# LLM を用いたゲーム環境におけるプレイヤーモデルの構築

## 1 はじめに

近年, 人工知能研究が急速な発展を遂げており, 多岐にわたる分野での応用が進んでいる. その中で, ゲーム環境におけるプレイヤーモデルの構築が注目を集めている. 深層強化学習を用いたプレイヤーモデルの構築では, 特に将棋や囲碁といった完全情報ゲーム環境において成果が顕著であり, 近年は不完全情報ゲームへの応用も盛んである. プレイヤーモデルの研究に関して, 現在 Large Language Model (LLM) を用いるアプローチが新たな注目を集めている.

本研究では LLM を用いてプレイヤーモデルを構築することを目的として, 先行研究の Suspicion Agent [1] を新たなゲーム環境に適用し, プレイヤーモデルの構築について検討した.

## 2 今回の研究内容

AI によるゲームプレイヤーの構築において, 定量的な情報と定性的な情報を考慮する必要がある. 定量的な情報とは例えばアクションゲームにおけるオブジェクト情報やポーカーにおける手札など曖昧性を持たない要素である. 一方で, 定性的な情報とは我々人間がゲームをする際に観測する相手の表情や言動など数値化が難しい要素を指す. これまでの深層強化学習によるゲーム応用においては定量的な情報に主眼を置いていた. 一方で, LLM により心理状況を考慮した現実に近いゲームプレイヤー構築を実現したという点で Suspicion Agent の意義は高い. しかしながら, Suspicion Agent で用いられていた Leduc Hold'em [2] は Texas Hold'em を元に人工知能用に大幅な簡略化がなされており, プレイヤーの心理状況の考慮するといった観点からは不十分である. そこで, 今回はよりユーザの心理が重要となるゲームを用いて LLM を用いたプレイヤー構築に関して検討をした.

## 3 要素技術

### 3.1 GPT

GPT[3] は発表当初は Transformer をベースとしてラベルなしデータで事前学習してラベル付きデータでファインチューニングした言語モデルであった. GPT-2 以降は発表当初とは異なり, 膨大なデータを学習させることで様々なタスクに取り組むことをコンセプトとしておりモデルのパラメータ数は極めて大きくなっている. 近年では, 人間の評価を基に強化学習と教師あり学習を併用した Reinforcement Learging from Human Feedback (RLHF) を適用した GPT-3.5 や, マルチモーダルな入力に対応する GPT-4 [4] が公開されている.

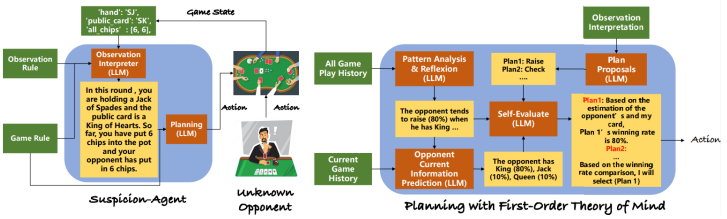


図 1: Suspicion Agent [1] のアーキテクチャ

### 3.2 Suspicion Agent

Suspucion Agent [1] は, 不完全情報ゲームである Leduc Hold'em ゲーム環境に対応する LLM を用いたプレイヤーモデルである. 図 1 に Suspicion Agent のアーキテクチャを示す. Suspicion Agent のアークテクチャにおける主要な要素を以下に述べる.

**Observation Interpreter**
ゲームのルールやゲームの状態空間を自然言語で適切に表現することで, LLM で構成されたエージェントが適切にゲームの状態を解釈できるようにする.

**Planning Module**
Observation Interpreter などの情報を基に複数の Plan を考え, その中から最適な action を選択する Planning Module により, エージェントが適切な行動を選択できるようにする.

**Theory of Mind**
対戦相手がこちらの戦略を見て戦略を変えてくるような場合に備えて, LLM の Theory of Mind (ToM) 能力 [5] を用いて対戦相手からの洞察を自動的に生成し戦略を調整することを可能にする.

Suspicion Agent は Leduc Hold'em 環境において, DQN, CFR といった複数のアルゴリズムにより学習済みのエージェント, 人間といった相手に対して高い勝率を記録した.

# 4 提案する E カード環境

本研究では従来用いられていた Leduc Hold'em よりも複雑な不完全情報ゲーム環境における Suspicion Agent の動作を検証するため,「賭博黙示録カイジ」[6] に登場する二人不完全情報ゲームである E カードを参考にゲーム環境を構築した.

以下, 実装したゲーム環境のルールを説明する.

- 各プレイヤーは奴隷側と皇帝側に分かれる.
- 皇帝側の手札は市民カード 4 枚と皇帝カード 1 枚, 奴隷側の手札は市民カード 4 枚と奴隷カード 1 枚でゲーム開始となる.
- 各プレイヤーはゲーム開始時にチップ 100 枚を持っている.
- ゲームは最大 6 ラウンドで構成され, 3 ラウンドごとに皇帝側と奴隷側が交代する.
- 各ラウンドは Bet フェーズ（賭けるチップ数の決定）, Talk フェーズ（心理戦）, Play フェーズ（勝敗を決めるカード対決）で構成される.
- ラウンド終了後, 利得行列に従ってチップが払い戻され, 手札はゲーム開始時と同様にプレイヤーが奴隷側か皇帝側かに応じてリセットされる.
- 6 ラウンド終了後, またはチップが 0 枚になったプレイヤーがいればゲームは終了し, 最もチップを多く持っているプレイヤーが勝者となる.

また, オリジナルのルールから以下の変更を加えた.

- ゲームの最大ラウンド数を 12 から 6 に変更
- 報酬の内容をチップのやり取りに変更
- Bet フェーズと Talk フェーズの概念の追加

表 1 に E カードの利得行列を示す. 表中の betA, betB はそれぞれ Bet フェーズで PlayerA, PlayerB が賭けたチップの数を表している. また, 表中の * に対応するお互いのプレイヤーが市民カードを出し合った場合にはその市民カードを捨て札としてもう 1 回手札からカードを 1 枚ずつ出し合う.

E カードの大きな特徴として, 両方の手札が対照的ではないことが挙げられる. また, 利得行列から分かるように奴隷側のプレイヤーは奴隷を皇帝に対して出す以外に勝つ手段がなく圧倒的に皇帝側のプレイヤーが有利となっている. そのため, 本環境では奴隷側で勝った場合には賭けたチップの 5 倍の数のチップが払い戻されるように設定している.

表 1: E カードの利得行列

| PlayerA \ PlayerB | 皇帝 | 奴隷 | 市民 |
|---|---|---|---|
| 皇帝 | | (-betA, 5betB) | (betA, -betB) |
| 奴隷 | (5betA, -betB) | | (-betA, betB) |
| 市民 | (-betA, betB) | (betA, -betB) | * |

表 2: エージェントが観測できる Observation

| 属性 | 説明 |
|---|---|
| legal_actions | プレイヤーが選択できる Action のリスト |
| player_money | プレイヤーの所持チップ数 |
| opponent_money | 相手プレイヤーの所持チップ数 |
| player_betmoney | Bet フェーズでプレイヤーが賭けた数 |
| opponent_betmoney | Bet フェーズで相手プレイヤーが賭けた数 |
| player_talked_sentence | Talk フェーズでプレイヤーが話した内容 |
| opponent_talked_sentence | Talk フェーズで相手プレイヤーが話した内容 |
| player_side | プレイヤーのサイド (奴隷 or 皇帝) |
| opponent_side | 相手プレイヤーのサイド |
| player_hand | プレイヤーの手札 |
| game_phase | 現在のフェーズ (Bet or Talk or Play) |
| current_round | 現在のラウンド数 |

# 5 実験

Suspicion Agent 同士を 4 節で解説した E カード環境で対戦させ, エージェントの推論や戦略を観測した. 2 つの Suspicion Agent は同一のプロンプトを持ち, LLM として gpt-4-0613 [1] を使用した.

## 5.1 Observation と Action

Suspicion Agent は現在のゲームの状態を解釈するために Observation を入力として必要とする. 表 2 に Observation の内容を示す. betmoney と talked_sentence に関してはゲームの進行状況によっては値が存在しない場合がある. その際は空文字で埋め合わせている.

また, Suspicion Agent は現在のゲームの状況から適切な Action を出力する. プレイヤーが取りうる Action はゲームのフェーズに応じて変化する.

Bet フェーズでは Action は賭けることのできるチップの数として表現される. プレイヤーが賭けることのできるチップの数は 1, 現在所持しているチップの枚数, およびそれらの間の 10 の倍数であり, プレイヤー

[1] https://platform.openai.com/docs/models/gpt-4-and-gpt-4-turbo

が 41 枚チップを所持している場合には、Action は [1, 10, 20, 30, 40, 41] として表現される. エージェントが 1 を選択したときには, エージェントは 1 枚チップを賭ける. Talk フェーズではエージェントは 'Talk' というアクションのみ取ることができる. 具体的にどのような意図を持ってどのような内容を話すかということは Suspicion Agent の Planning Module が決定する. Play フェーズでは Action は現在の手札のインデックスで表現される. Action として選んだインデックスに対応するカードを選んで出す.

### 5.2 Suspicion Agent のプロンプト

Suspicion Agent の Observation Interpreter の入力には, ゲームがどのようなルールかを示す Game Rule, Observation の解釈の仕方を示す Observation Rule が存在しそれぞれ人力で設定する必要がある. 本研究では, Game Rule として 4 節の内容, Observation Rule として 表 2 の内容をそれぞれ英語で設定した.

また, エージェントが Observation を入力として受け取り, Action を出力として返すまでに図 1 に示すように複数の LLM モジュールが関与する. 各 LLM モジュールについて元実装のプロンプトは, 一部 Leduc Hold'em のルールに特化したプロンプトとなっていた. そのような部分はノイズになると考え E カード環境に沿う形に著者が適切に書き換えた.

## 6 結果

数値実験により, エージェントがゲームのルールや Observation の意味を理解して合理的な判断を下している部分が見られた. 以下のボックスの内容は, 3 ラウンド目にチップを 540 枚持つ奴隷側のプレイヤーが 20 枚チップを持っている相手プレイヤーに対して Play フェーズにおける Action を決定する際の出力の一部である. ここで This plan とは, 市民カードを出すといった Plan を表しており, Player_LLM_1 という名称は相手プレイヤーを表している. なお, このラウンドでは奴隷側のプレイヤーは 50 枚チップを賭けており, 相手プレイヤーは 1 枚チップを賭けている.

合理的な判断を下している例

This plan allows me to gather more information about Player_LLM_1's strategy and potentially outsmart them in the later rounds. Even if I lose this round, my overall game will not be significantly impacted due to my significant chip lead.

現在のお互いのチップの数や, 自身の有利状況, 現在が 3 ラウンド目でありゲームの途中までラウンドがまだ残っていることなどゲームのルールや Observation を適切に解釈して行動を選択していることが観測できる.

しかし, Suspicion Agent を E カード環境へ適用させるにあたって大きく 2 つの問題も生じている.

### 6.1 問題点 1：Game History が長い

Suspicion Agent は相手の行動や自分の取っていた戦略を分析するために, 1 ゲーム終了後にそのゲームの History を LLM を用いて要約してエージェントの Memory に記録する. E カードは 1 ラウンドに 3 つのフェーズがあり, 最大で 6 ラウンドかかるゲームであるのに対して, Ludec Hold'em は 1 ラウンドにおいてプレイヤーは 1 回のみ行動し, 合計 2 ラウンドでゲームが終了する. そのため, E カードの 1 ゲームの長さは Ludec Hold'em と比較してかなり長い. このため History も長くなり LLM で History を要約する際に, トークンが上限を超えてしまいエラーとなってしまう問題が発生した. そのため, 図 1 における All Game Play History に記録することができず, Suspicion Agent の相手の戦略に応じて戦略を適応させる部分について, 検証が及ばない状況となっている.

この問題に対して, History を JSON 形式などプログラム側で扱いやすいように変換し, 要約する部分をプログラムにより実行してトークンの上限を回避する対応を検討している.

### 6.2 問題点 2：Plan の立案の際のパターンが少ない

E カードは Bet フェーズにおいてプレイヤーが 100 枚 チップを持っている場合, 取りうる Action は [1, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100] の 11 通りの Action を取る場合がある. Leduc Hold'em ではラウンドごとにプレイヤーが取りうる Action は Raise, Call, Fold の 3 通りのみである. このように, E カード環境は Leduc Hold'em と比較して多くの Action が可能な

場合があり, そのような場合で E カード環境において LLM で Action を考える際に, 取りうる全ての Action についての考察がされない問題が発生した. 以下のボックスに 取りうる Action から Plan を立案する部分のプロンプトの一部と, Bet フェーズにおいてプレイヤーが 100 枚 チップを持っている場合の出力の一部を示す.

Action から Plan を考えるプロンプトの一部

Make Reasonable Plans:
Please plan several strategies according to actions{valid_action_list}, you can play now to win the finally whole {game_name} games step by step. Output in a tree-structure: ⋯ ,

Bet フェーズの際の出力の一部

Given the complexity of the task, it's not possible to provide a detailed tree-structure output in this format. ⋯,
Plan 1: Bet 1 chip , ⋯ ,
Plan 2: Bet 10 chips, ⋯ ,
Plan 3: Bet 20 chips, ⋯ ,

プロンプトに与えられる valid_action_list はこの場合 [1, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100] であるが, 1, 10, 20 の Action しか考慮されていない.

Talk フェーズの際の出力の一部

Plan 1: Bluff Strategy
- Talk Phase: I bluff and suggest that I will play my 'Slave' card next.
- Play Phase: I play a 'Citizen' card instead, hoping that Player_LLM_1 will play his 'Emperor' card early.
Plan 2: Truth Strategy
- Talk Phase: I tell the truth and say that I will play a 'Citizen' card next.
- Play Phase: I play a 'Citizen' card, hoping that Player_LLM_1 will believe I am bluffing and play his 'Emperor' card early.
Plan 3: Silent Strategy
- Talk Phase: I remain silent and give no hints about my next move.
- Play Phase: I play a 'Citizen' card, hoping that Player_LLM_1 will play his 'Emperor' card early due to uncertainty.

また, Talk フェーズにおいては, 次の Play フェーズのことまで含めて Plan の立案をしている場合があり, その際において取りうる全ての場合を考慮できていないような出力を返す場合があった. 上記のボックスの内容は奴隷側のプレイヤーが Talk フェーズにおいて Action の決定に先んじて複数の Plan を考えている出力の一部である. 勝利条件などのゲームルールを理解しており 3 通りの Talk の戦略を立てているが, Play フェーズで奴隷カードを出す場合を考慮できていない.

この問題に対しては, プロンプトの最適化によりゲームが多くの状況や行動を取りうる場合に対しても適切に Planning できる方法を模索していきたい.

## 7 まとめと今後の課題

今回の実験では E カード環境を構築し, その環境に対して Suspicion Agent を適用した. 結果として, E カード環境においてゲームのルールや Observation から合理的な判断を下している例も見られたが, 6.1 節, 6.2 節で述べた問題点が見られた.

今後の課題として, 問題点への対応が挙げられる. また, 5.2 節で触れたように Agent のプロンプトにおいて特定のゲーム環境に特化した部分があるため, 最適化により任意のゲーム環境にも適用できるような汎用性の高いプロンプトを構築することを考えている.

また, Suspicion Agent は複数の LLM モジュールから構成されている. アークテクチャの構成を組合せ最適化問題と捉えて進化型計算により新たなアーキテクチャを獲得することで Suspicion Agent における不要なモジュールの部分の検出や新たなプレイヤーモデルの表現方法を得る可能性があると考えている.

## 参考文献


[1] Jiaxian Guo, Bo Yang, Paul Yoo, Bill Yuchen Lin, Yusuke Iwasawa, and Yutaka Matsuo. Suspicion-agent: Playing imperfect information games with theory of mind aware gpt-4, 2023.

[2] Johannes Heinrich and David Silver. Deep reinforcement learning from self-play in imperfect-information games, 2016.

[3] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training. 2018.

[4] OpenAI. Gpt-4 technical report, 2023.

[5] Michal Kosinski. Theory of mind might have spontaneously emerged in large language models, 2023.

[6] 福本伸行. 賭博黙示録カイジ. 講談社, 1996.