Dynamic Discounted Counterfactual Regret Minimization

Hang Xu*, **Kai Li*,†**, **Haobo Fu**,
Qiang Fu, **Junliang Xing†**, **Jian Cheng**

{xuhang2020, kai.li, jian.cheng}@**ia.ac.cn**,
{haobofu, leonfu}@**tencent.com**, jlxing@**tsinghua.edu.cn**

Imperfect information games

- Imperfect-information games (IIGs) model strategic interactions between players with hidden information.
- The hidden information is omnipresent in real-world decision-making problems, such as medical treatment, negotiation, and security, making the research on IIGs theoretically and practically crucial.

Nash Equilibrium

- We focus on solving in two-player zero-sum IIGs.
- Nash equilibrium[3]: No player can benefit from unilaterally deviating from the equilibrium.

Counterfactual Regret Minimization (CFR)

- The family of CFR[8] is the most successful approaches to computing Nash equilibrium in IIGs.
- CFR iteratively minimizes both players' regrets so that the time-averaged strategy approaches the Nash equilibrium.
- Repeat T iterations for each information set I:
  - Compute the instantaneous regret $r^t(I, a)$ using strategy $\sigma^t(I)$.
  - Update the cumulative regret $R^t(I, a) = R^{t-1}(I, a) + r^t(I, a)$.
  - Compute the next strategy $\sigma^{t+1}(I, a) \sim \max(0, R^t(I, a))$.
  - Cumulate the strategy $C^t(I, a) = C^{t-1}(I, a) + \pi^{\sigma^t}(I)\sigma^t(I, a)$.
  - Compute the average strategy $\bar{\sigma}^t(I, a) \sim C^t(I, a)$.

CFR Variants

- CFR assigns equal weights to every iteration. One key to improving performance is weighting each iteration non-uniformly.
- CFR+[6]:
  - Cumulate the strategy $C^t(I,a) = C^{t-1}(I,a) + t * \pi^{\sigma^t}(I)\sigma^t(I,a)$.
- LinearCFR[1]
  - Update the cumulative regret $R^t(I,a) = R^{t-1}(I,a) + t * r^t(I,a)$.
  - Cumulate the strategy $C^t(I,a) = C^{t-1}(I,a) + t * \pi^{\sigma^t}(I)\sigma^t(I,a)$.
- DCFR[2]
  - Update the cumulative regret

$$R^t(I,a) = \begin{cases} R^{t-1}(I,a)\frac{(t-1)^\alpha}{(t-1)^\alpha+1} + r^t(I,a), & \text{if } R^{t-1}(I,a) > 0 \\ R^{t-1}(I,a)\frac{(t-1)^\beta}{(t-1)^\beta+1} + r^t(I,a), & \text{otherwise,} \end{cases}$$

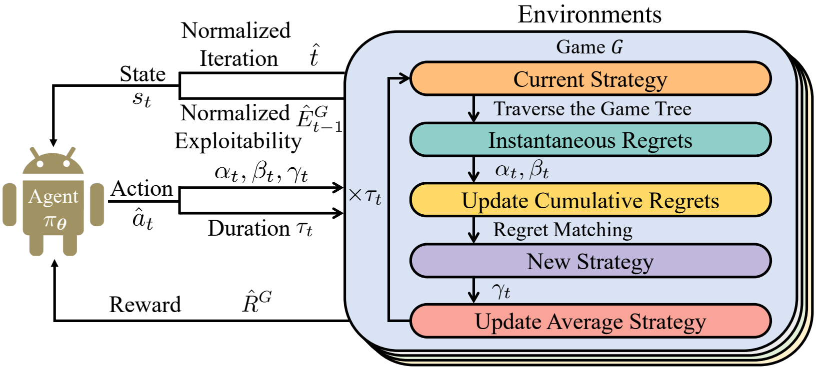
  - Cumulate the strategy $C^t(I,a) = C^{t-1}(I,a)\left(\frac{t-1}{t}\right)^\gamma + \pi^{\sigma^t}(I)\sigma^t(I,a)$.

Motivation

- The discounting CFR variants have obtained remarkable performance in solving IIGs, but exploiting a fixed and manually-specified discounting scheme.
- Pre-determined schemes are not flexible enough, thus inevitably limiting the convergence performance.
- We argue that an ideal scheme should fulfill two criteria:
  - Be automatically learned rather than manually designed.
  - Adjust the weights dynamically instead of using fixed weights
- We propose a novel Dynamic Discounted CFR (DDCFR) framework that weights each iteration using a dynamic, automatically-learned discounting scheme.

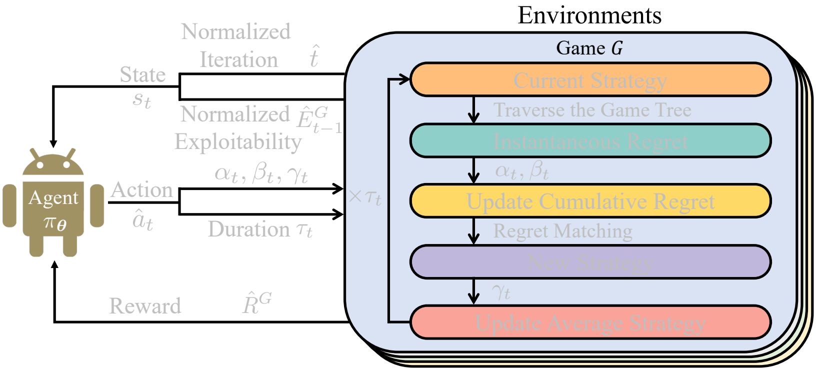
High-level Idea

- DDCFR encapsulates CFR's iteration process into an environment and regard the discounting scheme as an agent interacting with it.
- The interaction process constitutes an MDP $(G, S, A, P^G, \hat{R}^G)$



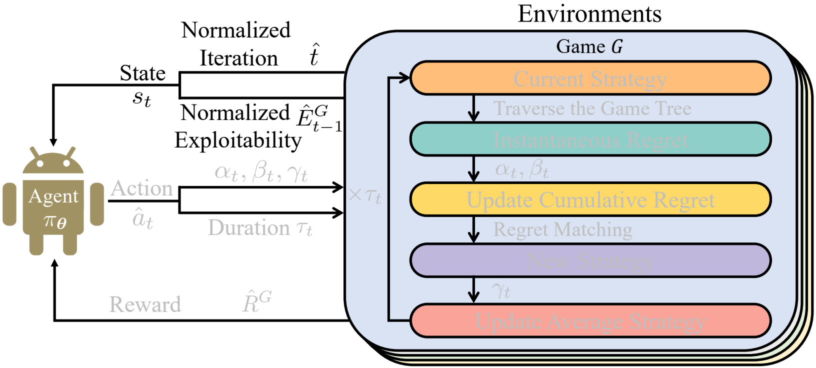
MDP for CFR's Iteration

- The game G: an IIG to be solved.



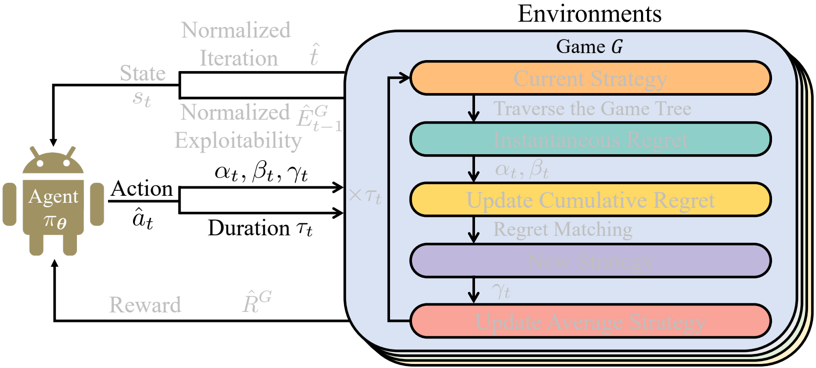
MDP for CFR's Iteration

- The state space S: help the agent make good decisions, and make the learned scheme applicable to different games. It consists of the normalized iteration $\hat{t}$ and the normalized exploitability $\hat{E}^G_{t-1}$.



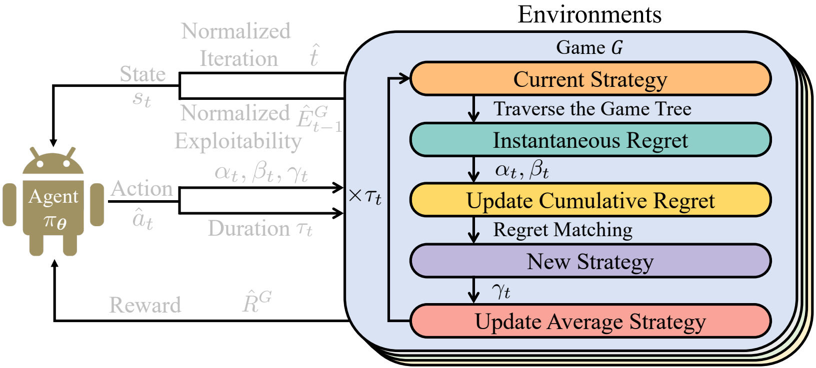
MDP for CFR's Iteration

- The action space A: $\hat{a}_t = [\alpha_t, \beta_t, \gamma_t, \tau_t]$. $\alpha_t, \beta_t, \gamma_t$ are used to determine the discounting weights. τ_t is the duration for how long to use these discounting weights.



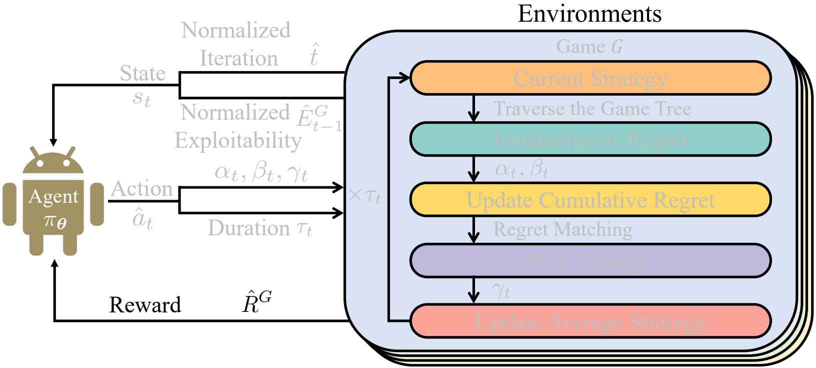
MDP for CFR's Iteration

- The state transition P^G: DDCFR uses the discounting weights calculated by $\alpha_t, \beta_t, \gamma_t, \tau_t$ for τ_t iterations, and the state transitions from s_t to $s_{t+\tau_t}$.



MDP for CFR's Iteration

- the reward function $\hat{R}^G$: the agent receives a reward $\hat{R}^G = \log E_1^G - \log E_T^G$ at the end of the iteration process. E_t^G is the exploitability of the average strategies at the iteration t.



Optimization Objective

- In each game G, the objective is to maximize the final reward, represented as $f^G(\boldsymbol{\theta}) = \hat{R}^G$.
- DDCFR's overall objective is to maximize the average sum of the rewards across the training games $\mathbb{G}$, $f(\boldsymbol{\theta}) = \frac{1}{|\mathbb{G}|} \sum_{G \in \mathbb{G}} f^G(\boldsymbol{\theta})$.
- By optimizing $f(\boldsymbol{\theta})$, our ultimate goal is to learn a generalizable discounting policy that applies to new games.

Theoretical Analysis

- DDCFR is guaranteed to converge to a Nash equilibrium as long as $\alpha_t, \beta_t, \gamma_t$ are within a certain range.

Theorem

Assume that conduct DDCFR T iterations in a two-player zero-sum game. If DDCFR selects hyperparameters as follows: $\alpha_t \in [0, 5]$ for $t < \frac{T}{2}$ and $\alpha_t \in [1, 5]$ for $t \geq \frac{T}{2}$, $\beta_t \in [-5, 0], \gamma_t \in [0, 5]$, the weighted average strategy profile is a $6|\mathcal{I}|\Delta \left(\frac{8}{3}\sqrt{|\mathcal{A}|} + \frac{2}{\sqrt{T}}\right)/\sqrt{T}$-Nash equilibrium.

- The theorem signifies that numerous dynamic discounting schemes converge in theory.
- We then describe how to efficiently optimize the policy to find a well-performing scheme in practice.

Evolution Strategies (ES)

- ES[5;7] has demonstrated its efficacy as a scalable alternative to RL in tackling these challenges.
- As a black box optimization technique, ES is indifferent to the distribution of rewards and tolerant of arbitrarily long time horizons.
- Besides, ES is easy to implement and is highly scalable and efficient to use on distributed hardware.

Method and Acceleration Techniques

- Evolution Strategies (ES)[5]
  - Generate a population of perturbed network parameters $\{\boldsymbol{\theta}^m + \delta\boldsymbol{\epsilon}_i\}_{i=1}^N$, where δ denotes the noise standard deviation and $\boldsymbol{\epsilon}_i \sim \mathcal{N}(0, I)$.
  - Evaluate the performance $f(\boldsymbol{\theta}^m + \delta\boldsymbol{\epsilon}_i)$ of each perturbed parameter $\boldsymbol{\theta}^m + \delta\boldsymbol{\epsilon}_i$.
  - Approximate the gradient estimation with samples.

$$\frac{1}{\delta} * \frac{1}{N} \sum_{i=1}^{N} f(\boldsymbol{\theta}^m + \delta\boldsymbol{\epsilon}_i)\boldsymbol{\epsilon}_i$$

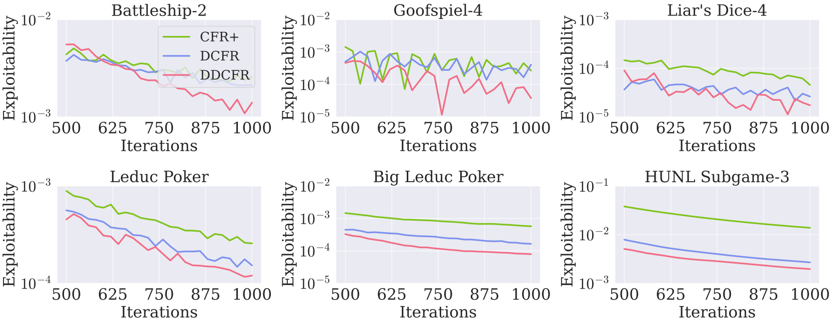
  - Update the parameter using stochastic gradient ascent.

$$\boldsymbol{\theta}^{m+1} \leftarrow \boldsymbol{\theta}^m + \frac{lr}{\delta \cdot N} \sum_{i=1}^{N} f(\boldsymbol{\theta}^m + \delta\boldsymbol{\epsilon}_i)\boldsymbol{\epsilon}_i$$

- Acceleration Techniques
  - Antithetic estimator[4].
  - Fitness shaping[7].
  - Parallelism.

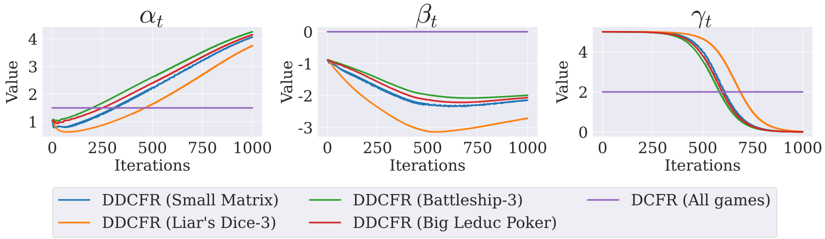
Comparison To Discounting CFR Variants

- DDCFR achieves competitive performance on training games and unseen testing games against the other CFR variants, thanks to the learned dynamic discounting scheme's ability to adjust the discounting weights on the fly using information available at runtime.



Learned Dynamic Discounting Scheme

- We visualize the actions of the learned discounting scheme during the iteration process.
- The learned discounting scheme behaves differently in various games yet exhibits a similar trend. Compared with DCFR's fixed discounting scheme (we can view DCFR as a special case of DDCFR, where $\alpha_1 = 1.5, \beta_1{=}0, \gamma_1{=}2, \tau_1{=}\infty$), it is more aggressive in the earlier iterations and becomes more moderate as the iteration progresses.



Conclusion

- We present DDCFR, the first equilibrium-finding framework that discounts prior iterations using an automatically-learned dynamic scheme.
- We first formulate CFR's iteration process as a carefully designed MDP and transform the discounting scheme learning problem into a policy optimization problem.
- We then exploit a scalable ES-based algorithm to optimize the discounting policy efficiently.
- The learned discounting policy exhibits strong generalization ability, achieving competitive performance on both training games and new testing games.

Thanks!

[1] Noam Brown, Adam Lerer, Sam Gross, and Tuomas Sandholm. Deep counterfactual regret minimization. In *International Conference on Machine Learning*, pages 793–802, 2019.

[2] Noam Brown and Tuomas Sandholm. Solving imperfect-information games via discounted regret minimization. In *AAAI Conference on Artificial Intelligence*, pages 1829–1836, 2019.

[3] Jr John F Nash. Equilibrium points in n-person games. *Proceedings of the National Academy of Sciences of the United States of America*, 36(1):48–49, 1950.

[4] Yurii Nesterov and Vladimir Spokoiny. Random gradient-free minimization of convex functions. *Foundations of Computational Mathematics*, 17(2):527–566, 2017.

[5] Tim Salimans, Jonathan Ho, Xi Chen, Szymon Sidor, and Ilya Sutskever. Evolution strategies as a scalable alternative to reinforcement learning. *arXiv preprint arXiv:1703.03864*, 2017.

[6] Oskari Tammelin. Solving large imperfect information games using CFR+. *arXiv preprint arXiv:1407.5042*, 2014.

[7] Daan Wierstra, Tom Schaul, Tobias Glasmachers, Yi Sun, Jan Peters, and Jürgen Schmidhuber. Natural evolution strategies. *Journal of Machine Learning Research*, 15(1):949–980, 2014.

[8] Martin Zinkevich, Michael Johanson, Michael Bowling, and Carmelo Piccione. Regret minimization in games with incomplete information. In *Advances in Neural Information Processing Systems*, pages 1729–1736, 2007.