The background of the page features a large, faint, circular seal of Tsinghua University. The seal contains the university's name in both Chinese and English, along with the founding year 1911 and a central star.

2020 年全国大学生数字冰壶 人工智能挑战赛参赛报告

队员： 沈晓腾

孙纪翔

李炜

相良儒

王惠生

指导教师： 任艳频

2020 年全国大学生数字冰壶人工智能挑战赛参赛报告

沈晓腾 2018010867 孙纪翔 2018010844 李炜 2017011653

相良儒 2018010163 王惠生 2017010295

1. 介绍

本次数字冰壶人工智能挑战赛，使用 unity 模拟冰壶比赛场地，同时与选手程序通讯，将选手的决策作用于冰壶上，进行模拟冰壶投掷，并将得到的结果反馈给选手。选手可以根据得到的信息调整自己的决策，选手决策包含打出冰壶的速度、横坐标偏移量及角速度，冰壶运动的摩擦模型为根据现实情况进行仿真，冰壶之间的碰撞则是依赖于 unity 自带的碰撞模型。

冰壶比赛的规则与国际规则接近，一局中双方选手轮流各投掷 8 壶，共计 16 壶打完后，每一方的得分为己方在大本营内的冰壶中，比对方离大本营中心最近的一壶更靠近中心的冰壶数量。另外还有自由防守区规则：在前 4 壶中，如果将场上对方的壶击飞至场外，则撤下新投掷的一壶，其余所有壶恢复到本次投掷之前的状态。

初赛及模拟赛每场会进行 4 局，淘汰赛及决赛每场会进行 8 局。在初赛及初赛之前，每局如果后手得分较多则下一局交换先后手，否则先后手不变，初赛后每局如果后手得分较多或平局则交换先后手，否则先后手不变。

2. 文献调研

2.1 冰壶比赛及策略

冰壶是一项冬季奥林匹克运动会项目，运动员需将对方的壶撞走，并把自己的壶尽量留在比赛场地的圆心中。每支参赛队伍由五人组成，除一名替补队员外，其余队员按照投掷顺序分别称为一垒、二垒、三垒和四垒。

在 2020 年全国大学生数字冰壶人工智能挑战赛（以下简称“冰壶挑战赛”）中，需要控制的因素包括三方面：投掷策略、擦冰策略和自由防守区策略。下面将分别对其进行简要介绍。

中国女子冰壶队成立于 2003 年，虽然成立时间晚，在短短几年之间就跻身世界强队。以中国女子冰球队为例，我们调研了冰壶比赛的投掷制胜策略。冰壶比赛中通常后手方进攻得分，先手方防守保分。作为传统的进攻模式，后手边路进攻可作为有效

的进攻打法，其中一、二垒的发挥至关重要，占位、旋投和打甩是形成 2 分局面的关键。而后手中路进攻的成功要靠全队的共同发挥，旋投是后手中路进攻最主要的打法。后手的优势在于打开中路的情况下，能通过最后一投控制局势。如果中路有障碍，就失去了后手控分的作用，当对手旋投成功后只能被动得 1 分，甚至被偷分。因此在有后手的情况下，尽量不要主动采用中路进攻，后手边路进攻的效果好于中路进攻。

除投掷策略外，另一个影响比赛结果的因素是擦冰。擦冰不会对速度控制起到决定性的作用，但能够通过减少冰面阻力，使冰壶轨迹更流畅。旋投、打甩等策略离不开擦冰的辅助。擦冰改变冰壶的运动轨迹的机制一般认为有两种：摩擦生热提高冰面温度、扫除冰面的雾和碎片使冰面更光滑，前者的效用更为显著。

冰壶比赛的一大看点在于，队员不仅可以选强势进攻，还可以机智防守，特别是先手局保分的情况下，合理运用自由防守区规则有可能实现偷分反超，防守型战术是先手方的最佳策略。先手方投掷的前两壶中，最常见的战术为采用中路布局；后手方投掷的前两支壶中，第一壶通常为边路占位布局，第二壶常见的布局则采用进营战术，或选择击打清空。冰壶比赛不同阶段的比分决定着战术的变化。比分落后时，可采用进攻型战术打法，扭转不利局势；比分领先时，应采用防守型战术打法，不给对手赶超机会。

2.2 冰壶相关算法

对于冰壶比赛的对阵编排问题，丁蕊等设计了单亲遗传算法的靶向自交叉算子和定点-随机自交叉算子，对多约束问题进行逐层优化，使算法快速收敛。与粒子群算法和经典遗传算法相比，该算法能够根据冰壶比赛的不同情况快速找出满足其多约束条件的最优解，并且执行效率和求解精度能够满足现实要求。

对于对抗策略的生成算法有基于规则的算法、基于粒子群的算法、基于蒙特卡洛和监督学习网络结合的策略生成方法和基于强化学习的方法等。其中基于规则的算法，主要理论依据为上述提到的投掷策略、擦冰策略和自由防守区策略。

粒子群算法是一种进化算法，可以在搜索空间内进行最优化问题的求解。粒子群算法收敛快，但容易陷入局部最优，通过引入惯性权重可以使搜索初期倾向全局搜索，以此寻找全局最优。

蒙特卡洛博弈树是一种随机试验的方法，其所耗费的成本远小于一般的极大极小值算法，其最大的优点就是搜索效率高，然而正确率却较低。在冰壶比赛中，将蒙特卡洛博弈树与监督学习网络相结合，监督学习以粒子群算法为训练样本，可以在一定程度上提高搜索正确率。

前面两种方法更多地为我们提供了搜索的思路，在本次冰壶挑战赛中直接将所有可能的策略枚举一遍，并选择最优解，其运算时间也满足大赛组的设置要求。基于强化学习的方法将在 4.3.1 强化学习 DDPG 部分详细说明。

2.3 冰壶物理模型

本次冰壶挑战赛比赛程序中冰壶轨迹的物理模型采用了 Mark R.A. Shegelski 等人的理论。

初始阶段，冰壶在运动过程中受到冰面的滑动摩擦力，冰面受到冰壶的反作用力导致冰融化，这一点与擦冰的原理相同，其结果是，冰壶与冰面接触区域后方存在液态水薄膜，当遇到固态冰面时，冰壶同时受到干摩擦和湿摩擦的影响，其中湿摩擦与冰壶的运动速度有关。随着冰壶速度逐渐减慢，干湿摩擦区的分布发生变化。在接近停止的阶段，冰壶运动速度足够慢时，将由于表面张力等原因拖曳一些液体膜到冰壶的前面，湿摩擦主要出现在前半环，与初始阶段相反。



图 1 冰壶轨迹模型

碰撞模型则采用 unity 中的物理规则，这部分与物理规律略有差异，将在 3 物理模型复现部分具体说明。

3. 物理模型复现

在虚拟的冰壶比赛中，为了保证投出的壶按照我们所预期的状态进行运动，我们有必要得到虚拟冰壶场地的物理模型，即冰壶的运动模型、冰壶之间的碰撞模型以及擦冰动作引入的影响。在使用所有的非强化学习方法进行冰壶投掷决策时，皆需要得到尽量准确而简单的物理模型从而为决策提供依据。在与全国众多队伍进行 AI 对抗的过程中，我们意识到只要能得到一个准确的物理模型，利用一些比较简单的策略就能够保证绝大多数的胜利。

方法与过程

最简单的得到物理模型的方法是直觉上的，根据物理运动学的运动定律以及动量守恒等基础科学原理，通过进行实验拟合得到动摩擦因数、冰壶碰撞弹性系数等，这样通过简单的匀变速直线运动规律即可完成物理模型构建。但事实上，通过实验我们得知该虚拟冰壶场的物理模型不是最初想象的理想状态，碰撞不符合动量守恒、运动非匀变速、冰面的摩擦也具有随机的各向异性。因此在拿到冰壶场地客户端经过数次测试后，我们便抛弃了这种方法。

通过文献调研，我们了解到了客户端所使用的物理模型，即根据 Mark R.A. Shegelski 等人的理论，对冰壶的运动进行三个阶段的划分，并根据冰面的不同状态，分为干摩擦和湿摩擦两种摩擦属性。我们这里假设在每个阶段内部，模型遵循不同的线性变化规则，类似于对整体的运动过程进行分段线性拟合。这里我们通过实验数据拟合得到了不同阶段之间分隔点的速度阈值、不同阶段干湿摩擦的参数变化、各个参数随着速度的变化等各个参数。此处的物理模型不再是套用公式对整体轨迹进行建构，而是通过设置时间粒度，通过时间序列的参数模拟来得到冰壶运动的轨迹。通过这种方法，我们在冰壶正常运动和冰壶之间正碰时，可以基本上做到准确，实际位置与期望达到的目标相比误差很小。

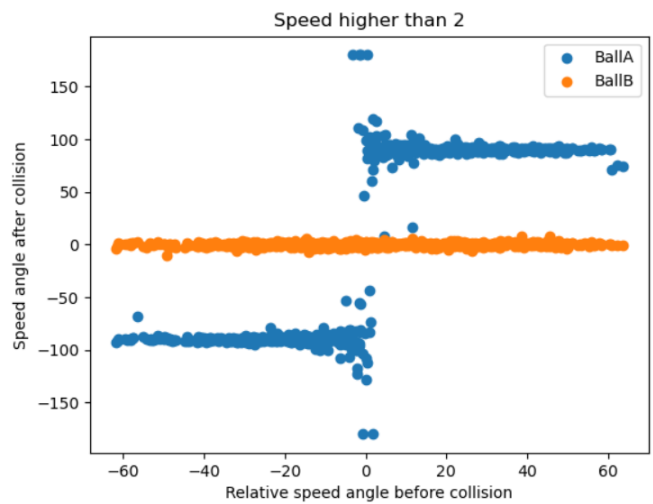


图 2 速度大于 2 的碰撞角度变化

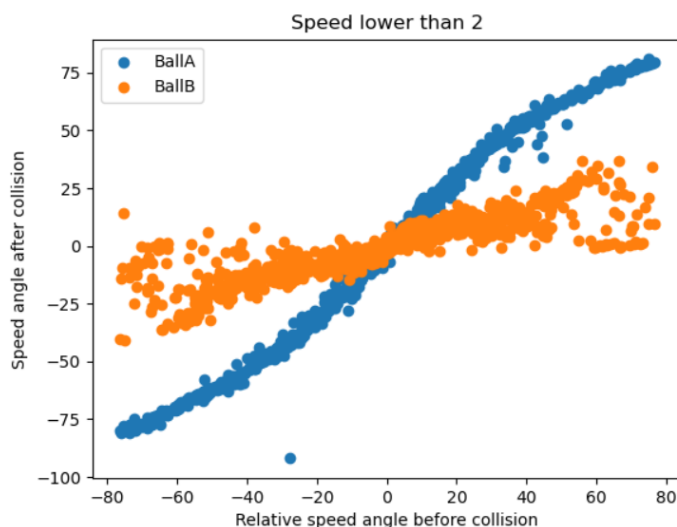


图 3 速度小于 2 的碰撞角度变化

上方两图是我们从大量对局数据中提取出了碰撞的情况绘制的碰撞前后角度变化情况。可以看出若碰撞时的相对速度大于 2，则近似为完全弹性碰撞，不论碰前角度如何，碰后两壶速度均朝平行碰撞时径向或切向的方向（横坐标为 0 时，即正碰的情况下壶 A 角度不稳定是因为它实际上静止了，无法测量角度导致，这个误差不会产生实质影响）。若碰撞速度小于 2，发生的则是不完全弹性碰撞，可以看出碰撞壶（壶 A，蓝色线）角度曲线相对稳定，而被碰壶角度波动较大，极端情况下波动达到 60 度以上，完全不在可预计范围内。

通过实验发现，冰壶的低速斜碰不满足直觉上的物理规律，过程中我们尝试了用线性模型拟合、通过单层网络训练得到对应的参数进行拟合，但是效果均不理想。最后我们也并没有在冰壶斜碰上达到比较高的精度。由于冰场本身摩擦的随机性，使得斜碰后冰壶的飞出方向波动性很大，很难得到比较满意的结果。之后根据冰壶平台的数据记录，我们发现，unity 内部的碰撞模型当中碰撞的判定有可能发生了多次，这个次数可能根据内部代码实现方式不同而无法预估，我们是几乎不可能在自己的物理模型中复现这一点的（这相当于拟合仿真程序本身的 bug），因此我们决定放弃该部分的拟合，转为将其视为固定误差，用引入误差区间的方法进行调和，这将在 4.2 搜索算法中介绍。

4. 算法

4.1 基于规则

基于规则的策略是一种实现简单且效果较好的方法，在实际的冰壶比赛中采用的也是基于规则的方法。经过试验，以进攻碰撞为主，阻挡防守为辅的策略有着最好的效果。

具体来说，基于规则的策略是：投前五个受自由防守规则保护的球时，占住自由防守区的中间线，或挡在对手的冰壶前，这样在后续对碰时会有较好效果。后 11 个球以进攻为主，找到对手离中心最近的冰壶，判断直线路径上是否有其他冰壶阻挡，若没有，则高速直碰以达到完全弹性碰撞的效果，若有，则分别判断左右两条曲线路径上是否有壶和边界存在，选择壶较少的一边进行曲线发球，即给球一个角速度。当当前局面已经具有 2 分以上优势时，采取防守策略，将冰壶挡在己方壶前。对于最后一球进行特判处理，如已可得分则避免碰到己方冰壶。

经过调参，该方案有了较精准的直碰和斜碰效果，尤其在对手不采用类似的进攻策略时，先手最多失分为 1 分，后手情况常可以取得极高的分数优势。

4.2 搜索

搜索方法可以说是在 AI 对抗中最为常用的一种方法，它是非常简单直接地将所有可能的策略都枚举一遍，模拟经过此操作后出现的局面，找出对我方最有利的局面。不同于基于规则的方式完全是机械地执行人类所想到的策略，这种方法往往能够找出让人意想不到的优质决策。

在本次比赛的冰壶对战中，搜索算法的使用有着一些障碍：

决策空间近似为连续空间，难以搜索

由于决策对于精度的要求并不是非常严格，决策的速度或者角度上有一点点的偏差并不会太大的影响，我们采用直接对决策均匀地进行离散化的方式减少枚举量。

速度方面，低速情况下对速度的精度要求相对较高，因此在 2.6 到 4.5 之间每隔 0.05 取一个决策点，高速情况下则在 4.5 到 7.5 之间每隔 0.5 取一个决策点，总共约有 45 个决策点。

横坐标方面，首先在 -1.7 到 1.7 之间每隔 0.05 取一个决策点总共有约 70 个决策点，之后考虑到可能会有正面碰撞其他冰壶，以做到击飞效果的情况，需要再额外增加场上所有冰壶的横坐标，最多会有 15 个，因此横坐标部分有约 85 个决策点。

转角方面，由于前面横坐标其实已经提供了足够高的精度，冰壶出手时的自转角速度只设定了-10、0、10 三个极限决策点，通常情况下这三个点已经可以满足需求。

以上是对于决策中三个量的处理，由于它们之间是组合关系，极限情况下决策的数量会达到 $45 \times 85 \times 3$ ，**大约有 1 万个决策**，而在确定了一个决策后，物理模型的单次推演往往需要 10ms 左右的时间，如果 1 万个决策全部推演必然导致超时。

事实上，这 1 万个决策有很大一部分是明显会导致冰壶未抵达大本营附近就出界，或者没有与任何冰壶发生碰撞就直接飞出界外，这些决策都是不必要推演的，因此我们采取**哈希的方法进行剪枝处理**。具体来说，进行过一次决策推演后，我们先对场上各冰壶的坐标进行信息提取，得到一个代表场面状况的数据，第二次决策推演通常会与上一次拥有相同的横坐标和角度值，仅仅是速度有所增加，将第二次决策推演后场面状况的信息与第一次进行比对，如果发现提速过后得到的局面与上次相同，就认为继续提速下去也不会有优质决策出现，可以停止本次搜索，回退改变横坐标与角速度。在实际测试中，这个方法可以将决策速度提高 10 倍以上。

回合数	1	2	3	4	5	6	7	8
普通搜索用时(s)	1.27	2.66	5.13	11.82	20.36	36.16	>60 (超时)	>60 (超时)
加入哈希算法用时(s)	0.05	0.13	0.47	0.98	1.16	1.58	1.96	1.94

随着回合数增加，局面变得复杂，物理模型推演用时大幅增加，只有加入剪枝才可以确保不出现超时的情况。

最后为了进一步提高精度，在粗略确定决策之后，还会在决策周围进行小范围、高精度的搜索，才能够得出最终的决策。

局面推演存在困难，物理模型误差较大

本次比赛使用的物理模型并没有明确地公布出来，仅仅是提供了一个 unity 实现的对战平台，其中具体的物理参数并不对外公布，尤其是冰壶的碰撞模型完全使用了 unity 内置模型，更是难以拟合。我们最终尽力实现出了一个模型，但拟合效果并不尽如人意。尤其冰壶旋转时角速度对速度方向的影响、低速情况下冰壶的斜碰等等误差较大。

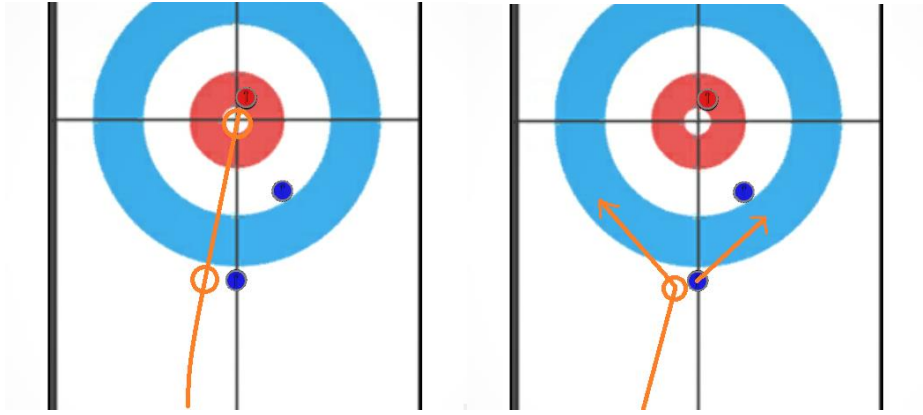


图 4 左为期望效果，击飞红方冰壶，右为实际效果，
由于角度估算存在细微误差，碰撞歪斜

我们引入误差范围的概念，对于某一参数（如冰壶碰撞时的切向摩擦力）设定一个可信的参数范围。在对给定决策进行局面推演时，对于同一个决策反复多做几次推演（一般为 10 次），每次都让参数都在可信的范围内随机取值，取其得到局面估价的最小值作为本次决策的价值。

这是一个在拟合存在误差情况下的折中办法，确保 AI 不会因为走“险招”而出现失误，但同时也丧失了一定打出精彩操作的能力。

最后总结一下**单步搜索算法的流程**：

- 按照横坐标、角速度、速度的顺序依次循环枚举决策。
- 取一组固定参数进行物理推演，如果得到局面的哈希值与上次决策相同，则退出速度循环的枚举。
- 重复 10 次局面推演，每次都随机重新取一组物理参数（误差区间内），取其最小值作为本决策的价值，取价值最大的决策。
- 得到决策后，在决策附近小范围内在进行一次精度更高的搜索作为修正，输出最终得到的决策。

上述单步搜索算法就是最终的在比赛中使用的方法，而除此之外还尝试了**多步搜索**的方法，其原理为：

设定搜索步数 N （单步搜索等同于 $N=1$ ）。

设 $\text{Value}(N, \text{state}, \text{action})$ 为 state 局面下，使用 N 步搜索方法得到的 action 决策价值，若当前轮到我方决策，当前局面下得出的决策价值：

$$\text{Value}(N, \text{state}, \text{action}) = \min_{\text{action}} \{ \text{Value}(N-1, \text{nextstate}, \text{action}) \}$$

其中 nextstate 为 state 下进行 action 后得到的价值，即假设对方能够做出最小化我方价值的操作，而我们目标则是最大化这个最小值，即取令 $\text{Value}(N, \text{state}, \text{action})$ 最

大的 action。轮到敌方决策时同理，其目标为假设我方能够最大化价值的情况下，要最小化这个最大值。因此这种搜索方法也被称为极大极小搜索，或 α - β 搜索。

具体应用到本次比赛中，却因为**搜索量巨大**难以施展，因为其时间复杂度为决策空间的 N 次方，随 N 的增大指数级增长。

最终通过不断剪枝实现了一个仅考虑对方少数操作的两步搜索算法，其在对战特定 AI 时能够有绝佳表现，但一旦遇到对方的决策在自己搜索空间之外的对手就不尽如人意，通用性不强，最终未予使用。

4.3 深度学习

我们尝试了多种强化学习方法，但最终没能得到较好的成果。

4.3.1 强化学习 DDPG

本次比赛是一个连续空间决策的问题，因此首先想到采用 DDPG 的方法。

其核心思想被称为 Actor-Critic，即在 DDPG 中存在两个神经网络，一个是 Actor，负责根据局面输出动作，是核心的决策网络，另一个是 Critic，负责根据局面及决策判断本次决策的价值，进而引导 Actor 网络做出价值更高的决策。

训练流程

在训练中，Actor 网络会根据当前局面输出一个决策，游戏环境模块根据这个决策给出奖励信息，Critic 需要以环境给出的奖励为目标进行学习，尽可能让自己的输出接近环境给出的奖励。当 Critic 能够较好地估算一个决策的价值之后，Actor 网络就需要开始进行学习，让自己输出的决策能够在 Critic 网络中得到较高的价值。

另外为了让 AI 能做到“考虑大局”，即引入未来奖励的影响，最大化总奖励，Critic 实际学习的价值函数为：

$$\text{Value}(\text{state}) = \text{reward} + \text{gamma} * \text{Value}(\text{nextState})$$

上式意味着 Critic 网络在关注当前 reward 的同时，还会以 gamma 的权重关注未来（nextState）的价值，我们可以通过调整 gamma 的大小来改变 AI 对这两者的关注度。

奖励设置

游戏的奖励设置对训练效果有很大影响，实际的冰壶比赛中，我们只在一局游戏结束时才知道当前的胜负情况，才能确切地给予奖励，若在中间设置奖励要保证奖励的总和是固定的。于是我们引入局面估价，本次给予的奖励是操作后的局面价值减去

操作前的局面价值，而局面估价则是根据离中心最近的一批冰壶来确定。这样一来就保证了所有奖励的总和就是最终局面的估价。

训练效果

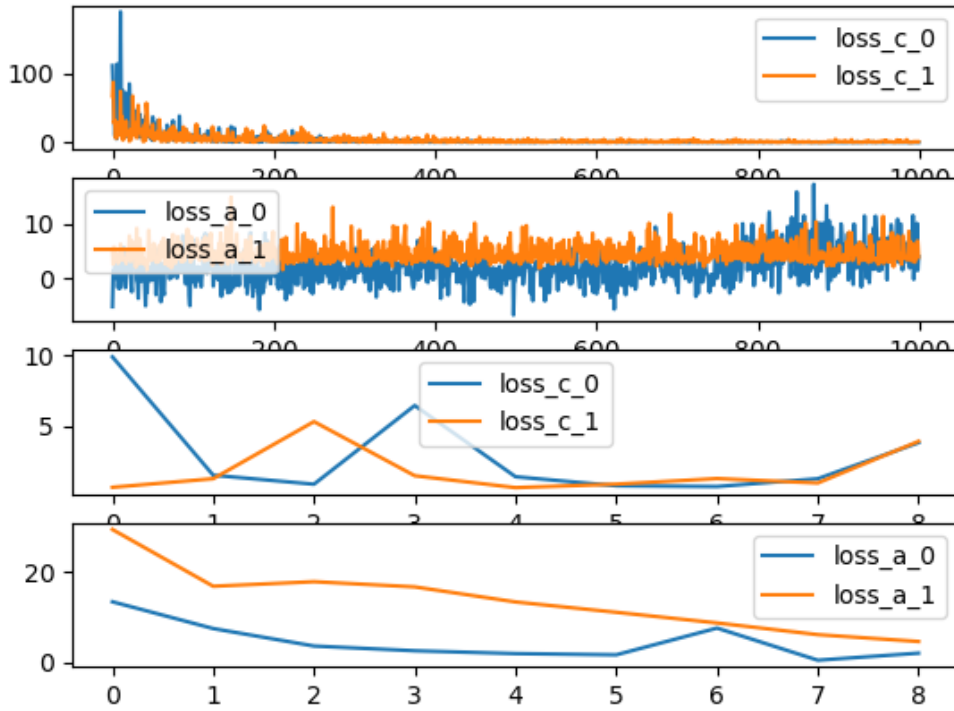


图 5 正式训练时没有记录数据，临时复现没有时间训练太多轮，这里只作为展示
其中蓝色先为先手 AI，橙色线为后手 AI

从上图可以看出单轮训练中，网络 Critic 网络很快收敛，能够对局面做出合适的估价，但 Actor 网络则没有明显的变化。后续训练中，训练了较多轮的情况下仍然发现 Critic 网络对于 Actor 网络给出的决策的评价波动幅度较大，没有明显的收敛趋势，同时观察 AI 的决策，决策水平仍然较低，因此认为训练没有起到效果，暂时放弃了进一步的尝试。

DQN

该部分其实是对于 DDPG 的一点修改，在 DDPG 中，我们发现 Critic 网络的估价其实已经有了初步的效果，于是选择使用离散化进行搜索的方式来替代这个 Actor 网络，这样一来决策空间被离散，算法本质上变成了 DQN 算法，但仍未能取得较好的效果。

4.3.2 RNN

RNN 方法是在复现物理模型的过程中，想到尝试使用神经网络来替我们完成物理模型的单调的参数调整工作。这源于我们发现对战平台的 `record` 文件中存放了冰壶从抛出到静止全过程所有冰壶的坐标数据，正好可以使用 RNN 进行训练。

RNN 是一种会保存隐藏层数据的神经网络，类似于隐马尔可夫链，能够从数据中提取出隐藏的状态，对于本问题来说，这些数据是冰壶的速度、角速度数据，它们未被记录在 `record` 中。如此一来，就可以使用 RNN 逐帧推演物理模型。

但我们最终没有实践这一方法，主要有两个原因：

- a. 一次投掷在物理模型中包含 6000 帧数据，如果使用 RNN，光是输入的数据量就需要达到平方级别，已经是千万左右，复杂度太高会导致搜索量大幅降低，这样反而不能够得到优质的决策。
- b. RNN 处理单帧可能有比较好的效果，但在 6000 帧的问题中，很可能出现误差积累，最终我们真正需要的最后一帧数据误差变得极大。

出于上述顾虑，加之比赛时间有限，我们最终没有尝试使用 RNN，但这一点可以作为之后继续探索的方向。

5. 比赛过程

5.1 时间线

比赛分为模拟赛阶段和正式赛阶段。我方参与的所有比赛整体时间线如下表格所示。比赛共有 40 余支队伍报名，经校内预选赛选拔后剩余 14 支队伍。14 支队伍被分为 A、B、C、D 四个小组，组内进行单循环赛，小组前 2 名出线组成八强队伍。八强队伍按照一定的规则进行两两对决，赢者组成四强队伍，即 1/4 决赛。由于强化学习 AI 效果不佳，所以最终我们采用的主要为基于搜索的 AI。整个挑战赛均在官方提供的 Unity 比赛程序上进行。

时间	比赛名称	采用 AI
7.8-7.30	模拟赛	前三次规则 AI，最后一次搜索 AI
7.30	校内预选赛（单循环）	搜索 AI
8.6	初赛（下半区 B 组）	策略-搜索 AI
8.11	八强赛（1/4 决赛）	防守策略-搜索 AI

5.2 各阶段比赛情况

模拟赛中，我方（“华水队”）基本一直处于排行榜前列，基于规则的冰壶 AI 取得了良好的效果。预选赛时，我方首次启用了基于搜索的冰壶 AI，根据比赛结果可以看出，AI 尽管在面对对手以撞为主的策略时束手无策，但在面对对手以防御为主的策略时可以取得不错的进攻效果，这可能是得益于搜索算法把握更全面，并且投掷更准确。预选赛中很多队伍并不会利用冰壶旋转击飞，因此搜索 AI 取得了一定的优势。

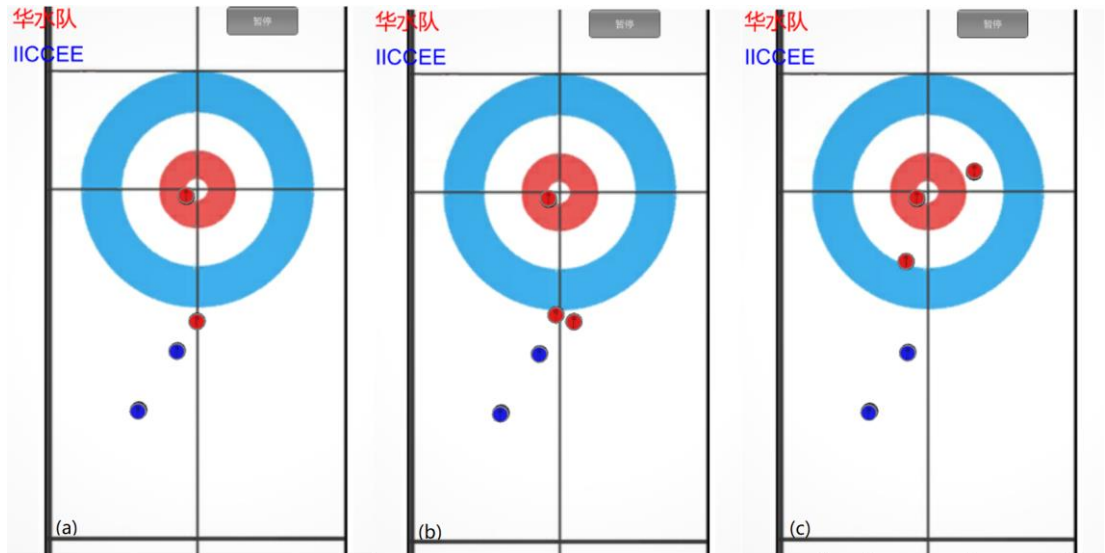


图 7 预选赛：我方利用一个冰壶成功将另一个冰壶也带入大本营内
(a)我方投掷前；(b)我方冰壶滑行并撞击；(c)我方投掷结束后

在面对一般防守策略时，我方善于破防并且以大比分取得比赛胜利，预选赛（单循环赛）积分情况如下所示。

积分表如下：

积分表						
	华水队	IICCEE	不会起名	Coming	积分	净胜球
华水队	-	10:0	4:0	4:2	9	16
IICCEE	0:10	-	6:1	7:0	6	2
不会起名	0:4	1:6	-	0:2	0	-11
Coming	2:4	0:7	2:0	-	3	-7

图 8 清华组预选赛积分表（来源于赛事直播）

进入初赛后，我方发现了冰壶规则上的缺陷，即平局并不交换先后手。但是后手相对来讲具有优势，容易得分，所以我方策略为后手时一直逼平，最后一局得分（或某局拿到 2 分），取得胜利；而先手时发挥我方容易击打出多飞的优势，只让对手一分并逼平整场比赛。所以胜负基本取决于赛前决定先后手的猜拳结果。在小组内单循

环赛中，我方在猜拳中取得了 2 次胜利，所以最终在 3 场小组赛中取得了 2 胜 1 平的战绩，没有出现失误，以小组第 1 名身份进入 8 强。积分表如下所示。

B 组积分表						
	摩擦力规划局	我说的队	华水队	IICCEE	积分	净胜球
摩擦力规划局	-	4:2	1:2	6:2	6	5
我说的队	2:4	-	0:1	1:8	0	-10
华水队	2:1	1:0	-	2:2	7	2
IICCEE	2:6	8:1	2:2	-	4	3

图 9 B 组初赛积分表

从 1/4 决赛开始，赛制被调整，平局仍然交换先后手，所以初赛的策略不能被采用。1/4 决赛中，我方对手为厦门大学的“夏天要加冰”队，在之前的模拟赛的交手记录中，我方不占优势。我方使用调整过的具有防守策略的搜索 AI 进行比赛，可是防守模式表现不佳，并且比赛中多次出现失误，导致最终结果告负，遗憾止步 8 强。失误包括：将自由防守区内的对方冰壶击飞并触发了自由防守区规则，浪费了投壶机会；击飞冰壶时自己也飞出了大本营，导致后手优势被浪费；仿真不准确，导致投壶出现失误，例如 1/4 决赛最后一球，如下图所示。

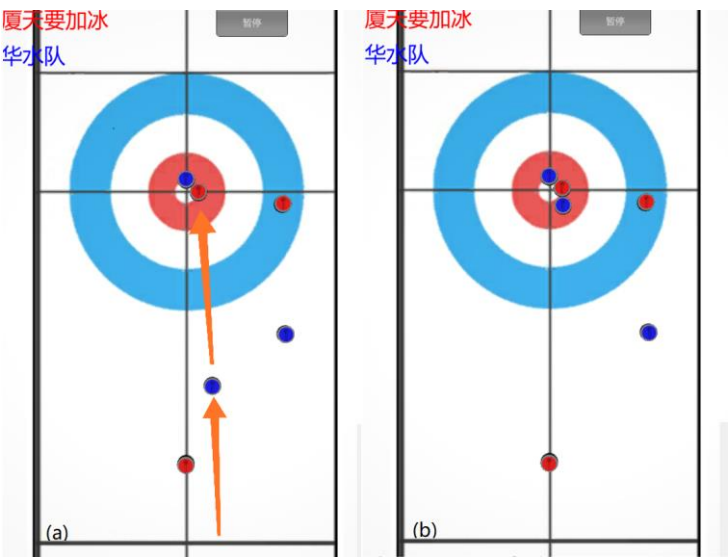


图 10 1/4 决赛最后一球

(a)预期轨迹：撞开红壶并得到 2 分 (b)最终结果：未能撞开并失掉 1 分

注意到“夏天要加冰”队习惯于将己方的冰壶贴附到对方冰壶上，这利用了冰壶 Unity 比赛程序的碰撞模型缺陷（低速碰撞被认为完全非弹性碰撞）。这一操作使得单独击飞其冰壶变得较为困难，而且我方搜索 AI 的仿真精度不够，这种情况下容易出现失误。

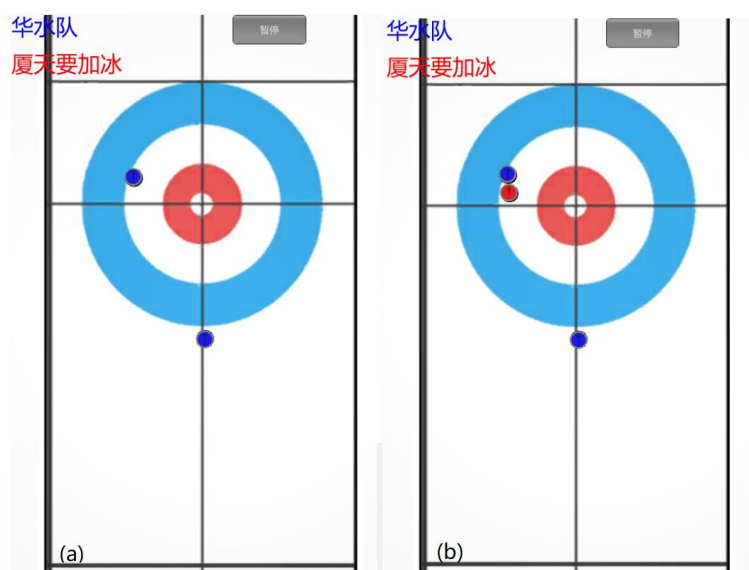


图 11 1/4 决赛“夏天要加冰”队（红方）将冰壶黏附于我方冰壶

6. 总结

实际冰壶比赛中，利用好自由防守区规则往往可以形成一道较好的防线进而创造更多的得分机会。但是在本次 AI 冰壶挑战赛中，由于冰壶控制较为准确，自由防守区规则很难被利用。本次比赛中，双方往往交替换壶，因为在曲线撞击可能的情况下，防守往往不起作用，除非利用较多的冰壶可以从各个方向上进行封锁。所以本次 AI 冰壶挑战赛的各队策略与实际冰壶比赛差异很大。另一方面，本次比赛程序中对冰壶场地的仿真情况也与实际情况差异较大。实际中冰壶速度有一定的限制，且冰壶间发生的一般是完全弹性碰撞，但在比赛程序中，低速碰撞会被认为是完全非弹性碰撞，这使得利用己方冰壶准确控制敌方冰壶落点成为可能，因为低速碰撞下两个冰壶会“粘”在一起。例如“夏天要加冰”队的策略很明显地体现了这一点。

我方的搜索 AI 由于能考虑到更全面的情况而在比赛初期具有优势，但搜索 AI 较大的缺点是评估局面的函数连续性较好，对于一些细节如冰壶“粘”在一起的情况处理不够恰当，并且也不擅长利用这些细节。局面上某些微小的物理差别在实际中可能会最终导致完全不同的结果。这种不连续性可能也是强化学习方法很难产生较好效果的原因。另外，搜索 AI 由于采用评价函数体系，所以其行为具有一定的不可预测性，也没有更宏观的规则策略，也是在某些场次比赛中失败的原因。所以基于规则方法的 AI 如果有更好的布局策略，做好细节处理，给搜索 AI 造成干扰与失误，也可以战胜看似更全面的搜索 AI。以上提到的点也是搜索 AI 可以改进的地方，例如：针对对手做有针对性的调整；结合规则方法，加入策略布局；加强仿真精度等等。虽然本次比赛已经结束，但本次比赛中收获的宝贵经验也可以借鉴至其他竞赛中。

参考文献

- [1]Mark R. A. Shegelski, Ross Niebergall, Mark A. Walton. The motion of a curling rock.[J].Canadian Journal of Physics, 1996, Vol. 74, No. 9-10 : pp. 663-670
- [2]金晶,姜宇,李丹丹,沈毅.基于冰壶机器人的人工智能实验教学设计与实践[J].实验技术与管理,2020,37(04):210-212+230.
- [3]柳悦.浅谈冰壶比赛中的投壶技术特点[J].科技资讯,2019,17(31):251+253.
- [4]刘家楠. 中国女子冰壶队历届冬奥会比赛战术运用的研究[D].首都体育学院,2019.
- [5]李征宇,冯伟,王珂.对中国女子冰壶队比赛后手进攻战术的研究[J].北京体育大学学报,2015,38(07):128-132+139.
- [6]穆亮. 冰壶竞技战术研究[D].东北师范大学,2015.
- [7]王文龙,孙耀威.冰壶投壶和擦冰技术特点的研究述评[J].青少年体育,2019(03):139-140.
- [8]汪宇峰,苏和,穆亮. 我国优秀男子冰壶运动员擦冰有效性研究[C]. 中国体育科学学会 (China Sport Science Society) .2015 第十届全国体育科学大会论文摘要汇编(三).中国体育科学学会 (China Sport Science Society) :中国体育科学学会,2015:270-271.
- [9]邵蔚. 基于强化学习的冰壶比赛策略生成方法研究[D].哈尔滨工业大学,2018.
- [10]丁蕊,董红斌,邢薇,刘文杰,孔飞.一种求解冰壶比赛对阵多约束问题的逐层优化算法[J].电子学报,2017,45(03):632-637.
- [11]田雨. 无碰撞冰壶运动轨迹仿真系统的研制[C]. 中国体育科学学会 (China Sport Science Society) .2015 第十届全国体育科学大会论文摘要汇编(二).中国体育科学学会 (China Sport Science Society) :中国体育科学学会,2015:974-975.
- [12]李妍,王珂,于亮,徐乐.冰壶自由防守区投壶布局研究[J].体育文化导刊,2012(01):62-65.

相关代码

- 1. 物理模型复现: <https://github.com/HUA-water/CurlingSimulator>
- 2. 单步搜索: <https://github.com/HUA-water/Curling/tree/search>
- 3. 两步搜索: <https://github.com/HUA-water/Curling/tree/search-2>
- 4. 强化学习: <https://github.com/HUA-water/Curling/tree/ReinforcementLearning>

5. 初赛使用的策略: <https://github.com/HUA-water/Curling/tree/search-fool>