Deep Counterfactual Regret Minimization

Noam Brown*, Adam Lerer*1, Sam Gross1, Tuomas Sandholm23

*Equal Contribution
1Facebook AI Research
2Carnegie Mellon University
Counterfactual Regret Minimization (CFR)  
[Zinkevich et al. NeurIPS-07]

- **CFR** is the leading algorithm for solving partially observable games
  - Iteratively converges to an equilibrium
  - Used by *every* top poker AI in the past 7 years, including *Libratus*
  - *Every single one* used a *tabular* form of CFR

- This paper introduces a **function approximation** form of CFR using deep neural networks
  - Less domain knowledge
  - Easier to apply to other games
Example of Monte Carlo CFR [Lanctot et al. NeurIPS-09]

- Simulate a game with one player designated as the **traverser**
Example of Monte Carlo CFR [Lanctot et al. NeurIPS-09]

• Simulate a game with one player designated as the traverser
• After game ends, traverser sees how much better she could have done by choosing other actions
Example of Monte Carlo CFR [Lanctot et al. NeurIPS-09]

• Simulate a game with one player designated as the traverser
• After game ends, traverser sees how much better she could have done by choosing other actions
• This difference is added to the action’s regret. In future iterations, actions with higher regret are chosen with higher probability
Example of Monte Carlo CFR [Lanctot et al. NeurIPS-09]

• Simulate a game with one player designated as the traverser
• After game ends, traverser sees how much better she could have done by choosing other actions
• This difference is added to the action’s regret. In future iterations, actions with higher regret are chosen with higher probability
• Process repeats even for hypothetical decision points
Prior Approach: Abstraction in Games

- Requires extensive domain knowledge
  - Several papers written on how to do abstraction just in poker
  - Difficult to extend to other games
Deep CFR

• **Input**: low-level features (visible cards, observed actions)
• **Output**: estimate of action regrets

• On each iteration:
  1. Collect samples of action regrets, add to a buffer
  2. Train a network to predict regrets
  3. Use network’s regret estimates to play on next iteration
Deep CFR

• **Input**: low-level features (visible cards, observed actions)

• **Output**: estimate of action regrets

• On each iteration:
  1. Collect samples of action regrets, add to a buffer
  2. Train a network to predict regrets
  3. Use network’s regret estimates to play on next iteration

• **Theorem**: With arbitrarily high probability, Deep CFR converges to an \( \epsilon \)-Nash equilibrium in two-player zero-sum games, where \( \epsilon \) is determined by prediction error
Experimental results in limit Texas hold’em

- Deep CFR produces superhuman performance in heads-up limit Texas hold’em poker
  - ~10 trillion decision points
  - Once played competitively by humans

- Deep CFR outperforms Neural Fictitious Self Play (NFSP), the prior best deep RL algorithm for partially observable games \[\text{Heinrich & Silver arXiv-15}\]
  - Deep CFR is also much more sample efficient

- Deep CFR is competitive with domain-specific abstraction algorithms
Conclusions

• Among algorithms for non-tabular solving of partially-observable games, Deep CFR is the fastest, most sample-efficient, and produces the best results.

• Uses less domain knowledge than abstraction-based approaches, making it easier to apply to other games.