IME 模型在构建模仿 AI 方法的后续研究方面，有几点需要继续深入探究。以下几点可能会影响到模仿效果，需要进行多组对比实验：

- 游戏性质，比如：随机性/确定性，双人/多人，完成信息/非完全信息等等；
- 被模仿 AI 的算法以及随机性；
- 游戏过程中的对称局面以及决策顺序；
- 训练过程中，训练数据是否去重，数据量量级；
- 网络结构、网络参数量。

由于时间关系，本文不能对这些因素进行更加全面的对比探究，后续研究可以先从改进模仿方法入手，在获得更好的模仿AI的基础上，研究以上因素对模仿效果的影响。

IME 模型在模仿相似度的评价模型后续研究方面，可以进一步探究以下几点：

- 关键局面的选取，是从训练数据中选取，还是与训练数据完全不同的残局；
- 单步动作相似度和残局胜负相似度之间是否相关；
- 探索更多相似度指标，增加评估角度。

模仿 AI 可以用于新游戏冷启动、玩家掉线 AI 代打、陪玩 AI 等等业界游戏场景。模仿 AI 也可以用于提升玩家 AI 的水平，一种是通过模仿自己发现漏洞，一种是模仿对手，在展开博弈树时选择对手更可能的动作，从而减小搜索规模，提升找到最优解的效率。作者期待在未来的模仿 AI 研究中，能发现更多的 AI 个性风格。

参考文献

[1] McKevitt P. Daniel Crevier, AI: The Tumultuous History of the Search for Artificial Intelligence. London and New York: Basic Books, 1993. Pp. xiv+ 386. ISBN 0-465-02997-3.£ 17.99, $27.50[J]. The British Journal for the History of Science, 1997, 30(1): 101-121.

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

[3] Laird J, VanLent M. Human-level AI's killer application: Interactive computer games[J]. AI magazine, 2001, 22(2): 15-15.

[4] Newborn M. Kasparov versus Deep Blue: Computer chess comes of age[M]. Springer Science & Business Media, 2012.

[5] Silver D, Huang A, Maddison C J, et al. Mastering the game of Go with deep neural networks and tree search[J]. nature, 2016, 529(7587): 484-489.

[6] Brown N, Sandholm T. Superhuman AI for heads-up no-limit poker: Libratus beats top professionals[J]. Science, 2018, 359(6374): 418-424.

[7] Li J, Koyamada S, Ye Q, et al. Suphx: Mastering mahjong with deep reinforcement learning[J]. arXiv preprint arXiv:2003.13590, 2020.

[8] Liu Z, Hu M, Zhang Z. A Solution to China Competitive Poker Using Deep Learning[J]. 2018.

[9] Ortega J, Shaker N, Togelius J, et al. Imitating human playing styles in super mario bros[J]. Entertainment Computing, 2013, 4(2): 93-104.

[10] Van Hoorn N, Togelius J, Wierstra D, et al. Robust player imitation using multiobjective evolution[C]//2009 IEEE Congress on Evolutionary Computation. IEEE, 2009: 652-659.

[11] Silver D, Schrittwieser J, Simonyan K, et al. Mastering the game of go without human knowledge[J]. nature, 2017, 550(7676): 354-359.

[12] Shaker N, Togelius J, Yannakakis G N, et al. The turing test track of the 2012 mario ai championship: entries and evaluation[C]//2013 IEEE Conference on Computational Inteligence in Games (CIG). IEEE, 2013: 1-8.

[13] Zhou H, Zhang H, Zhou Y, et al. Botzone: an online multi-agent competitive platform for ai education[C]//Proceedings of the 23rd Annual ACM Conference on Innovation and Technology in Computer Science Education. 2018: 33-38.

[14] Gao C, Hayward R, Müller M. Move prediction using deep convolutional neural

networks in hex[J]. IEEE Transactions on Games, 2017, 10(4): 336-343.

[15] Runarsson T P, Lucas S M. Preference learning for move prediction and evaluation function approximation in Othello[J]. IEEE Transactions on Computational Intelligence and AI in Games, 2014, 6(3): 300-313.

[16] Devlin S, Anspoka A, Sephton N, et al. Combining gameplay data with monte carlo tree search to emulate human play[C]//Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment. 2016, 12(1).

[17] Bindewald J M, Peterson G L, Miller M E. Clustering-based online player modeling[M]//Computer Games. Springer, Cham, 2016: 86-100.

[18] Renman C. Creating human-like AI movement in games using imitation learning[J]. 2017.

[19] Spronck P, Ponsen M, Sprinkhuizen-Kuyper I, et al. Adaptive game AI with dynamic scripting[J]. Machine Learning, 2006, 63(3): 217-248.

[20] Tang Z, Zhu Y, Zhao D, et al. Enhanced Rolling Horizon Evolution Algorithm with Opponent Model Learning: Results for the Fighting Game AI Competition[J]. arXiv preprint arXiv:2003.13949, 2020.

[21] Zhou Y, Li W. Discovering of Game AIs' Characters Using a Neural Network based AI Imitator for AI Clustering[C]//2020 IEEE Conference on Games (CoG). IEEE, 2020: 198-205.

[22] Hernndez-Orallo J. AI evaluation: past, present and future[J]. arXiv preprint arXiv:1408.6908, 2014.

[23] Gorman B. Imitation learning through games: theory, implementation and evaluation[D]. Dublin City University, 2009.

[24] Hingston P. The 2k botprize[C]//2009 IEEE Symposium on Computational Intelligence and Games. IEEE, 2009: 1-1.

[25] Khaustov V, Bogdan G M, Mozgovoy M. Pass in Human Style: Learning Soccer Game Patterns from Spatiotemporal Data[C]//2019 IEEE Conference on Games (CoG). IEEE, 2019: 1-2.

[26] Togelius J, De Nardi R, Lucas S M. Towards automatic personalised content creation for racing games[C]//2007 IEEE Symposium on Computational Intelligence and Games. IEEE, 2007: 252-259.

[27] Khalifa A, Isaksen A, Togelius J, et al. Modifying MCTS for Human-Like General Video Game Playing[C]//IJCAI. 2016: 2514-2520.

[28]Zha D, Lai K H, Cao Y, et al. Rlcard: A toolkit for reinforcement learning in card games[J]. arXiv preprint arXiv:1910.04376, 2019.

[29]王政飞. 一种定量评估斗地主游戏 AI 智能水平的方法——蒙特卡洛当量[硕士学位论文]. 北京大学学位论文数据库, 2020.

[30]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 Conference-China. 2017: 1-5.

[31]Li W, Zhou H, Wang C, et al. Teaching AI algorithms with games including Mahjong and FightTheLandlord on the Botzone online platform[C]//Proceedings of the ACM Conference on Global Computing Education. 2019: 129-135.

附录 A　在学期间发表的论文与获得的奖励

本文作者在研究生阶段发表了如下论文：

1. Zhou Y, Li W. Discovering of Game AIs' Characters Using a Neural Network based AI Imitator for AI Clustering[C]//2020 IEEE Conference on Games (CoG). IEEE, 2020: 198-205.（第一作者）
2. Li W, Zhou H, Wang C, et al. Teaching AI algorithms with games including Mahjong and FightTheLandlord on the Botzone online platform[C]//Proceedings of the ACM Conference on Global Computing Education. 2019: 129-135.（第六作者）

并获得了如下奖项：

1. 2021 年搜狐秒针奖学金

附录 B　本人在研期间的其他工作

本文作者在研期间，作为学生主席举办了麻将人工智能比赛（IJCAI 2020 Mahjong AI Competition）。这一比赛由北京大学信息科学技术学院计算机科学技术系网络与信息系统研究所人工智能研究室与国际人工智能联合会议（IJCAI, International Joint Conferences on Artificial Intelligence）联合举办，国际化程度高，面向全球编程爱好者举办的比赛。作者承担前期文书申请、复式赛制引入与游戏环境维护、比赛创建、后期资料整理以及研讨会召开等工作。比赛使用的游戏是中国传统民俗文化游戏——四人麻将，具有三千多年历史，因其较高的趣味性、竞技性、观赏性，传播至世界各地，培养了广泛的群众基础。麻将在传播到其他地区时，也“入乡随俗”，根据当地人喜好改编的规则玩法中，也隐含着当地的人情风貌。

本次麻将比赛在国标麻将规则的基础上，根据平台限制及公平性原则进行了一定的修改。国标麻将是是 1998 年中国国家体育总局制定的一套麻将规则。国标麻将是四人回合制非完全信息游戏，要获得游戏的胜利，需要尽快地让牌符合特定的胡牌牌型，即为“番种”，一共有 81 种番种，不同番种有不同番数。国标麻将的正式比赛一共打 16 局，分坐在东南西北的 4 家轮流做庄，所坐的位置即为门风，4 个玩家的门风各不相同。此外还有一个通用的场风，也是东南西北风中的一个。门风场风与特定番种判定相关。

麻将人工智能比赛的规则在以下方面做出了改动：

- 将打 16 局改为只打一局。
- 借鉴国际麻将联盟（Mahjong International League）的复式赛制，将四人共享牌山改为四人私有牌山，每个人只能从自己的牌山里取牌，这样当四人换位置时，不会因为吃碰杠牌带来取牌的变化。
- 对于被匹配的 4 个 bot，采取全排列座次安排，也即有$Q_1 = A_4^4 = 24$种。
- 在上述 24 种全排列座次安排的基础上，重复 4 次，这 4 次的场风依次为东南西北风，且初始手牌不同，故每次被匹配到的 4 个 bot 会进行$Q_2 = A_4^4 \times 4 = 96$场。
- 匹配计分时，将重复对局分数加起来进行排序，如果分数加和有相同的，则重新进行，直到没有重复分数和的情况，然后按分数和从大到小分别赋以 4、3、2、1 作为匹配分。最后的排名按照匹配分的加和进行排序。

平台实现的国标麻将为四人游戏，对局平均用时为 5 分钟，比赛进行时评测机数量为 32，初赛参赛队伍为 36 支。

$$N_{player} = 4, N = 36, D = 5, P = 32 \qquad (\mathrm{B}-1)$$

循环赛明显不满足技术要求，对于四人游戏，要求每 4 人进行一次匹配，那么将有匹配数$Q_{cycle} = C_{36}^{4} = 58905$，按照技术限制这个值不能超过 40000，故在需要评测的 bot 数量较多、游戏规则需要的人数超过 2 时，一般不采用循环赛。

实际比赛中比赛分为模拟赛（不计分）、初赛、决赛三阶段。

- 模拟赛每周举办，采用效率和 bot 参与度都较高的瑞士轮，方便选手进行 AI 算法测试迭代。
- 初赛为三轮积分赛，采用瑞士轮加复式赛制，按 20%、30%、50%计入初赛成绩，使用Q_1重复数，则每次积分赛的匹配总数及评测总时间为：

$$M_1 = \frac{N * \lceil log_2 N \rceil}{N_{player}} = \frac{36 * 6}{4} = 54 \qquad (\mathrm{B}-2)$$

$$T_1 = \frac{M_1 Q_1 D}{1440P} = \frac{54 * 24 * 5}{1440 * 32} == 0.14(day) = 3.36(hour) \qquad (\mathrm{B}-3)$$

初赛选出前 16 支队伍进入决赛。

- 决赛考虑到比赛奖金设置，分为两个阶段。第一阶段 16 支队伍进行 24 倍标准瑞士轮加复式赛制，使用Q_2重复数，则匹配总数及评测总时间为：

$$M_2 = \frac{N_2 * \lceil log_2 N_2 \rceil}{N_{player}} = \frac{16 * 96}{4} = 384 \qquad (\mathrm{B}-4)$$

$$T_2 = \frac{M_2 Q_2 D}{1440P} = \frac{384 * 96 * 5}{1440 * 32} == 4(day) \qquad (\mathrm{B}-5)$$

第一阶段的前 4 支队伍进入第二阶段，由于此时只有 4 支队伍，故匹配数$M_3 = 1 * 96 = 96$，相应的重复数可以使用较大值$Q_3 = 128$，则评测总时间为：

$$T_3 = \frac{M_3 Q_3 D}{1440P} = \frac{96 * 128 * 5}{1440 * 32} = 1.33(day) = 32(hour) \qquad (\mathrm{B}-6)$$

比赛奖金设置为第一名第一档，第二名到第十六名第二档，对于评估出头部选手的要求比较高，故第一阶段到第二阶段的过程可视为淘汰赛。

从以上赛制可以看出，初赛每轮积分赛都能在一天之内完成测评，让选手能更好更快地进行优化。决赛两阶段之间，不允许选手改变 bot，并能在会议开始前一天完成测评，充分考虑了选手开发和评测公平效率的时间需求。决赛两个阶段的排名（见附图 1

及附图 2）也显示出了赛制的稳定性，进入第二阶段的 4 支队伍排名和第一阶段的前 4 排名一致。

排名	队伍名	账号	排名分
1	SuperJong	yata	1052.01
2	ALONG	yue	991.00
3	清澄高校	gameai_platform	979.00
4	地锅鸡	luyd_cpp	979.00
5	Luma Pools	TheWitness	978.00
6	国士无双	kczno1	973.00
7	雀圣	metaphysics	963.99
8	你能卡掉我我当场把屏幕吃掉	infinityedge	962.99
9	重在参与	工行卡十六号噶	961.00
10	点个大的	humanfy	950.00
11	King of gambler	komqaq	937.99
12	DX	zasfs	936.99
13	Test	huluBrother	935.99
14	自所小分队吧	啊咪咪小熊	928.99
15	just for fun	emmmm	924.99
16	我好菜啊啊	woaixuexi	904.99

附图 1　IJCAI2020 麻将人工智能比赛初赛排名

排名	队伍名	账号	排名分
1	**SuperJong**	**yata**	**1338.00**
2	ALONG	yue	1314.00
3	清澄高校	gameai_platform	1281.00
4	地锅鸡	luyd_cpp	1186.98

附图 2　IJCAI2020 麻将人工智能比赛决赛排名

致谢

本论文的完成，除了我个人的工作安排计划得当，更要感谢我周围的人，给予我最大的帮助与支持、关怀与包容。在北大度过的时间里，研究生的这三年给我留下了不可磨灭的印象。在这个过程中，我作为懵懵懂懂的学术新手，逐渐窥探到了学术殿堂的富丽，为之心折，心向往之。

首先我要着重感谢我的导师，李文新教授，您是我学术之路上的引路人，是我人生道路的启发者。您总以温和而严谨、细致负责的态度对待您的学生，与您讨论的时光是我在实验室中最美好的回忆。您总是很耐心地为我答疑解惑，为了让我学术力更进一步，提了很多很有价值也很有操作性的意见。我从您这里得到的教导，在之后的人生中，我将奉为圭臬。“宝剑锋从磨砺出”，要持续不断地努力，才能有所精进。

我也要大力感谢实验室的师兄师姐、师弟师妹们，总能从你们这里得到关心爱护，在失落时重振信心，在激动时分享喜悦，能玩到一起说到一起，也能共同进步。林舒师兄总能一针见血地指出我目前遇到的问题实质，洪星星师兄经常给我们分享最新科研动态，李昂和鲁云龙时常和我讨论与我的研究相关的问题，帮助我理清思路。还有已经毕业了的师兄们，周昊宇、王政飞、王鑫超，已经毕业了的师姐们，倪燎、张艺，在我刚踏入实验室探究课题的时候，他们给予了我莫大的帮助。

感谢我的舍友，没有你们，我的研究生生涯会失去多少乐趣！虽然我们是不同的研究方向，但你们总能让我了解你们的进展、最前沿技术以及研究趣事。每一天醒来，我带着喜悦的心情投入课题，与你们分享。不可否认，你们极大地丰富了我在读研究生时的精神世界！

感谢我的父母还有我的弟弟在研究生的最后这几个月里，一直在关心我。我们沟通顺畅，父母对我的研究很感兴趣，而我也能给你们讲解我所做的课题。今年弟弟即将小升初，希望你未来也能找到自己喜欢的方向，深入钻研！

最后，我不知要如何道出我的感谢，周昊宇，我的师兄，也是我的男友。回想起那一个个埋头苦干的日子，你在身边始终陪伴我、鼓励我、安慰我，我都快要落泪了。虽然担负着极大的压力，但因为有你，我没有被压垮。这几年，我们都有所成长，互相成为了对方的精神支撑，成为了更好的自己。

如今我即将翻开人生新的一页，再次感谢那些帮助过我、关心过我的可爱人们！我会继续努力，不负青春！

北京大学学位论文原创性声明和使用授权说明

原创性声明

本人郑重声明：所呈交的学位论文，是本人在导师的指导下，独立进行研究工作所取得的成果。除文中已经注明引用的内容外，本论文不含任何其他个人或集体已经发表或撰写过的作品或成果。对本文的研究做出重要贡献的个人和集体，均已在文中以明确方式标明。本声明的法律结果由本人承担。

论文作者签名：周墨杉　　日期：2021 年 5 月 18 日

学位论文使用授权说明

（必须装订在提交学校图书馆的印刷本）

本人完全了解北京大学关于收集、保存、使用学位论文的规定，即：

- 按照学校要求提交学位论文的印刷本和电子版本；
- 学校有权保存学位论文的印刷本和电子版，并提供目录检索与阅览服务，在校园网上提供服务；
- 学校可以采用影印、缩印、数字化或其它复制手段保存论文；
- 因某种特殊原因需要延迟发布学位论文电子版，授权学校□一年/□两年/□三年以后，在校园网上全文发布。

（保密论文在解密后遵守此规定）

论文作者签名：周墨杉　　导师签名：

日期：2021 年 5 月 18 日