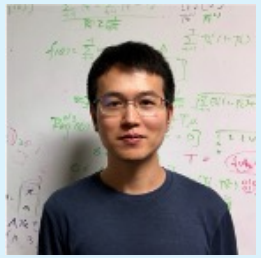
Near-Optimal No-Regret Learning for General Convex Games






Gabriele Farina*, Ioannis Anagnostides*, Haipeng Luo, Chung-Wei Lee, Christian Kroer, Tuomas Sandholm

gfarina@{cs.cmu.edu | meta.com}

NeurIPS 2022

Paper overview

What regret guarantees can be obtained for no-regret learning agents playing against each other?

Problem with a rich history:

Paper overview

What regret guarantees can be obtained for no-regret learning agents playing against each other?

Problem with a rich history:

- Question first formulated by Daskalakis et al. [2011] for two-player zero-sum matrix games

Paper overview

What regret guarantees can be obtained for no-regret learning agents playing against each other?

Problem with a rich history:

- Question first formulated by Daskalakis et al. [2011] for two-player zero-sum matrix games
- Since then: considerable interest in extending guarantees to more general setting

Paper overview

What regret guarantees can be obtained for no-regret learning agents playing against each other?

Problem with a rich history:

- Question first formulated by Daskalakis et al. [2011] for two-player zero-sum matrix games
- Since then: considerable interest in extending guarantees to more general setting
- Chiang et al. [2012] and Rakhlin and Sridharan [2013] pioneered the framework of **optimism**

Paper overview

What regret guarantees can be obtained for no-regret learning agents playing against each other?

Problem with a rich history:

- Question first formulated by Daskalakis et al. [2011] for two-player zero-sum matrix games
- Since then: considerable interest in extending guarantees to more general setting
- Chiang et al. [2012] and Rakhlin and Sridharan [2013] pioneered the framework of **optimism**
- Syrgkanis et al. [2015] crystallized the **RVU property** and established dynamics with $O(T^{1/4})$ per-player regret in convex games

Paper overview

What regret guarantees can be obtained for no-regret learning agents playing against each other?

Problem with a rich history:

- Question first formulated by Daskalakis et al. [2011] for two-player zero-sum matrix games
- Since then: considerable interest in extending guarantees to more general setting
- Chiang et al. [2012] and Rakhlin and Sridharan [2013] pioneered the framework of **optimism**
- Syrgkanis et al. [2015] crystallized the **RVU property** and established dynamics with $O(T^{1/4})$ per-player regret in convex games
- Chen and Peng [2020] improves to $O(T^{1/6})$ in *two-player* games

Paper overview

- Daskalakis et al. [2021] shows that in matrix games one can achieve $O(\log^4 T)$ by using the OMWU algorithm
  - *Exponential improvement* over the guarantees obtained using traditional techniques within the no-regret framework!

Paper overview

- Daskalakis et al. [2021] shows that in matrix games one can achieve $O(\log^4 T)$ by using the OMWU algorithm
  - *Exponential improvement* over the guarantees obtained using traditional techniques within the no-regret framework!
- Anagnostides et al. [2022] extendes the technique of Daskalakis et al. [2021] to swap regret in matrix games

Paper overview

- Daskalakis et al. [2021] shows that in matrix games one can achieve $O(\log^4 T)$ by using the OMWU algorithm
  - *Exponential improvement* over the guarantees obtained using traditional techniques within the no-regret framework!
- Anagnostides et al. [2022] extendes the technique of Daskalakis et al. [2021] to swap regret in matrix games
- Farina et al. [2022] show that in certain classes of polyhedral games (including extensive-form games) one can run OMWU on the (exponentially many) vertices in polynomial time thanks to a **kernel trick** ($\rightarrow$ Kernelized OMWU)

Paper overview

- Daskalakis et al. [2021] shows that in matrix games one can achieve $O(\log^4 T)$ by using the OMWU algorithm
  - *Exponential improvement* over the guarantees obtained using traditional techniques within the no-regret framework!
- Anagnostides et al. [2022] extendes the technique of Daskalakis et al. [2021] to swap regret in matrix games
- Farina et al. [2022] show that in certain classes of polyhedral games (including extensive-form games) one can run OMWU on the (exponentially many) vertices in polynomial time thanks to a **kernel trick** ($\rightarrow$ Kernelized OMWU)
- Piliouras et al. [2022] proposes learning dynamics that guarantee $O(1)$ regret for a **subsequence** of iterates in matrix games

Paper overview

So far these results have only been limited to certain classes of **games with structured strategy spaces**

- Mostly normal-form games
- Extensive-form games via **kernelized** multiplicative weights

Paper overview

So far these results have only been limited to certain classes of **games with structured strategy spaces**

- Mostly normal-form games
- Extensive-form games via **kernelized** multiplicative weights

Yet, many fundamental models in economics and multiagent systems require more **general, convex** strategy sets

Paper overview

So far these results have only been limited to certain classes of **games with structured strategy spaces**

- Mostly normal-form games
- Extensive-form games via **kernelized** multiplicative weights

Yet, many fundamental models in economics and multiagent systems require more **general, convex** strategy sets

Q: Can $O(\text{polylog}\, T)$ regret be attained in general convex and compact strategy sets while retaining efficient strategy updates?

Paper overview

Q: Can $O(\text{polylog}\, T)$ regret be attained in general convex and compact strategy sets while retaining efficient strategy updates?

- We answer in the **positive**, and give an uncoupled learning algorithm with $O(\log T)$ per-player regret in general convex games
  - In adversarial settings: usual $O(\sqrt{T})$ regret bound

Paper overview

Q: Can $O(\text{polylog}\, T)$ regret be attained in general convex and compact strategy sets while retaining efficient strategy updates?

- We answer in the **positive**, and give an uncoupled learning algorithm with $O(\log T)$ per-player regret in general convex games
  - In adversarial settings: usual $O(\sqrt{T})$ regret bound
- Per-iteration complexity:
  - $O(\log \log T)$ with access to local proximal oracle
  - $O(\text{poly}\, T)$ with access to only a linear optimization oracle

Paper overview

Q: Can $O(\text{polylog}\, T)$ regret be attained in general convex and compact strategy sets while retaining efficient strategy updates?

- We answer in the **positive**, and give an uncoupled learning algorithm with $O(\log T)$ per-player regret in general convex games
  - In adversarial settings: usual $O(\sqrt{T})$ regret bound
- Per-iteration complexity:
  - $O(\log \log T)$ with access to local proximal oracle
  - $O(\text{poly}\, T)$ with access to only a linear optimization oracle
- In special cases where prior results apply, our algorithm improves over the state-of-the-art regret bounds in terms of the dependence on either **the number of iterations** or **dimension of the strategy sets**

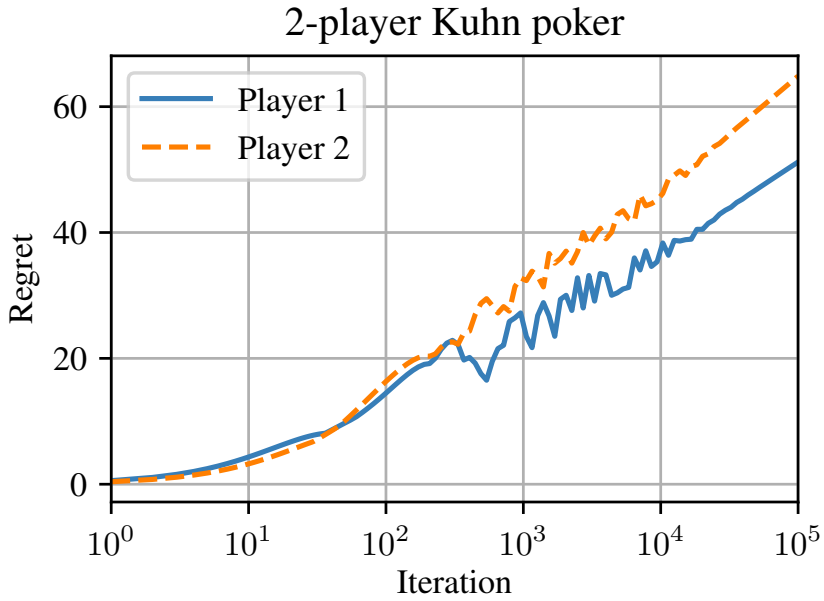
Comparison table

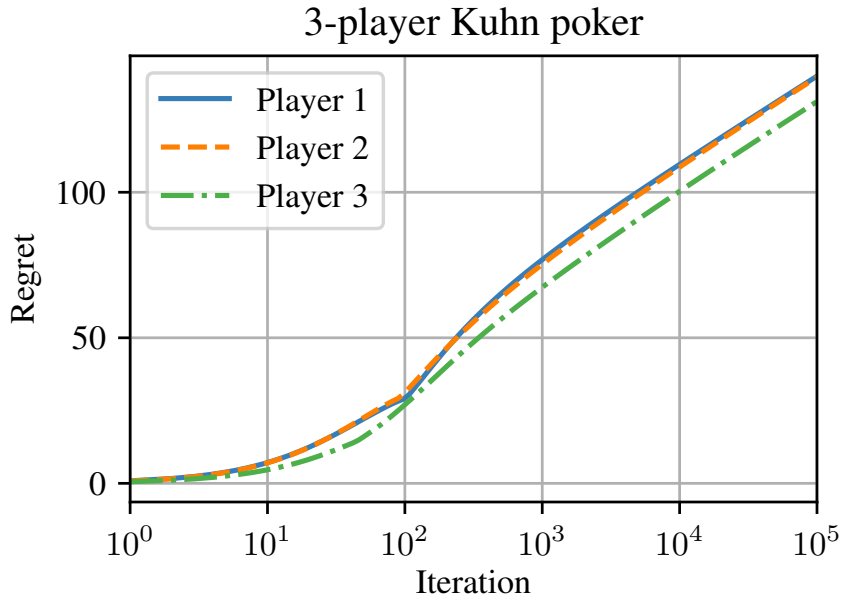
Method	Applies to	Regret bound	Cost per iteration
OFTRL / OMD [Syrgkanis et al., 2015]	General convex set	$O(\sqrt{n}\,\mathfrak{R}\,T^{1/4})$	Regularizer- & oracle- dependent
OMWU [Daskalakis et al., 2021]	Simplex Δ^d	$O(n \log d \log^4 T)$	$O(d)$
Clairvoyant MWU [Piliouras et al., 2022]	Simplex Δ^d	$O(n \log d)$ ⚠ subsequence only!	$O(d)$
Kernelized OMWU [Farina et al., 2022]	Polytope $\Omega = \mathrm{co}\mathcal{V}$ with $\mathcal{V} \subseteq \{0,1\}^d$	$O(n \log \lvert\mathcal{V}\rvert \log^4 T)$	$d \times$ cost of kernel
`LRL-OFTRL` [**This talk**]	General convex set $\mathcal{X} \subseteq \mathbb{R}^d$	$O(nd\lVert\mathcal{X}\rVert_1^2 \log T)$	Oracle-dependent: • $O(\log\log T)$ proximal oracle calls • $O(\mathrm{poly}\, T)$ linear opt. oracle calls

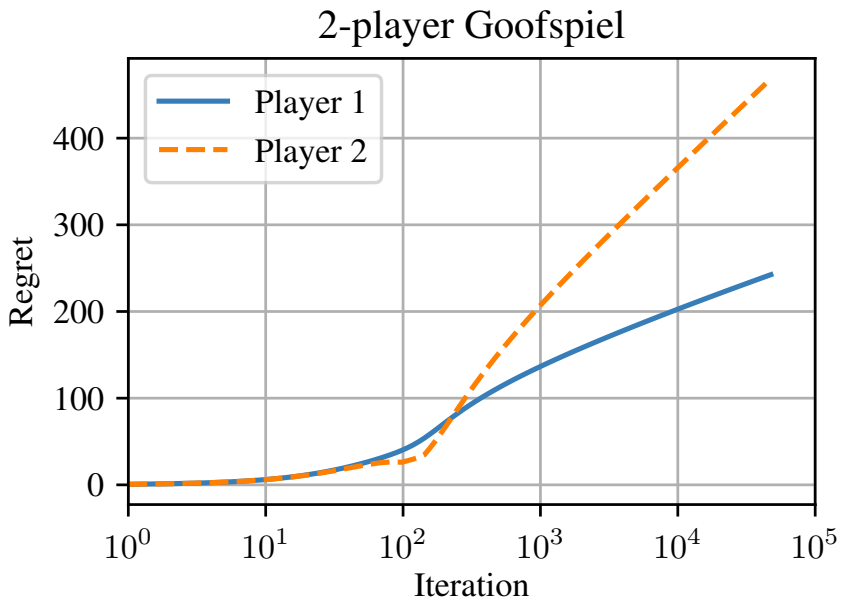
where:

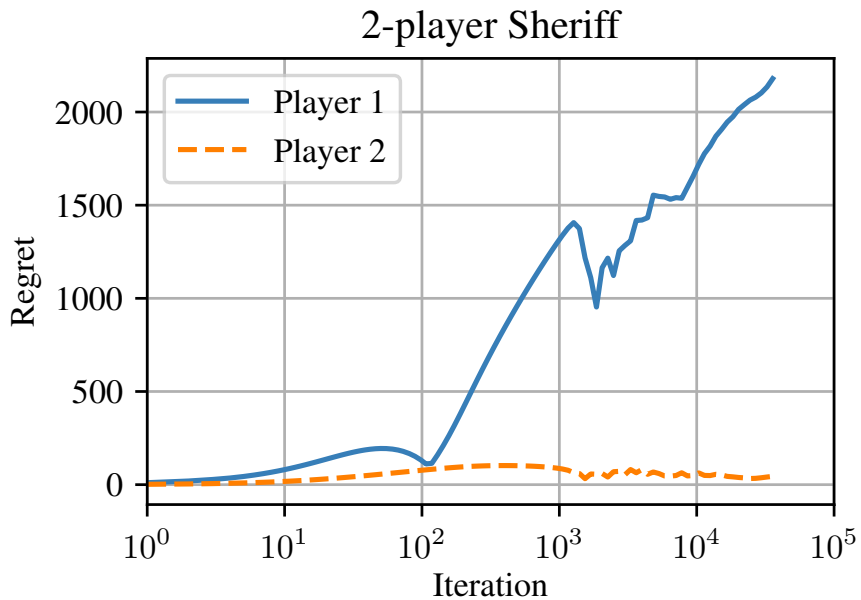
- n: number of players
- T: number of iterations/repetitions of the game
- $\mathfrak{R}$: regularizer-dependent parameter
- $\mathrm{co}\mathcal{V}$: convex hull of $\mathcal{V}$
- $\lVert\mathcal{X}\rVert_1$: upper bound on $\max_{\boldsymbol{x}\in\mathcal{X}} \lVert\boldsymbol{x}\rVert_1$

Experimental results (log **x-axis**)









Thank you!

More video content: You can also find the recording of a long talk on this on the website of the Simons Institute