PokerGPT: An End-to-End Lightweight Solver for Multi-Player Texas Hold'em via Large Language Model

Chenghao Huang‡ , Yanbo Cao‡ , Yinlong Wen† , Tao Zhou‡ , and Yanru Zhang‡§

‡University of Electronic Science and Technology of China
§Shenzhen Institute of Advanced Study, UESTC, China
†Sichuan Optical Character Technology Co., Ltd

Background

Representative prior works, such as DeepStack and Libratus heavily rely on counterfactual regret minimization (CFR) to tackle heads-up no-limit Poker.

However, this method has the following drawbacks:

- Significant computational and storage resources are required
- It is difficult to expand this to multiplayer
- Game tree compression unavoidably results in the loss of crucial information for decision making
- Depend on human expert knowledge

Purpose

To construct an AI solver for Poker solutions, named PokerGPT by utilize a lightweight LLM and data from real Poker games.

LLM

By adopting an end-to-end learning method, this method is easy to train and significantly more cost-effective than conventional methods.

Method

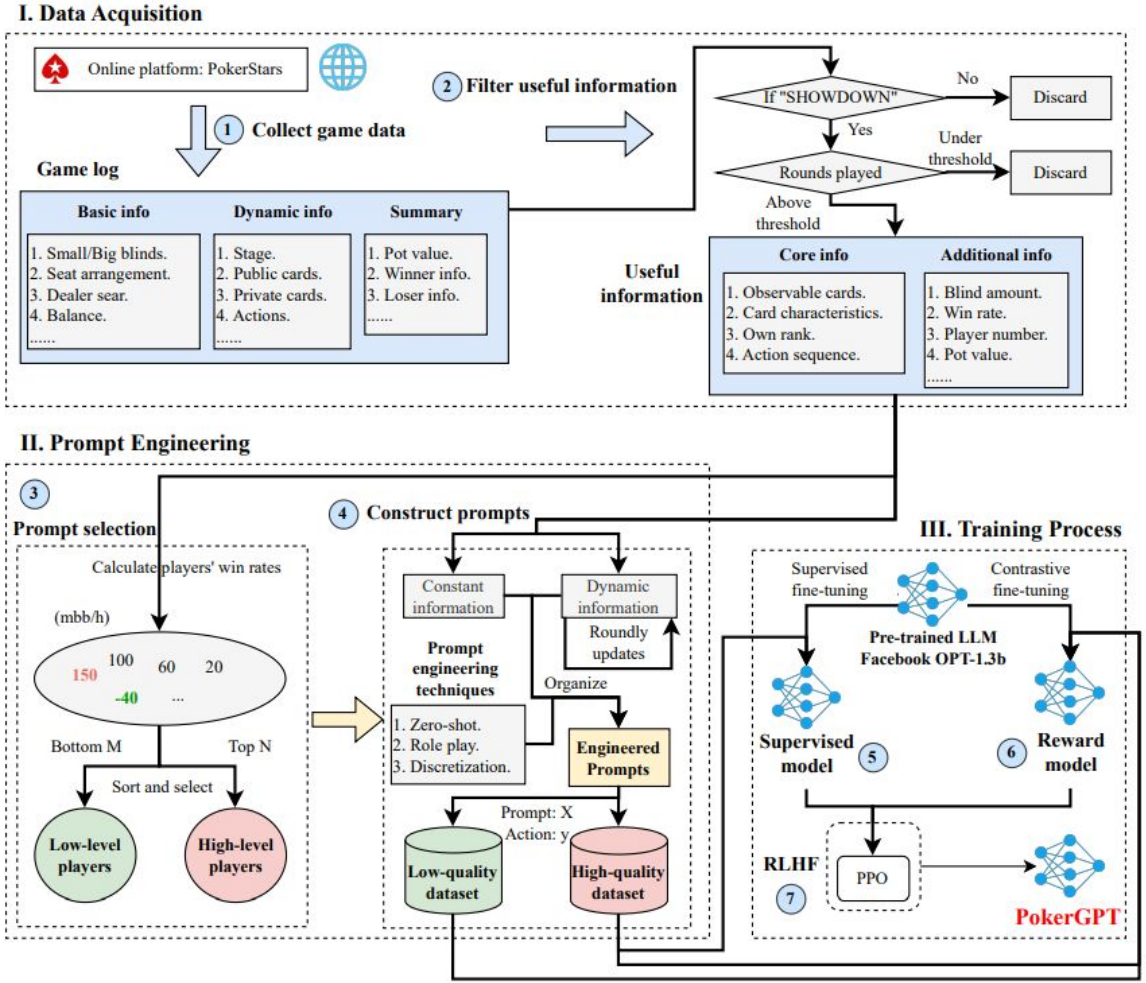
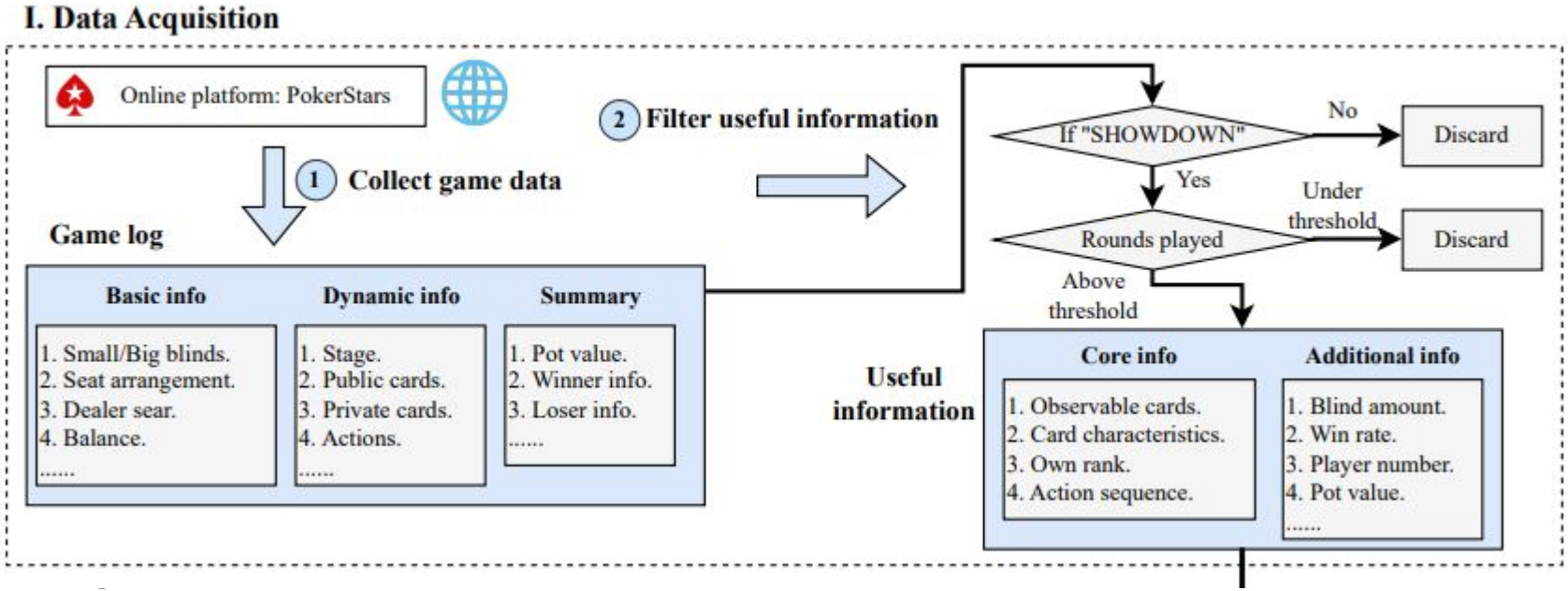


Fig. 4: The framework of PokerGPT, consisting of data acquisition, prompt engineering, and training.

Method

Data Acquisition



Method

Data Description

Basic information: This includes the big blind and small blind settings, details of each player's seat assignment, their respective chip counts, and the current dealer position.

Dynamic information:

1. The “PREFLOP” stage, each player receives two private cards, and the first betting round begins.
2. The “FLOP”, “TURN”, and “RIVER” stages proceed, community cards are revealed in sequence, and the remaining players place their bets.
3. The “SHOWDOWN” stage, private cards are revealed, and the final decisive result is determined.

Summary: An overview of the game pool size, the community cards, and each player’s earnings is recorded.

Method

Data Description

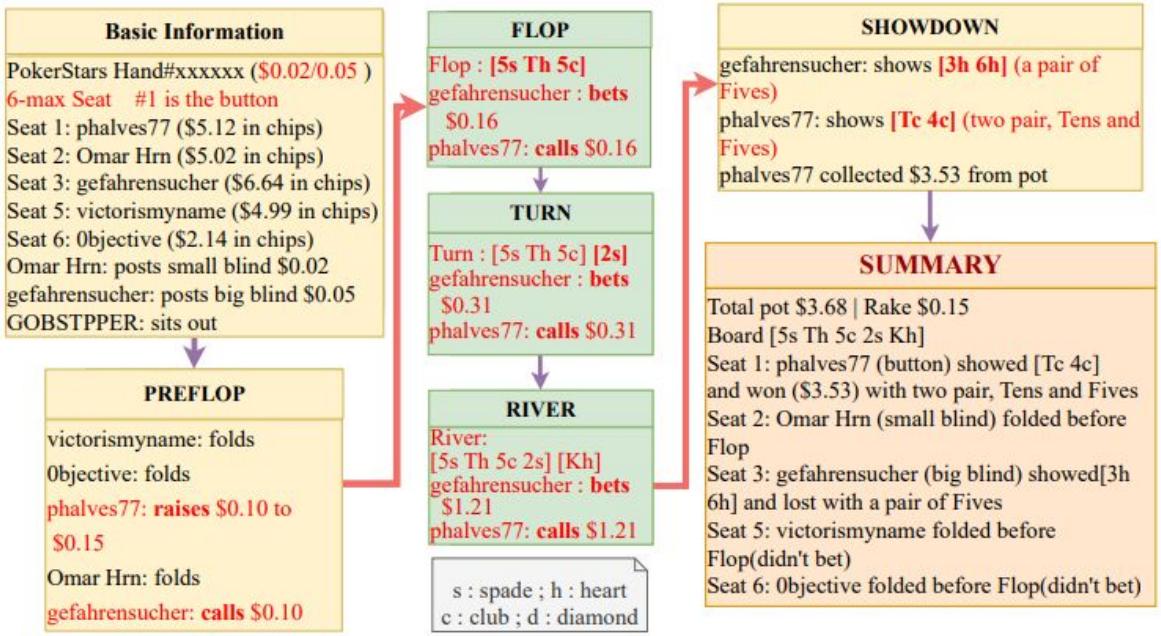
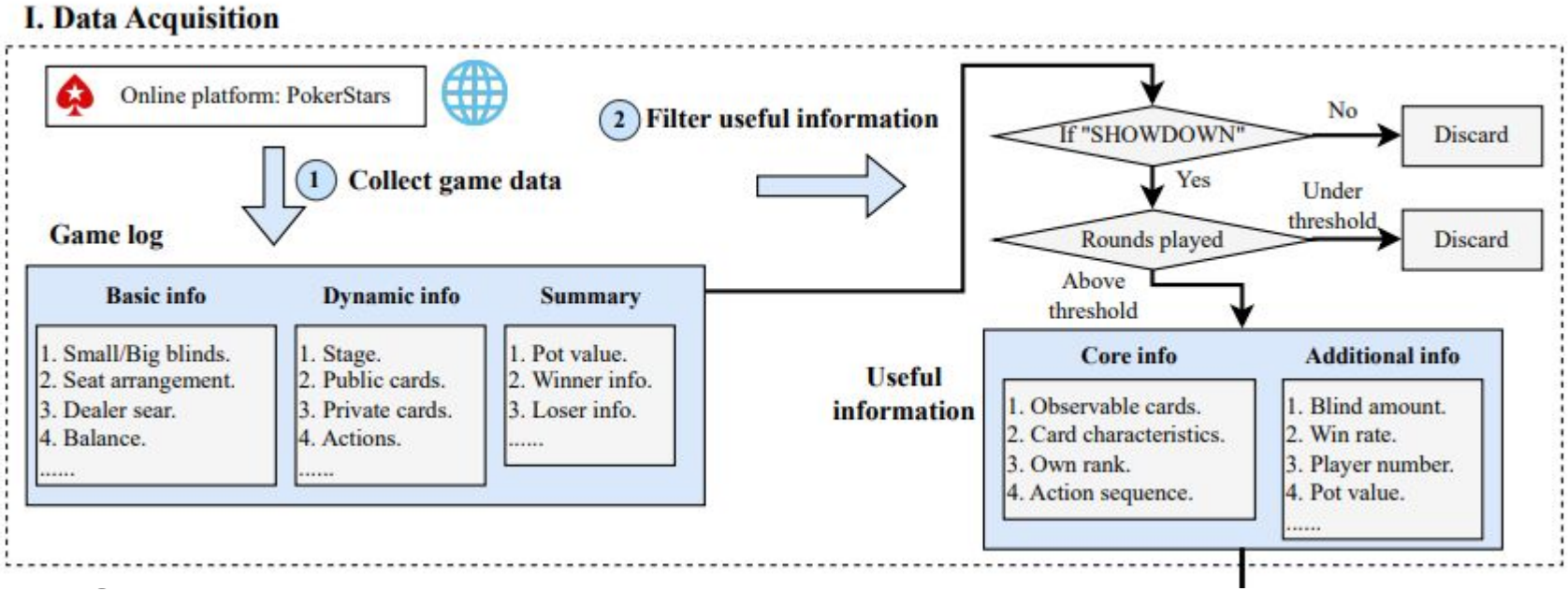


Fig. 5: An example of Poker game log.

Method

Data Acquisition



Method

Information Filtering

Core information of utmost significance is crucial to be included in the prompt, listed as follows:

- ***Observable cards***: It is necessary to know the 2 private cards and public cards exhibited.
- ***The characteristics of cards***:

  -“Suit” means 2 private cards with the same color.

  -“High” means possessing one card higher than ‘9’.

  -“Close” means the difference of 2 cards is less than 5, which can form a “straight”

- ***The rank of the hand***: Since it may be difficult for the LLM to learn to recognize the rank of the hand, we simply provide the rank in the prompt.
- ***Action sequence***: Analyzing players’ behavioral patterns plays a pivotal role in understanding the psychology and strategy of opponents.

Method

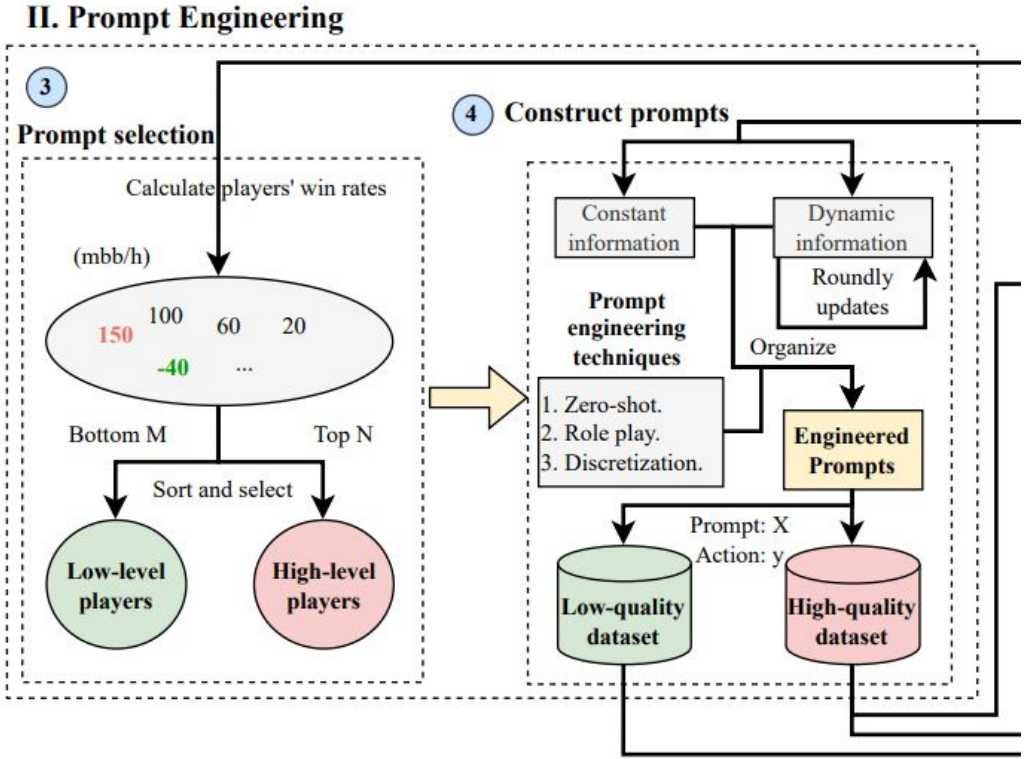
Information Filtering

Additional information also assists the learning and understanding of our model, including:

- ***Amounts of blinds*:** Small and big blinds is important for evaluating both the revenue and the risk of Poker games.
- ***Positions of players***: The positional context of a player relative to the dealer, coupled with the player's actions, serves as an indicator of the hand's strength.
- ***Number of players***: The variation in the number of players within a game intensely changes the game structure.
- ***Balance***: They record each player's balance through the initial amount and the amount each has invested in the pot.
- ***Discard***: This helps their model to determine the remaining players.
- ***Pot value***: This helps LLM understand the concept of odds.
- ***Stage***: The same action in different stages generally has different purposes.

Method

Prompt Engineering



Method

Prompt Selection

Make raw data into prompts

- Data Filtering: Excludes the “SHOWDOWN” stage (when private cards are revealed) and cases where a clear winner cannot be determined.
- Calculate win rate in milli-big-blinds per hand (mbb/h): Select top performers for high-quality training data and Collect data from low performing players as negative examples

Method

Prompt Engineering

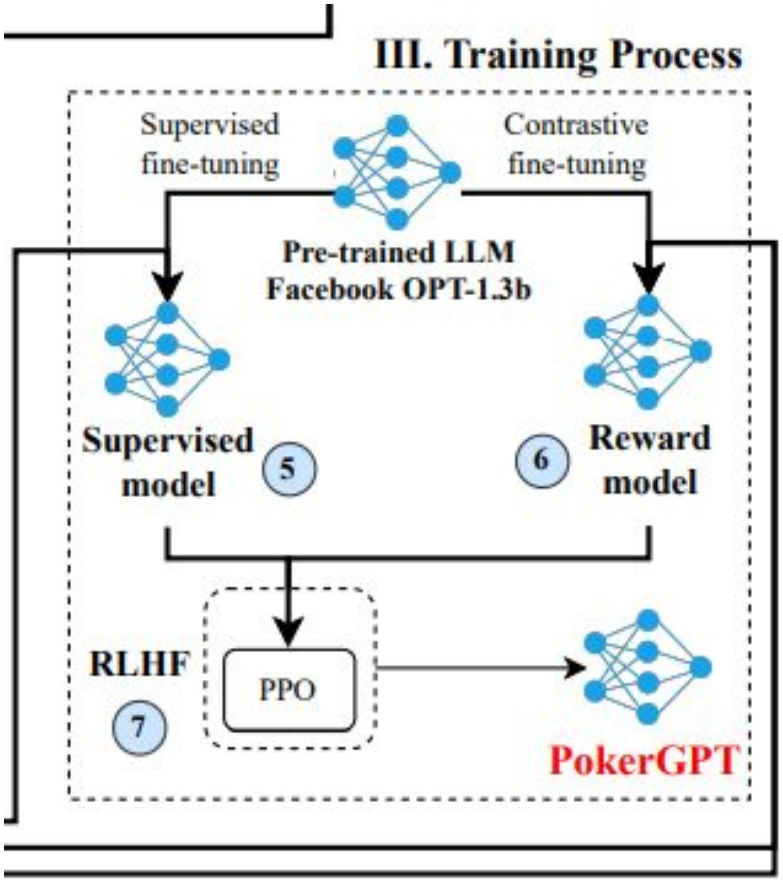
Prompt Instance

You are an experienced gambler. Now you need to assist me to make decisions in Texas Hold'em games. You have been provided with a series of observable information:
Player amount: [6], Currency: USD, Blind value: [0.02/0.05], Order: ['2', '3', '5', '6', '7', '9'], Seat 2 is small blind.
My cards: ['Th', 'Ah'], the characteristics of my cards: ["suit", "high", "close"], My seat: [Seat 2]
Stage: "PREFLOP", Public cards: ['**' '**' '**' '**' '**']
My rank: ["High"], Money: [3.92], Action: []
Seat 3: ['**', '**'], Money: [2.33], Action: [], Discard: [False]
Seat 5: ['**', '**'], Money: [5.54], Action: [], Discard: [False]
Seat 6: ['**', '**'], Money: [3.75], Action: [], Discard: [False]
Seat 7: ['**', '**'], Money: [4.22], Action: [], Discard: [False]
Seat 9: ['**', '**'], Money: [1.47], Action: ["raises 0.05 to 0.1"], Discard: [False]
The pot value is [0.17]
The actions can be: ["fold", "raise", "call"]. What should I do? If I choose to "bet" or "raise", then how much? Choose a number from {(0, 0.05, 0.15, 0.3, 0.5, 1, 2.5, *3.92*}.

Fig. 6: An instance of our prompt.

Method

Training Process



Method

Training Process

1. Supervised finetune a pre-trained LLM on small datasets to provide specific prior knowledge to this model
2. Train a reward model using data separated by quality.
3. Apply RLHF to make the fine-tuned LLM in step (1) able to understand various human text assisted by the reward model in step (2).

Experiment

Fundamental Step

- Dataset: Processed 1M + games → obtained 120K high-quality samples
  Split: 90% training, 10% testing
- Framework: Used DeepSpeed-Chat for efficient training and GPU distribution
- Model: Fine-tuned Facebook's OPT-1.3B(GPT-3 level, Hugging Face)
- Environment: Trained on a single NVIDIA GeForce RTX 3090
- Hyperparameters: Followed DeepSpeed-Chat defaults, mini-batch size = 4(due to hardware limits)

Experiment

Comparison with previous works

They compare their performance against Slumbot with 2 previous works.

TABLE II: Performance comparison against the latest 2 models.

Model	Win Rate (mbb/h)	Training Time (h)	Response Speed (s)	Player Amount	Parameter Amount	Main Tech	Year
ReBel [13]	45±5	-	-	2	-	DRL+CFR	2020
AlphaHoldem [14]	111±16	580	0.017	2	8.6 million	DRL	2022
PokerGPT	**158±49**	**9.5**	5.4	**2 or more**	1.3 billion	LLM	2023

Experiment

Ablation Results

They verify whether filtering useful information and optimizing prompts can improve the performance of PokerGPT.

TABLE I: Datasets under different setup for fine-tuning.

Dataset	Technique	Win Rate Threshold	Amount
I	Raw	-	1 million
II	Info filtering	-	1 million
III	Win rate sorting	> 1500 mbb/h	50000
IV	Win rate sorting	$600 \sim 1200$ mbb/h	50000
V	Win rate sorting	$0 \sim 500$ mbb/h	50000
VI	Win rate sorting	< 0 mbb/h	50000

Experiment

Ablation Results

To evaluate supervised fine-tuning, action prediction ("bet", "fold", "raise", "check") is treated as a multi-label classification task. Model performance is measured using macro F1 score, which equally weights all action types regardless of their frequency.

$$\text{macro-F1} = \frac{1}{N}\sum_{i=1}^{N} \text{F1}_i, \tag{1}$$

$$\text{F1}_i = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}}, \tag{2}$$

Experiment

Ablation Results

To evaluate value-related prediction accuracy, all amounts are converted to USD and expressed as big blind multiples. Bet/raise values are discretized into set multiples {0, 1, 3, 6, 10, 20, 50, 100, all-in}, excluding amounts exceeding the player’s balance. Prediction accuracy is assessed by frequency analysis and mean squared error (MSE).

Experiment

Ablation Results

Perplexity in Natural Language Processing (NLP) is a measure of how well a probability model predicts a sample.

Assume we have a sequence W with N words, a perplexity is formulated as:

$$\text{perplexity}(W) = P(w_1, w_2, ..., w_N)^{-\frac{1}{N}}, \qquad (3)$$

where W = (w1, w2, ..., wN) is the sequence of words, and P is the probability.

In the context of language models, a lower perplexity generally indicates the model is better at predicting the next word in a sentence.

Experiment

Multiplayer evaluation

To evaluate PokerGPT in multi-player settings, it was tested over 1,000 games per player count.

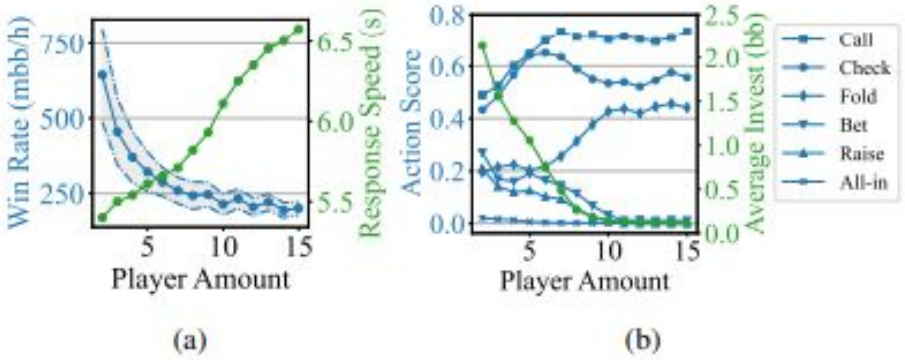


Fig. 11: (a) The trends of win rate and response speed when player amount changes. The dashed lines are the standard deviations. (b) Strategy variation during player amount changes.

Experiment

Analysis of interaction

PokerGPT

System (constant information): You are an experienced gambler. Now you need to assist me to make decisions in Texas Hold'em games. You have been provided with a series of observable information:
Player amount: [6], Currency: USD, Blind value: [0.02/0.05], Order: ['2', '3', '5', '6', '7', '9'], Seat 2 is small blind.
My cards: ['Th', 'Ah'], the characteristics of my cards: ["suit", "high", "close"], My seat: [Seat 2]

Dynamic information:
Stage: "PREFLOP", Public cards: ['**' '**' '**' '**' '**']
My rank: ["High"], Money: [3.92], Action: []
Seat 3: ['**', '**'], Money: [2.33], Action: [], Discard: [False]
Seat 5: ['**', '**'], Money: [5.54], Action: ["fold"], Discard: [True]
Seat 6: ['**', '**'], Money: [3.75], Action: ["fold"], Discard: [True]
Seat 7: ['**', '**'], Money: [4.22], Action: ["fold"], Discard: [True]
Seat 9: ['**', '**'], Money: [1.47], Action: ["raises 0.05 to 0.1"], Discard: [False]
The pot value is [0.17]
The actions can be: ["fold", "raise", "call"]. What should I do? If I choose to "bet" or "raise", then how much? Choose a number from {(0, 0.05, 0.15, 0.3, 0.5, 1, 2.5, *3.92*}.

User:
Q1: What should I do?
PokerGPT: You should call.

Q2: Please be aggressive.
PokerGPT: You should raise to 0.5.

Information updates:
Stage: "FLOP", Public cards: ['7h' '4h' '2h' '**' '**']
My rank: ["Flush"], Money: [3.82], Action: ["call"]
Seat 3: ['**', '**'], Money: [2.23], Action: ["call"], Discard: [False]
.....
Seat 9: ['**', '**'], Money: [1.37], Action: ["raises 0.05 to 0.1"], Discard: [False]
The pot value is [0.3]
The actions can be: ["fold", "check", "bet"]. What should I do? If I choose to "bet" or "raise", then how much? Choose a number from {(0, 0.05, 0.15, 0.3, 0.5, 1, 2.5, *3.82*}.

User:
Q3: What should I do?
PokerGPT: You should check.

Q4: Should I all-in?
PokerGPT: As you have a flush, all-in is fine.

Information updates:
Stage: "FLOP", Public cards: ['7h' '4h' '2h' '**' '**']
My rank: ["Flush"], Money: [3.82], Action: ["call", "check"]
Seat 3: ['**', '**'], Money: [2.23], Action: ["call", "check"], Discard: [False]
Seat 9: ['**', '**'], Money: [1.15], Action: ["raises 0.05 to 0.1", "bets 0.22"], Discard: [False] The pot value is [0.52]
.....

The actions can be: ["fold", "call", "raise"]. What should I do? If I choose to "bet" or "raise", then how much? Choose a number from {(0, 0.05, 0.15, 0.3, 0.5, 1, 2.5, *3.82*}.

User:
Q5: What should I do?
PokerGPT: You should raise to 0.5.

Q6: Please guess Seat 9's rank.
PokerGPT: Seat 9 likes to raise and bet. Therefore, he might be three of a kind or flush.

Q7: Please guess Seat 9's cards.
PokerGPT: Seat 9 likes to raise and bet. Therefore, he might be there of a kind or flush. But there is no evidence what cards Seat 9 has.

Information updates:
Stage: "TURN", Public cards: ['7h' '4h' '2h' 'Ks' '**']
My rank: ["Flush"], Money: [3.6], Action: ["call", "check", "call"] Seat 3: ['**', '**'], Money: [2.23], Action: ["call", "check", "fold"], Discard: [True]
.....
The actions can be: ["fold", "check", "bet"]. What should I do? If I choose to "bet" or "raise", then how much? Choose a number from {(0, 0.05, 0.15, 0.3, 0.5, 1, 2.5, *3.6*}.

User:
Q8: What should I do?
PokerGPT: You should bet 1.

Q9: Please tell me my winning probability.
PokerGPT: Sorry, I can't calculate the probability. However, as you have flush, you are very likely to win.

Information updates:
... My Action: ["call", "check", "call", "bet 0.9"]
.....
Seat 9: ['**', '**'], Money: [1.37], Action: ["raises 0.05 to 0.1", "bets 0.22", "fold"], Discard: [True]
The pot value is [1.42]

Player 2 wins $1.42 with "Flush".

Fig. 12: The question-answering instance of PokerGPT.

Conclusion and future work

- The study explores using large language models (LLMs) for playing Poker.
- PokerGPT is developed through prompt engineering and RLHF fine-tuning on a lightweight LLM.
- It supports any number of players and allows user interaction.
- The model achieves competitive win rates with lower training cost and faster inference compared to previous methods.
- PokerGPT addresses prior challenges in Poker AI and shows potential in imperfect-information games (IIGs).
- Future work includes improving decision explainability using:
    - Human-written action explanations
    - Action probabilities from CFR models
    - Multiple decision candidates

Thank you for your attention!