## A. Introduction of DouDizhu

As the most popular card game in China, DouDizhu has attracted hundreds of millions of players with many tournaments held every year. DouDizhu is known to be easy to learn but challenging to master. It requires careful planning and strategic thinking. DouDizhu is played among three players. In each game, the players will first bid for the Landlord position. After the bidding phase, one player will become the Landlord, and the other two players will become the Peasants. The two Peasants play as a team to fight against the Landlord. The objective of the game is to be the first player to have no cards left. In addition to the huge state/action spaces and incomplete information, the two Peasants need to cooperate to beat the Landlord. Thus, existing algorithms for poker games, which usually operate on small games and are only designed for two players, are not applicable in DouDizhu. In what follows, we first give an overview of the game rule of DouDizhu and then analyze the state/action spaces of DouDizhu. Readers who are familiar with the game may skip Section A.1. Readers who are not familiar with DouDizhu may also refer to Wikipedia [9] for more introduction.

### A.1. Rules

DouDizhu is played with one pack of cards, including the two jokers. Suits are irrelevant in DouDizhu. The cards are ranked by Red Joker, Black Joker, 2, A, K, Q, J, 10, 9, 8, 7, 6, 5, 4, 3. Each game has three phases as follows.

- **Dealing:** A shuffled pack of 54 cards will be dealt to the three players. Each player will be dealt 17 cards, and the last three leftover cards will be kept on the deck, face down. These three cards will be dealt to the Landlord, which are decided in the bidding phase.
- **Bidding:** The three players will analyze their own cards without showing to other players. The players decide whether they would like to bid the Landlord based on their hand cards' strength. There are many versions of bidding rules. In this paper, we consider a version adopted in most online DouDizhu app. The first bidder will be randomly chosen. The first bidder will then decide whether she bids. If the first bidder does not bid, the other players will become the bidder in turn until someone bids. If no one bids, a new pack of cards will be dealt to the players. If one chooses to bid, the other players will decide whether she accepts the bid or she wants to outbid. Each player only has one chance to outbid. The last player who bids or outbids will become the Landlord. Once the Landlord is settled, the Landlord will be dealt with the three cards on the deck. The other two players will play as the Peasants to fight against the Landlord.
- **Card-Playing:** In this phase, the players will play cards in turn starting from the Landlord. The first player can choose either category of cards such as Solo, Pair, etc. (detailed in the next paragraph). Then the next player must play the cards in the same category with a higher rank. The next player can also choose "PASS" if she does not have a higher rank in hand or she does not want to follow the category. If all the other players choose "PASS," the player who first plays the category can freely play cards in other categories. The players will play cards in turn until one player has no cards left. The Landlord wins if she has no cards left. The Peasant team wins if either of the peasants has no cards left. The two Peasants need to cooperate to increase the possibility of winning. A Peasant may still win a game by helping the other Peasant win even if she has terrible hand cards.

One challenge of DouDizhu is the rich categories, which consist of various combinations of cards. For some categories, the player can choose a kicker card, which can be any card in hand. One will usually choose a useless card as a kicker card so that she can more easily go out of hand. As a result, the player needs to carefully plan how to play the cards to win a game. The categories in DouDizhu are listed as follows. Note that Bomb and Rocket defy the category rules and can dominate all the other categories.

- **Solo:** Any single card.
- **Pair:** Two matching cards of equal rank.
- **Trio:** Three individual cards of equal rank.
- **Trio with Solo:** Three individual cards of equal rank with a Solo as the kicker.
- **Trio with Pair:** Three individual cards of equal rank with a Pair as the kicker.

[9] https://en.wikipedia.org/wiki/Dou_dizhu

- **Chain of Solo:** $\geq$Five consecutive individual cards.
- **Chain of Pair:** $\geq$Three consecutive Pairs.
- **Chain of Trio:** $\geq$Two consecutive Trios.
- **Plane with Solo:** $\geq$Two consecutive trios with each has a distinct individual kicker card.
- **Quad with Pair:** Four-of-a-kind with two sets of Pair as the kicker.
- **Bomb:** Four-of-a-kind.
- **Rocket:** Red and Black jokers.

### A.2. State and Action Space of DouDizhu

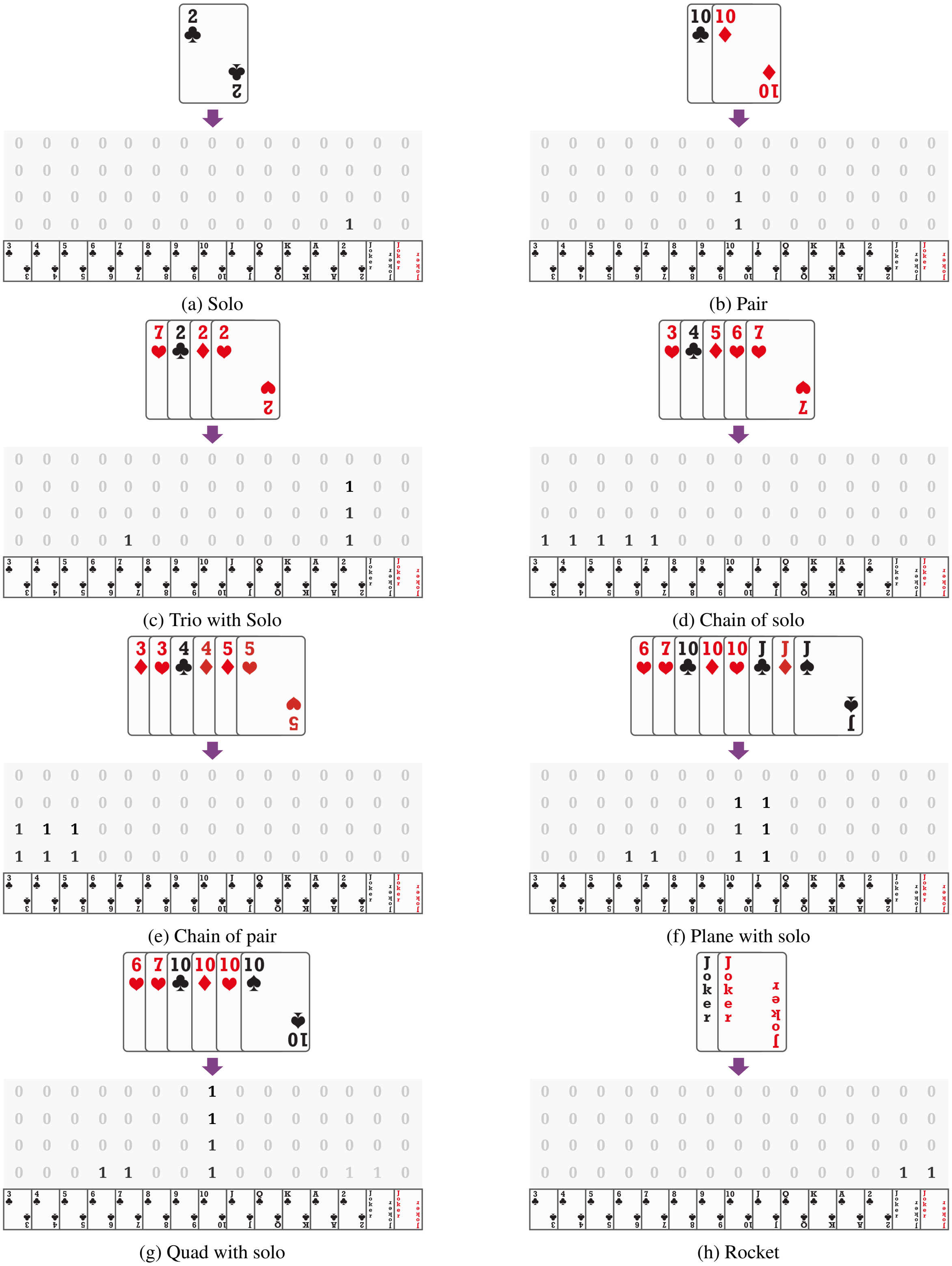
According to the estimation in RLCard (Zha et al., 2019a), the number of information sets in DouDizhu is up to $10^{83}$ and the average size of each information set is up to $10^{23}$. While the number of information sets is smaller than that of No-limit Texas Hold'em ($10^{126}$), the average size of each information set is much larger than that of No-limit Texas Hold'em ($10^4$). Different from Hold'em games, the state space of DouDizhu can not be easily abstracted. Specifically, every card matters in DouDizhu towards winning. For example, the number of cards of rank 2 in the historical moves is crucial since the players need to decide whether their cards will be dominated by other players with a 2. Thus, a very slight difference in the state representation could significantly impact the strategy. While the size of the state space of DouDizhu is not as large as that of No-limit Texas Hold'em, learning an effective strategy is very challenging since the agents need to distinguish different states accurately. DouZero approaches this problem by extracting representations and learning the strategy automatically with deep neural networks.

DouDizhu suffers from an explosion of action space due to the combinations of cards. We summarize the action space in Table 3. The size of the action space of DouDizhu is $27,472$, which is much larger than Mahjong ($10^2$). It is also much more complicated than No-limit Texas Hold'em, whose action space can be easily abstracted. Specifically, in DouDizhu, every card matters. For example, for the action type `Trio with Solo`, wrongly choosing the kicker may directly result in a loss since it could potentially break a chain. Thus, it is difficult to abstract the action space. This poses challenges for reinforcement learning since most of the algorithms only work well on small action space. In contrast to the previous work that abstracts the action space with heuristics (Jiang et al., 2019), DouZero approaches this issue with Monte-Carlo methods, which allow flexible exploration of the action space to potentially discover better moves.

*Table 3.* Summary of the action space of DouDizhu. We follow the summary provided in RLCard (Zha et al., 2019a).

| **Action Type** | **Number of Actions** |
|---|---|
| Solo | 15 |
| Pair | 13 |
| Trio | 13 |
| Trio with Solo | 182 |
| Trio with Pair | 156 |
| Chain of Solo | 36 |
| Chain of Pair | 52 |
| Chain of Trio | 45 |
| Plane with Solo | $21,822$ |
| Plane with Pair | $2,939$ |
| Quad with Solo | $1,326$ |
| Quad with Pair | 858 |
| Bomb | 13 |
| Rocket | 1 |
| Pass | 1 |
| Total | $27,472$ |

## B. Additional Examples of Card Representations



(a) Solo

| 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | J | Q | K | A | 2 | Joker | Joker |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |

(b) Pair

| 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | J | Q | K | A | 2 | Joker | Joker |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

(c) Trio with Solo

| 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | J | Q | K | A | 2 | Joker | Joker |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |

(d) Chain of solo

| 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | J | Q | K | A | 2 | Joker | Joker |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

(e) Chain of pair

| 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | J | Q | K | A | 2 | Joker | Joker |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

(f) Plane with solo

| 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | J | Q | K | A | 2 | Joker | Joker |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |

(g) Quad with solo

| 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | J | Q | K | A | 2 | Joker | Joker |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 |

(h) Rocket

| 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | J | Q | K | A | 2 | Joker | Joker |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |

*Figure 11.* Additional examples of encoding different types of cards.

## C. Additional Details of Feature Representation and Neural Architecture

### C.1. Action and State Representation

The input of the neural network is the concatenated representation of state and action. For each $15 \times 4$ card matrix, we first flatten the matrix into a 1-dimensional vector of size 60. Then we remove six entries that are always zero since there is only one black or red joker. In other words, each card matrix is transformed into a one-hot vector of size 54. In addition to card matrices, we further use a one-hot vector to represent the other two players' current hand cards. For example, for Peasant, we use a vector of size 17, where each entry corresponds to the number of hand cards in the current state. For the Landlord, the vector's size is 20 since the Landlord can have at most 20 cards in hand. Similarly, we use a 15-dimension vector to represent the number of bombs in the current state. For historical moves, we consider the most recent 15 moves and concatenate the representations of every three consecutive moves; that is, the historical moves are encoded into a $5 \times 162$ matrix. The historical moves are fed into an LSTM, and we use the hidden representation in the last cell to represent the historical moves. If there are less than 15 moves historically, we use zero matrices for the missing moves. We summarize the encoded features of Landlord and each Peasant in Table 4 and Table 5, respectively.

*Table 4.* Features of the Landlord.

| | **Feature** | **Size** |
|---|---|---|
| Action | Card matrix of the action | 54 |
| State | Card matrix of hand cards | 54 |
| | Card matrix of the union of the other two players' hand cards | 54 |
| | Card matrix of the most recent move | 54 |
| | Card matrix of the the played cards of the first Peasant | 54 |
| | Card matrix of the the played cards of the second Peasant | 54 |
| | One-hot vector representing the number cards left of the first Peasant | 17 |
| | One-hot vector representing the number cards left of the second Peasant | 17 |
| | One-hot vector representing the number bombs in the current state | 15 |
| | Concatenated matrix of the most recent 15 moves | $5 \times 162$ |

*Table 5.* Features of the Peasants.

| | **Feature** | **Size** |
|---|---|---|
| Action | Card matrix of the action | 54 |
| State | Card matrix of hand cards | 54 |
| | Card matrix of the union of the other two players' hand cards | 54 |
| | Card matrix of the most recent move | 54 |
| | Card matrix of the most recent move performed by the Landlord | 54 |
| | Card matrix of the most recent move performed by the other Peasant | 54 |
| | Card matrix of the the played cards of the Landlord | 54 |
| | Card matrix of the the played cards of the other Peasant | 54 |
| | One-hot vector representing the number cards left of the Landlord | 20 |
| | One-hot vector representing the number cards left of the other Peasant | 17 |
| | One-hot vector representing the number bombs in the current state | 15 |
| | Concatenated matrix of the most recent 15 moves | $5 \times 162$ |

### C.2. Data Collection and Neural Architecture of Supervised Learning

In order to train an agent with supervised learning, we collect user data internally from a popular DouDizhu game mobile app. The users in the app have different leagues, which represent the strengths of the users. We filter out the raw data by only keeping the data generated by the players of the highest league to ensure the quality of the data. After filtering, we obtain 226,230 human expert matches. We treat each move as an instance and use a supervised loss to train the networks. The problem can be formulated as a classification problem, where we aim at predicting the action based on a given state, with a total of $27,472$ classes. However, we find in practice that most of the actions are illegal, and it is expensive to iterate over all the classes. Motivated by Q-network's design, we transform the problem into a binary classification task, as shown in Figure 12. Specifically, we use the same neural architecture as DouZero and add a Sigmoid function to the output. We then use binary cross-entropy loss to train the network. We randomly sample $10\%$ of the data for validation purposes and use the rest for training. We transform the user data into positive instances and generate negative instances based on the legal moves that are not selected. Eventually, the training data consists of $49,990,075$ instances. We further find that the data is imbalanced, where the number of negative instances is much larger than that of positive instances. Thus, we adopt a re-weighted cross-entropy loss based on the distribution of positive and negative instances. We find in practice that the re-weighted loss can improve the performance. We set the batch size to be 8096 and train 20 epochs. The prediction is made by choosing the action that leads to the highest score. We output the model that has the highest accuracy on the validation data. We do this process three times for the three positions, respectively. We plot the validation accuracy w.r.t. the number epochs in Figure 13. The network can achieve around $84\%$ accuracy for all positions.

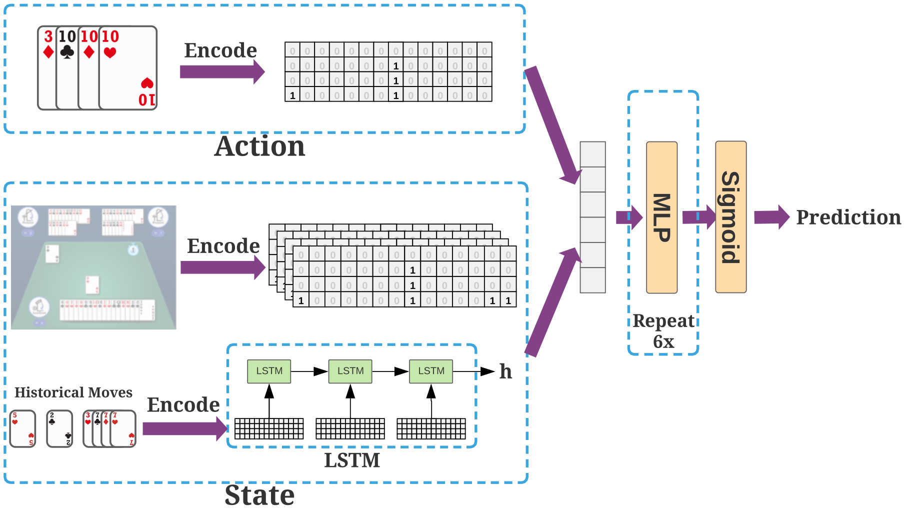
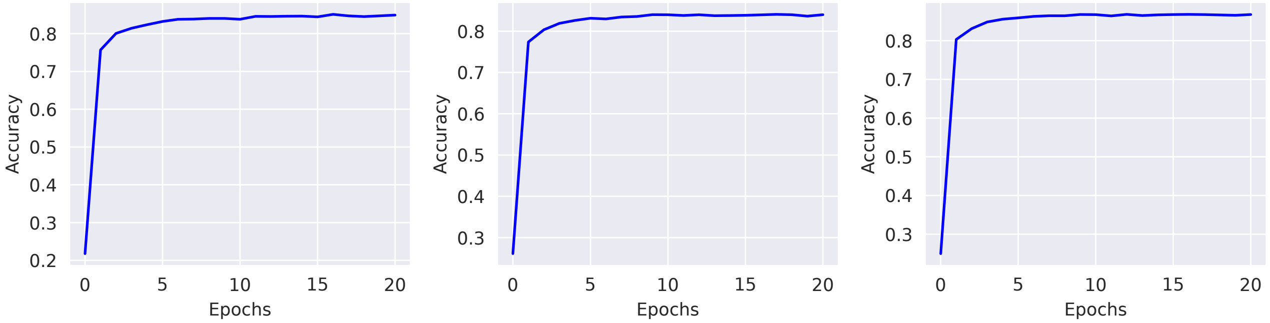


*Figure 12.* We use the same neural architecture as DouZero for SL. We add a Sigmoid function to the output and transform the problem into a binary classification task. The agent will perform the action that leads to the highest prediction score.



(a) Landlord (b) LandlordUp (c) LandlordDown

*Figure 13.* Accuracy w.r.t. the number of training epochs of SL for the three positions. LandlordUp stands for the Peasant that moves before the Landlord. LandlordDown stands for the Peasant that moves after the Landlord.

### C.3. Neural Architecture and Training Details of Bidding Network

The bidding phase's goal is to determine whether a player should become the landlord based on the strengths of the hand cards. This decision is much simpler than card-playing since the agent only needs to consider the hand cards and the other players' decisions, and we only need to make a binary prediction, i.e., whether we bid. At the beginning of the bidding phase, a randomly chosen player will decide whether to bid or not. Then the other two players will also choose whether to bid. If only one player bids, then that player will become the landlord. Suppose two or more players bid, the player who bids first will have the priority to decide whether she wants to become the landlord. We extract 128 features to represent hand cards and the players' moves, as summarized in Table 6. For the network architecture, we use a (512, 256, 128, 64, 32, 16) MLP. Like the supervised card playing agent, we add a Sigmoid function to the output and train the network with binary cross-entropy loss. We plot the validation accuracy w.r.t. the number epochs in Figure 14. The network can achieve $83.1\%$ accuracy.

*Table 6.* Features of the bidding network.

| **Feature** | **Size** |
|---|---|
| Card matrix of hand cards | 54 |
| A vector representing solos of ranks 3 to A | 12 |
| A vector representing pairs of ranks 3 to 2 | 13 |
| A vector representing trios of ranks 3 to 2 | 13 |
| A vector representing bombs of ranks 3 to 2 and the rocket | 14 |
| The number of cards of rank 2 and the jokers | 10 |
| A vector encoding historical bidding moves | 12 |
| Total | 128 |

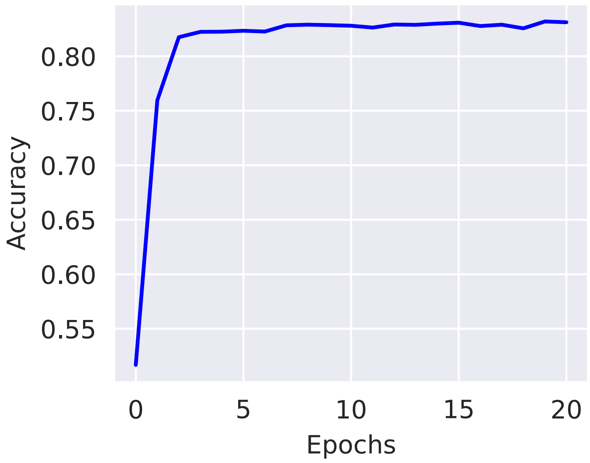


*Figure 14.* Accuracy w.r.t. the number of training epochs of the bidding network.

# D. Additional Results of DouZero

## D.1. Full WP and ADP Results for Landlord and Peasants

We report the results for Landlord and Peasants in Table 7 and Table 8 for WP and ADP, respectively. We observe that the advantage of DouZero for Peasants tends to be larger than that of Landlord. A possible explanation is that the two Peasants agents in DouZero have learned cooperation skills, which could be hardly covered by the heuristics and other algorithms.

*Table 7.* WP of DouZero and the baselines. L: WP of A as Landlord; P: WP of A as Peasants. If the average WP of L and P is higher than 0.5, we conclude that A outperforms B and highlight both L and P in boldface. The algorithms are ranked according to the number of the other algorithms that they beat.

| Rank | B | DouZero | | DeltaDou | | SL | | RHCP-v2 | | RHCP | | RLCard | | CQN | | Random | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | A | L | P | L | P | L | P | L | P | L | P | L | P | L | P | L | P |
| 1 | DouZero | .4159 | .5841 | **.4870** | **.6843** | **.5692** | **.7494** | **.6844** | **.8303** | **.7253** | **.8033** | **.8695** | **.9089** | **.7686** | **.8513** | **.9858** | **.9920** |
| 2 | DeltaDou | .3166 | .5130 | .4120 | .5880 | **.5130** | **.7211** | **.6701** | **.8165** | **.7048** | **.7899** | **.8563** | **.8955** | **.7326** | **.8351** | **.9871** | **.9960** |
| 3 | SL | .2506 | .4308 | .2789 | .5130 | .4072 | .5928 | **.5370** | **.6857** | **.5831** | **.6810** | **.7605** | **.8650** | **.6450** | **.7428** | **.9599** | **.9927** |
| 4 | RHCP-v2 | .1697 | .3156 | .1835 | .3299 | .3143 | .4630 | .4595 | .5405 | **.5134** | **.5165** | **.6813** | **.7018** | **.6313** | **.6116** | **.9519** | **.9821** |
| 5 | RHCP | .1967 | .2747 | .2101 | .2952 | .3190 | .4179 | .4835 | .4866 | .4971 | .5029 | **.6718** | **.6913** | **.6416** | **.5640** | **.9092** | **.9725** |
| 6 | RLCard | .0911 | .1305 | .1045 | .1437 | .1350 | .2395 | .2982 | .3187 | .3087 | .3282 | .4465 | .5535 | **.5839** | **.4603** | **.9314** | **.9539** |
| 7 | CQN | .1487 | .2314 | .1649 | .2674 | .2572 | .3550 | .3884 | .3687 | .4360 | .3584 | .5397 | .4161 | .5238 | .4762 | **.8566** | **.9213** |
| 8 | Random | .0080 | .0142 | .0040 | .0129 | .0073 | .0401 | .0179 | .0481 | .0025 | .0908 | .0461 | .0686 | .0787 | .1434 | .3461 | .6539 |

*Table 8.* ADP of DouZero and the baselines. L: ADP of A as Landlord; P: ADP of A as Peasants. If the average ADP of L and P is higher than 0, we conclude that A outperforms B and highlight both L and P in boldface. The algorithms are ranked according to the number of the other algorithms that they beat.

| Rank | B | DouZero | | DeltaDou | | SL | | RHCP-v2 | | RHCP | | RLCard | | CQN | | Random | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | A | L | P | L | P | L | P | L | P | L | P | L | P | L | P | L | P |
| 1 | DouZero | -0.435 | 0.435 | **-0.342** | **0.858** | **0.287** | **1.112** | **1.436** | **1.888** | **1.492** | **1.850** | **2.222** | **2.354** | **1.368** | **2.001** | **3.254** | **2.818** |
| 2 | DeltaDou | 0.342 | -0.858 | -0.476 | 0.476 | **0.268** | **1.038** | **1.297** | **1.703** | **1.312** | **1.715** | **2.270** | **2.648** | **1.218** | **1.849** | **3.268** | **2.930** |
| 3 | SL | -1.112 | -0.287 | -1.038 | -0.268 | -0.364 | 0.364 | **0.564** | **1.142** | **0.658** | **1.114** | **1.652** | **1.990** | **0.878** | **1.196** | **3.026** | **2.415** |
| 4 | RHCP-v2 | -1.888 | -1.436 | -1.703 | -1.297 | -1.142 | -0.564 | -0.209 | 0.209 | **0.074** | **0.029** | **1.011** | **1.230** | **0.750** | **0.677** | **2.638** | **2.624** |
| 5 | RHCP | -1.850 | -1.492 | -1.715 | -1.312 | -1.114 | -0.658 | -0.029 | -0.074 | -0.007 | 0.007 | **1.190** | **1.328** | **0.927** | **-0.432** | **2.722** | **2.717** |
| 6 | RLCard | -2.354 | -2.222 | -2.648 | -2.270 | -1.990 | -1.652 | -1.230 | -1.011 | -1.328 | -1.190 | -0.266 | 0.266 | **0.474** | **-0.138** | **2.630** | **2.312** |
| 7 | CQN | -2.001 | -1.368 | -1.849 | -1.218 | -1.196 | -0.878 | -0.677 | -0.750 | 0.432 | -0.927 | 0.138 | -0.474 | 0.056 | -0.056 | **1.832** | **1.992** |
| 8 | Random | -2.818 | -3.254 | -2.930 | -3.268 | -2.415 | -3.026 | -2.624 | -2.638 | -2.717 | -2.722 | -2.312 | -2.629 | -1.991 | -1.832 | -0.883 | 0.883 |

## D.2. Comparison of Using WP and ADP as Objectives

In our experiments, we find that the agents will learn different styles of card playing strategies when using WP and ADP as objectives. Specifically, we observe that the agents trained with WP play more aggressively about bombs even if it will lose. We visualize this phenomenon in Appendix F.4. The possible explanation is that a bomb will not double the points so that playing a bomb or rocket will not harm WP. Aggressively playing bombs may benefit WP since they will dominate other payers, which allows them to play hand cards freely. In contrast, the agents trained with ADP tend to be very cautious of playing bombs since improperly playing a bomb may double the ADP loss if the agents lose the game in the end.

To better interpret the differences between WP and ADP, we show the results of the agents trained with ADP and WP against the baselines. In Table 9, we report the results of DouZero trained with ADP using WP as the metric. We observe that the performance is slightly worse than that in Table 7. In Table 10, we show the results of DouZero trained with WP using ADP as the metric. Similarly, the ADP result is slightly worse than thate in Table 8. We observe similar results if considering the bidding phase (see Table 11 and Table 12). Finally, we launch a head-to-head competition of these two agents in Table 13. The results again verify that the agents trained with WP are better in terms of WP and vice versa. The above results suggest that WP and ADP are indeed different and encourage different card playing strategies.

In addition to WP and ADP, some other metrics could also be adopted in real-word DouDizhu completions. For example, some apps allow users to double the base score at the beginning of a game. We argue that we should adjust the objectives to achieve the best performance according to different scenarios.

*Table 9.* WP of DouZero against baselines when using ADP as the reward. L: WP of A as Landlord; P: WP of A as Peasants. If the average WP of L and P is higher than 0.5, we conclude that A outperforms B and highlight both L and P in boldface.

| B | DouZero (ADP) | | DeltaDou | | SL | | RHCP-v2 | | RHCP | | RLCard | | CQN | | Random | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| A | L | P | L | P | L | P | L | P | L | P | L | P | L | P | L | P |
| DouZero (ADP) | .4281 | .5719 | **.4177** | **.6319** | **.5039** | **.6815** | **.6615** | **.7543** | **.6950** | **.7628** | **.8416** | **.8668** | **.7198** | **.8280** | **.9801** | **.9895** |

*Table 10.* ADP of DouZero against baselines when using WP as the reward. L: ADP of A as Landlord; P: ADP of A as Peasants. If the average ADP of L and P is higher than 0, we conclude that A outperforms B and highlight both L and P in boldface.

| B | DouZero (WP) | | DeltaDou | | SL | | RHCP-v2 | | RHCP | | RLCard | | CQN | | Random | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| A | L | P | L | P | L | P | L | P | L | P | L | P | L | P | L | P |
| DouZero (WP) | -0.411 | 0.411 | **-0.360** | **0.664** | **0.224** | **1.001** | **1.252** | **1.880** | **1.378** | **1.794** | **2.094** | **2.298** | **1.418** | **1.872** | **2.947** | **2.518** |

*Table 11.* DouZero against DeltaDou and SL when using ADP as reward with bidding network.

| | DouZero (ADP) | DeltaDou | SL |
|---|---|---|---|
| WP | **0.535** | 0.477 | 0.407 |
| ADP | **0.323** | -0.004 | -0.320 |

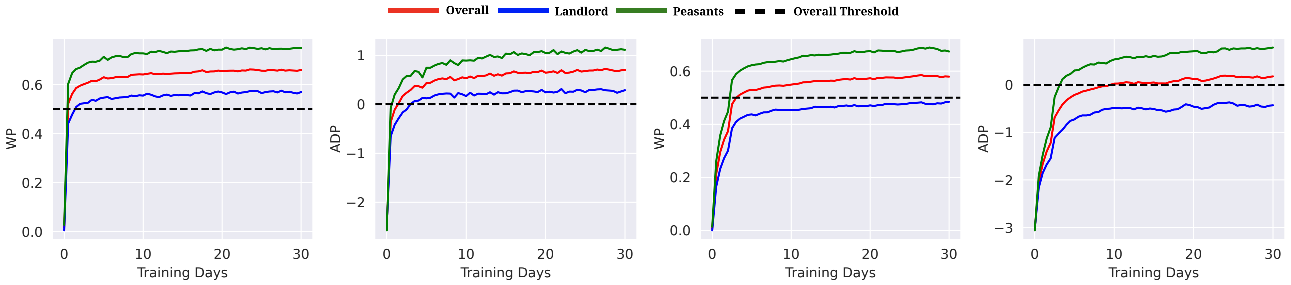
*Table 12.* DouZero against DeltaDou and SL when using WP as reward with bidding network.

| | DouZero (WP) | DeltaDou | SL |
|---|---|---|---|
| WP | **0.580** | 0.461 | 0.381 |
| ADP | **0.315** | 0.075 | -0.390 |

*Table 13.* Head-to-head comparison between using ADP and WP as objectives. DouZero (ADP) outperforms DouZero (WP) in terms of ADP but is worse than DouZero (WP) in terms of WP. The agents tend to learn different skills with different objectives.

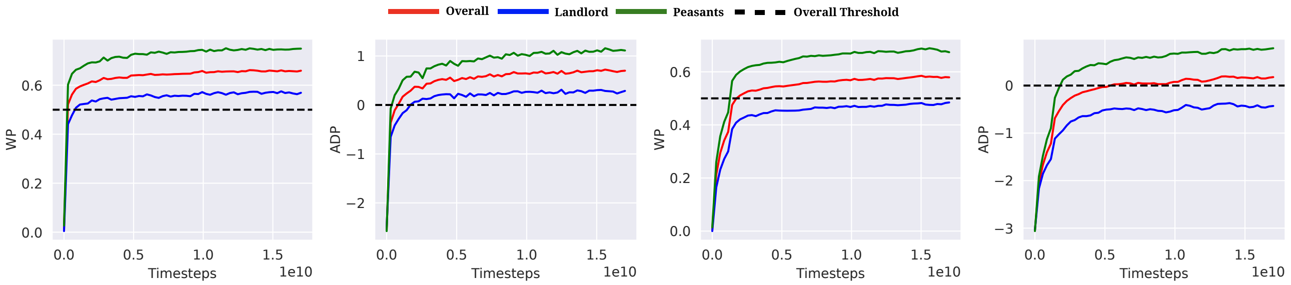
| | ADP | | WP | |
|---|---|---|---|---|
| | L | P | L | P |
| DouZero (ADP) vs DouZero (WP) | **-0.3101** | **0.4476** | .3617 | .5151 |

## D.3. Additional Results of Learning Progress



(a) WP against SL (b) ADP against SL (c) WP against DeltaDou (d) ADP against DeltaDou

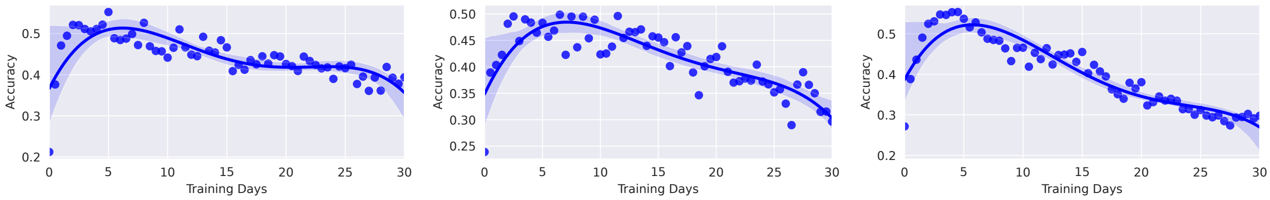
*Figure 15.* WP and ADP of DouZero against SL and DeltaDou w.r.t. the number of training days. DouZero outperforms SL with 2 days of training, i.e., the overall WP is larger than the threshold of 0.5 and the overall ADP is larger than the threshold of 0, and surpasses DeltaDou within 10 days, using a single server with four 1080 Ti GPUs and 48 processors.



(a) WP against SL (b) ADP against SL (c) WP against DeltaDou (d) ADP against DeltaDou

*Figure 16.* WP and ADP of DouZero against SL and DeltaDou w.r.t. the number of training timesteps, i.e., the number of actions played by the agent. DouZero outperforms SL with around $5 \times 10^8$ training timesteps, i.e., the overall WP is larger than the threshold of 0.5 and the overall ADP is larger than the threshold of 0, and surpasses DeltaDou within $5 \times 10^9$ training timesteps, using a single server with four 1080 Ti GPUs and 48 processors.

## D.4. Full Results of DouZero on Expert Data

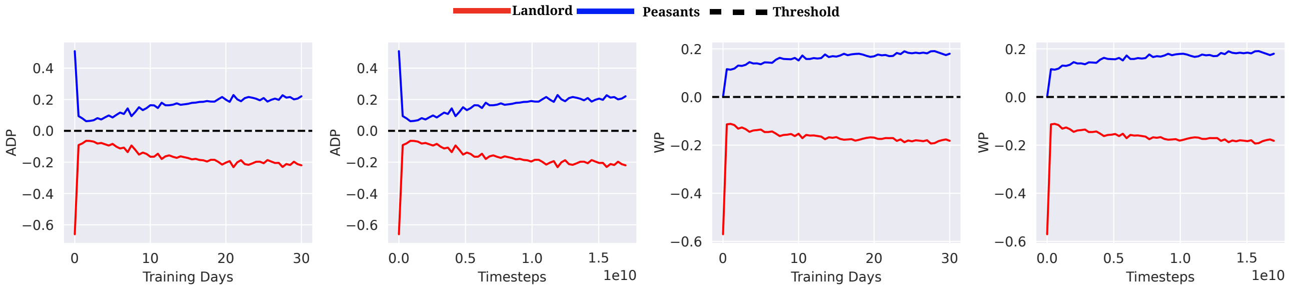


(a) Landlord (b) LandlordUp (c) LandlordDown

*Figure 17.* Accuracy for the three positions on the human data w.r.t. the number of training days for DouZero. We fit the data points with a polynomial with four terms for better visualizing the trend. LandlordUp stands for the Peasant that moves before the Landlord. LandlordDown stands for the Peasant that moves after the Landlord. The accuracies of SL for the Landlord, LandlordUp, LandlordDown are 83.3%, 86.1%, 83.1%, respectively. DouZero aligns with human expertise in the beginning training stages but discovers novel strategies beyond human knowledge in the later training stages.

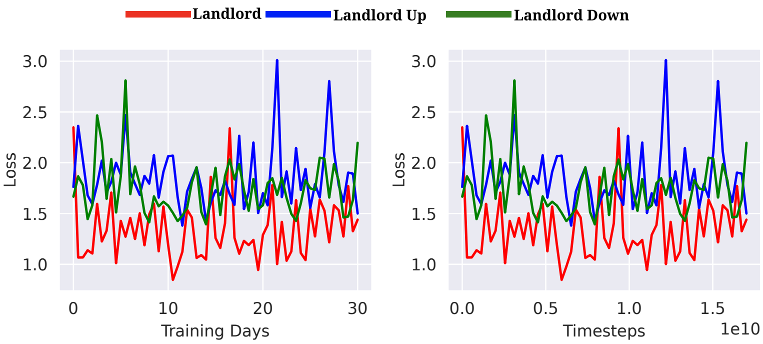
## D.5. Training Curves of DouZero

In this work, we train three DouZero agents with self-play for all the positions (i.e., the Landlord and the two Peasants) without considering the bidding phase. For each episode, a deck will be randomly generated. Then the three agents will perform self-play with partially-observed states. The generated episodes will be passed to the learner process to update the three DouZero agents. In what follows, we show the self-play rewards and the losses throughout the training process.



(a) ADP w.r.t. training days (b) ADP w.r.t. timesteps (c) WP w.r.t. training days (d) WP w.r.t. timesteps

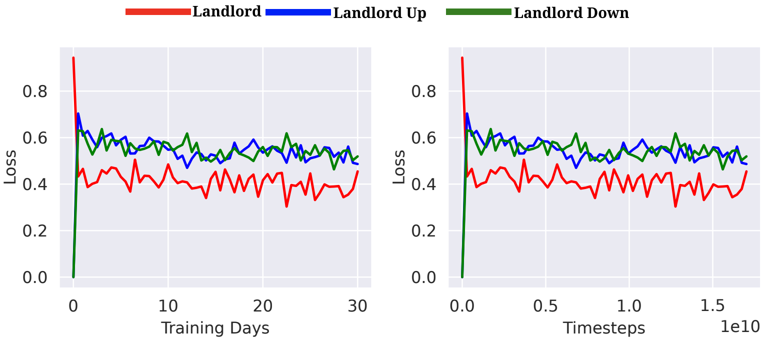
*Figure 18.* ADP and WP w.r.t. training days and the number of timesteps for the Landlord and Peasants of DouZero during the training progress. At the early stage, the Peasants win the Landlord by a small margin. The peasants become stronger and stronger compared with the Landlord in the later training stages.



(a) Loss w.r.t. training days

(b) Loss w.r.t. timesteps

*Figure 19.* Losses for different positions for DouZero trained with ADP as rewards. LandlordUp stands for the Peasant that moves before the Landlord. LandlordDown stands for the Peasant that moves after the Landlord.



(a) Loss w.r.t. training days

(b) Loss w.r.t. timesteps

*Figure 20.* Losses for different positions for DouZero trained with WP as rewards. LandlordUp stands for the Peasant that moves before the Landlord. LandlordDown stands for the Peasant that moves after the Landlord.

# E. More Details of Botzone

Botzone is a comprehensive multi-game, multi-agent online game AI platform hosted by AILab, Peking University[10]. Besides DouDizhu, Botzone supports more than 20 games, including Go, Mahjong, and Ataxx, to name a few. Botzone currently has more than 3,500 users and has hosted various games, from in-class and campus contests within Peking University to nation- and worldwide game AI competitions such as the recent IJCAI 2020 Mahjong AI competition, which attracted top AI researchers worldwide.

On the Botzone platform, users upload their bot program as a virtual agent to compete with other bots in a selected game. In order to do so, a user can either nominate opponents and manually start a game or add their bots to the Botzone Elo system, where games among bots will be scheduled automatically. A bot added to the Botzone Elo system will also be associated with the so-called Elo rating score and will be added to the rank list (leaderboard) of the game she plays. Botzone assigns an initial Elo rating score of 1000 to a new bot in the Elo, and the score is updated after each time the bot played an Elo rating game, as long as the bot remains in the Elo system.

## E.1. Interacting with Botzone

Botzone provides a “Judge” program that runs in the background and interacts with bot programs of users. A bot receives a request to act from the “Judge” every time he has to take an action, i.e., it is her turn to play cards. In DouDizhu, the input sent to the bot contains three parts of information: her own hand cards, the public card, and a sequence of cards played by each player. This is consistent with the incomplete information known to a human player in a common DouDizhu game setting.

The platform also sets constraints on the file size, memory, and running time of bots. Each decision made by the bot has to be completed within 1 second (or 6 seconds for a Python program) with no more than 256 MB of memory. The model size is limited to 140 MB.

## E.2. Botzone Ranking Rules

Botzone maintains a leaderboard for each game, which ranks all the bots in the Botzone Elo system by their Elo rating scores in descending order. Every five minutes, Botzone schedules match with randomly selected bots, with priority given to new bots with recent updates. The Elo rating score of participating bots will be updated after the match.

In the Botzone Elo of DouDizhu (named “FightTheLandlord” on the Botzone platform), each game is played by two bots, with one bot acting as the Landlord and the other as Peasants. A pair of games are played simultaneously, in which the two bots will play different roles; that is, the bot who plays the Landlord in one game will play the Peasants in another, and vice versa. The two games form a match, and the Elo rating of each bot is updated according to the match outcome, as well as their relative ratings.

The game score of a bot is mainly determined by whether she wins or loses a game. The winning bot receives a score of two points, while the losing bot receives a score of zero[11]. To encourage more complicated card-playing strategies, Botzone associates small points with categories of the cards played by a bot and adds that to the game score. The total game score is thus the *winning point* (2 for the winner and 0 for the loser) plus the *card-playing advantage*, the sum of weights assigned according to categories of cards (Table 14) played by a bot in the entire game divided by 100. Since by convention, two Peasants always receive the same game score, Peasants’ game score is the average of their individual scores.

The match score is determined by the sum of game scores of the two games in the round, which further determines how the Elo rating will change for each player. If the match score of one bot is higher than the other, then the bot is considered the winner of this match. The winning bot will receive an increase in its Elo rating, while the same amount of rating points will be taken off from the losing bot.

## E.3. Discussion of Ranking Stability

Although Elo rating is generally considered a stable measurement of relative strength among a pool of players in games like Chess and Go, DouDizhu Elo ranking on Botzone suffers from some fluidity. This could be attributed to the nature of the high variance of the game and also the design of Botzone Elo. Firstly, the game outcomes of DouDizhu relies on the luck of

[10] https://wiki.botzone.org.cn/index.php?title=%E9%A6%96%E9%A1%B5/en

[11] https://wiki.botzone.org.cn/index.php?title=FightTheLandlord

*Table 14.* Summary of weights assigned to each category in Botzone.

| **Action Type** | **Weight** |
|---|---|
| Solo | 1 |
| Pair | 2 |
| Trio (or with Solo / Pair) | 4 |
| Chain of Solo | 6 |
| Chain of Pair | 6 |
| Chain of Trio | 8 |
| Plane with Solo / Pair | 8 |
| Quad with Solo / Pair | 8 |
| Space shuttle A (2 consecutive Quad) | 10 |
| Bomb | 10 |
| Rocket | 16 |
| Space shuttle B (more than 3 consecutive Quads) | 20 |
| Pass | 0 |

initial hand cards. In particular, if a player with bad hand cards is playing the Landlord, he will have a low chance to win a game. In practice, the bidding phase could compensate for this randomness, such that the player with bad initial hand cards can choose not to bid for the Landlord. However, Botzone does not incorporate the bidding phase into the game playing; rather, the Landlord position is specified even before the dealing phase happened. Secondly, although two bots exchange roles between games in each match, these two games are not initialized with the same hand cards. It is not rare to see that one bot was assigned bad initial hand cards in both games, making it infeasible for her to win the match. Finally, Elo rating games are not scheduled as frequently on Botzone, potentially due to limited server resources. We observe that, on average, DouZero has the chance to play one Elo rating game about every 2 hours. As such, it might take a long time for a bot to achieve a stable ranking. With bots continuously added to or leaving the Elo system, it might be just impossible to observe absolute stable ranking. Nonetheless, since ranked top on Botzone for the first time on October 30, 2020, DouZero has remained in the top-5 most of the time (at the time of submission deadline, DouZero was still ranked first with around 1600 points). While the rank of DouZero is impacted by the high variance of the BotZone platform, DouZero has maintained an Elo rating score of at least 1480 points during the months between October 30, 2020, to the ICML submission deadline, suggesting that DouZero has at least 95% chance of winning in a match with an average bot.

## F. Additional Case Studies

In this section, we conduct case studies for DouZero. We show both per-step decision with figures and the logs of full games. For simplicity, we use “T” to denote “10”,“P” to denote “PASS”, “B” to denote Black Joker, and “R” to denote Red Joker. Each move is represented as “position:move”, where position can be “L” for Landlord, “U” for LandlordUp (i.e., the Peasant that moves before the Landlord), “D” for LandlordDown (i.e., the Peasant that moves after the Landlord). For example, ”L:3555” means Landlord plays 3555, and “U:T” means LandlordUp plays 10. The initial hands are represented as “H:Landlord Hand; LandlordDown Hand; LandlordUp Hand”. The moves and the hands are separated by “,”. **Note that except Section F.4, we focus on the agent with WP as objective. Thus, the agents tend to ambitiously play bombs even when they will lose, and will not try to play more bombs when they think they will win.**

### F.1. Strategic Thinking of the Agents

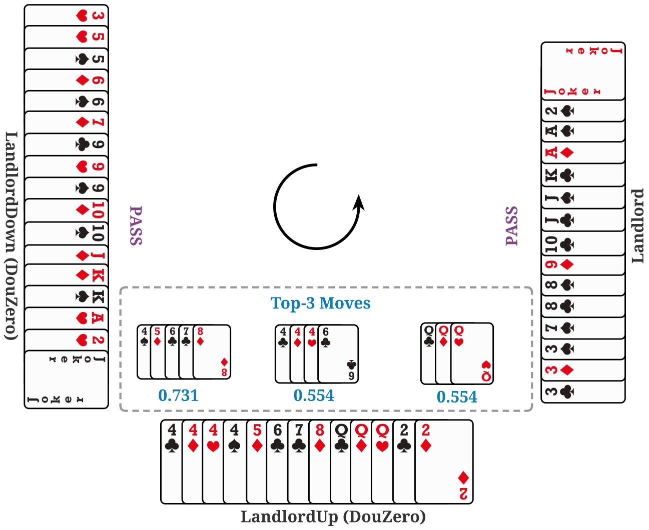


*Figure 21.* Case 1: Strategic thinking (turn 7). The LandlordUp has 4444 in hand. However, DouZero strategically chooses to break the bomb and play 45678. This is because playing this Chain of Solo can empty the hand more quickly. **Full game:** H:333456778889TJJKAA2R; 355667999TTJKKA2B; 4445678TJQQQQKA22, L:45678, D:P, U:TJQKA, L:P, D:P, U:45678, L:789TJ, D:P, U:P, L:3338, D:999J, U:4QQQ, L:P, D:P, U:22, L:P, D:P, U:4.

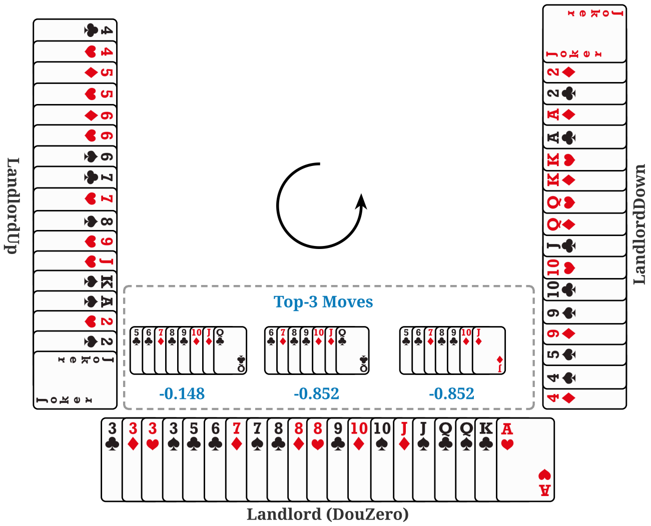


*Figure 22.* Case 2: Strategic thinking (turn 1). In this case, 56789TJQ is a very good move because the agent can play another Chain of Solo afterwards, i.e., TJQKA. **Full game:** H:333356778889TTJJQQKA; 44599TTJQQKKAA22R; 44556667789JKA22B, L:56789TJQ, D:P, U:P, L:TJQKA, D:P, U:P, L:7, D:J, U:K, L:P, D:P, U:44, L:88, D:99, U:22, L:3333.

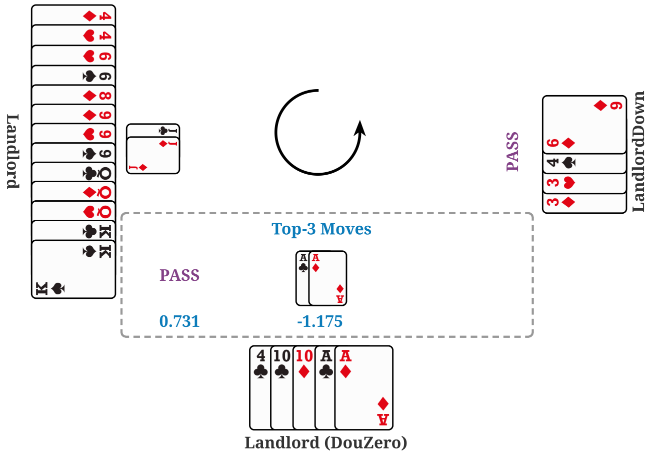


*Figure 23.* Case 3: Strategic thinking (turn 31). The Landlord has two pairs (AA and TT) and a small Solo in hand. DouZero chooses not to play AA in this turn. This is a nice move because there is a pair of K out there. If the Landlord plays AA, then the TT will be dominated by the KK in later turns. DouZero patiently chooses PASS and wins the game eventually. **Full game:** H:33455556889TTJAA22BR; 334677778TTJQKA22; 44668999JJQQQKKKA, L:33, D:22, U:P, L:BR, D:P, U:P, L:6, D:8, U:A, L:2, D:7777, U:P, L:P, D:TJQKA, U:P, L:5555, D:P, U:P, L:9, D:T, U:P, L:J, D:P, U:K, L:2, D:P, U:P, L:88, D:P, U:JJ, L:P, D:P, U:44999, L:P, D:P, U:66QQQ, L:P, D:P, U:KK, L:AA, D:P, U:P, L:TT, D:P, U:P, L:4.

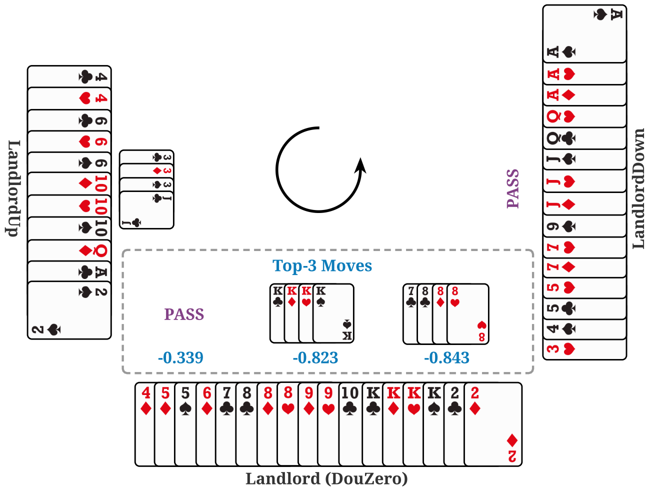


*Figure 24.* Case 4: Strategic thinking (turn 13). When the LandlordUp plays 333J, DouZero chooses not to play 888. This is a nice move because playing 888 will break a Chain of Solo, i.e., 456789T. **Full game:** H:45567788899TQKKKK22B; 34557789JJJQQAAA2; 333446669TTTJQA2R, L:7, D:8, U:9, L:Q, D:2, U:P, L:B, D:P, U:R, L:P, D:P, U:333J, L:P, D:P, U:666Q, L:P, D:P, U:TTTA, L:P, D:P, U:44, L:55, D:77, U:P, L:88, D:QQ, U:P, L:22, D:P, U:P, L:49KKKK, D:P, U:P, L:6789T.

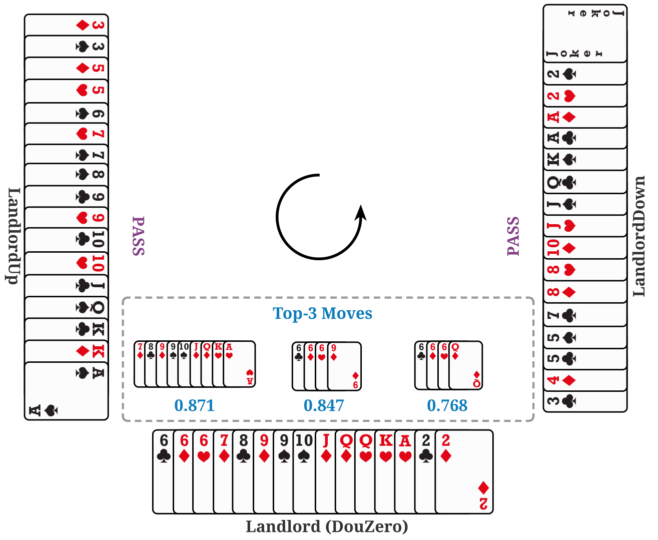


*Figure 25.* Case 5: Strategic thinking (turn 4). DouZero can choose a move from many possible legal moves. The top-1 move is nice because it is a very long Chain of Solo. The second and the third moves are also very good because they choose a good kicker. **Full game:** H:34446667899TJQQKA22R; 3455788TJJQKAA22B; 3355677899TTJQKKA, L:3444, D:P, U:P, L:789TJQKA, D:P, U:P, L:6669, D:P, U:P, L:22, D:P, U:P, L:Q, D:B, U:P, L:R.

## F.2. Cooperation of Peasants

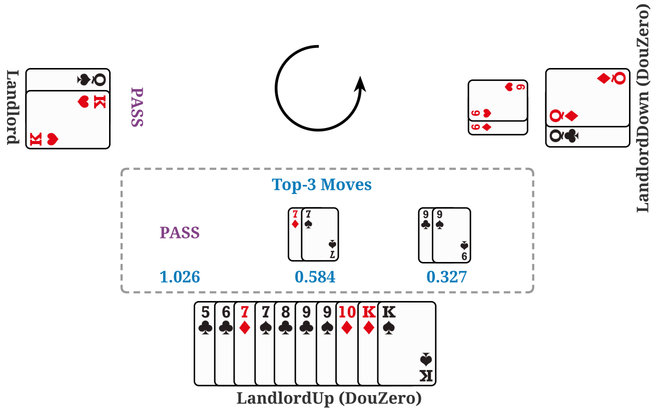


*Figure 26.* Case 1: Cooperation of Peasants (turn 29). Although the Landlord has much better hand than the Peasants (a Bomb plus a Rocket), the Peasants manage to win the game with cooperation. In this turn, the LandlordUp chooses PASS so that the LandlordDown can empty her hand. DouZero has learned not to fight against the teammate. **Full game:** H:3335556799JJJJQK22BR; 44445677899TQKKAA; 3667888TTTQQKAA22, L:3336, D:P, U:3888, L:JJJJ, D:P, U:P, L:5557, D:P, U:7TTT, L:BR, D:P, U:P, L:99, D:AA, U:22, L:P, D:P, U:AA, L:22, D:4444, U:P, L:P, D:Q, U:K, L:P, D:P, U:66, L:P, D:P, U:QQ.

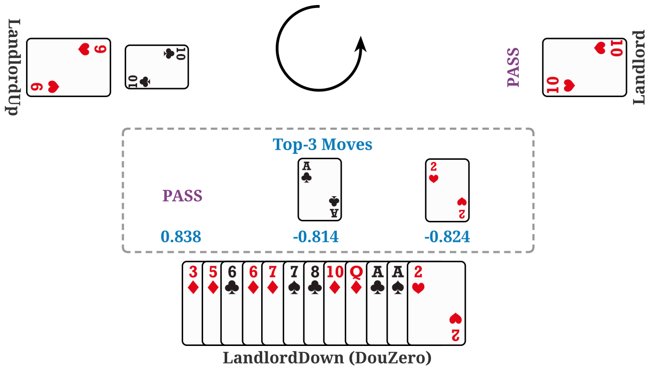


*Figure 27.* Case 2: Cooperation of Peasants (turn 33). The LandlordUp plays a T. DouZero chooses PASS because there is no card out there larger than T in rank. This suggests that DouZero has learned to reason about the cards that have not been played. **Full game:** H:334444556689TTJJJQ2R; 3577889TQQKKKKAA2; 356677899TJQAA22B, L:33444455, D:7788KKKK;, U:P, L:P, D:3, U:J, L:2, D:P, U:B, L:R, D:P, U:P, L:89TJQ, D:P¡ U:P, L:66, D:QQ, U:P, L:P, D:5, U:2, L:P, D:P, U:99, L:JJ, D:AA, D:P, L:P, D:2, U:P, L:P, D:T, U:P, L:P, D:9.

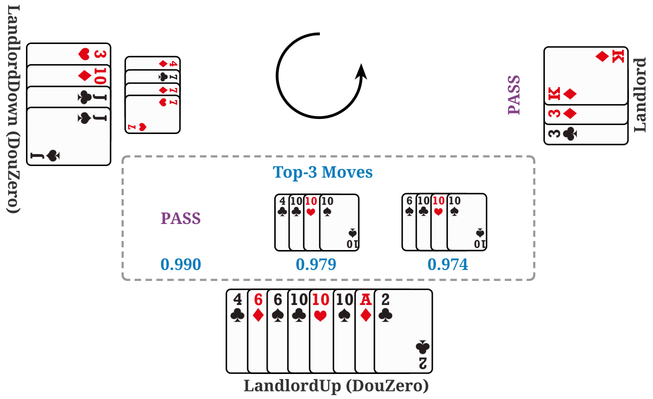


*Figure 28.* Case 3: Cooperation of Peasants (turn 36). The LandlordDown plays a Trio with Solo. While LandlordUp has a very good hand and can win the game by itself, DouZero chooses PASS to let her teammate win. **Full game:** H:3344566788999JQK222; 34577788TJJQQQKAA; 3455669TTTKKAA2BR; L:45678, D:P, U:P, L:4999, D:5QQQ, U:P, L:6222, D:P, U:BR, L:P, D:P, U:55, L:JJ, D:P, U:KK, L:P, D:P, U:9, L:Q, D:K, U:A, L:P, D:P, U:3, L:8, D:A, U:P, L:P, D:88, U:P, L:P, D:A, U:P, L:P, D:4777, U:P, L:P, D:3, U:T, L:K, D:P, U:A, L:P, D:P, U:T, L:P, D:P, U:T, L:P, D:J, U:P, L:P, D:J, U:P, L:P, D:T.

### F.3. Comparison of the Models in Early Stage and Later Stage

*Table 15.* Case 1: Comparison of DouZero in early stage and later stage. DouZero plays the Landlord position on the same deck with the same opponents. DouZero loses in the early stage but wins the game in the later stage. DouZero in the later stage tends to perform a better planning of the hand. Specifically, DouZero in the later stage tends to first play small pairs, such as 88 and TT, so that it can easily empty the hand later.

| | Logs |
|---|---|
| Early stage | H:455557777889TTKKAA22; 3446668999QQKA2BR; 333468TTJJJJQQKA2, L:88, D:QQ, U:P, L:KK, D:P, U:P, L:455559, D:P, U:46JJJJ, L:P, D:P, U:3338, L:P, D:3666, U:P, L:P, D:44999, U:P, L:P, D:8, U:K, L:A, D:2, U:P, L:P, D:K, U:P, L:A, D:BR, U:P, L:P, D:A |
| Later stage | H:455557777889TTKKAA22; 3446668999QQKA2BR; 333468TTJJJJQQKA2, L:TT, D:QQ, U:P, L:KK, D:P, U:P, L:88, D:99, U:TT, L:AA, D:P, U:P, L:477779, D:P, U:46JJJJ, L:P, D:P, U:3338, L:5555, D:BR, U:P, L:P, D:3666, U:P, L:P, D:8, U:K, L:2, D:P, U:P, L:2 |

*Table 16.* Case 2: Comparison of DouZero in early stage and later stage. DouZero plays the Landlord position on the same deck with the same opponents. In the early stage, DouZero keeps playing PASS because it does not want to break the Rocket. As result, DouZero loses the game. In the later stage, DouZero smartly breaks the Rocket to dominate other Solos, and eventually wins the game.

| | Logs |
|---|---|
| Early stage | H:3355556778889JJQKABR; 344446679JJQQKA22; 367899TTTTQKKAA22, L:9, D:K, U:P, L:A, D:2, U:P, L:P, D:3, U:9, L:K, D:A, U:P, L:P, D:4, U:Q, L:P, D:P, U:6789T, L:P, D:P, U:3TTT, L:P, D:P, U:KK, L:P, D:P, U:AA, L:P, D:P, U:22 |
| Later stage | H:3355556778889JJQKABR; 344446679JJQQKA22; 367899TTTTQKKAA22, L:33, D:66, U:KK, L:P, D:P, U:6789T, L:P, D:P, U:3TTT, L:P, D:P, U:9, L:Q, D:K, U:P, L:A, D:2, U:P, L:P, D:3, U:Q, L:K, D:A, U:P, L:B, D:P, U:P, L:6, D:7, U:A, L:R, D:P, U:P, L:555577JJ, D:P, U:P, L:8889 |

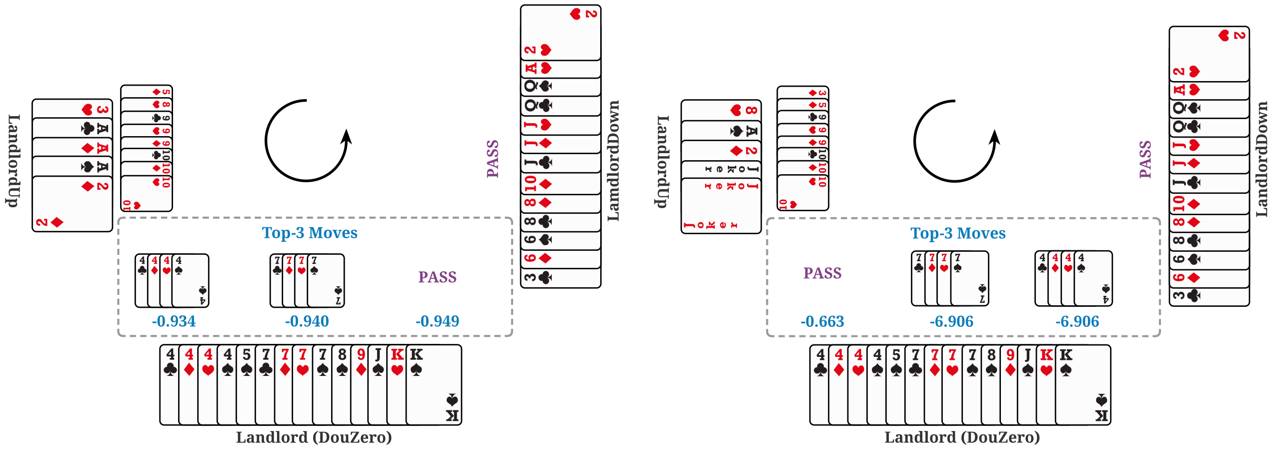
*Table 17.* Case 3: Comparison of DouZero in early stage and later stage. DouZero plays the Landlord position on the same deck with the same opponents. In the early stage, the first move of DouZero is 9. However there is a 4 in the hand, which causes troubles in later turns. In the later stage, DouZero plays in a different style by starting with 33 and finally wins the game. Although playing 9 seems to be not bad, it may lead to losing the game later.

| | Logs |
|---|---|
| Early stage | H:33455556889TTJAA22BR; 334677778TTJQKA22; 446688999JJQQQKKKA, L:9, D:T, U:A, L:2, D:7777, U:P, L:P, D:TJQKA, U:P, L:5555, D:P, U:P, L:6, D:8, U:P, L:J, D:2, U:P, L:R, D:P, U:P, L:88, D:P, U:JJ, L:AA, D:P, U:P, L:33, D:P, U:66, L:TT, D:P, U:KK, L:P, D:P, U:8999, L:P, D:P, U:QQQK, L:P, D:P, U:44 |
| Later stage | H:33455556889TTJAA22BR; 334677778TTJQKA22; 446688999JJQQQKKKA, L:33, D:22, U:P, L:BR, D:P, U:P, L:6, D:8, U:A, L:2, D:7777, U:P, L:P, D:TJQKA, U:P, L:5555, D:P, U:P, L:9, D:T, U:P, L:J, D:P, U:K, L:2, D:P, U:P, L:88, D:P, U:JJ, L:P, D:P, U:44999, L:P, D:P, U:66QQQ, L:P, D:P, U:KK, L:AA, D:P, U:P, L:TT, D:P, U:P, L:4 |

*Table 18.* Case 4: Comparison of DouZero in early stage and later stage. DouZero plays the LandlordUp and LandlordDown positions on the same deck with the same opponent. While DouZero wins both games. DouZero in the later stage looks more reasonable. In turn 16, DouZero in the early stage chooses to break a pair of 2, which is hard to be explained. DouZero in the later stage tends to play more smoothly and wins the games quickly.

| | Logs |
|---|---|
| Early stage | H:3455567788TJJQKKKKA2; 33446678899JQQQA2; 3456799TTTJAA22BR, L:345678, D:P, U:P, L:TJQKA, D:P, U:P, L:55KKK, D:P, U:P, L:7, D:P, U:R, L:P, D:P, U:2, L:P, D:P, U:34567, L:P, D:P, U:J, L:2, D:P, U:B, L:P, D:P, U:2, L:P, D:P, U:AA, L:P, D:P, U:99TTT |
| Later stage | H:3455567788TJJQKKKKA2; 33446678899JQQQA2; 3456799TTTJAA22BR, L:345678, D:P, U:P, L:TJQKA, D:P, U:P, L:55KKK, D:P, U:BR, L:P, D:P, U:99TTT, L:P, D:P, U:34567, L:P, D:P, U:AA, L:P, D:P, U:22, L:P, D:P, U:J |

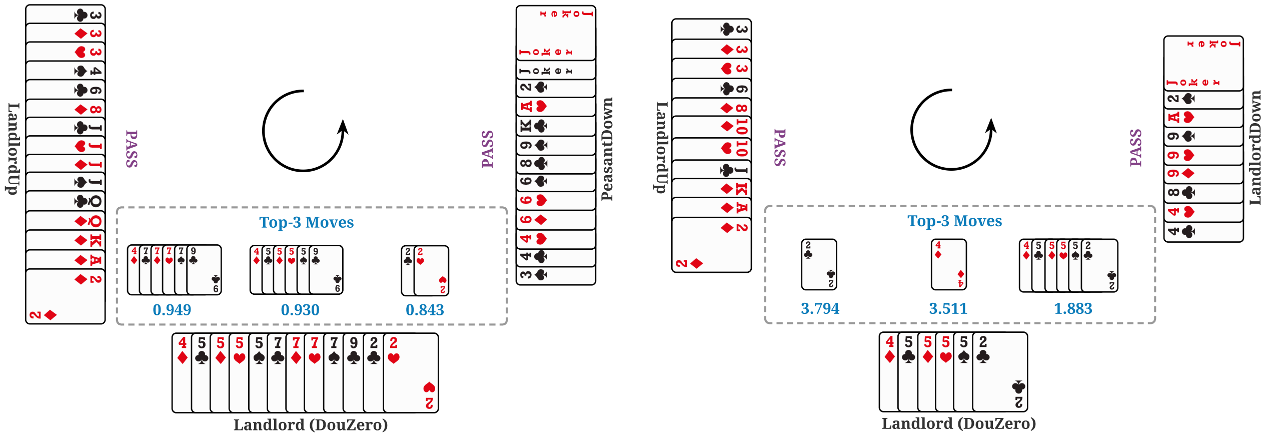
## F.4. Comparison of Using WP and ADP as Objectives



(a) WP (turn 13)

(b) ADP (turn 10)

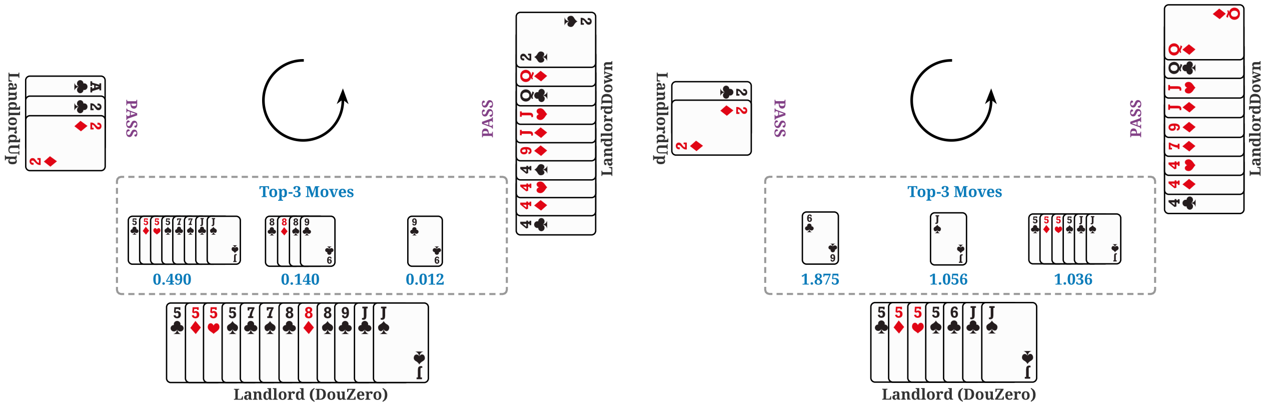
*Figure 29.* Case 1: Comparison of using WP and ADP as objectives. The two agents play the same deck twice. It is difficult for the Landlord to win this game since the LandlordUp has a very good hand. When LandlordUp plays a Plane with Solo, WP agent tends to ambitiously play bombs because playing bombs has no cost. However, ADP agent tends to be very cautious of playing bombs since it may lead to a larger loss of ADP. **Full game of (a):** H:3344445666689JQQKK22; 3556688TJJJQQKKA2; 35668999TTTAAA2BR, L:33, D:55, U:66, L:QQ, D:KK, U:P, L:22, D:P, U:BR, L:P, D:P, U:58999TTT, L:7777, D:P, U:P, L:8, D:T, U:2, L:P, D:P, U:3AAA. **Full game of (b):** H:3344445666689JQQKK22; 3556688TJJJQQKKA2; 35668999TTTAAA2BR, L:33, D:55, U:66, L:KK, D:P, U:AA, L:P, D:P, U:35999TTT, L:P, D:P, U:8, L:J, D:A, U:2, L:P, D:P, U:BR, L:P, D:P, U:A.



(a) WP (turn 13)

(b) ADP (turn 25)

*Figure 30.* Case 2: Comparison of using WP and ADP as objectives. The two agents play the same deck twice. While both agents win this game, they have different styles. The WP agent plays Quad with Solo instead of a bomb because it can empty the hand more quickly. This is reasonable since playing one more bomb will not double WP. In contrast, the ADP agent first plays a 2 so that it can play a bomb later. The ADP agent will try to play every bomb when it thinks it can win the game. **Full game of (a):** H:455557777889TTKKAA22; 3446668999QQKA2BR; 333468TTJJJJQQKA2, L:TT, D:QQ, U:P, L:KK, D:P, U:P, L:88, D:99, U:TT, L:AA, D:P, U:P, L:477779, D:P, U:46JJJJ, L:P, D:P, U:3338, L:5555, D:BR, U:P, L:P, D:3666, U:P, L:P, D:8, U:K, L:2, D:P, U:P, L:2. **Full game of (b):** H:455557777889TTKKAA22; 3446668999QQKA2BR; 333468TTJJJJQQKA2, L:88, D:QQ, U:P, L:KK, D:P, U:P, L:TT, D:P, U:QQ, L:AA, D:P, U:P, L:9, D:K, U:P, L:2, D:B, U:P, L:P, D:3666, U:4JJJ, L:7777, D:P, U:P, L:2, D:R, U:P, L:5555, D:P, U:P, L:4.



(a) WP (turn 34)

(b) ADP (turn 43)

*Figure 31.* Case 3: Comparison of using WP and ADP as objectives. The two agents play the same deck twice. In this game, the Landlord is very likely to win. The WP agent chooses a more conservative move by playing Quad with Pair to quickly empty the hand. In contrast, the ADP agent plays a small Solo first so that it can play 5555 later to double the ADP. **Full game of (a):** H:3355556778889JJQKABR; 344446679JJQQKA22; 367899TTTTQKKAA22, L:33, D:66, U:KK, L:P, D:P, U:6789T, L:P, D:P, U:3TTT, L:P, D:P, U:9, L:Q, D:K, U:P, L:A, D:2, U:P, L:P, D:3, U:Q, L:K, D:A, U:P, L:B, D:P, U:P, L:6, D:7, U:A, L:R, D:P, U:P, L:555577JJ, D:P, U:P, L:8889. **Full game of (b):** H:3355556778889JJQKABR; 344446679JJQQKA22; 367899TTTTQKKAA22, L:33, D:66, U:KK, L:P, D:P, U:6789T, L:P, D:P, U:3TTT, L:P, D:P, U:9, L:Q, D:K, U:P, L:A, D:2, U:P, L:P, D:3, U:Q, L:K, D:A, U:P, L:P, D:4, U:A, L:P, D:P, U:A, L:R, D:P, U:P, L:9, D:2, U:P, L:B, D:P, U:P, L:77888, D:P, U:P, L:6, D:7, U:2, L:5555, D:P, U:P, L:JJ.

## F.5. Bad Cases

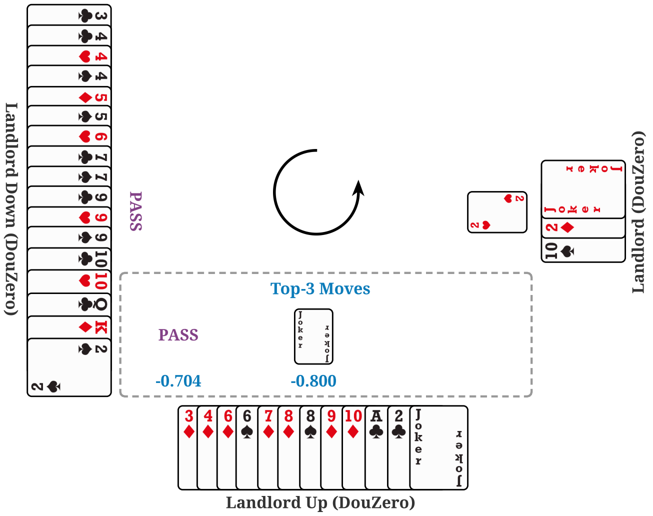


*Figure 32.* Case 1: Bad case (turn 21). The Landlord plays a 2 with only 3 cards left. While the LandlordUp has Black Joker in hand, DouZero chooses not to play it. Although the Peasants will lose whatever the LandlordUp plays in this specific case, playing the Black Joker should have a larger chance to win (if the Red Joker happens to be in the hand of the LandlordDown). Thus, it is worth a try to play Black Joker. **Full game:** H: 3556788TQQQKKKAAA22R; 344455677999TTQK2; 334667889TJJJJA2B; L:55, D:P, U:P, L:88, D:P, U:P, L:3AAA, D:P, U:P, L:7KKK, D:P, U:P, L:6QQQ, D:P, U:JJJJ, L:P, D:P, U:3, L:2, D:P, U:P, L:T, D:K, U:A, L:2, D:P, U:B, L:R.

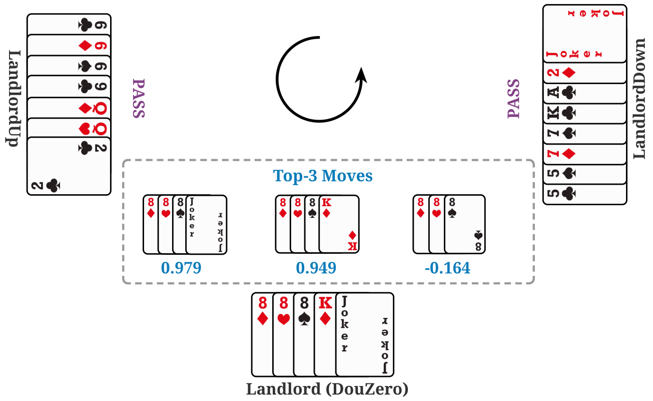


*Figure 33.* Case 2: Bad case (turn 22). This could be a bad case. DouZero aggressively chooses Black Joker as the kicker instead of K. While it is true that DouZero can win whichever kicker it chooses, choosing Black Joker is risky because this knowledge could be generalized to other cases with neural networks and results in losing a game. In fact, choosing K as the kicker will always be at least as good as Black Joker given the current hand. **Full game:** H:333578889TJQKKAAA22B; 444455779JJJQKA2R; 3566667899TTTQQK2, L:3335, D:9JJJ, U:P, L:7AAA, D:4444, U:P, L:P, D:Q, U:K, L:P, D:P, U:3TTT, L:P, D:P, U:56789, L:9TJQK, D:P, U:P, L:22, D:P, U:P, L:888B, D:P, U:P, L:K.

### F.6. Randomly Selected (rather than Cherry-Picked) Full Games

*Table 19.* Case 1: Randomly selected (rather than cherry-picked) full self-play games of DouZero. DouZero plays the Landlord, LandlordUP and LandlordDown positions. All DouZero agents were trained using WP as objectives.

| Logs |
|---|
| H:335556788899TTJKKA2R; 6677789TTJJQQKA2B; 334444569JQQKAA22, L:33, D:66, U:QQ, L:KK, D:P, U:AA, L:P, D:P, U:9, L:T, D:A, U:P, L:2, D:B, U:P, L:P, D:9, U:J, L:A, D:2, U:P, L:R, D:P, U:P, L:5559, D:777J, U:P, L:P, D:J, U:K, L:P, D:P, U:5, L:6, D:8, U:4444, L:P, D:P, U:33, L:88, D:TT, U:22, L:P, D:P, U:6 |
| H:34455777889JQQKA222B; 345668899TTTJQKAA; 33456679TJJQKKA2R, L:44, D:P, U:JJ, L:QQ, D:AA, U:P, L:P, D:3, U:T, L:J, D:P, U:Q, L:K, D:P, U:A, L:P, D:P, U:9, L:A, D:P, U:2, L:B, D:P, U:P, L:3, D:4, U:6, L:9, D:Q, U:P, L:2, D:P, U:R, L:P, D:P, U:34567, L:P, D:P, U:KK, L:22, D:P, U:P, L:77788, D:66TTT, U:P, L:P, D:J, U:P, L:P, D:9, U:P, L:P, D:9, U:P, L:P, D:8, U:P, L:P, D:5, U:P, L:P, D:K, U:P, L:P, D:8 |
| H:34455666789TTJQQKA2R; 335688899TJQA222B; 34457779TJJQKKKAA, L:445566, D:P, U:P, L:6789T, D:89TJQ, U:9TJQK, L:TJQKA, D:P, U:P, L:Q, D:A, U:P, L:2, D:B, U:P, L:R, D:P, U:P, L:3 |
| H:334447779TTTJQKKK22R; 3345566678TJQKA2B; 556888999JJQQAAA2, L:33444, D:55666, U:888JJ, L:TTT22, D:P, U:55AAA, L:P, D:P, U:QQ, L:P, D:P, U:6999, L:QKKK, D:P, U:P, L:R, D:P, U:P, L:777J, D:P, U:P, L:9 |
| H:455577899TTJJKKAA22B; 3346789TJJQQAA22R; 334456667889TQQKK, L:4, D:P, U:K, L:A, D:2, U:P, L:P, D:6789TJ, U:P, L:P, D:4, U:Q, L:A, D:2, U:P, L:B, D:R, U:P, L:P, D:33, U:88, L:TT, D:QQ, U:P, L:KK, D:AA, U:P, L:22, D:P, U:P, L:JJ, D:P, U:P, L:77, D:P, U:P, L:99, D:P, U:P, L:5558 |
| H:3345666889JQQQKAA22B; 4457777899TJQAA2R; 33455689TTTJJKKK2, L:5, D:9, U:J, L:K, D:2, U:P, L:B, D:R, U:P, L:P, D:89TJQ, U:P, L:P, D:44, U:P, L:88, D:AA, U:P, L:22, D:7777, U:P, L:P, D:5 |
| H:334679999TTQQKKKAA2B; 3455677888JQKA22R; 344556678TTJJJQA2, L:33, D:55, U:TT, L:QQ, D:P, U:P, L:TT, D:P, U:JJ, L:AA, D:22, U:P, L:P, D:4, U:Q, L:2, D:P, U:P, L:469999, D:P, U:P, L:7KKK, D:P, U:P, L:B |
| H:3455667899TTTJJQQKAB; 33567788JJQQKA22R; 3444567899TKKAA22, L:6, D:K, U:A, L:B, D:P, U:P, L:34567, D:P, U:56789, L:89TJQ, D:P, U:P, L:5, D:6, U:A, L:P, D:P, U:T, L:A, D:2, U:P, L:P, D:5, U:9, L:J, D:P, U:K, L:P, D:P, U:K, L:P, D:P, U:22, L:P, D:P, U:3444 |
| H:345578TTJJJQKKKA2222; 3334566667789QQKR; 445788999TTJQAAAB, L:3, D:8, U:Q, L:2, D:P, U:B, L:P, D:P, U:J, L:Q, D:K, U:P, L:A, D:P, U:P, L:4, D:9, U:T, L:P, D:P, U:88, L:TT, D:QQ, U:P, L:P, D:4, U:9, L:2, D:6666, U:P, L:P, D:5, U:9, L:K, D:R, U:P, L:P, D:33377 |
| H:3334577889TTJQKAAA2B; 345566789TJQQKKA2; 445667899TJJQK22R, L:5, D:6, U:K, L:P, D:A, U:P, L:2, D:P, U:P, L:T, D:K, U:P, L:B, D:P, U:R, L:P, D:P, U:456789TJQ, L:P, D:P, U:4, L:K, D:2, U:P, L:P, D:3, U:9, L:A, D:P, U:2, L:P, D:P, U:J, L:A, D:P, U:2, L:P, D:P, U:6 |

*Table 20.* Case 2: Randomly selected (rather than cherry-picked) full games of DouZero played with other agents in Botzone. DouZero plays the position as noted in the left column. All DouZero agents were trained using WP as objectives.

| Position | Logs |
|---|---|
| Landlord | H:33455778999TTTQKKAA2; 34567788TJQKKA2BR; 344566689JJJQQA22, L:48999TTT, D:BR, U:P, L:P, D:34567, U:P, L:P, D:TJQKA, U:P, L:P, D:88, U:QQ, L:KK, D:P, U:22, L:P, D:P, U:3666, L:P, D:P, U:5JJJ, L:P, D:P, U:44, L:AA, D:P, U:P, L:2, D:P, U:P, L:77, D:P, U:P, L:33, D:P, U:P, L:55, D:P, U:P, L:Q |
| Peasants | H:34456779TTJKKKAAAA22; 34455668888JQQQ2R; 335677999TTJJQK2B, L:34567, D:P, U:9TJQK, L:P, D:P, U:T, L:J, D:Q, U:P, L:A, D:2, U:P, L:P, D:445566, U:P, L:P, D:38888Q, U:P, L:P, D:Q, U:P, L:P, D:R, U:P, L:P, D:J |
| Landlord | H:334455668899TJQQAA2B; 34567777TJJKKKAA2; 34568899TTJQQK22R, L:33445566, D:P, U:P, L:9, D:T, U:P, L:Q, D:A, U:P, L:P, D:777JJ, U:P, L:P, D:34567, U:P, L:89TJQ, D:P, U:9TJQK, L:P, D:P, U:3, L:8, D:A, U:P, L:2, D:P, U:R, L:P, D:P, U:88, L:AA, D:P, U:22, L:P, D:P, U:4, L:B |
| Peasants | H:34455677889TTJJQKK2R; 33557899JQQQKKA22; 344666789TTJAAA2B, L:345678, D:P, U:6789TJ, L:789TJQ, D:P, U:P, L:4, D:7, U:A, L:2, D:P, U:B, L:R, D:P, U:P, L:5, D:A, U:P, L:P, D:JQQQ, U:P, L:P, D:55, U:P, L:KK, D:22, U:P, L:P, D:K, U:A, L:P, D:P, U:A, L:P, D:P, U:3, L:T, D:P, U:2, L:P, D:P, U:66, L:P, D:P, U:44, L:P, D:P, U:T |
| Peasants | H:3355667789TJJQA222BR; 334446788TQQKKAA2; 455678999TTJJQKKA, L:33, D:88, U:TT, L:P, D:P, U:45678, L:89TJQ, D:P, U:P, L:556677, D:QQKKAA, U:P, L:P, D:4, U:A, L:2, D:P, U:P, L:J, D:P, U:Q, L:A, D:P, U:P, L:2, D:P, U:P, L:2, D:P, U:P, L:BR |
| Landlord | H:334566778899TTQQKA2B; 3455689TJJQKKAA22; 344567789TJJQKA2R, L:456789T, D:89TJQKA, U:P, L:P, D:3, U:4, L:K, D:A, U:2, L:P, D:P, U:34567, L:6789T, D:P, U:P, L:33, D:55, U:P, L:QQ, D:22, U:P, L:P, D:4, U:J, L:B, D:P, U:R, L:P, D:P, U:789TJQKA |
| Landlord | H:33455668TTTJJJQKAA22; 3445677789QQKKA2R; 3456788999TJQKA2B, L:4, D:A, U:P, L:P, D:4, U:9, L:Q, D:K, U:2, L:P, D:P, U:89TJQKA, L:P, D:P, U:B, L:P, D:P, U:3456789 |