NTU CSIE FAI 2024 Final Report

The Methods I have tried

Due to the challenges in obtaining or generating training data, training a supervised model for my poker agent was not an ideal option. Instead, I employed rule-based conditions to determine the agent's actions. The methods I explored are as follows:

1. **All In Every Time (AIET)**
   My initial intuition was that many players would fold if they lacked confidence in their hole cards or faced aggressive raises. The AIET strategy involves the agent going all-in every time it is their turn. Experimental results showed that, despite its simplicity, this strategy performed tolerably well.

2. **Winning Rate Driven (WRD)**

   The WRD series of strategies base the agent's actions on its estimated winning rate.

   Version 1 (V1): In this simple strategy, if the winning rate is above a higher threshold (0.5), the agent raises an amount proportional to its winning rate. If the winning rate is above a lower threshold (0.3), it calls. Otherwise, it folds. Experimental results indicated that this strategy was slightly less effective than AIET.

   Version 2 (V2): An improved version of V1, V2 introduced two new features: if the agent had a sufficient stack to win the game, it would fold to conserve its lead. Additionally, in the late game (last five rounds), the agent would significantly increase its raise amount to turn the tables. This version showed considerable improvement over V1.

   Version 3 (V3): The ultimate version of WRD, V3, introduced several new features and removed conflicting ones. The most crucial feature was that if an action (call/fold) would lead to losing the game, the agent would go all-in as a last resort. This feature leveraged the fear of many opponents who fold against an all-in bet. Additionally, if the winning rate exceeded 97%, the agent would seize the opportunity to go all-in. V3's structure allowed multiple criteria to decide whether to all-in, raise, call, or fold, resulting in significant performance improvements over V2.

Wednesday (6/12) morning, I found that all my code on the WorkStation2 were deleted by VS Code with no apparent reason (I didn't do anything), so I can't provide the exact parameters and criteria for the above agents, the only thing I have is the result of the experiments.

3. **Winning Rate Driven with Risk Aversion (WRD-RA)**
   WRD-RA is a variant of WRD inspired by the observation that my agents often lost even with a winning rate higher than 97% (which will go all-in). It would only go all-in if folding would result in losing the game or if the winning rate was high, but calling and losing that round would also mean losing the game. Another key feature was that WRD-RA would avoid raising excessively or allowing the pot size to become too large, thereby inducing opponents to call and lose more stacks. These strategies significantly improved performance, achieving over a 90% winning rate against baselines 0-3, and more than 50% against baselines 4-7.
   The drop in winning rate from baselines 0-3 to baselines 4-7 was due to the latter's design, which included an all-in strategy at the "preflop" stage when facing imminent loss. This created a critical issue: at that point, the agent had only two options, fold or leave it to chance. If WRD-RA left it to chance, the final outcome had little correlation with previous successes, making much of the effort in

vain.
To address this, WRD-RA introduced a new feature: **leverage**. The philosophy behind leverage is that when WRD-RA is going to win, baselines 4-7 would frequently go all-in regardless of their hole cards. When facing an all-in, WRD-RA would fold without sufficient confidence, despite often having better hole cards. The leverage feature lowers the thresholds for calling and all-in when WRD-RA is on the verge of winning, aiming to catch the opponent's bluffs and secure a victory.
The experimental results showed that with full leverage, the winning rate against baseline 7 increased by over 10%, while the winning rates against baselines 4-6 slightly decreased. This decrease may be due to the instability introduced by leaving more decisions to chance. To mitigate this, I developed a third version: **WRD-RA with half leverage**. In this version, leverage is activated only when the agent is on the verge of winning and it is the late game, as opponents are more likely to make irrational decisions under pressure. This adjustment aims to catch opponents' bluffs while preserving the agent's stability.

Experiment Results

The table below shows the result of each method combatting against baselines.

The gaming configuration of the experiments is the same as the default (max_round=20, initial_stack=1000, small_blind_amount=5), and for AIET, WRD V1, V2, and V3, I ran every game 25 times to make the probability distribution closer to real probability distribution. For WRD-RA series, I ran every baseline 100 times.

Baseline / Agent	AIET	WRD V1	WRD V2	WRD V3	WRD-RA w/o leverage	WRD-RA w/ leverage
Baseline0	0.6	0.72	0.96	0.72	0.91	0.9
Baseline1	0.56	0.52	0.88	0.8	0.93	0.92
Baseline2	0.64	0.56	0.68	0.84	0.9	0.96
Baseline3	0.44	0.36	0.6	0.84	0.93	0.94
Baseline4	0.56	0.4	0.4	0.48	0.56	0.5
Baseline5	0.36	0.16	0.2	0.44	0.54	0.49
Baseline6	0.4	0.44	0.56	0.4	0.66	0.63
Baseline7	0.44	0.44	0.4	0.6	0.45	0.56

Submitted Agent