基于反绎学习的掼蛋扑克游戏博弈求解

陈仲石, 张申哲, 谭睿睿, 仲崇阳, 欧存勖, 朱天睿, 张绍群*, 周志华*

（计算机软件新技术全国重点实验室, 南京大学, 南京 210023）

摘　要: 掼蛋扑克游戏（简称掼蛋）是一种国内流行的卡牌游戏, 具有状态和动作空间大、规则复杂、非完全信息博弈等特点. 本文基于反绎学习范式（Abductive Learning）开发了一个掼蛋人工智能博弈策略 ABL-GD. 该策略结合基于对局经验的机器学习和基于专家知识、游戏规则等知识的逻辑推理, 实现动作空间约简、出牌动作合规、博弈性能提升. 其主要组件包括: 1）反绎学习网络, 利用对局信息和知识库对其他玩家的手牌信息进行估计; 2）决策模型, 根据知识库对动作空间进行约简, 并利用对局信息及反绎学习网络估计值预测候选动作的概率分布; 3）不一致性最小化器, 根据博弈规则及专家知识, 从候选动作集中选择出最终的输出动作. ABL-GD 策略采用预监督训练+强化学习的方式进行训练. 本文通过博弈实验和消融实验, 验证了 ABL-GD 策略的有效性. 本文收集并标注掼蛋牌谱及人类专家对局, 构建一个专家-牌谱数据集, 在该数据集上的实验验证 ABL-GD 策略已达到了接近人类专家的水平.

关键词: 掼蛋扑克游戏; 反绎学习; 反绎学习网络; 动作约简; 不一致性最小化

中图分类号: TP391　　**文献标志码:** A

Solving Guandan Poker Games by Abductive Learning

CHEN Zhong-Shi, ZHANG Shen-Zhe, TAN Rui-Rui, ZHONG Chong-Yang, OU Cun-Xu, ZHU Tian-Rui, ZHANG Shao-Qun, ZHOU Zhi-Hua**

(National Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210023)

Abstract: Guandan, a popular card game in China, retains the characteristics of a vast state-action space, complex rules, and an imperfect information nature. This paper presents ABL-GD, an AI strategy for playing Guandan. The proposed strategy follows the abductive learning paradigm, effectively combining machine learning based on game data and logical reasoning based on expert knowledge, thereby simplifying the action space, making the card-playing actions compliant, and improving game performance. The ABL-GD strategy comprises (1) an ABductive Learning Network (ABLN) that excels at inferring card information of other players by exploiting game information and knowledge base, (2) a decision-making model that simplifies the action space based on the knowledge base and uses the game information and ABLN estimates to predict the probability distribution of candidate actions, and (3) an inconsistency minimizer that enables interpretable actions, adhere to the playing rules and pre-determined knowledge. We train the ABL-GD strategy using the methodology of pre-supervised training + reinforcement learning. The empirical results from game and ablation experiments verify the effectiveness of the ABL-GD strategy over its competitors. This paper formally builds an expert hand record dataset by collecting and annotating the poker hand records and human expert games. The empirical investigations on this dataset show that the ABL-GD strategy has reached a level close to that of human experts through experiments.

Keywords: Guandan; Abductive Learning; Abductive Learning Network; Action Simplification; Inconsistency Minimization

---

收稿日期: 2025-03-08

基金项目: 江苏省前沿引领技术基础研究重大项目（BK20232003）

第一作者: 陈仲石（2005—）, 男, 本科生, E-mail: 221900070@smail.nju.edu.cn

*__通信作者:__ 张绍群（1992—）, 男, 助理教授, 博士生导师, E-mail: zhangsq@lamda.nju.edu.cn

*__通信作者:__ 周志华（1973—）, 男, 教授, 博士生导师, E-mail: zhouzh@lamda.nju.edu.cn

引用格式: 陈仲石, 张申哲, 谭睿睿等. 基于反绎学习的掼蛋扑克游戏博弈求解[C]. 中国计算机学会人工智能会议. 2025.

Citation: CHEN Zhong-Shi, ZHANG Shen-Zhe, TAN Rui-Rui, et al. Solving Guandan Poker Games by Abductive Learning [C]. Proceedings of the 2025 CCF Conference on Artificial Intelligence. 2025.

棋牌游戏长期以来一直是衡量人工智能算法能力的重要基准. 早期研究主要聚焦于完全可观察的棋盘游戏, 即完全信息博弈 (Complete Information Game), 比如围棋等. 在这类游戏中, 所有玩家均能完全掌握当前游戏状态, 因此核心挑战在于应对庞大的状态空间 (State Space) 并计算最优策略. 例如, Alphago[1] 和 AlphaZero[2] 通过深度学习与蒙特卡洛树搜索的结合, 在围棋游戏中取得了突破性的成功. 随着研究的深入, 棋牌游戏研究的关注点逐渐转向非完全信息的博弈游戏 (Incomplete Information Game), 比如桥牌、德州扑克、麻将等. 这类游戏不仅具有庞大的状态空间, 还要求玩家在信息受限的情况下推测对手的状态与意图, 并做出决策. 近年来, 一系列研究工作在非完全信息游戏中取得了不错的效果, 例如 Libratus[3] 和 Pluribus[4] 在德州扑克战胜人类职业冠军, Suphx[5] 和 AlphaStar[6]、OpenAI-Five[7] 分别在麻将、星际争霸与 DOTA2 中取得超越职业选手的表现.

掼蛋扑克游戏 (简称掼蛋) 是一种国内流行的卡牌游戏. 相较于德州扑克、麻将等非完全信息博弈游戏, 掼蛋更具有挑战性, 具体原因有二: 其一、掼蛋的规则更为复杂, 例如在掼蛋中存在“升级”和“逢人配 (又成百搭)”等概念和玩法, 因此其信息集和动作空间更大[8]. 具体而言, 德州扑克的信息集和动作空间 (Action Space) 的数量级分别为 10^{14} 和 10^{0}, 斗地主的信息集和动作空间的数量级分别为 10^{54} 和 10^{4}, 而掼蛋则是达到了惊人的 10^{118} 和 10^{6}. 近些年来, 有一些研究尝试在博弈策略中对信息集和动作空间进行约简[9-11], 然而掼蛋的规则较为复杂, 其状态和动作空间通常难以约简, 或约简后状态和动作空间仍然很大. 其二、掼蛋是典型的非完全信息博弈游戏, 在使用机器学习方法训练过程中通常仅能利用每个玩家的局部观测值, 而非全局信息, 因而缺乏对对局全局信息的良好评估. 一些学者尝试使用带中心化价值函数的独立策略[12] (Independent Actor with Centralized Critic) 来缓解该问题. 每个智能体独立地决定自身的行动策略, 但共用一个中心化价值函数. 因此其输入包括所有智能体的观测信息, 从而对整体局势进行更好的评估. 然而一些测试[13-15] 表明一个中心化价值函数不一定能够改善合作, 相反会导致分散化行动策略的方差更高, 因而导致策略学习不稳定和较差的表现.

职业玩家在掼蛋游戏中通常采用两阶段的博弈过程: 首先, 根据当前牌面和游戏规则来估计其他玩家的牌型, 即残局信息; 然后, 根据牌面估计选择最有可能的策略进行出牌. 第一阶段是基于游戏规则的反绎推理过程, 而第二阶段可以通过机器学习方法进行预测来完成. 此决策流程与反绎学习[16](Abductive Learning) 非常契合. 反绎学习是一种以均衡互促方式融合机器学习和逻辑推理的学习范式, 其中逻辑推理可以有效约束学习中的信息集, 并为学习结果提供解释, 而机器学习可利用数据有效拟合博弈策略的状态和动作分布, 其输出可以被约束为与知识库 (Knowledge Base, KB) 一致[17]. 因此, 利用反绎学习开发掼蛋博弈策略时, 基于掼蛋游戏规则和专家经验的推理可以有效缩小掼蛋的信息集和动作空间, 从而对残局信息进行有效估计, 而机器学习可以有效利用残局信息和当前牌面预测出牌.

本文提出了一个掼蛋博弈策略 ABL-GD, 其整体框架如图 1 所示. 该策略采取反绎学习范式, 在传统基于对局数据进行机器学习预测的基础上, 有效利用专家知识和游戏规则等知识, 从而约简动作空间、确保出牌动作合规、提升博弈性能. 该策略由三个主要部分组成: 1）反绎学习网络 (ABductive Learning Network, ABLN), 根据对局情况和掼蛋知识库, 对包含其它玩家手牌信息在内的全局信息进行估计; 2）决策模型, 根据知识库对动作空间进行约简 (Action Simplification), 即将掼蛋出牌动作约简为抽象的伪动作 (Pseudo Action), 每个伪动作对应一个候选动作 (Candidate Action) 集合, 在此基础上对伪动作概率分布进行预测; 3）不一致性最小化器 (Inconsistency Minimizer), 根据最小化与知识库的不一致性原则, 从伪动作对应的候选动作集中推理出最终的输出动作.

实验阶段, 本文将 ABL-GD 与 4 个公开策略进行博弈实验, 实验结果验证了该策略的有效性. 本文通过收集并标注掼蛋牌谱及人类对局数据, 构建一个专家-牌谱数据集, 用以评估 ABL-GD 策略与人类专家的水平差距. 在该数据集上的实验结果表明 ABL-GD 达到了接近人类类专家的水平. 本文对 ABLN 进行了消融实验. 实验结果表明 ABL-GD 在 500 局的对弈中平均取得 60.9% 的胜率, 在小局的对弈中累计净得分 740 分, 验证了 ABLN 的有效性.

本文第 1 节简介掼蛋、博弈策略研究现状和反绎学习; 第 2 节介绍本文提出的 ABL-GD 策略的整体框架和实现细节; 第 3 节通过设计实验验证了 ABL-GD 的有效性; 第 4 节总结本文.

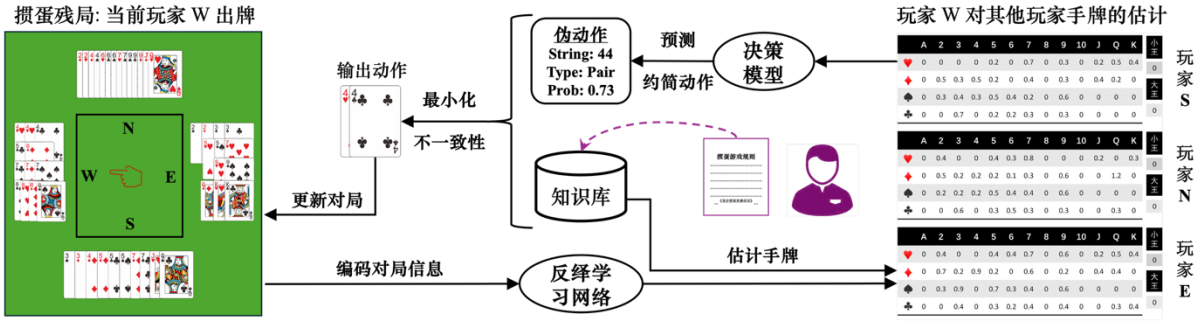


图 1　ABL-GD 策略的整体流程

Fig.1　Overview of ABL-GD

1　相关工作

1.1　掼蛋扑克规则介绍

掼蛋是一种国内流行的卡牌游戏. 在一局掼蛋比赛中, 4 个玩家使用 2 副牌, 共计 108 张, 与对家组成队伍采取 2 对 2 的形式进行博弈. 本文采用 2017 年国家体育总局发布的《淮安掼蛋竞赛规则》中的掼蛋规则, 下面进行简要介绍.

1）牌点和牌型. 在不考虑级牌的情况下, 掼蛋中牌点从小到大依次为: 2、3、4、5、6、7、8、9、10、J、Q、K、小王、大王, 牌型有: 单张、对子、三张、钢板、三带二、三连对、顺子、同花顺、炸弹、王炸. 其中 A 在搭配成三连对、二连三、顺子、同花顺时, 可视作 A 也可以视作 1, 视作 1 时 A 是牌点最小的牌, 例如 A2345 是最小的顺子. 其中炸弹张数越多则越大, 同样张数按照点数排序; 同花顺大于任意不超过 5 张的炸弹; 王炸可以覆盖任意牌型. 以 H、S、C、D 表示花色红心、黑桃、草花、方块.

2) 升级和逢人配. 掼蛋的一整局博弈可以分为若干小局, 每小局结束后, 仅第一个出牌完成的玩家所在的队伍可以升级: 如果队友是二游升 3 级; 为三游升 2 级; 为末游升 1 级. 每小局根据双方队伍等级中最高的决定当前牌局等级, 和当前牌局级别相同的牌被称为级牌, 级牌的牌点仅比小王和大王小, 例如若一局等级为5, 则所有牌点的大小从小到大依次为: 2、3、4、6、7、8、9、10、J、Q、K、5、小王、大王. 其中红桃级牌被称为逢人配, 可以作为除大小王以外的任意牌与其它牌组成牌型. 掼蛋对局中初始等级为2, 当一方队伍到达等级 A, 且在三局中一个玩家为头游, 另一个玩家为二游或三游, 则获胜; 若累计三局不双上则降级至等级 2, 其中等级 A 必打, 升级不可跳过等级 A.

3）进贡和抗贡. 从第 2 小局开始, 上一轮取得下游的玩家需要向上游的玩家进贡一张逢人配以外的最大的牌, 同时上游玩家向下游玩家还贡一张不大于 10 的牌, 如果上一局同一个队伍的玩家分别为三游和末游, 则需要向对方队伍分别进贡和还贡. 如果进贡方有两张大王, 则可以抗贡.

掼蛋由于使用两副牌进行对抗, 且有逢人配、升级等复杂设置, 因此信息集空间比其他扑克博弈游戏显著复杂, 这为掼蛋的博弈策略相关研究带来了挑战.

1.2　掼蛋游戏博弈算法研究现状

目前针对扑克游戏的博弈策略大致有如下两类: 1) 基于反事实遗憾最小化[9] (Counter Ractual Regret minimization, CFR) 的方法[3-4,10], 这类方法对游戏树进行深度优先搜索以计算策略的遗憾值, 通过最小化遗憾近似纳什均衡. 面对信息集状态和动作空间较大的情况, 这类方法通常需要对状态和动作空间进行约简[11]. 然而掼蛋的规则较为复杂, 其状态和动作空间通常难以约简, 或约简后状态和动作空间仍然很大. 2) 基于强化学习的方法[1-2, 5-8], 该方法使用深度神经网络拟合状态动作值函数或状态动作概率分布, 并在试错中收集奖励以更新网络. 然而, 由于掼蛋的动作空间相当大, 对 GPU 的显存需求较大[8], 且部分强化学习算法在大动作空间的条件下也难以对值函数进行较好的拟合. 对此, Douzero 提出深度蒙特卡洛[11] (deep Monte Carlo, DMC) 方法. 该方法在单个时间步的推理过程中使用价值网络对每个可选动作的值函数进行评估, 将观测信息和单个可选动作的编码信息结合一并作为神经网络的特征输入, 输出对该合法动作的值函数估计值, 通过蒙特卡洛采样对状态动作值函数进行拟合. 然而, 因为这种方法在每个时间步的推理中需要对所有可选动作逐个评估, 因此需要非常大的推理时间和训练开销. 综上, 受限于掼蛋复杂的游戏

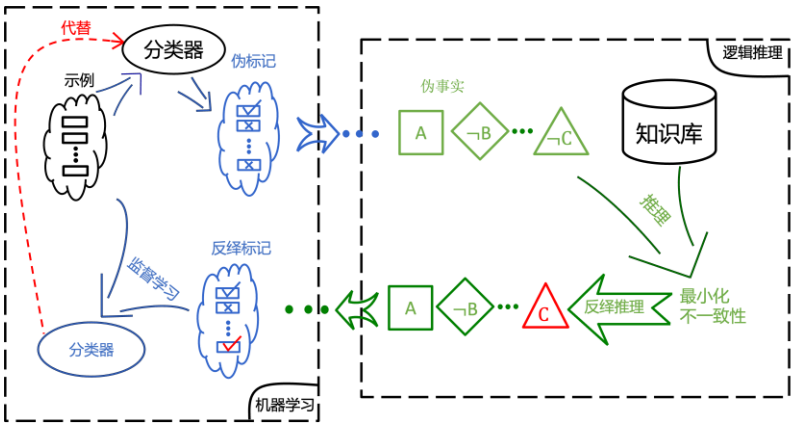


图 2　反绎学习范式[16]

Fig.2　Overview of Abductive Learning[16]

规则和大数量级的信息集空间，大多数现有的博弈策略在掼蛋中难以直接复用.

当前掼蛋博弈策略关注于如何处理掼蛋庞大的状态和动作空间. Danzero[8] 率先将 DMC 方法引入到掼蛋，实验结果表明基于强化学习的掼蛋博弈策略远优于基于启发式规则的博弈策略. 为了缓解 DMC 方法训练时间长的问题，葛等人[18]提出软深度蒙特卡洛，通过已有策略知识对模型进行软启动来减少 DMC 方法的训练时间，并通过软动作采样缓解 DMC 方法仅选择最大值动作的问题，但这种方法仍然需要较长的训练时间才能收敛. Danzero+[19] 通过结合 DMC 方法和 PPO 算法缓解 DMC 方法探索不充分的问题. 该方法根据预训练好的 DMC 模型从动作空间中筛选出估计值函数最大的 k 个动作来为 PPO 算法提供一个精简的动作空间，并由 PPO 算法从中选择出最后的动作，可以在一定程度上缓解 DMC 方法探索不充分的问题.

综上可知，现有掼蛋的博弈策略主要集中于如何使用强化学习等方法处理庞大的信息集状态和动作空间，而鲜少关注如何对全局信息数据进行有效推断以及利用专家知识对策略进行修正，因而往往会出现由于动作空间过大导致的计算复杂度高、博弈性能低等问题. 为了解决上述挑战，不仅需要依赖数据驱动的机器学习来选择最有可能的策略进行出牌，而且需要结合游戏规则和专家知识对信息集和动作空间进行有效约简和估计.

1.3　反绎学习

反绎学习是人工智能领域以均衡互促方式融合机器学习和逻辑推理的一种学习范式[16-17]. 与传统基于大量数据和标记 (Label) 的学习方式不同，反绎学习泛式融合了基于数据的学习和基于知识的推理. 反绎学习通常包含两个核心组件，如图 2 所示，即用于生成初步预测的神经网络等机器学习模型，以及提供逻辑规则与推理机制的知识库模块.知识库可以以命题逻辑、一阶逻辑、数学公式等多种形式呈现，具备可执行的符号推理能力.

反绎学习的基本流程如下. 先由机器学习模型将原始特征数据 $\boldsymbol{x} \in \mathrm{X}$ 映射为离散符号 $\boldsymbol{z} \subseteq \mathrm{Z}$，其中 X 和 Z 分别为学习模型的输入空间和输出空间，离散符号 $\boldsymbol{z}$ 常被称作伪标记. 然后由一个不一致性最小化器接收符号序列 $\boldsymbol{z}$，并基于一阶逻辑规则的知识库 KB 通过逻辑推理得到符合领域知识的最终输出 $y \in \mathrm{Y}$. 在推理阶段，不一致性最小化器在接收伪标记 $\boldsymbol{z}$ 后，根据游戏规则或者知识库 KB 将 $\boldsymbol{z}$ 修改为候选标记 $\overline{\boldsymbol{z}}$. 由于知识库 KB 可能存在多条满足逻辑约束的推理路径，通常存在多个候选标记 $\overline{\boldsymbol{z}}$，例如在本文提出的 ABL-GD 策略中，对决策模型预测出的一个伪动作可以通过 KB 推理出多个候选动作. 当存在多个反绎标记时，反绎学习通过最小化反绎标记和知识库之间的不一致性推理出最佳的反绎标记作为最终的输出 y. 反绎学习的基本流程可通过 Python 开源工具箱 ABLkit 实现[20]. 该工具箱提供了高灵活性、易用的接口，在速度和性能上均远超现有的神经符号学习方法.

反绎学习范式具有高度的通用性和灵活性：1) 不仅可以使用图像和文本分类器，还能采用如基于集成学习的随机森林等机器学习模型. 2) 知识库不局限于一阶逻辑和加法规则，也适用于法律规则[21]，甚至是互联网上现有的非形式化大规模

知识图谱[22]. 反绎学习知识库中的概念可通过数据驱动的方式进行动态增补[23]. 3) 反绎学习适用多种学习任务. 与半监督学习 (Semi-Supervised Learning) 结合的反绎学习可有效利用符号知识和未标记数据以提升性能[21]; ABLFast 提出了一种针对反绎学习的新颖一致性度量和拒绝推理策略, 有效地减少对标记数据的需求[24-25]. 此外, 反绎学习中不一致性最小优化器通常使用基于零阶优化的离散搜索方法, 故而计算成本随问题规模的增长呈指数级上升. ABL-Refl 通过引入反思机制[26], 利用领域知识快速识别并修正神经网络输出中的潜在错误, 从而提升反绎学习效率, 并荣获 AAAI 2025 Outstanding Paper Award.

2 基于反绎学习的掼蛋游戏博弈框架

本节介绍 ABL-GD 策略的整体框架, 如图 1 所示. 该策略在一个时间步的推理过程中主要包括五个步骤: 信息编码、估计手牌、预测伪动作、最小化不一致性和更新状态. 该框架首先对玩家可见的对局信息进行编码, 随后反绎学习网络根据编码后的状态信息和知识库估计各玩家的手牌统计量, 决策模型在此基础上对伪动作分布进行预测, 最后不一致性最小化器根据知识库从伪动作对应的候选动作集中推理出最终出牌动作, 并根据输出动作更新对局状态. 本文中所涉及掼蛋知识库包含两部分内容, 其一为掼蛋的游戏规则, 其二为总结的人类专家游戏偏好.

2.1 信息编码

掼蛋的对局信息通常包括两部分: 1) 状态信息, 包括手牌、等级、所有玩家已出卡牌等, 2) 所有玩家的历史动作信息. 本文使用一个 54 维向量指示掼蛋扑克游戏中的状态信息和出牌动作, 其中包括 4 种花色 SHDC, 13 个点数和大小王. 向量中每个位置的取值为 0、1 或 2, 表示该位置上的扑克牌的数量. 为了方便展示, 本文通常采用一个 2 维索引指示向量, 比如红心 2 的位置维索引为 H2 , 即该向量的第 6 个位置; 若索引 H2 位置上的数值为2, 则表示该玩家持有或者打出两张红心 2 的扑克牌. 本文使用 4 个 54 维的向量来记录对其他玩家手牌信息的估计概率. 其余对局信息, 包括当前对局等级、玩家编号、逢人配数量等, 的编码均采用一位标量进行有效编码.

根据是否与对局序列有关, 本文将掼蛋扑克牌局的对局信息分为两部分: 序列信息 o_t 和非序列信息 a^i, 如表 1 所示, 其中当前对局记为 t, i 指示玩家编号. ABL-GD 接收的对局信息为当前状态信息 $o_t \in O_t$ 和一定对局步长内智能体的历史动作信息 $\tau_t^i = (a^i, o_{t-7}^i, \dots, o_{t-1}^i) \in T^i$. 在步骤 1 信息编码阶段, 本文采用一个 4 层的全连接神经网络将当前状态信息 o_t 处理为编码向量 $h_o \in R^{d_o}$, 并采用一个 6 层的注意力网络[27]作为编码器, 将 τ_t 处理为编码向量 $h_\tau \in R^{d_\tau}$.

表 1 ABL-GD 状态编码

Tab.1 State encoding of ABL-GD

类型	含义	特征维度
序列信息	历史 7 步出牌动作	7×54
	历史 7 步出牌者编号	7×4
非序列信息	当前对局等级	13
	玩家编号	4
	逢人配数量	3
	手牌的矩阵表示	54
	四家剩余手牌数量	4×54
	四家已出卡牌矩阵表示	4×54

2.2 反绎学习网络与手牌估计

反绎学习网络 ABLN 以两个编码向量为输入, 拼接为 $h_o \oplus h_\tau \in R^{d_o+d_\tau}$, 估计其他玩家的手牌信息. 本文采用一个 432-512-512-512-162 架构的全连接神经网络 $f_{\text{abln}}: R^{d_o+d_\tau} \to R^{3\times 54}$ 来实现 ABLN, 其中 3 表示其他玩家的数量, 每个隐藏层以 ReLU 函数激活, 最后一层采用 Sigmoid 函数作为激活函数. 该网络的预测结果需经过放缩 $\hat{y} = 2 * \text{Sigmoid}(f_{\text{abln}}(h_o \oplus h_\tau))$, 以确保在合法范围内.

算法 1 ABLN 的计算过程

Alg.1 Computational procedure of ABLN

输入: 当前时间步智能体 i 的观测信息, 一定时间步长内的历史对局信息

输出: 对其他玩家手牌统计量的估计结果

for episode = 1 to max_episodes
  for t = 0 to T
    $o_i \leftarrow$ 智能体 i 观测的当前状态信息
    $\tau_t^i \leftarrow$ 智能体 i 观测的历史动作信息
    $y^i \leftarrow$ 其他玩家的真实手牌标记信息
    前向传播计算手牌估计值 $\hat{y}$
    根据标记 y 计算损失 $\text{MSELoss}(\hat{y}, y)$
    反向传播梯度更新
  end for
end for

ABLN 通过监督学习进行训练, 其计算流程如算法 1 所示. 本文采用均方误差损失 (Mean Squared Error, MSE) 作为损失函数, 即算法 1 中的 MSELoss$(\hat{y}, y)$, 其中 y 表示其他玩家的真实手牌.在实际训练中, 通常要先对 ABLN 进行预训练以实现 ABL-GD 整体训练的快速收敛. 由于目前掼蛋缺少良好的对局数据集, 本文使用随机策略的博弈数据作为 ABLN 的预训练数据.

2.3 约简和伪动作预测

决策模型用于预测玩家伪动作的概率分布, 其预测流程如图 1 所示. 决策模型接收编码信息及反绎学习网络的估计结果, 对伪动作概率分布进行预测. 决策模型由一个行动策略和一个价值函数构成, 其中行动策略输出对伪动作概率分布 $\pi_\theta(a^i|\boldsymbol{s})$ 的预测结果, 价值函数输出状态价值 $V_\theta(\boldsymbol{s})$ 的估计值, $\boldsymbol{s}$ 指示伪动作. 决策模型采用 PPO 算法[28] 进行训练. 注意, 为了增强预测性能和加速收敛, 决策模型的输入不仅包括反绎学习网络 ABLN 输出的手牌估计值, 还包括当前状态信息 o_t 和历史动作信息 τ_t^i.

掼蛋的动作空间巨大, 其数量级达到了 10^6, 这导致了博弈中搜索成本高、计算开销大的问题, 并要求决策模型的输出维度必须涵盖完整的动作空间. 虽然多数情况下可选动作集合远小于动作空间大小, 但是批量计算的成本在实际应用中仍不可接受[8].

表 2 ABL-GD 伪动作设计

Tab.2 Pseudo actions of ABL-GD

牌型	实例	动作维度
Single	A	15
Pair	22	15
Triple	333	13
Plate	333444	13
Tube	223344	12
Full House	999JJ	13
Straight	34567	10
Bomb	4444	65
Straight Flush	\	10
Joker Bomb	\	1
PASS	\	1

为解决掼蛋游戏动作空间规模大的问题, 本文采取反绎学习范式, 将动作空间约简为抽象的伪动作标记, 由决策模型计算伪动作的概率分布. 具体而言, ABL-GD 策略将同一牌型同一牌点的所有出牌动作压缩到同一个伪动作标记. 以图 1 为例并考虑等级 8 的情况下, 可以实现伪动作“44, Pair”的出牌包括 H4H4、H4D4 等不同花色的动作, 以及 H4H8 等使用逢人配的动作, 共计数十种出牌结果. 本文统称这数十种出牌动作为候选动作. 实际上出牌的花色仅影响同花顺的保有, 对大多数出牌结果影响不大, 且多数状态下玩家手中的逢人配数量为 0. 因此本文认为掼蛋的动作空间存在极大程度的冗余, 且该冗余主要由花色和逢人配导致. 根据出牌动作将多个候选动作约简为一个伪动作可以极大减少花色和逢人配对动作空间大小的影响, 约简后的伪动作空间只包含 168 个标记, 如表 2 所示.

2.4 最小化不一致性

本小节设计不一致性最小化器, 即根据输出动作与知识库的不一致性最小化原则, 从伪动作对应的候选动作集合中推理出 ABL-GD 的输出动作. 该最小化不一致性的过程可被建模为如下优化问题

$$\begin{aligned} \max_{\bar{\boldsymbol{z}}} \quad & Score(\bar{\boldsymbol{z}}) \\ \text{s.t.} \quad & Consistent(KB \cup \{\bar{\boldsymbol{z}}\}) \end{aligned}$$

其中 $Score(\hat{\boldsymbol{z}})$ 表示推理模型根据知识库 KB 分配给伪动作 $\boldsymbol{z}$ 对应候选动作 $\bar{\boldsymbol{z}}$ 的分数, $Consistent(KB \cup \{\bar{\boldsymbol{z}}\})$ 确保候选动作 $\bar{\boldsymbol{z}}$ 符合知识库 KB.

不一致性最小化器的目标在于识别出一个符合知识库且得分最高的候选动作 $\bar{\boldsymbol{z}}$ 作为 ABL-GD 的输出动作. 具体流程如下: 不一致性最小化器在接收到伪动作标记 $\boldsymbol{z}$ 的概率分布后, 选取其中概率最大的伪动作, 对其对应的候选动作进行打分. 对不满足游戏规则的候选动作设置分数为最低分, 满足游戏规则的候选动作根据专家知识设定的规则给出分数, 最终选出分数最高的候选动作作为最终的结果. 若概率最大的伪动作对应的候选动作均为最低分, 即均为不满足游戏规则的非法动作, 则在剩下的伪动作中选取概率最高伪动作再次进行最小化不一致性推理.

如图 3 所示例子，决策模型预测出了概率分布最高的伪动作“44, Pair”, 而其对应的候选动作有 H4H4、H4D4 等不同的出牌, 其中满足掼蛋游戏规则的仅有 H4H4、H4C4 两种候选动作. 不一致性最小化器将根据游戏规则和当前玩家所处持

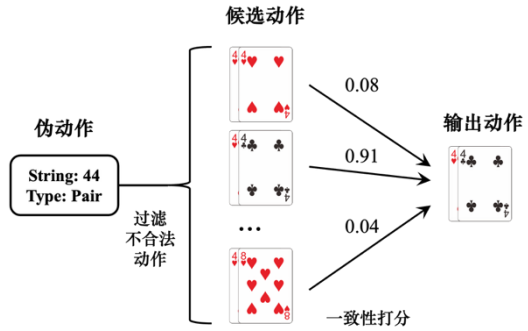


图 3　不一致性最小化示例

Fig.3　An example of inconsistent minimization

手牌过滤掉不合适的候选动作，并根据专家知识对两种合适的候选动作分别进行打分，最终选择分数更高的 H4C4 作为策略输出的动作结果.

2.5　更新和结算

将不一致性最小化器推理出的输出动作作为 ABL-GD 策略的输出动作，即可更新对局状态.

接下来介绍 ABL-GD 掼蛋策略的结算. 当每小局结束、新的小局开始时，需要根据掼蛋规则进行进贡、还贡环节，ABL-GD 策略的进还贡模块采用第一届“中国人工智能博弈算法大赛”的冠军算法使用的规则设计. 小局结束时，对智能体给予奖励. ABL-GD 的奖励设计与掼蛋的升级方式相同. 本文设计的奖励机制如表 3 所示.

表 3　ABL-GD 奖励设计

Tab.3　Reward functions of ABL-GD

完牌顺序	队伍 1 奖励	队伍 2 奖励
1—1—2—2	+3	−3
1—2—1—2	+2	−2
1—2—2—1	+1	−1
2—1—1—2	−1	+1
2—1—2—1	−2	+2
2—2—1—1	−3	+3

本文以自博弈的方式对 ABL-GD 进行训练，博弈策略整体流程如算法 2 所示，其中决策模型中的子网络均采用 PPO 算法[28] 进行训练.

2.6　ABL-GD 的实战示例

本小节给出了 ABL-GD 策略在某实战对局种的示例. 该示例处于等级 8，残局牌面如图 4(a) 所示，当前轮到玩家 W 出牌. 玩家 W 采用 ABL-GD 策略. 根据策略流程，ABL-GD 编译玩家 W 的手牌、所有玩家已出牌信息、过去 7 步的出牌动作以及其他对局信息作为策略输入，并利用知识库估计其他玩家剩余手牌. 估计结果被呈现为三个子表格，每个表格记录了 54 张牌仍被某玩家持有的概率，如图 4(b) 所示.

由于逢人配、复杂规则等因素，以往 AI 策略通常很难对残局牌面实现精准估计. 然而，人类玩家在博弈时通常可以比较精准地判断某些关键手牌在某个玩家手中. 根据图 4(b)，易发现 ABLN 对其他玩家手牌的估计相对比较准确，比如对手玩家 N 中 D2、H2、D4、H7、S9、HQ，合作玩家 E 中 D2、S3、S5、H7、S9、H9、HK、CK 等. 可见，反绎学习的引入提升了策略的可解释性，且 ABLN 的估计值比较符合真实手牌规律，极大地提升了后续预测的准确率. 值得一提的是，ABLN 的估计值也存在一些误差，该误差随着对局的进程将不断降低.

算法 2　基于反绎学习的掼蛋博弈策略伪代码

Alg.2　Pseudocode of the ABL-GD strategy

输入： ABNL 参数 $\{A_i\}_{i=1}^n$，行动策略参数 $\{\pi_i\}_{i=1}^n$，价值函数参数 $\{V_i\}_{i=1}^n$，经验缓存 $\{B_i\}_{i=1}^n$ 和 $\{D_i\}_{i=1}^n$

输出： 策略参数

初始化经验缓存为空

随机初始化 ABNL 参数、行动策略和价值函数参数

for episode = 1 to max_episodes

  for t = 0 to T

    $o_i \leftarrow$ 智能体 i 观测的当前状态信息

    $\tau_t^i \leftarrow$ 智能体 i 观测的历史动作信息

    $y^i \leftarrow$ 其他玩家的真实手牌标记信息

    ABLN 前向传播计算手牌估计值 $\hat{y}$

    选取伪动作 $\boldsymbol{z}^i \sim \{\pi_i\}_{i=1}^n$

    知识库推理出最终动作 $\overline{\boldsymbol{z}}^i$

    存储样本 $\{\tau_t^i, \overline{\boldsymbol{z}}^i, y^i\}$ 至经验缓存 $\{B_i\}_{i=1}^n$

  end for

  获得环境奖励 $r = (r_1, r_2, ..., r_n)$

  for i = 1 to n

    for $\{\tau_t^i, \overline{\boldsymbol{z}}^i, y^i, r^i\}$ in B_i

      存储 $\{\tau_t^i, \overline{\boldsymbol{z}}^i, y^i, r^i\}$ 进 D_i

    end for

  end for

  清空经验缓存 B_i

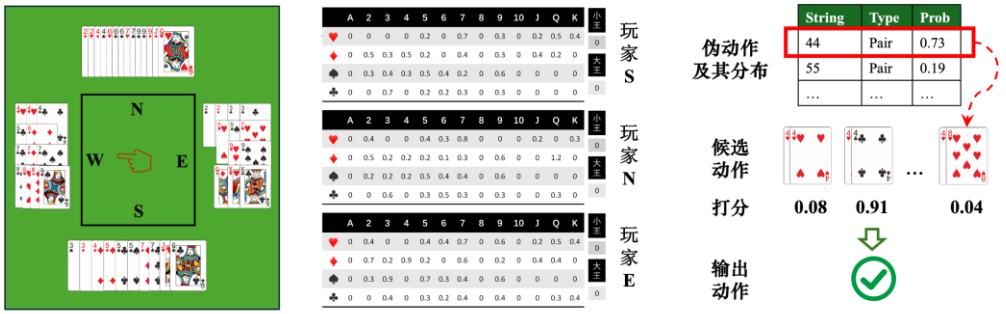
  while $D_i.length > batch\_size$

    从 D_i 采样，监督学习更新 $\{A_i\}_{i=1}^n$

    从 D_i 采样，更新 $\{\pi_i\}_{i=1}^n$ 和 $\{V_i\}_{i=1}^n$

  end while

end for



（a）当前对局记录　　（b）对其他玩家手牌估计　　（c）预测及决策

图 4　ABL-GD 策略对局示例

Fig.4　Empirical example of ABL-GD

根据上述对局信息和手牌估计, ABL-GD 策略将对约简之后的动作实现预测. 如第 2.3 小节所述, 决策模型的输出为伪动作及其分布. 当前示例下的伪动作及其分布如图 4(c) 所示. 首先选择最大可能的伪动作 “44, Pair”, 并查找其对应的候选动作. 通过知识库和玩家 W 的手牌过滤, 当前可操作的候选动作有 H4H4、H4C4、H8H4、H8C4. 接下来, ABL-GD 策略通过最小化不一致性对候选动作进行打分 (以概率形式呈现). 此处, H4C4 的分值最大. 因此, 玩家 W 采用 ABL-GD 策略, 打出 H4C4 作为输出动作.

表 4　ABL-GD 超参数配置

Tab.4　Hyperparameters in ABL-GD

超参数名称	含义	数值
lr_rl	强化学习学习率	1e-4
lr_sl	监督学习学习率	4e-6
Optimizer	优化器	Adam
buffer size	经验回放缓冲区大小	32768
batch size	批量大小	16384
train freq	两次更新间采样时间步数量	16384
MLP layer	多层线性网络层数	4
MLP node	多层线性网络隐藏层维度	512
d_{attn}	注意力网络隐藏层维度	256
d_{τ}	注意力网络输出特征维度	216
d_o	观测信息全连接网络输出特征维度	216
E	PPO 算法裁剪系数	0.2
γ	GAE 奖励折扣因子	0.996
λ	GAE 衰减系数	0.98
\	可选伪动作数量	168
\	最大采样时间步数	4.0×10^8

3　实验

本节通过三组实验系统评估 ABL-GD 掼蛋博弈策略的有效性, 其超参数和相关配置如表 4 所示. 第 3.1 小节将 ABL-GD 策略和 4 个基于启发式规则的掼蛋博弈算法以及随进行对局测试, 以验证 ABL-GD 策略的有效性. 第 3.2 小节构建了一个专家-牌谱数据集, 在该数据集上的实验验证 ABL-GD 策略已接近人类专家的水平. 第 3.3 小节对反绎学习网络 ABLN 进行消融实验, 将本文提出的策略与不含 ABLN 的同结构策略进行博弈测试, 以验证 ABLN 对于 ABL-GD 策略的影响.

3.1　博弈测试

本小节将 ABL-GD 与现有的开源掼蛋策略进行博弈测试以测试其性能. 具体而言, 本小节将 ABL-GD 与四个来自第一届“中国人工智能博弈算法大赛”的基于启发式规则的策略进行对抗, 在本小节中分别记为 BSL1、BSL2、BSL3、BSL4.

表 5　ABL-GD 与其他掼蛋策略博弈对抗的胜率

Tab.5　Win rate of ABL-GD against contenders

胜率	对抗算法					
(%)	random	BSL4	BSL3	BSL2	BSL1	Ours
random	\	0	0	0	0	0
BSL4	100	\	25.4	1.2	0.5	0
BSL3	100	74.6	\	0.4	0	0
BSL2	100	98.8	99.6	\	45.2	14.8
BSL1	100	99.5	100	54.8	\	18.6
Ours	100	100	100	85.2	81.4	\

具体的博弈测试方案设置如下: 每个策略单独运行两个智能体实例组成一支队伍, 每两支队伍之间进行 1000 局测试, 统计博弈胜率. 博弈实

验结果如表 5 所示. 观察易知,ABL-GD 对战基于启发式规则算法的博弈胜率远超 50%、平均可以达到 80%以上. 这表明相较于基于启发式规则的掼蛋策略, 本文提出的 ABL-GD 策略更加有效.

3.2 专家-牌谱数据集

为评估 ABL-GD 与人类专家的水平差距, 本文基于人工收集, 根据掼蛋牌谱书籍及高水平业余人类玩家对局数据, 构建了一个专家-牌谱数据集. 该数据集包括 200 盘对局记录, 对局记录包括手牌、等级、对局胜负及得分、出牌记录等信息, 并根据人类专家意见对其中每个出牌动作的水平进行标注, 共计 15645 个有效动作标记.

本小节在此数据集上设计实验, 测试本文提出的 ABL-GD 策略与人类专家的水平差距. 测试方案如下: 策略在任意环节介入历史对局, 对下一步的出牌进行预测, 并将预测结果与数据集中人类专家的真实对局结果进行比较. 为了衡量策略是否能够获得与人类专家相近的对局结果, 本小节采用单步动作预测准确率和对局得分预测准确率来衡量 ABL-GD 策略与人类专家的水平差异. 通常当策略出牌与专家出牌的牌型和牌点一致时, 认为策略成功预测专家出牌动作. 由于掼蛋动作空间和随机性较大, 很多情况下一手出牌就可以造成局势的巨大变化, 因此对局得分和专家动作预测的实际准确率通常难以达到 100%.

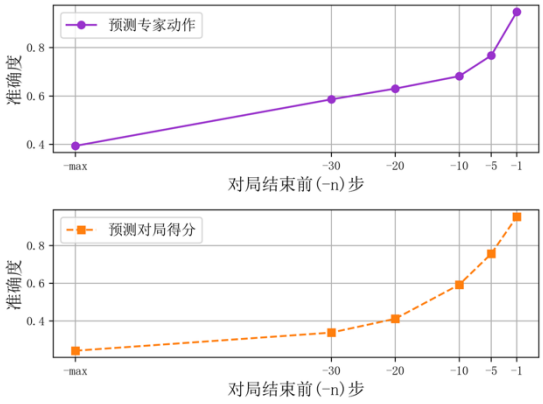


图 5 ABL-GD 策略的预测准确度变化曲线

Fig.5 Accuracy curves of ABL-GD on the dataset

测试结果如图 5 和表 6 所示. 观察易知, ABL-GD 策略的介入节点越接近对局结束, 对人类专家出牌动作以及对局得分的预测精度就越高. 当距离对局结束还有 30 步时, ABL-GD 对专家动作的预测准确率已超过 50%; 当距离对局结束还有 20 步时, 策略对专家动作的预测准确率能够达到 63.05%, 对得分的预测准确率达到 41.24%; 对局前一步时, 策略对专家动作的预测准确率能够达到 94.79%, 对得分的预测准确率达到 95.37%. 该实验结果说明 ABL-GD 的决策结果与人类专家比较接近, 进而说明该策略能够达到与人类专家相近的水平.

表 6 ABL-GD 策略在专家-牌谱数据集上的预测准确度

Tab.6 Prediction accuracy of ABL-GD on the dataset

对局结束前 -n 步	预测准确度 (%)	
	专家动作	对局得分
-max	39.36	24.29
-30	58.60	33.89
-20	63.05	41.24
-10	68.19	59.32
-5	76.74	75.70
-1	94.79	95.37

3.3 ABLN 消融实验

本小节通过消融实验验证反绎学习网络 ABLN 的有效性. 实验方案设置如下: 首先训练一个不含 ABLN 的 ABL-GD 策略, 命名为 ABL-GDoABLN, 该策略直接通过局部观测信息进行出牌预测, 然后将两种策略进行博弈测试.

表 7 ABL-GD 与 ABL-GDoABLN 对抗博弈的胜率

Tab.7 Win rate of ABL-GD vs ABL-GDoABLN

算法	对抗算法	胜率 (%)
ABL-GD	ABL-GDoABLN	60.9±0.7
ABL-GDoABLN	ABL-GD	39.1±0.7

本小节首先将两个策略在 500 局完整对局上进行博弈测试, 统计其胜率, 并重复 5 次测试以计算平均胜率和标准差. 实验结果如表 7 所示, ABL-GD 与 ABL-GDoABLN 对战, 在完整对局上的博弈胜率平均为 60.9%. 该实验结果说明有 ABLN 的策略性能要好于没有 ABLN 的策略, 验证了反绎学习网络 ABLN 估计牌局对 ABL-GD 策略性能有益.

表 8 ABL-GD 与 ABL-GDoABLN 对抗博弈的累计得分

Tab.8 Cumulative score of ABL-GD vs ABL-GDoABLN

分值及比例	ABL-GD	ABL-GDoABLN
累计得分	5181	4441
3 分占比 (%)	40.3	18.6
2 分占比 (%)	27.5	39.7
1 分占比 (%)	32.2	41.7

此外, 本小节还测试了两种策略在 5000 小局对抗中的累计得分表现, 并统计两种策略的累计得分和小分占比情况. 实验结果如表 8 所示. 观

察易知, 1）ABL-GD 对战 ABL-GDoABLN 的博弈小局胜率接近 50%; 2）在获胜的小局当中, ABL-GD 获得 3 分, 即双上 (队伍分别以上游和二游完牌) 的占比明显提高, 因此 ABL-GD 在累计得分上远超过 ABL-GDoABLN. 该实验结果表明反绎学习网络 ABLN 有助于策略在完整对局上以更快的速度“升级”, 因而使得策略在完整对局上有更强劲的表现.

4 总结

本文提出了一种基于反绎学习的掼蛋扑克游戏博弈策略 ABL-GD. 该策略由反绎学习网络、决策模型和不一致性最小化器三个主要部分组成, 有效结合基于对局数据的机器学习和基于专家知识、游戏规则等知识库的逻辑推理, 从而实现动作空间约简、出牌动作合规、博弈性能提升. 通过博弈实验和消融实验, 验证了 ABL-GD 策略的有效性. 本文通过收集并标注掼蛋牌谱及人类对局数据, 构建一个专家-牌谱数据集. 在该数据集上的测试结果证明了 ABL-GD 策略已达到了人类专家的水平.

参考文献 (References)

[1] Silver D, Hubert T, Schrittwieser J, et al. A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play[J]. Science, 362(6419), 1140-1144. 2018.

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

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

[4] Brown N, Sandholm T. Superhuman AI for multiplayer poker[J]. Science, 365(6456): 885-890. 2019.

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

[6] Vinyals O, Babuschkin I, Czarnecki W M, et al. Grandmaster level in StarCraft II using multi-agent reinforcement learning[J]. Nature, 575(7782): 350-354. 2019.

[7] Berner C, Brockman G, Chan B, et al. Dota 2 with large-scale deep reinforcement learning[J]. arXiv:1912.06680, 2019.

[8] Lu Y, Zhao Y, Zhou W, et al. Danzero: Mastering guandan game with reinforcement learning[C]// Proceedings of the 2023 IEEE Conference on Games, 1-8. 2023.

[9] Zinkevich M, Johanson M, Bowling M, & Piccione C. Regret minimization in games with incomplete information[C]//Advances in Neural Information Processing Systems 20, 1729-1736, 2007.

[10] Sandholm T. Abstraction for solving large incomplete-information games[C]//Proceedings of the AAAI Conference on Artificial Intelligence, 29(1). 2019.

[11] Zha D, Xie J, Ma W, et al. Douzero: Mastering Doudizhu with self-play deep reinforcement learning [C]//Proceedings of the 38th International Conference on Machine Learning, 12333-12344. 2021.

[12] Lowe R, Wu Y I, Tamar A, et al. Multi-agent actor-critic for mixed cooperative-competitive environments[C]// Advances in Neural Information Processing Systems 30, 6382-6393. 2017.

[13] Lyu X, Xiao Y, Daley B, et al. Contrasting centralized and decentralized critics in multi-agent reinforcement learning[J]. arXiv:2102.04402, 2021.

[14] Lin T, Huh J, Stauffer C, et al. Learning to ground multi-agent communication with autoencoders[C]//Advances in Neural Information Processing Systems 34, 15230-15242. 2021.

[15] Lo Y L, Sengupta B, Foerster J, et al. Learning multi-agent communication with contrastive learning[J]. arXiv preprint arXiv:2307.01403, 2023.

[16] Zhou Z-H. Abductive learning: Towards bridging machine learning and logical reasoning[J]. Science China Information Sciences, 62(7): 76101. 2019.

[17] Zhou Z-H & Huang Y-X. Abductive learning[M]//Neuro-Symbolic Artificial Intelligence: The State of the Art. 353-369. 2021.

[18] 葛振兴, 向帅, 田品卓, 等. 基于深度强化学习的掼蛋扑克博弈求解[J]. 计算机研究与发展, 61(01):145-155. 2024.

[19] Zhao Y, Lu Y, Zhao J, et al. Danzero+: Dominating the guandan game through reinforcement learning[J]. IEEE Transactions on Games, 2024.

[20] Huang Y-X, Hu W-C, Gao E-H, & Jiang Y. ABLkit: A Python toolkit for abductive learning. Frontiers of Computer Science, to appear. 2024.

[21] Huang Y-X, Dai W-Z, Yang J, et al. Semi-Supervised Abductive Learning and Its Application to Theft Judicial Sentencing [C]//Proceedings of the 20th International Conference on Data Mining, 1070-1075. 2020.

[22] Huang Y-X, Sun Z, Li G, et al. Enabling Abductive Learning to Exploit Knowledge Graph[C]//Proceedings of the 32nd International Joint Conference on Artificial Intelligence, 3839-3847. 2023.

[23] Huang Y-X, Dai W-Z, Jiang Y, & Zhou Z-H. Enabling Knowledge Refinement upon New Concepts in Abductive Learning[C]//Proceedings of the 37th AAAI Conference on Artificial Intelligence, 7928-7935. 2023.

[24] Huang Y-X, Dai W-Z, Cai L.-W, et al. Fast Abductive Learning by Similarity-based Consistency Optimization[C]//Advances in Neural Information Processing Systems 34, 26574-26584. 2021.

[25] 黄宇轩 & 姜远. 带拒绝推理的反绎学习方法. 计算机研究与发展[J], 已录用. 2024.

[26] Hu W-C, Dai W-Z, Jiang Y, & Zhou Z-H. Efficient rectification of neuro-symbolic reasoning inconsistencies by abductive reflection[C]// Proceedings of the AAAI Conference on Artificial Intelligence, 17333-17341. 2025.

[27] Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need[C]//Advances in Neural Information Processing Systems 30. 2017.

[28] Schulman J, Wolski F, Dhariwal P, et al. Proximal policy optimization algorithms[J]. arXiv:1707.06347, 2017.