Ideas:

Fine tuning LLM for poker advice. Implement simulation, or attach to existing simulation?

How can we have a poker bot explain its decisions in natural language?

Once the bot is trained, create a decision tree or a rule-based system that maps the bot’s decisions to human-understandable explanations

**Step 1**: Find a poker environment/simulation for bot to interact in. (RLcard looks most promising)

**Step 2**: Either train a poker bot using Reinforcement Learning Techniques like DQN or CFR

**Step 3**: Implement explainability - Try Methods like SHAP and LIME, could also try hooking up an LLM - (fine tuning, Retrieval Augemented Generation, or just Zero-shot)

**Step 4**: Determine End Goal: If we can Achieve step 3, create a poker tutor that explains it’s strategies, and/or guides you in a virtual game.

**Step 5**: iF we have time, implement a visual interface

Useful Resources: - RL Card

* **Step 1: Use Explainable AI (XAI) Methods**:
  + Apply techniques like SHAP (Shapley Additive Explanations) or LIME (Local Interpretable Model-agnostic Explanations) to interpret the bot’s decisions. These methods can help identify which features (e.g., hand strength, opponent’s betting pattern) influenced the decision the most.

Using LLM for explanability? Retrieval Augmented Generation

* **Step 2: Translate Explanations**:
  + Convert the insights from SHAP or LIME into natural language. For example, if SHAP identifies hand strength as the key factor for a raise, the bot might say, “My hand’s strength was the main reason I decided to raise.”

Need an existing poker simulation for training our agent, and to test the explaining

Plan:

Start with training an agent on a simple game tic tack to, uno, checkers,