Presentation outline (if you want to add ur own script lines add em here)

Introduction, framing the problem

* Deep learning has seen huge success in perfect information games
  + Applying these concepts/strategies into hidden information and strategic deception remains an open challenge

So what’s the difference between the two?

Complete: All players know everything about the game: players, actions, sequence, and payoffs for all outcomes and game outcomes can be mapped from step one. For example, Google’s AlphaZero to play chess

In comparison incomplete Information games are where at least one player lacks full knowledge about some relevant aspect of the game (e.g., other players' payoffs or characteristics).

Why Traditional Methods don’t work

Approach to the Problem: simplest way possible to explore characteristics of the problem

Imagine a game of poker, but there are no river cards and you are only dealt a single card between 1-10 with higher being the “strongest” hand. You would then play poker as normal, small blind, big blinds, check, betting, folding and all the other mechanics which make up poker. (might take this out) Raises occur in increments of 5 chips, with a maximum bet cap of 100 per round, ensuring that the game remains focused on decision-making under uncertainty rather than resource hacking.

With this simplified game we can categorize the complete and incomplete information as follows:

\*show table\*

Baseline, using a DQN

With a new game, we want to establish a baseline of the performance of previous architecture. To do this, we implemtened a Deep Q-Network (DQN) agent using a multi-layer perceptron (MLP) for Q-learning. Q-learning finds an optimal action-selection policy for any finite Markov decision process (MDP), assigning values to each state. The model contains 4 MLP layers with Tanh activation for non-linearity. To balance exploration and exploitation, the epsilon-greedy strategy [6] is used: the agent chooses a random action with probability ε and the greedy (best-estimated) action with probability 1 − ε. The value of ε decays over time, initially favoring exploration and later shifting toward exploitation.

Baseline is poop, why is it bad, why our model might be good

Our Approach + Results

In games with hidden information, determining the optimal strategy goes beyond simply knowing the current state, as the value of an action is intertwined with the uncertain actions of other players. Q-learning attempts to address this by maintaining a specific belief about opponent strategies and using these beliefs to evaluate states. However, the challenge lies in the fact that players adopt diverse strategies, meaning a belief effective against one opponent may be inaccurate against another. Consequently, a model relying on a fixed belief system may lack robustness, performing well against certain players while struggling significantly against others due to this inherent variability in opponent behavior

Computation time

Since q-learning involves a tree search, even search algorithms such as Monte-Carlo search algorithms still require huge levels of computation, with research showing that since Q-learning directly parameters and updates value functions without explicitly modelling the environment, a lot of input data would be needed to develop a sufficient model-algorithm

<https://proceedings.neurips.cc/paper_files/paper/2018/file/d3b1fb02964aa64e257f9f26a31f72cf-Paper.pdf>

Our Model

* Local minima easier to find in two smaller functions compared to one big thing
* Mixed strategy

Reflection

* Tehcnically, a NN can learn anything, but being an AI engineer is about finding the best way for the network to learn