PPOker: Adaptive Clipping in a Multiagent Environment

Katie Huntley
(kah294)

Ananya Jajodia
(aj477)

GitHub: https://github.com/kh31514/PPOker

1 Introduction

Poker, the popular betting game since the 18th century, presents an interesting challenge in the context of robot learning. Unlike most card games, success in poker is heavily dependent on bluffing, a tactic in which players mask their abilities or intentions, and imperfect information, where some aspects of the game (the opponent's card and the cards that will be on the table) are hidden from the player. A robot, unable to feel or express such emotion, has no way to read other players' body language and must overcome the lack of information to make game-winning decisions.

In the following paragraphs, we propose a way to overcome this imperfect information to train a successful poker agent. We utilize proximate policy optimization (PPO) combined with adaptive observation-based clipping to train an agent to compete against two opponent strategies.

2 Approach

We train and deploy agents within the PettingZoo Texas Hold'em No Limit environment in which two players aim to win chips by forming the best five-card hand or by forcing opponents to fold through betting. At their turn, players can take one of the following actions as determined by the game state: fold, check and call, raise half pot, raise full pot, or go all in. The environment is zero-sum, where the winner and loser of each round receives a reward of +raised chips/2 and -raised chips/2 respectively. Compared to the ordinary Texas Hold'em environment, the No Limit environment places no limit on the amount of each raise or the number of raises. We chose the No Limit environment rather than the ordinary environment because we found the No Limit environment to be more common among existing research. To employ PPO, we utilize the StableBaselines 3 library.

We train agents using invalid action masking and self-play. Invalid action masking ensures that agents adhere to the rules of Texas Hold'em poker. Self-play allows agents to play against copies of themselves, and thus develop robust strategies.

We use four different strategies to train agents as outlined below:

Model Name	Description
Default clipping	We use the standard clipping value of 0.2 from Stable-Baseline3's Maskable PPO. In "Proximal Policy Optimization Algorithms", the clipping rate of 0.2 was observed to yield the highest reward on average across many environments (Schulman et al., 2017) (1)
No clipping	We manually override the standard clipping value of 0.2 with 0. Recall that the PPO objective function is $L^{CLIP}(\theta) = \hat{\mathbb{E}}_t[\min(r_t(\theta)\hat{A}_t, \text{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t)]$. With a clipping value of 0, the objective function is entirely dependent on the advantage, subjecting the model to potentially large policy updates.
Decay clipping	We explore the effects of decay clipping as proposed in "Decaying Clipping Range in Proximal Policy Optimization" (3). We begin training with the standard clipping value of 0.2 and decrease over time at a rate of 0.99 per step taken. For example after the first training step, the clipping value decreases from 0.2 to 0.2 * 0.99.
Chip clipping	We propose a new strategy for clipping based on current rewards. We adjust the clipping value based on the current observation space. More specifically, the clipping value is player's chips / 100 + 0.1 where 100 is the maximum number of chips a player can hold. In this methodology, the policy updates more radically as the player increases their chip count.

Regardless of clipping rate, we use the following default hyperparameters across all trials:

- Batch size = 64
- Number of epochs = 10
- Learning rate = 0.0003

We use a class, SB3ActionMaskWrapper, to adapt the Stable-Baselines3 algorithm to the PettingZoo environment. Stable-Baselines3 is primarily designed for single-agent environments while the PettingZoo Texas Hold'em No Limit environment employs two agents. The wrapper class resolves this discrepancy by standardizing the observation and action spaces for each agent.

We train each agent for a total of 32768 time steps and evaluate at increments of 2048. We evaluate agents against two different opponent strategies which we call random and call-focused. In the random strategy, at each turn, the opponent chooses a valid action at random. In the call-focused strategy, at each turn the opponent checks and calls until no longer possible at which they choose a valid action at random.

Each trained agent plays each type of opponent in a series of 1000 rounds (environment seeds 0 through 1000 for consistency). At the conclusion of each round, the agent with the highest total reward is deemed the winner.

3 Hypothesis

We expect default clipping to have conservative policy updates leading to a more conservative play style. This policy is also expected to do well generally.

We expect no clipping to have less conservative updates which should allow it to explore potential actions better but will also result in a less conservative play style and ultimately a worse performance.

We expect decay clipping to begin with a more conservative policy updates and have greater updates later in training. This may lead to early convergence as well as a more diverse play style.

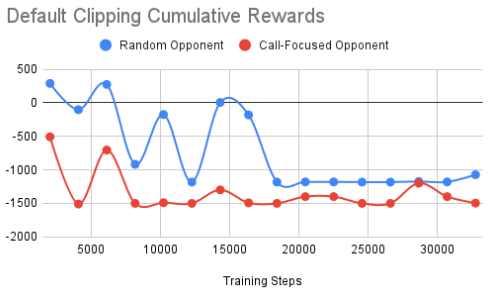
Finally, we expect clip chipping to have smaller updates on steps where the player has a low amount of chips and greater policy updates when the player has more chips. This may result in more conservative play when the player has a low number of chips and more exploratory play when the player has a large number of chips.

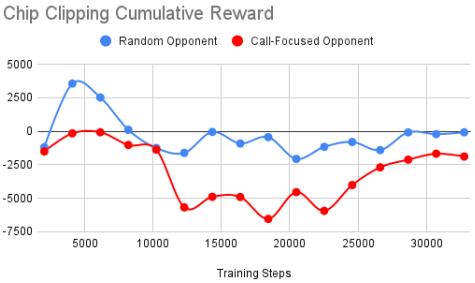
4 Results

Even a great poker player cannot expect to win every hand, so win percentage, although more widely understood, is a poor metric in the context of poker. As such, we choose cumulative reward to

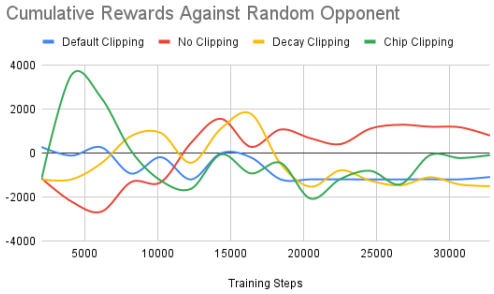
measure the effectiveness of each agent. A high cumulative reward indicates that the agent is able to recognize the strength of its hand at each round. With a promising hand, the agent should make bolder choices (higher bets) and attain a high, positive reward. With a less promising hand, the agent should make more conservative choices (lower bets) and attain a low, negative reward.

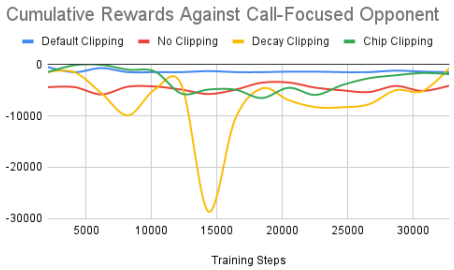
The graphs below outline how each model performs against the two types of opponents across training iterations. Note that the step of the y-axis varies across graphs.





The following graphs display the same results, but grouped by opponent rather than algorithm.



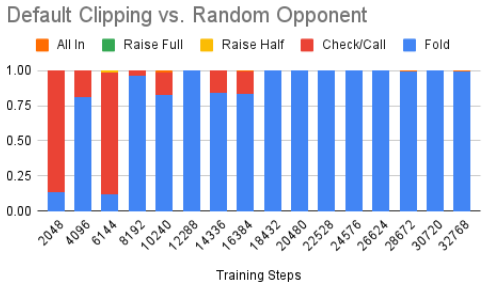


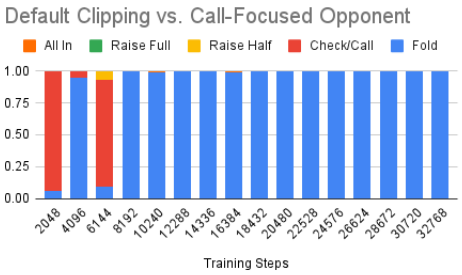
Cumulative Rewards at Conclusion of Training (After 32768 Steps)			
	Random Opponent	Call-Focused Opponent	Absolute Difference
Default Clipping	-1074	-1498	424
No Clipping	810	-4120	4930
Decay Clipping	-1491	-720	771
Chip Clipping	-74	-1883	1809

A successful model should attain a high cumulative reward. The negative cumulative rewards above indicate that the agents are losing more chips than they gain across the 1000 evaluation games.

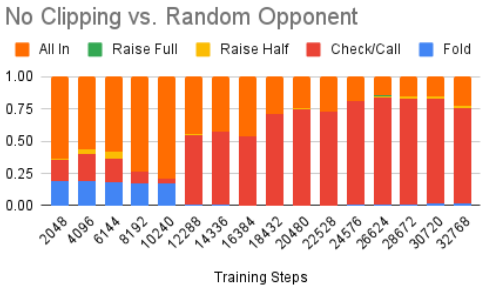
To further comprehend the agents' performances, we tallied the agents' action choices during evaluation. Since specific game lengths may vary, we observe the agents' actions by relative percentages. For example, we can ask "What percent of the time does the agent choose to fold?"

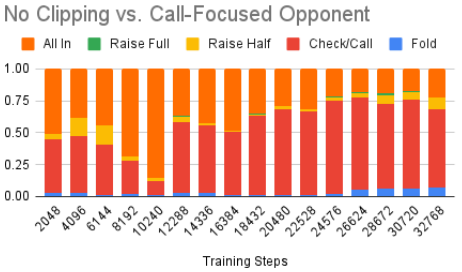
Default Clipping



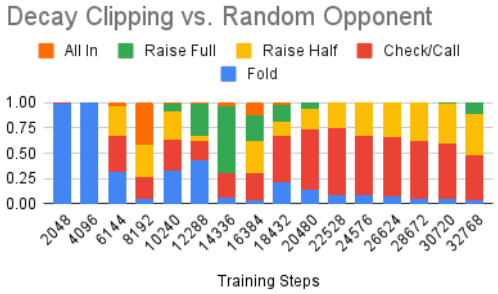


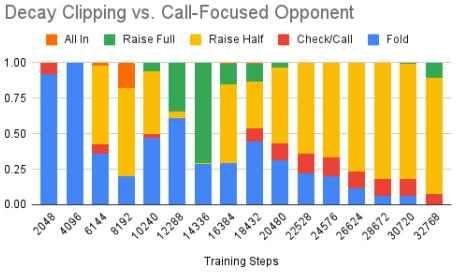
No Clipping



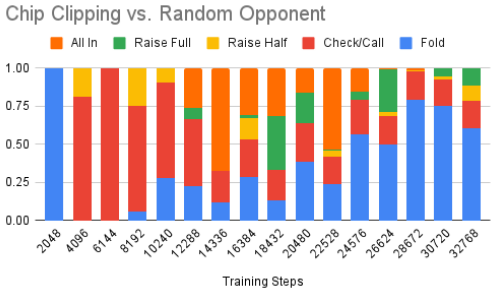


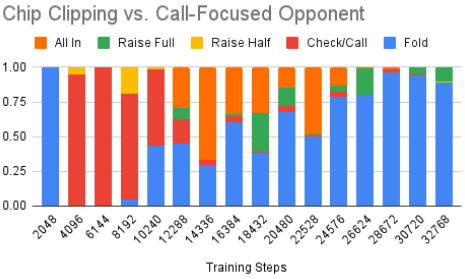
Decay Clipping





Chip Clipping





5 Analysis

Across all clipping techniques, the trained agents yielded a higher cumulative reward against the random opponent compared to the call-focused opponent. We see the highest cumulative reward of

3563 achieved by the chip clipping model at timestep 4096, and we see the lowest cumulative reward of -28618 achieved by the decay clipping model at timestep 14336 against the call-focused opponent.

-28618 is a large outlier, and the second lowest cumulative reward falls just below -10000 (-10644) in the following timestep (16384). Upon closer inspection, we can see that the model raises full pot a significant portion of the time ($\sim$0.71) at time step 14336, but shifts its strategy to raise half pot in the subsequent timestep.

Each model appears to develop its own strategy throughout the training process. At the conclusion of training, the default clipping agent tends to fold, the no clipping agent tends to check and call, the decay clipping agent tends to raise half pot, and the chip clipping agent tends to fold. The default clipping and the no clipping agents develop the least diverse strategies. They both converge to a strategy fairly quickly, showing little change in move distribution after the 12288th training step. This result aligns with expectations since the default clipping and no clipping algorithms use one uniform clipping value throughout the training process while the decay clipping and chip clipping algorithms update the clipping value dynamically.

Decay Clipping and Chip clipping show the most change in strategy during later timesteps, with chip clipping showing the most change. This is most likely a result of the dynamic clipping allowing larger policy updates at later training steps.

The no-clipping technique shows the most stratified data of any clipping technique. By the end of training, the absolute difference in cumulative reward against the random and call-focused opponent is 4930, more than all other model absolute differences combined. We attribute this result to lack of stability in training in the no clipping algorithm.

6 Conclusion

Based on our results, it is difficult to pinpoint how effectively each clipping technique performed, but nonetheless, we attempt to rank models as described below.

We consider the no clipping model to be the least successful model. Although the model achieves a positive cumulative reward against the random agent, the model shows little signs of extending its strategy to the call-focused opponent.

We consider the chip clipping model to be the most successful model. As compared to other models, the chip clipping model does not show any major outliers against either the random or call-focused opponent. By the end of training, the chip clipping model performs roughly evenly with the random opponent (cumulative reward of -74) and shows an upward trend in performing better against the call-focused opponent. The current chip clipping formula is current chips/100 + 0.1, and with further modifications, we believe the model has potential for higher cumulative rewards.

Across the board, no model performed as well as we were hoping for. Negative cumulative rewards signify that the agents are losing more chips than they gain across the 1000 evaluation games.The random opponent is likely to outperform all trained models from this experiment.

Based on our findings, we believe that PPO is generally not a good choice for information-imperfect games like poker. PPO typically assumes that the input to the policy network is the full observable state and cannot effectively infer hidden states. This, in turn, can result in suboptimal decision-making. Furthermore, poker incorporates delayed rewards; A player has few ways to evaluate their actions until the end of the game when the "river" stage when the fifth and final card is dealt face up in the middle of the table. Only then, does the player have the ability to gain chips. In the context of robot learning, an agent may struggle to correlate their current actions to the delayed reward.

Future studies can investigate other approaches such as Counterfactual Regret Minimization (CRM) or Recurrent A3C which may offer better ways to overcome imperfect-information situations.

7 Contributions

7.1 Katie Huntley (kah294)

Katie Huntley was responsible for the following aspects of the project:

- Implementing default clipping, no clipping, decay clipping, and chip clipping models in the PettingZoo Texas Hold'em No Limit environment.
- Implementing two opponent strategies (random and call-focused) to compete against the trained models.
- Experimenting with and tuning hyperparameters including total training steps, learning rate, batch size, epoch number.
- Gathering and plotting cumulative reward statistics across all model and opponent combinations.
- Gathering and plotting action distribution statistics across all model and opponent combinations.

7.1.1 Ananya Jajodia (aj477)

Ananya Jajodia was responsible for the following aspects of the project:

- Researching proximal policy optimization (PPO) and current clipping techniques and common values
- Researching reinforcement learning based poker bots and PPO application to the scenario
- Experimenting with and tuning hyperparameters including total training steps, learning rate, batch size, epoch number
- Helping with analysis of data and conclusions
- Formatting and editing final report

References

[1] Schulman, John, et al. *Proximal Policy Optimization Algorithms*. 28 Aug. 2017.

[2] Kashi, Alex, et al. *Building an Efficient Poker Agent Using RL*. May 2022.

[3] Farsang, Monika, and Luca Szegletes. *Decaying Clipping Range in Proximal Policy Optimization*. 1 July 2021.

[4] "SB3: Action Masked PPO for Connect Four." PettingZoo Documentation, Farama Foundation, 2023, `pettingzoo.farama.org/tutorials/sb3/connect_four/`.