

# 双升环境 使用说明

## 牌的基本表示

- 每一方的手牌、每轮出牌都可以用  $(2 \times 4 \times 14)$  的0-1数组表示
- 每一层  $(4 \times 14)$  矩阵) 表示一副牌，每一行  $(14 \text{维向量})$  表示一种花色，第0~12位表示点数2~A，第13位表示王牌，黑桃行的第13位表示小王，红桃行的第13位表示大王（如下表所示），其余两行留空
- 依照这种方法，可将每一张牌映射到矩阵上的唯一位置，因此可以将两副牌中的任意张牌组成的套牌用一个矩阵表示（矩阵中1的数量=牌的张数）

	2	3	4	5	6	7	8	9	0	J	Q	K	A	o
S♠														小王
H♥														大王
C♣														-
D♦														-

- 实际处理牌型时，为了压缩动作空间，会优先将每张牌映射至第一层
- 详见wrapper.py中关于将牌型转化为np.array的函数：

```
# Adding cards 1 by 1 to the given cardset
def add_card(self, cardset: np.array, cards):
    # cardset: np.array(2,4,14)
    # cards: list[str(2)], cardnames.
    for card in cards:
        card_pos = self.name2pos(card) # Getting positions in the matrix
        if cardset[0, card_pos[0], card_pos[1]] == 0: # Try placing on
Layer 1
            cardset[0, card_pos[0], card_pos[1]] = 1
        elif cardset[1, card_pos[0], card_pos[1]] == 0: # Layer 1
occupied
            cardset[1, card_pos[0], card_pos[1]] = 1 # Placing on Layer 2
        else:
            raise "More than two cards with same suits and points. Please
recheck."

    return cardset
```

## 状态与动作空间

- State\_shape: 128\*4\*14
  - major\_mat: (2\*4\*14) 本局主牌信息
  - deck\_mat: (2\*4\*14) 玩家手牌
  - hist\_mat: (8\*4\*14) 本轮历史出牌信息（依次为4名玩家）
    - 若本轮内已出牌的玩家不足4人，剩余部分补0
    - 除第一轮外，每轮首个出牌的玩家可获得
    - 每局的第一轮第一次出牌前，历史信息为空白（全0）
  - played\_mat: (8\*4\*14) 全局已出牌信息（4名玩家）
  - option\_mat: (108\*4\*14) 本回合可选牌组信息（54个可选动作）
    - 由程序解析当前玩家手牌生成并包装（不足54个会补0）
- Action\_shape: 54
  - 输出54个可选动作的概率
  - Action\_mask: 54
    - 许多情况下，可选动作不足54种，用于mask可选动作中的所有缺省项
  - 模型在state中提供的54个可选动作(option\_mat)中做出选择

## 奖励设计

- 按照双升规则，每轮出牌（即四名玩家各行动一次）都可能产生“分”，目前依照得“分”情况为玩家提供奖励
- 奖励设计可多样化

## 双升规则与botzone交互方式

- botzone上双升游戏规则及交互方式请见<https://wiki.botzone.org.cn/index.php?title=Tractor>
- 我们已经提供了botzone接口\_\_main\_\_.py

## 训练框架

Actor.py 采样：

- 从modelpool中调取模型（目前默认调用modelpool中最新的模型）
- 四名玩家均由模型控制，与环境交互获取数据存入Replaybuffer

Learner.py 学习：

- Replaybuffer达到一定数据量，learner启动，从Replaybuffer中采样更新网络参数
- 更新后的模型存入modelpool

## 使用方式

- 运行train.py启动多进程训练，模型会定期保存至checkpoint文件夹下
- 将模型上传至botzone存储空间，新建bot并将\_\_main\_\_.py, mvGen.py, wrapper.py和model.py打包上传作为bot代码