DouZero: Mastering DouDizhu with Self-Play Deep Reinforcement Learning

Demo

Daochen Zha†, Jingru Xie‡, Wenye Ma‡, Sheng Zhang*, Xiangru Lian‡, Xia Hu†, Ji Liu‡

†Department of Computer Science and Engineering, Texas A&M University
‡AI Platform, Kwai Inc.
**Georgia Institute of Technology*

Code

RLCard Code

DouDizhu: The Most Popular yet Challenging Poker game in China

- **Popularity.** There are more than **800 million** registered users and **40 million** daily active players on the Tencent mobile platform **alone** for DouDizhu.
- **Competition and Collaboration.** Two Peasants need to cooperate to fight against Landlord.
- **Large State/Action Space.** There are a large number of information sets and 27, 472 possible actions due to combinations of cards, where the legal actions vary significantly from turn to turn.

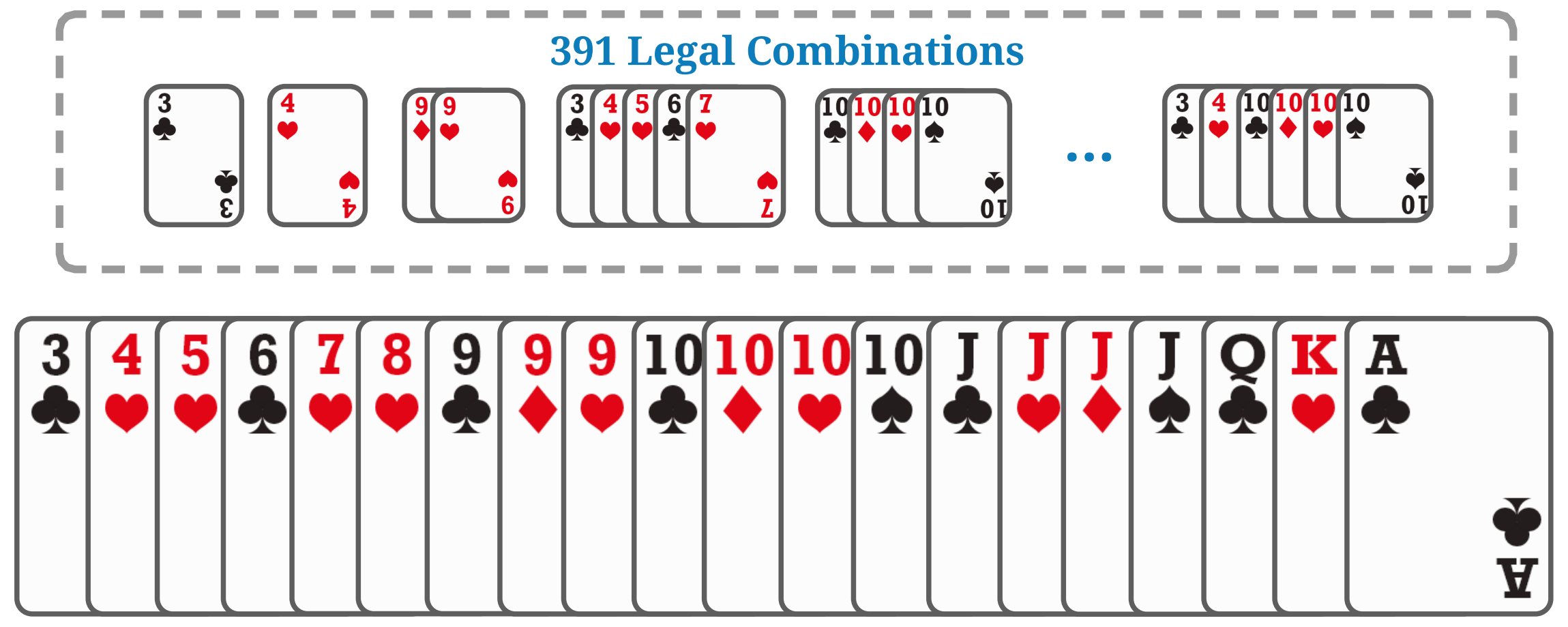


Figure 1:A hand and its corresponding legal moves.

Existing Algorithms are not Satisfactory

- **DQN and A3C.** The commonly used algorithms are mainly designed for simple and small action spaces. They are shown to not perform well in DouDizhu's large and complex action space in previous work.
- **Previous Efforts in DouDizhu.** They reduce the action spaces by using either hierarchical action space or heuristics for abstraction. However, both of them rely on human knowledge and could be sub-optimal. Moreover, they are computationally expensive. DeltaDou, the previous state-of-the-art, takes two months for training.

Traditional Monte-Carlo Methods

To optimize a policy π, Monte-Carlo Methods estimate Q-table $Q(s, a)$ by iteratively executing the following procedure:

- Generate an episode using π.
- For each s, a appeared in the episode, calculate and update $Q(s, a)$ with the return averaged over all the samples concerning s, a.
- For each s in the episode, $\pi(s) \leftarrow_a Q(s, a)$.

Deep Monte-Carlo (DMC)

We enhance traditional Monte-Carlo methods with deep neural networks, action encoding, and parallel actors.

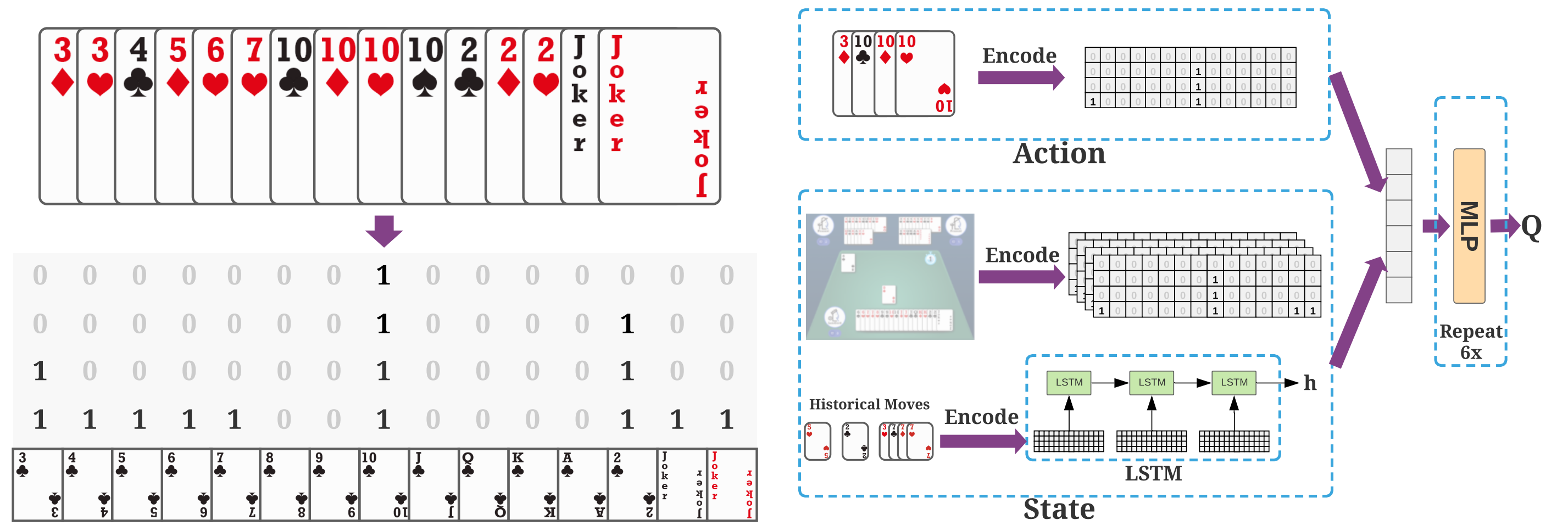


Figure 2:**Left:** Cards for both states and actions are encoded into a 4×15 one-hot matrix, where columns correspond to the 13 ranks and the jokers, and each row corresponds to the number of cards of a specific rank or joker. **Right:** The Q-network of DouZero consists of an LSTM to encode historical moves and six layers of MLP with hidden dimension of 512. The network predicts a value for a given state-action pair based on the concatenated representation of action and state.

While DMC is simple, it is very suitable for DouDizhu due to several nice properties. First, it is not susceptible to overestimation bias when action space is large. Second, actions can be easily encoded as features. Third, DMC can be efficiently parallelized so that it is fast in terms of wall-clock time.

The Strongest DouDizhu AI Up-to-Date

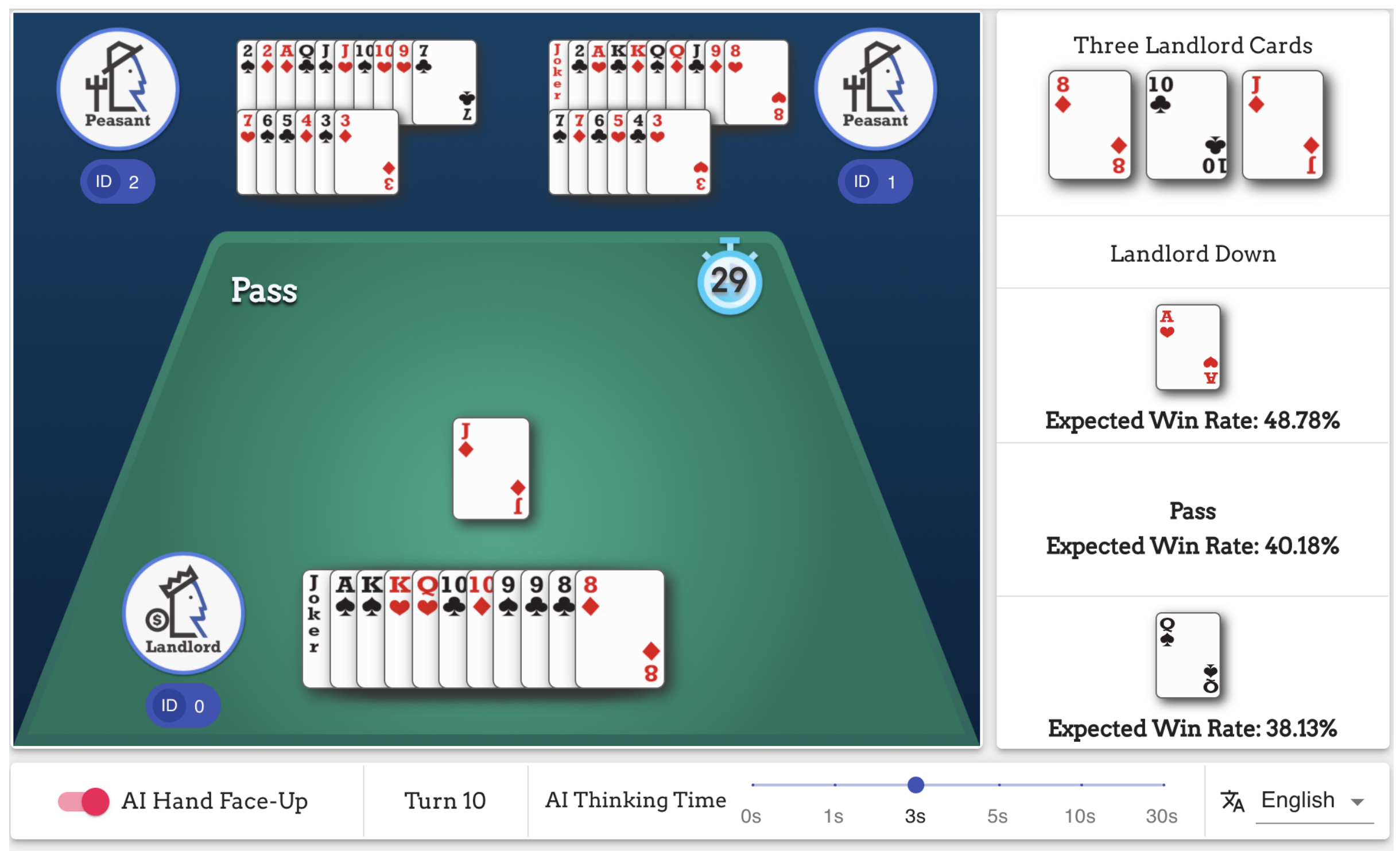
Given an algorithm `A` and an opponent `B`, we use two metrics to compare the performance of `A` and `B`:

- **WP** (Winning Percentage): The number of the games won by `A` divided by the total number of games.
- **ADP** (Average Difference in Points): The average difference of points scored per game between `A` and `B`.

Rank	B / A	DouZero		DeltaDou		SL		RHCP-v2		RHCP		RLCard		CQN		Random	
		WP	ADP	WP	ADP	WP	ADP	WP	ADP	WP	ADP	WP	ADP	WP	ADP	WP	ADP
1	DouZero	-	-	**0.586**	**0.258**	**0.659**	**0.700**	**0.757**	**1.662**	**0.764**	**1.671**	**0.889**	**2.288**	**0.810**	**1.685**	**0.989**	**3.036**
2	DeltaDou	0.414	-0.258	-	-	**0.617**	**0.653**	**0.745**	**1.500**	**0.747**	**1.514**	**0.876**	**2.459**	**0.784**	**1.534**	**0.992**	**3.099**
3	SL	0.341	-0.700	0.396	-0.653	-	-	**0.611**	**0.853**	**0.632**	**0.886**	**0.813**	**1.821**	**0.694**	**1.037**	**0.976**	**2.721**
4	RHCP-v2	0.243	-1.662	0.257	-1.500	0.389	-0.853	-	-	**0.515**	**0.052**	**0.692**	**1.121**	**0.621**	**0.714**	**0.967**	**2.631**
5	RHCP	0.236	-1.671	0.253	-1.514	0.369	-0.886	0.485	-0.052	-	-	**0.682**	**1.259**	**0.603**	**0.248**	**0.941**	**2.720**
6	RLCard	0.111	-2.288	0.124	-2.459	0.187	-1.821	0.309	-1.121	0.318	-1.259	-	-	**0.522**	**0.168**	**0.943**	**2.471**
7	CQN	0.190	-1.685	0.216	-1.534	0.306	-1.037	0.379	-0.714	0.397	-0.248	0.478	-0.168	-	-	**0.889**	**1.912**
8	Random	0.011	-3.036	0.008	-3.099	0.024	-2.721	0.033	-2.631	0.059	-2.720	0.057	-2.471	0.111	-1.912	-	-

Takeaways

- Simple Monte-Carlo (MC) methods can be made to deliver strong performance in a very hard domain.
- A reasonable experimental pipeline for DouDizhu domain with only days of training on 4 GPUs.
- Open-sourced everything at `https://github.com/kwai/DouZero`. It is also integrated in RLCard at `https://rlcard.org/`.
- An online demo at `https://douzero.org/`.


Peasant
ID 2
Peasant
ID 1
Pass
29
Landlord
ID 0
Three Landlord Cards
Landlord Down
Expected Win Rate: 48.78%
Pass
Expected Win Rate: 40.18%
Expected Win Rate: 38.13%
AI Hand Face-Up
Turn 10
AI Thinking Time
0s
1s
3s
5s
10s
30s
English