Learning and Computing Φ-Equilibria

EC 2026 Tutorial

Ioannis Anagnostides
Carnegie Mellon University

Gabriele Farina
MIT

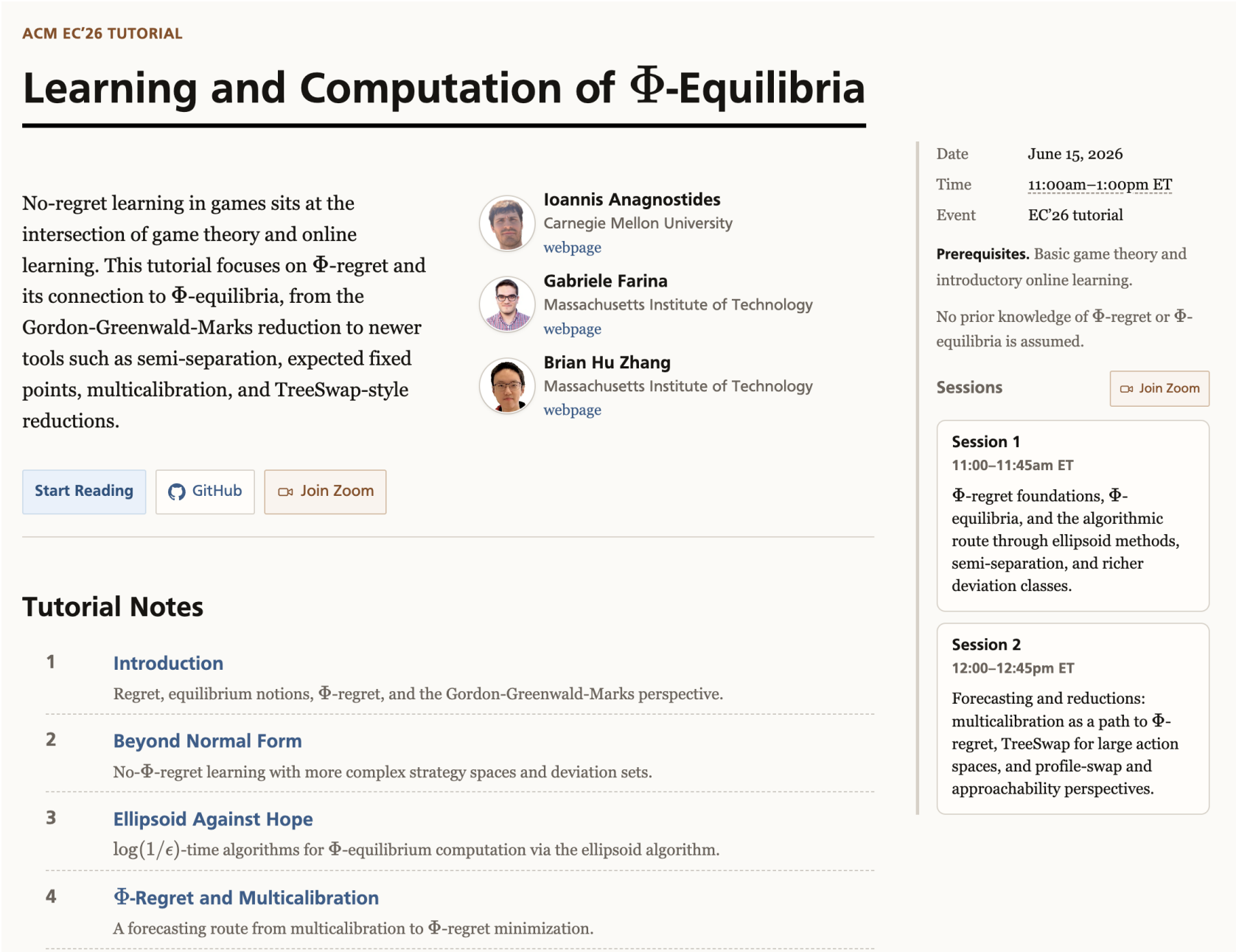
Brian Hu Zhang
MIT

At a glance

- Lot of research in the topic, very central to EC community
- Important connections with different areas (dynamics, games, complexity, online learning, economics, forecasting)
- We tried to systematize the main trends we observe and paint open questions / areas of opportunity

Online notes

- We put together notes for the tutorial
- Available online at ec26-phi-regret-tutorial.github.io



ACM EC'26 TUTORIAL

Learning and Computation of Φ-Equilibria

No-regret learning in games sits at the intersection of game theory and online learning. This tutorial focuses on Φ-regret and its connection to Φ-equilibria, from the Gordon-Greenwald-Marks reduction to newer tools such as semi-separation, expected fixed points, multicalibration, and TreeSwap-style reductions.

Ioannis Anagnostides
Carnegie Mellon University
webpage

Gabriele Farina
Massachusetts Institute of Technology
webpage

Brian Hu Zhang
Massachusetts Institute of Technology
webpage

Start Reading | GitHub | Join Zoom

Date June 15, 2026
Time 11:00am–1:00pm ET
Event EC'26 tutorial

Prerequisites. Basic game theory and introductory online learning.

No prior knowledge of Φ-regret or Φ-equilibria is assumed.

Sessions Join Zoom

Session 1
11:00–11:45am ET
Φ-regret foundations, Φ-equilibria, and the algorithmic route through ellipsoid methods, semi-separation, and richer deviation classes.

Session 2
12:00–12:45pm ET
Forecasting and reductions: multicalibration as a path to Φ-regret, TreeSwap for large action spaces, and profile-swap and approachability perspectives.

Tutorial Notes

1 **Introduction**
Regret, equilibrium notions, Φ-regret, and the Gordon-Greenwald-Marks perspective.

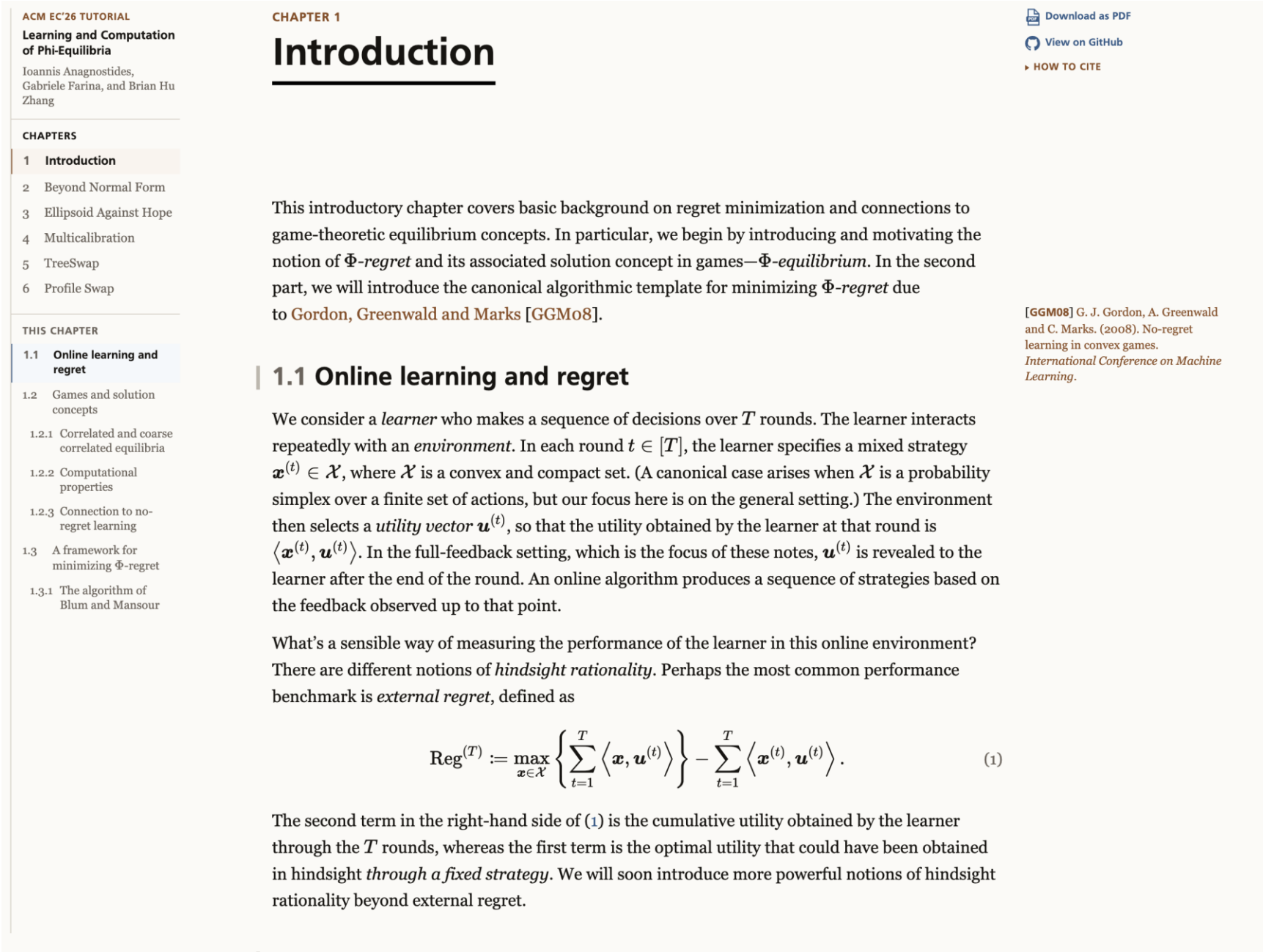
2 **Beyond Normal Form**
No-Φ-regret learning with more complex strategy spaces and deviation sets.

3 **Ellipsoid Against Hope**
$\log(1/\epsilon)$-time algorithms for Φ-equilibrium computation via the ellipsoid algorithm.

4 **Φ-Regret and Multicalibration**
A forecasting route from multicalibration to Φ-regret minimization.

Online notes

- We have both an HTML and a PDF version
- Feel free to report issues or open pull requests on GitHub!



ACM EC'26 TUTORIAL

Learning and Computation of Phi-Equilibria

Ioannis Anagnostides, Gabriele Farina, and Brian Hu Zhang

CHAPTERS

THIS CHAPTER

CHAPTER 1

Introduction

Download as PDF

View on GitHub

HOW TO CITE

This introductory chapter covers basic background on regret minimization and connections to game-theoretic equilibrium concepts. In particular, we begin by introducing and motivating the notion of Φ-*regret* and its associated solution concept in games—Φ-*equilibrium*. In the second part, we will introduce the canonical algorithmic template for minimizing Φ-*regret* due to Gordon, Greenwald and Marks [GGM08].

[GGM08] G. J. Gordon, A. Greenwald and C. Marks. (2008). No-regret learning in convex games. *International Conference on Machine Learning*.

1.1 Online learning and regret

We consider a *learner* who makes a sequence of decisions over T rounds. The learner interacts repeatedly with an *environment*. In each round $t \in [T]$, the learner specifies a mixed strategy $\boldsymbol{x}^{(t)} \in \mathcal{X}$, where $\mathcal{X}$ is a convex and compact set. (A canonical case arises when $\mathcal{X}$ is a probability simplex over a finite set of actions, but our focus here is on the general setting.) The environment then selects a *utility vector* $\boldsymbol{u}^{(t)}$, so that the utility obtained by the learner at that round is $\left\langle \boldsymbol{x}^{(t)}, \boldsymbol{u}^{(t)} \right\rangle$. In the full-feedback setting, which is the focus of these notes, $\boldsymbol{u}^{(t)}$ is revealed to the learner after the end of the round. An online algorithm produces a sequence of strategies based on the feedback observed up to that point.

What's a sensible way of measuring the performance of the learner in this online environment? There are different notions of *hindsight rationality*. Perhaps the most common performance benchmark is *external regret*, defined as

$$\mathrm{Reg}^{(T)} := \max_{\boldsymbol{x} \in \mathcal{X}} \left\{ \sum_{t=1}^{T} \left\langle \boldsymbol{x}, \boldsymbol{u}^{(t)} \right\rangle \right\} - \sum_{t=1}^{T} \left\langle \boldsymbol{x}^{(t)}, \boldsymbol{u}^{(t)} \right\rangle. \tag{1}$$

The second term in the right-hand side of (1) is the cumulative utility obtained by the learner through the T rounds, whereas the first term is the optimal utility that could have been obtained in hindsight *through a fixed strategy*. We will soon introduce more powerful notions of hindsight rationality beyond external regret.

Introduction

Online learning and regret minimization

Strategy set X

For rounds $t = 1, \dots, T$:

- Learner chooses strategy $\boldsymbol{x}^t \in X$
- Environment chooses utility $\boldsymbol{u}^t \in [0,1]^n$
- Learner observes $\boldsymbol{u}^t$, gets utility $\langle \boldsymbol{u}^t, \boldsymbol{x}^t \rangle$

How to measure performance in this online setting?

Online learning and regret minimization

Strategy set X

For rounds $t = 1, \dots, T$:

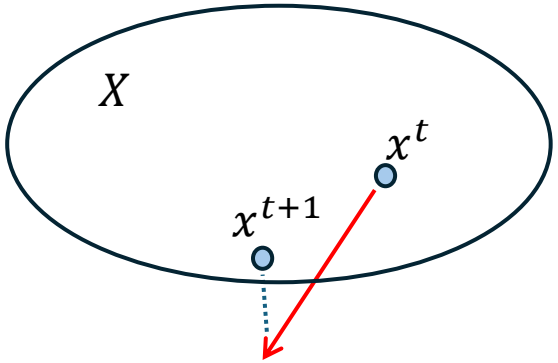
- Learner chooses strategy $\boldsymbol{x}^t \in X$
- Environment chooses utility $\boldsymbol{u}^t \in [0,1]^n$
- Learner observes $\boldsymbol{u}^t$, gets utility $\langle \boldsymbol{u}^t, \boldsymbol{x}^t \rangle$

How to measure performance in this online setting?

External regret

$$\max_{\boldsymbol{x} \in X} \sum_{t=1}^{T} \langle \boldsymbol{u}^t, \boldsymbol{x} - \boldsymbol{x}^t \rangle$$

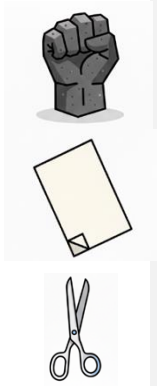
Always easy to guarantee sublinear:
projected gradient descent



Leads to convergence to coarse correlated equilibria!

External regret is not strong enough

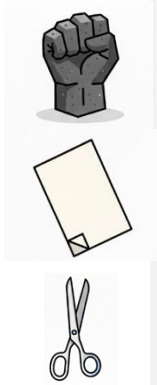
- This learner has zero external regret
- However, there is a "simple" behavior modification that would be profitable for the learner
- Can we call this learner rational?



$t=1$	$t=2$								$t=T$
1	0	0	1	0	0	1	0	0	1
2	0	1	0	0	1	0	0	1	0
3	1	0	0	1	0	0	1	0	0

External regret is not strong enough

- This learner has zero external regret
- However, there is a "simple" behavior modification that would be profitable for the learner
- Can we call this learner rational?

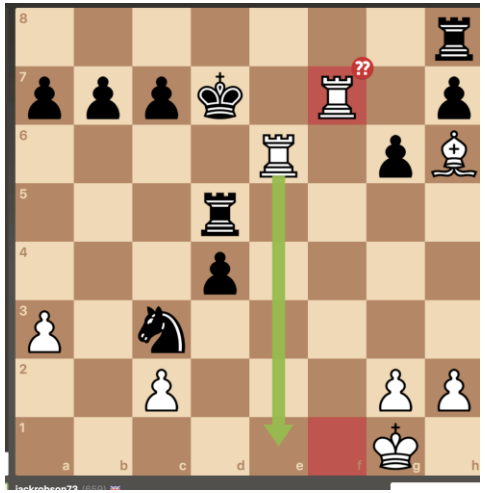


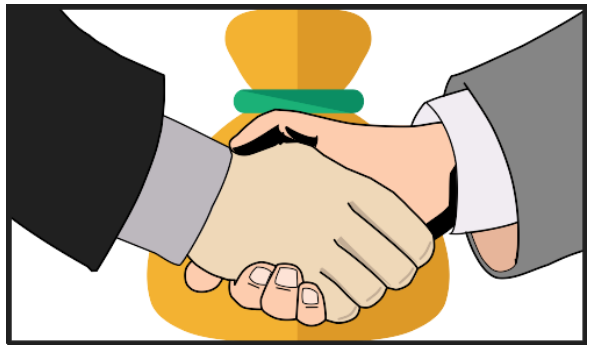
	$t=1$	$t=2$							$t=T$
	1	0	0	1	0	0	1	0	0	1
	2	0	1	0	0	1	0	0	1	0
	3	1	0	0	1	0	0	1	0	0

Φ-regret formalizes the meaning of behavior modification and hindsight rationality

Pitfalls of external regret

- Coarse correlated equilibria can be supported on *strictly dominated actions* (Viossat and Zapechelnyuk; JET 2013)
- No-external-regret algorithms can be manipulated (Deng, Schneider, and Sivan; NeurIPS 2019)
- No-external-regret algorithms can lead to collusive outcomes (Nisan and Zerah; 2026, Hartline, Wang, and Zhang; CSL 2025)





Beyond external regret

For a set of transformations Φ on X,

Φ regret

$$\max_{\phi \in \Phi} \sum_{t=1}^{T} \langle u^t, \phi(x^t) - x^t \rangle$$

The learner experiences no regret with respect to behavior modifications per Φ

Larger Φ → Stronger notion of learning → Stronger notion of equilibrium

External regret: Φ = **all constant** transformations $X \to X$.

Swap regret: Φ = **all** transformations $X \to X$.

Beyond external regret

For a set of transformations Φ on X,

Φ regret

$$\max_{\phi \in \Phi} \sum_{t=1}^{T} \langle u^t, \phi(x^t) - x^t \rangle$$

The learner experiences no regret with respect to behavior modifications per Φ

Larger Φ → Stronger notion of learning → Stronger notion of equilibrium

External regret: Φ = **all constant** transformations $X \to X$.

Swap regret: Φ = **all** transformations $X \to X$.

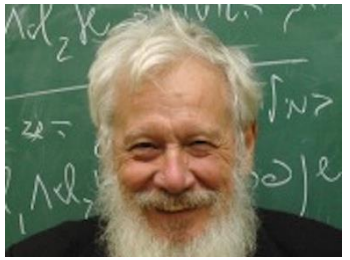
What is the largest Φ that is tractable in all games?

Normal-form games

X = Probability simplex

Coarse correlated equilibrium

Correlated equilibrium



[E.g., Blum and Mansour; JMLR 2007]

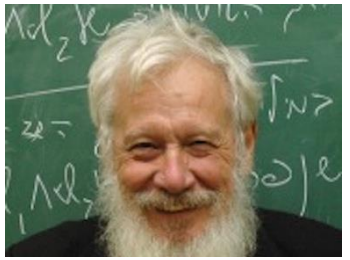
Normal-form games

X = Probability simplex

Coarse correlated equilibrium

Correlated equilibrium

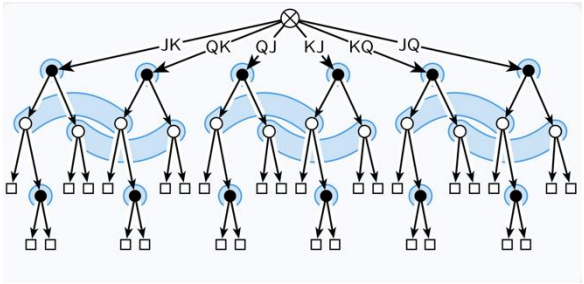
In most settings, the number of pure strategies is exponential



[E.g., Blum and Mansour; JMLR 2007]

Sequential games

X = Sequence-form polytope



External

Swap

Easy

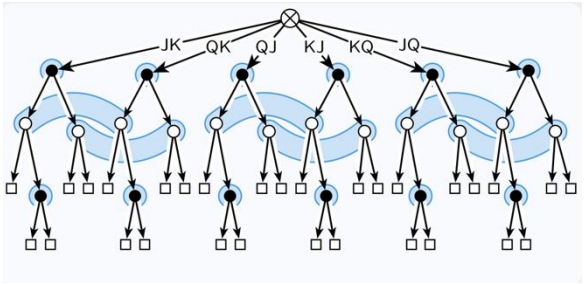
$\sqrt{T}$

Hard to *online learn* at meaningfully sublinear rate

[DFGSZ; 2025]

Sequential games

X = Sequence-form polytope



External

Swap

Easy

$\sqrt{T}$

(Coarse) trigger deviations

$\sqrt{T}$

Extensive form (coarse) correlated eq.

[FCMG; JACM 2022]

Linear transformations

$\sqrt{T}$

Linear correlated eq.

[FP; NeurIPS 2023]

Polynomial swap

$\sqrt{T}$

[ZAFS; NeurIPS 2024]

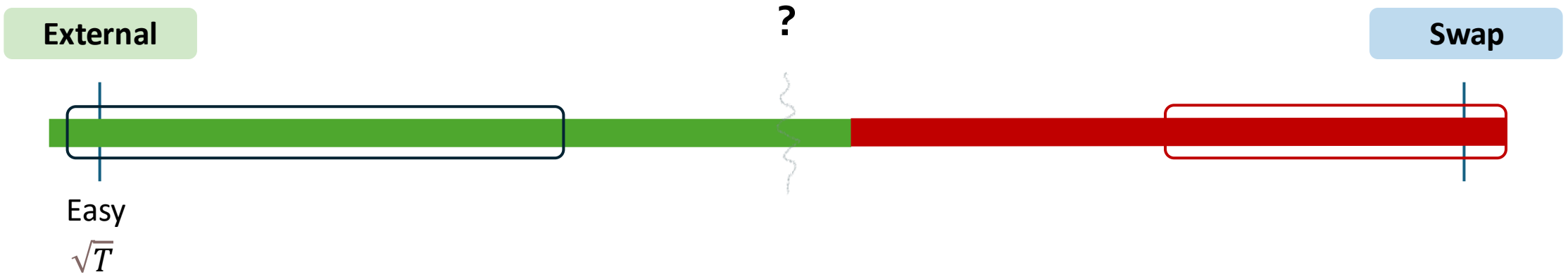
Hard to *online learn* at meaningfully sublinear rate

[DFGSZ; 2025]

The general case

X convex and compact

(Oracle access)



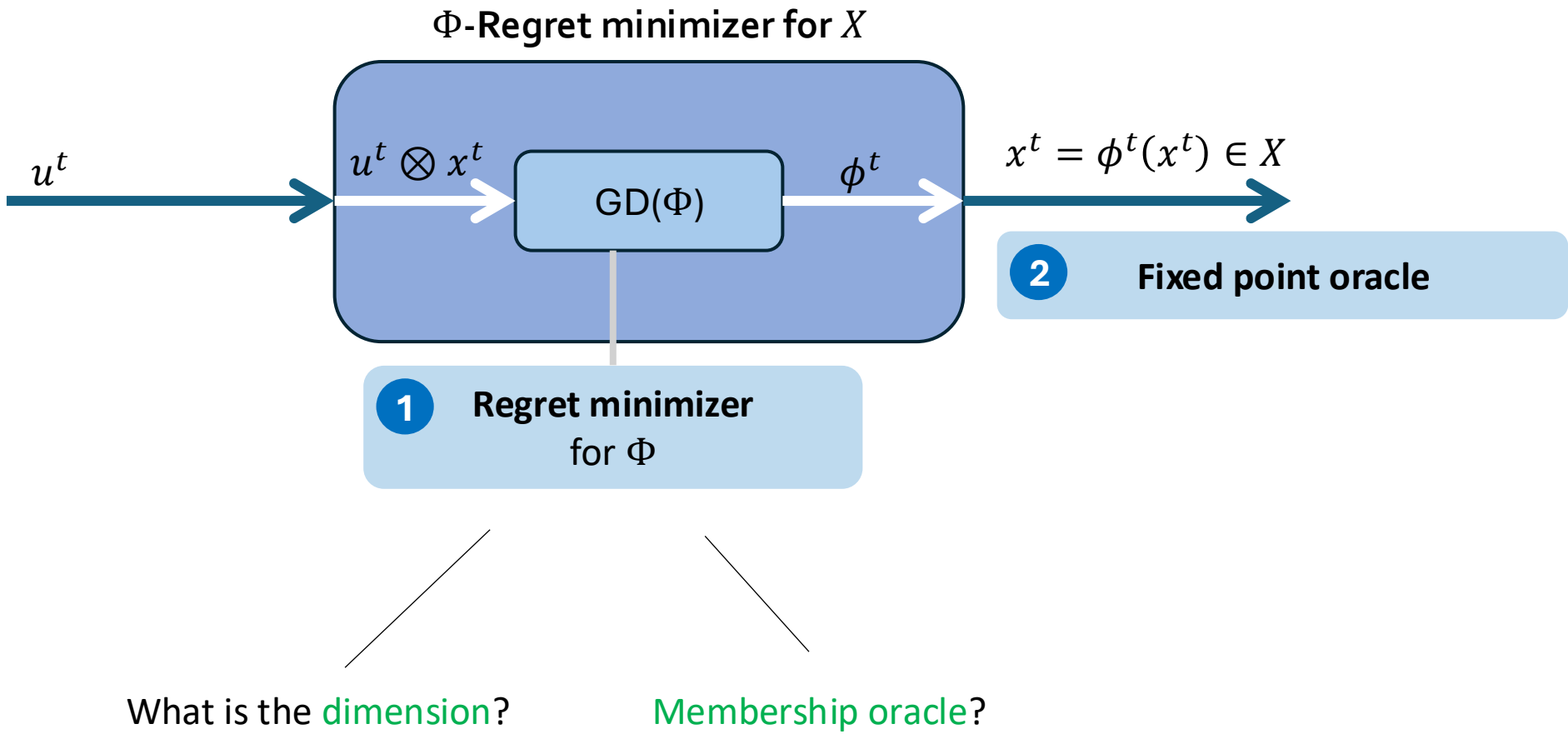
Q1: What is the strongest notion of Phi-regret that is easy to keep under control efficiently?

Q2: What is the strongest notion of Phi-equilibrium that can be computed efficiently?

GGM's construction

[Gordon, Greenwald and Marks; ICML 2008]

Reduces: Φ-regret minimization on X → external regret minimization on Φ



Blum and Mansour for swap regret

The key is to characterize the set of *endomorphisms* Φ

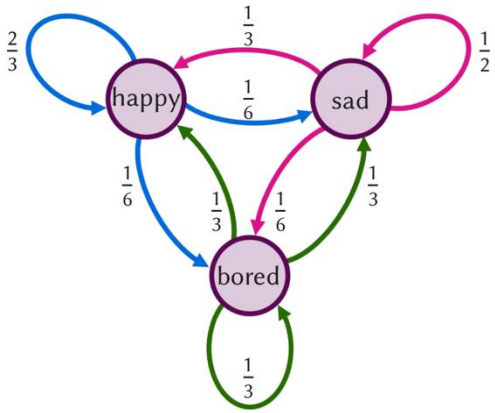
Blum and Mansour; JMLR 2007

Blum and Mansour for swap regret

The key is to characterize the set of *endomorphisms* Φ

For swap regret, this is the set of **stochastic matrices**

The fixed point is any stationary distribution of the Markov chain



Algorithm: Swap regret minimizer

Input: A regret minimizer R_a for each action $a \in \mathcal{A}$

`NextStrategy`():

  for each action $a \in \mathcal{A}$ **do**

    $\boldsymbol{x}_a^{(t)} := R_a$.`NextStrategy`()

  Set $\mathbf{M}^{(t)} := \left[\left(\boldsymbol{x}_a^{(t)}\right)_{a \in \mathcal{A}}\right]$

  return a fixed point $\boldsymbol{x}^{(t)} = \mathbf{M}^{(t)}\boldsymbol{x}^{(t)}$

`ObserveUtility`($\boldsymbol{u}^{(t)} \in \mathbb{R}^{\mathcal{A}}$):

  for each action $a \in \mathcal{A}$ **do**

    Set $\boldsymbol{u}_a^{(t)} := \boldsymbol{x}^{(t)}[a]\boldsymbol{u}^{(t)}$

    R_a.`ObserveUtility`$\left(\boldsymbol{u}_a^{(t)}\right)$

Blum and Mansour; JMLR 2007

Linear swap regret
and semi-separation

Linear swap regret

What if Φ contains **linear** endomorphisms?

$$\max_{\phi \in \Phi_{\text{linear}}} \sum_{t=1}^{T} \langle u^t, \phi(x^t) - x^t \rangle$$

Fixed point oracle

Linear swap regret

What if Φ contains **linear** endomorphisms?

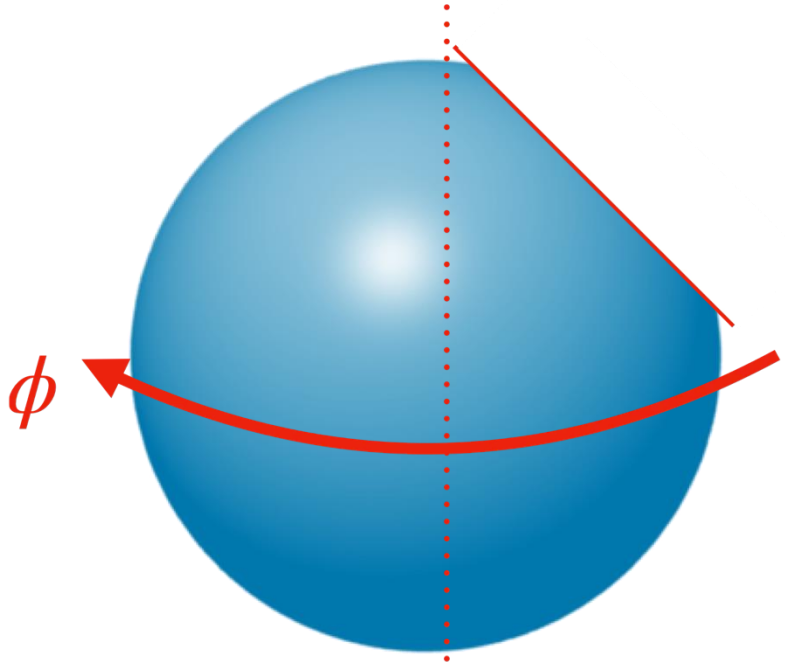
$$\max_{\phi \in \Phi_{\text{linear}}} \sum_{t=1}^{T} \langle u^t, \phi(x^t) - x^t \rangle$$

Fixed point oracle

✓

Regret minimizer for Φ

✗



✗ Given a linear function, it is computationally hard to ascertain whether it maps X to X [DFFPS; STOC 2025]

✗ We cannot apply the GGM framework

Semi-separation

[DFFPS; STOC 2025]

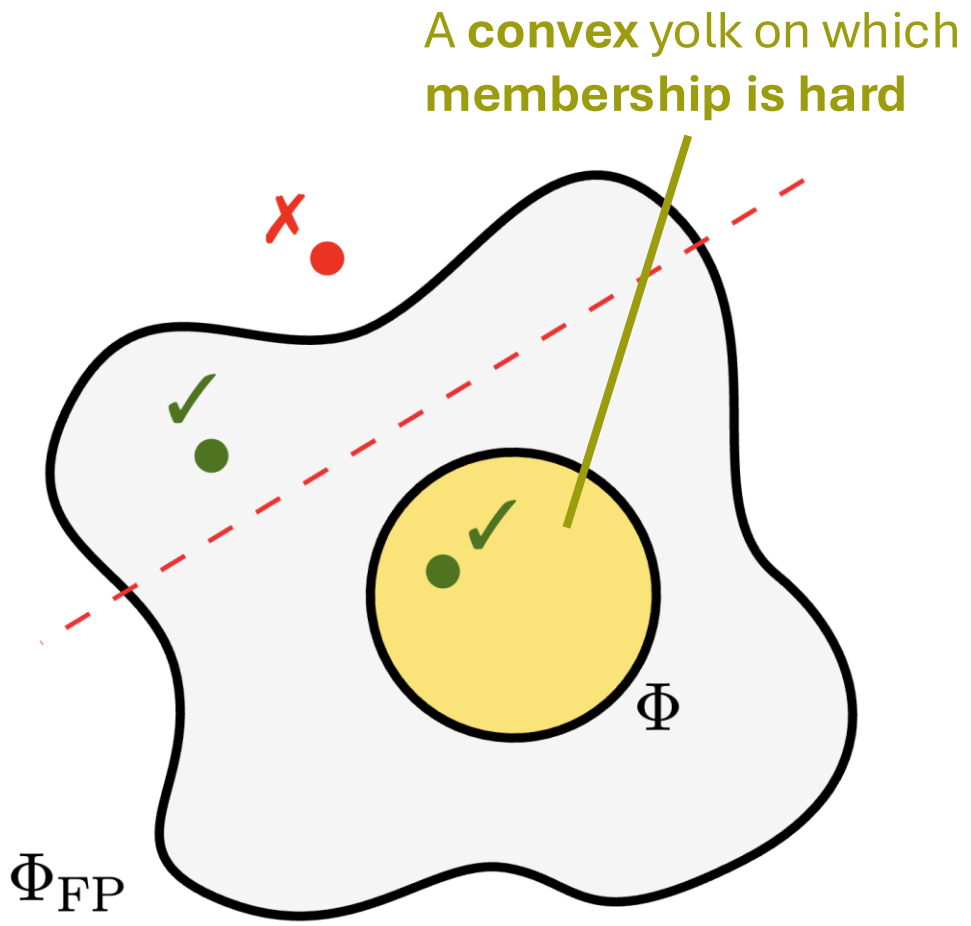
As long as $\phi^{(t)}$ **has a fixed point in X,** we can at least run the algorithm

We can perhaps relax the condition of $\phi^{(t)}$ being an endomorphism, and only check if it admits a fixed point

Semi-separation oracle

Given ϕ, either:

- Assert that ϕ has a fixed point in X; or
- Return a direction separating ϕ from Φ



A **nonconvex** white on which **membership** is **easy**

Constructing the semi-separation oracle [DFFPS; STOC 2025]

Given a linear transformation ϕ, either:
- Assert that ϕ has a fixed point in X; or
- return a direction separating ϕ from Φ

Easy for linear transformations:
start by minimizing $\|\phi(x) - x\|_2^2$

If the minimum is 0, the x^* that attains it is a fixed point of ϕ.

Else, it is easy to construct a separating direction starting from the quantities $u := \phi(x^*) - x^*$ and $\mathrm{p}^* := \arg\max_{p \in X} \langle u, p \rangle$

Using the semi-separation oracle

[DFFPS; STOC 2025]

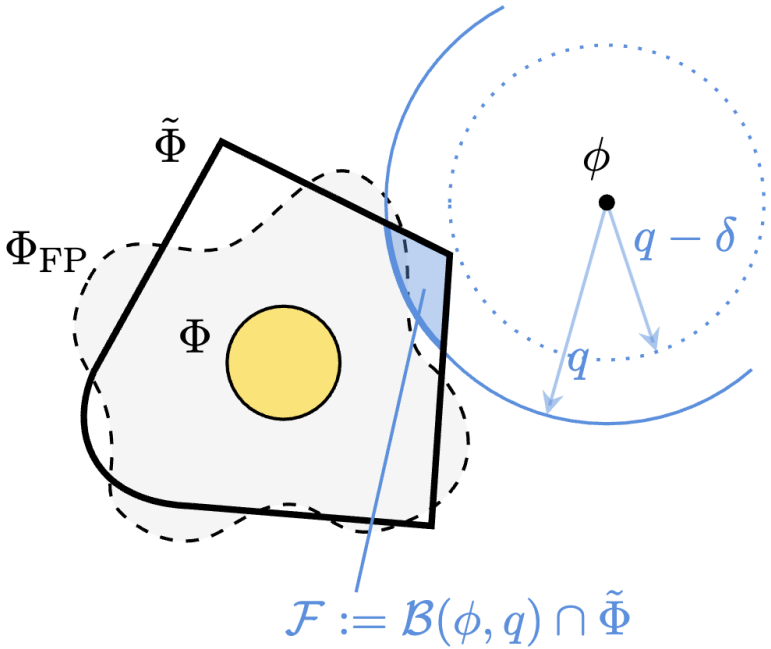
Key insight: as long as we can justify ϕ^t as a projected gradient descent step onto some convex compact $\widetilde{\Phi}^t \supseteq \Phi$, regret is minimized

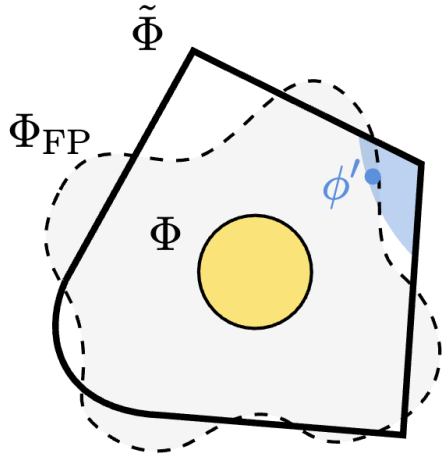
- At every iteration we use our semi-separation to construct a "shell" $\widetilde{\Phi}^t$ and a candidate projection ϕ^t
- The algorithm is called shell projection

The algorithm's idea

[DFFPS; STOC 2025]

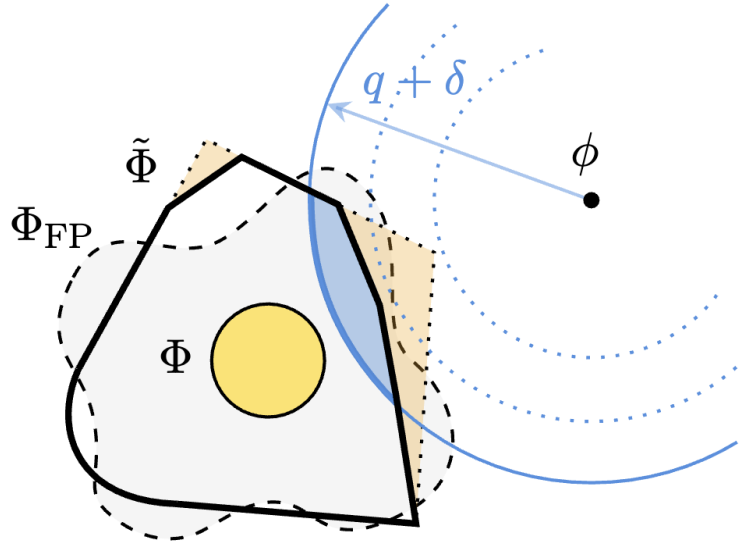
Case (I). ShellEllipsoid($\mathcal{F}$) returns a transformation $\phi' \in \Phi_{\mathrm{FP}} \cap \mathcal{F}$. We terminate and return ϕ'.





Terminate

Case (II). ShellEllipsoid($\mathcal{F}$) returns a separating frontier. We curtail $\tilde{\Phi}$ and increase the radius q.



Manipulability
and swap regret

Manipulability and swap regret

No-external-regret algorithms can be manipulated

Learner Optimizer

If the optimizer can asymptotically improve over it's Stackelberg value, the learner is **manipulable**

$$\text{Stack}(u_o) := \max_{y \in Y} \max_{x \in \text{BR}(y)} u_o(x, y)$$

Swap regret guarantees non-manipulability!

Deng, Schneider, and Sivan (NeurIPS 2019)

Manipulability and swap regret

No-external-regret algorithms can be manipulated

Learner Optimizer

If the optimizer can asymptotically improve over it's Stackelberg value, the learner is **manipulable**

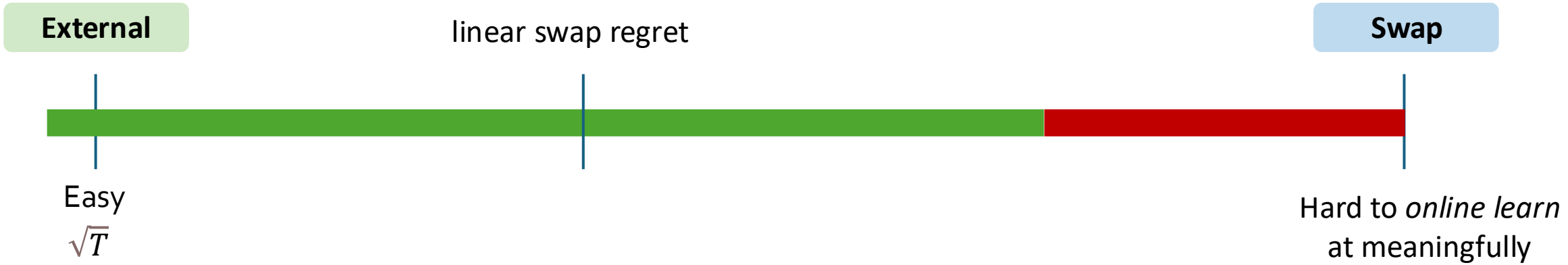
$$\text{Stack}(u_o) := \max_{y \in Y} \max_{x \in \text{BR}(y)} u_o(x, y)$$

Swap regret guarantees non-manipulability!

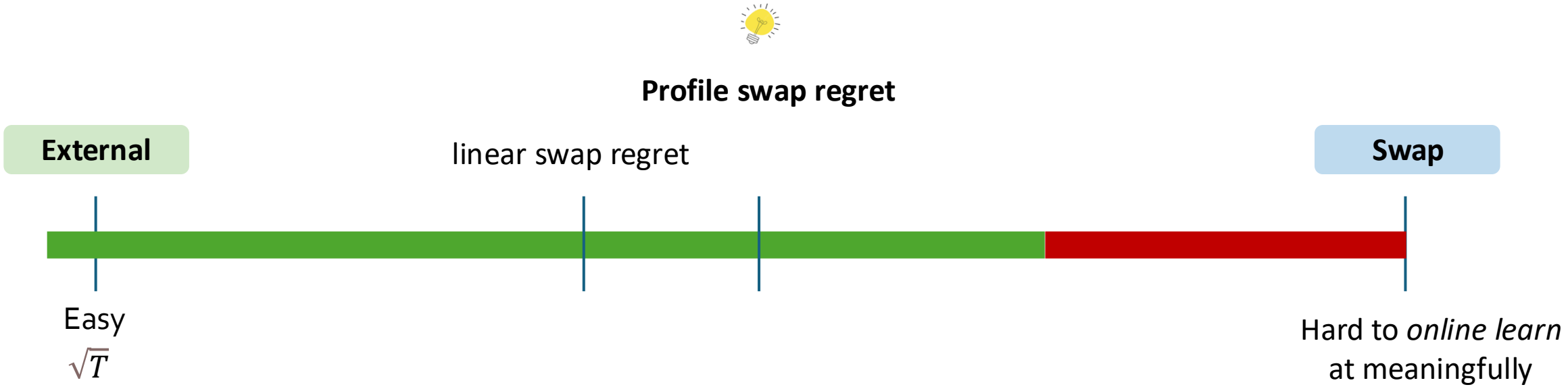
Deng, Schneider, and Sivan (NeurIPS 2019)

However, swap regret is intractable…

Profile swap regret



Profile swap regret



Even in complex strategy sets, such as extensive-form, there is an efficient algorithm that guarantees non-manipulability!

Arunachaleswaran, Collina, Mansour, Mohri, Schneider, and Sivan (EC 2025 best paper!)

Profile swap regret via approachability

- Simple way to minimize profile swap regret via **response-based approachability**!
- Bypasses the GMM framework
- It relies on solving a **bilinear zero-sum** game

Profile swap distance is equal to the loss of a suitable **approachability instance**

Algorithm [BS15]: Response-based approachability

Input: Horizon T, sets $\mathcal{X}, \mathcal{U}$, best-response map $b : \mathcal{U} \to \mathcal{X}$

Initialize $\mathbf{U}^{(0)} := 0 \in \mathbb{R}^{d \times (d+1)}$

for each round $t = 1, ..., T$ **do**

Compute maximin strategies $\left(\boldsymbol{x}^{(t)}, \boldsymbol{u}_*^{(t)}\right)$ for

$$\max_{\boldsymbol{x} \in \mathcal{X}} \min_{\boldsymbol{u} \in \mathcal{U}} \left\langle \mathbf{U}^{(t-1)}, (\boldsymbol{u} \otimes \boldsymbol{x}, \boldsymbol{u}) \right\rangle$$

Set $\boldsymbol{s}^{(t)} := \left(\boldsymbol{u}_*^{(t)} \otimes b\left(\boldsymbol{u}_*^{(t)}\right), \boldsymbol{u}_*^{(t)}\right)$

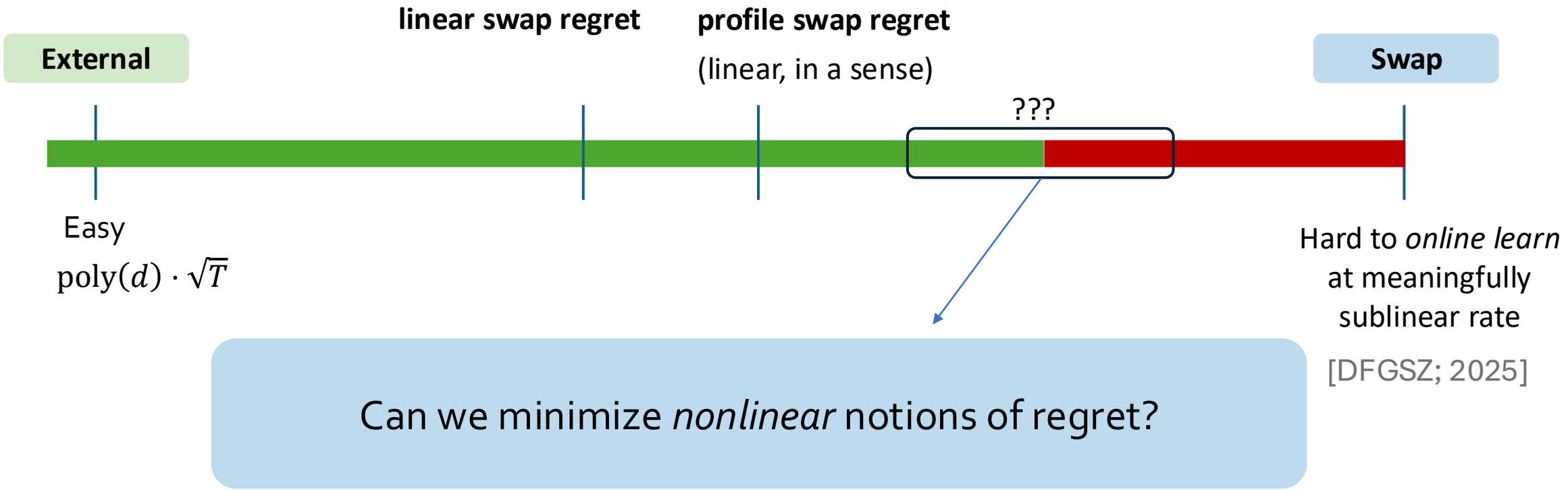
Play $\boldsymbol{x}^{(t)}$ and observe $\boldsymbol{u}^{(t)}$

Set $\kappa^{(t)} := \left(\boldsymbol{u}^{(t)} \otimes \boldsymbol{x}^{(t)}, \boldsymbol{u}^{(t)}\right)$

Update $\mathbf{U}^{(t)} := \mathbf{U}^{(t-1)} + \kappa^{(t)} - \boldsymbol{s}^{(t)}$

Nonlinear deviations

Beyond linear functions



k-dimensional deviation sets

Given *feature map* $q: X \to \mathbb{R}^k$, define

$$\Phi^q := \{\text{maps } \phi : X \to X \text{ of the form } \phi^{\mathbf{M}}(\boldsymbol{x}) \coloneqq \mathbf{M}q(x) \text{ where } \mathbf{M} \in \mathbb{R}^{d \times k}\}$$

Example (degree-ℓ polynomials):

$$q(\boldsymbol{x}) = (\text{all monomials in } \boldsymbol{x} \text{ of degree} \leq \ell) \in \mathbb{R}^{d^{O(\ell)}}$$

Recall semi-separation:
Given a transformation ϕ, either:

- find a fixed point of ϕ in X, or
- find $\boldsymbol{x} \in X$ with $\phi(\boldsymbol{x}) \notin X$

→ Problem: Computing fixed points is hard when ϕ is nonlinear

Expected fixed points

Definition: An ϵ-**expected fixed point** of a function $\phi: X \to \mathbb{R}^d$ is a distribution $\mu \in \Delta(X)$ such that

$$\left\| \mathop{\mathbb{E}}_{x \sim \mu} [\phi(\boldsymbol{x}) - \boldsymbol{x}] \right\|_2 \leq \epsilon$$

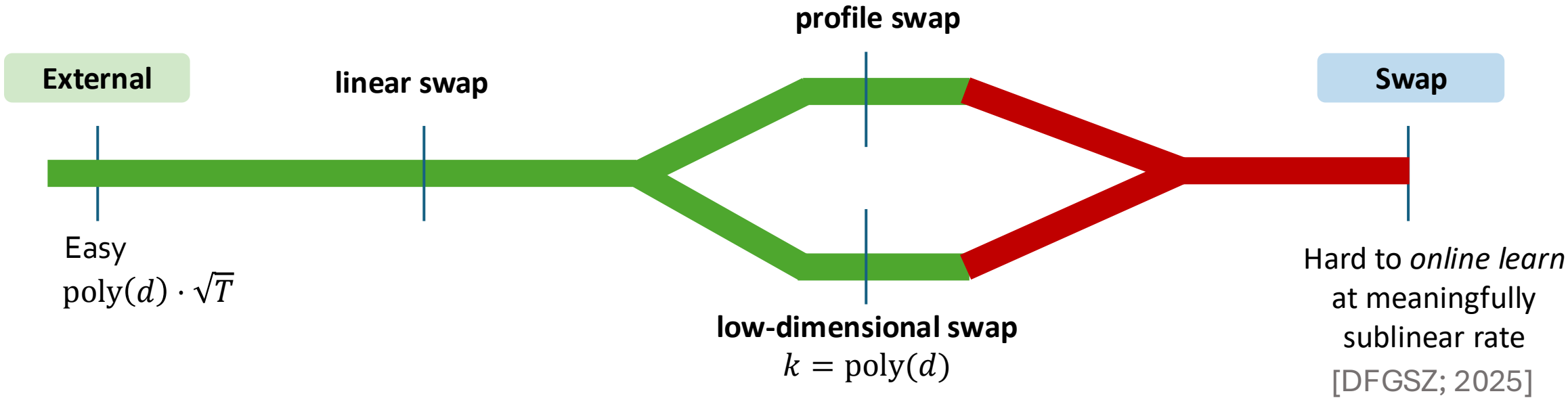
Observation #1: Semi-separation for expected fixed points is easy

Idea: Iterate ϕ. Set $\boldsymbol{x}_1$ = arbitrary, $\boldsymbol{x}_\ell = \phi(\boldsymbol{x}_{\ell-1})$ for $\ell > 1$

Then $\mathrm{Unif}\{\boldsymbol{x}_1, \dots, \boldsymbol{x}_L\}$ is an $O(1/L)$-EFP. If any $\boldsymbol{x}_\ell \notin X$, use it to separate.

Observation #2: The GGM framework only needs EFPs, not actual fixed points

⇒ **Theorem** [**ZA**TB**F**CS EC'25]:
Regret minimization is possible on Φ^q even for nonlinear feature map q!


External
linear swap
profile swap
Swap
Easy
$\mathrm{poly}(d) \cdot \sqrt{T}$
low-dimensional swap
$k = \mathrm{poly}(d)$
Hard to *online learn*
at meaningfully
sublinear rate
[DFGSZ; 2025]

TreeSwap

Slightly sublinear swap regret

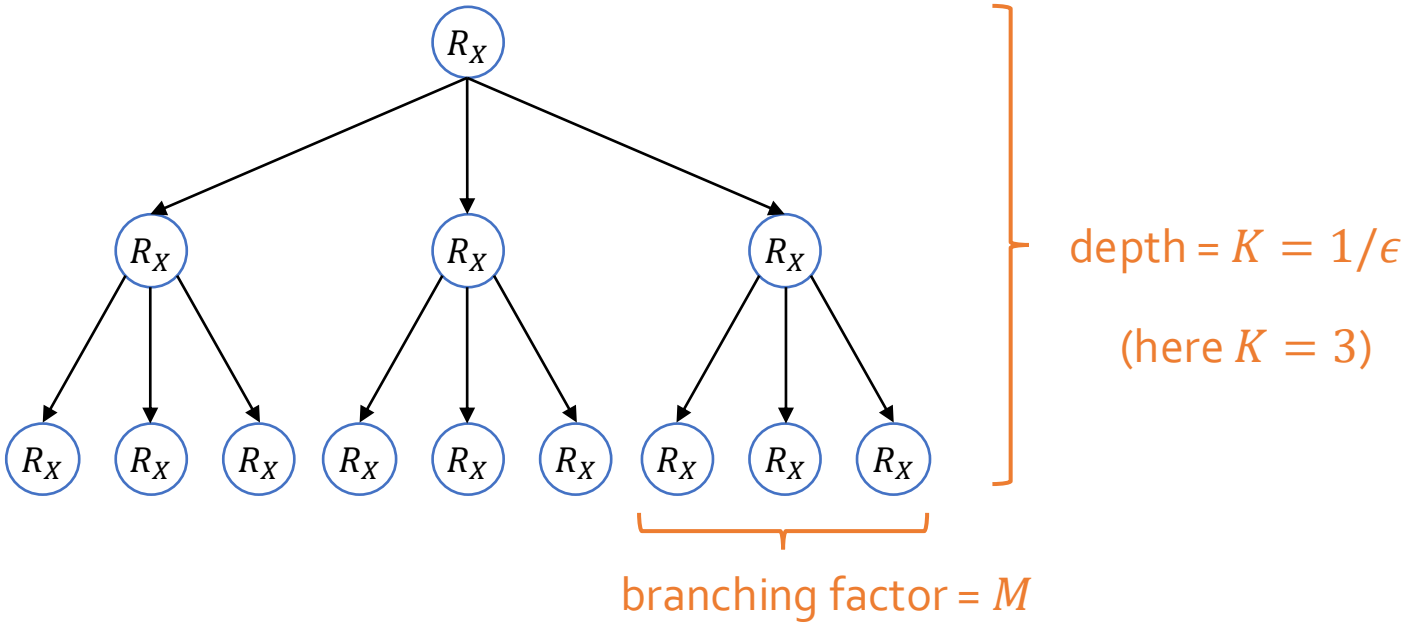
TreeSwap

Peng & Rubinstein (*STOC'24*); Dagan, Daskalakis, Fishelson, Golowich (*STOC'24*)

Given: External regret minimizer R_X on $X \subset [0,1]^n$ achieving ϵM regret after M steps (e.g., with gradient descent, $M = \text{poly}(d)/\epsilon^2$)

Goal: Build a **swap regret minimizer** on X

Idea:



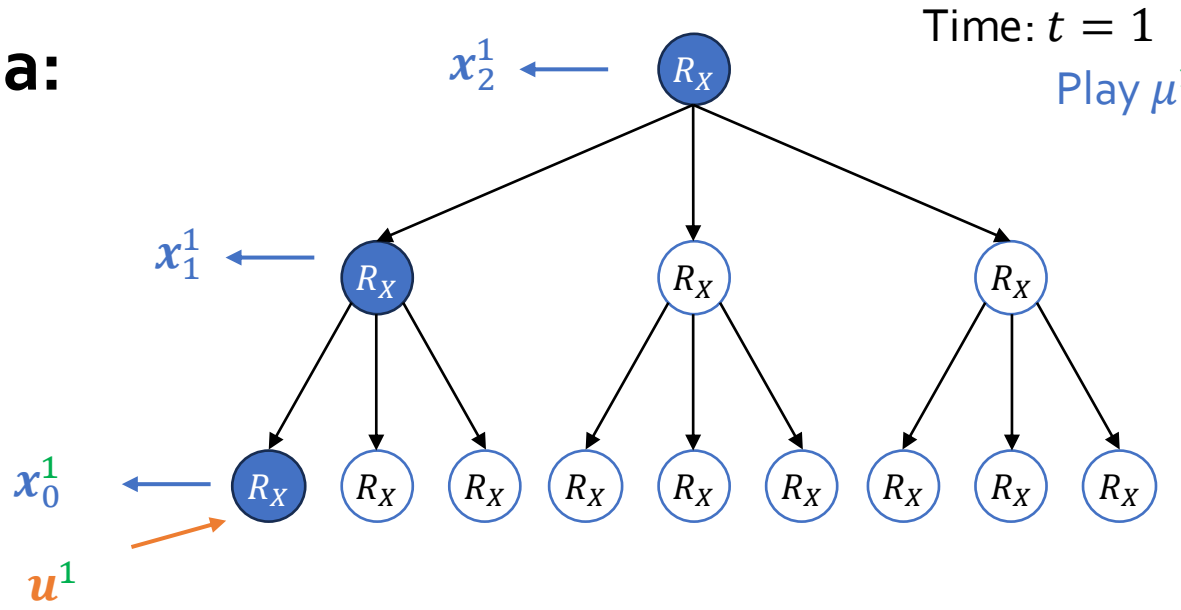
TreeSwap

Peng & Rubinstein (*STOC*'24); Dagan, Daskalakis, Fishelson, Golowich (*STOC*'24)

Given: External regret minimizer R_X on $X \subset [0,1]^n$ achieving ϵM regret after M steps (e.g., with gradient descent, $M = \text{poly}(d)/\epsilon^2$)

Goal: Build a **swap regret minimizer** on X

Idea:



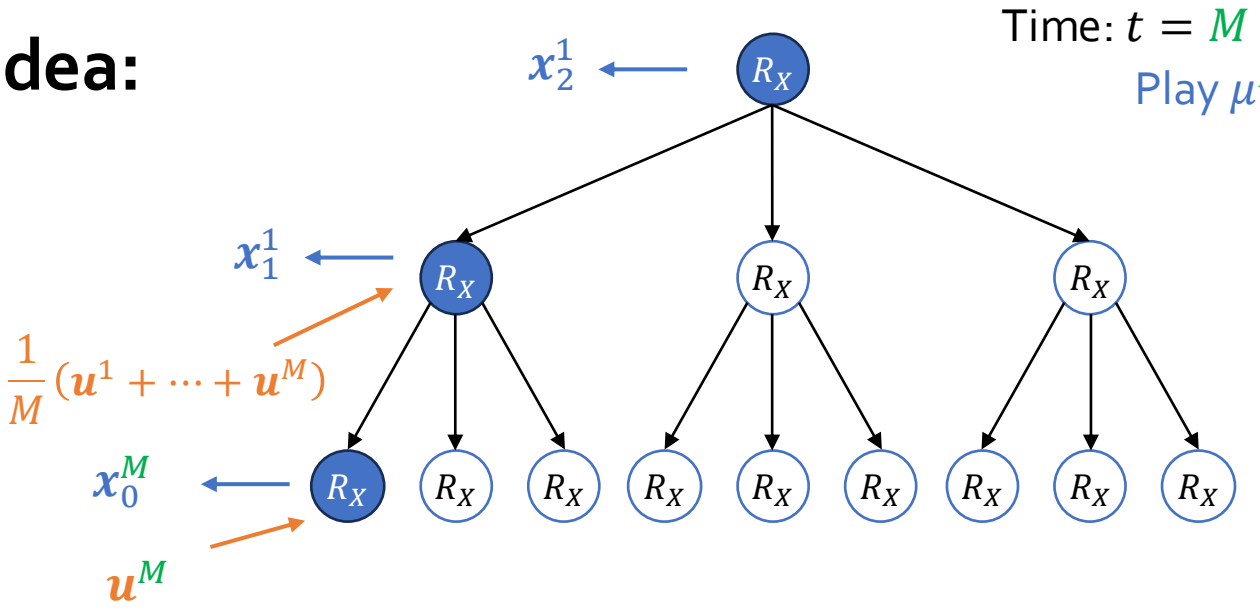
TreeSwap

Peng & Rubinstein (*STOC*'24); Dagan, Daskalakis, Fishelson, Golowich (*STOC*'24)

Given: External regret minimizer R_X on $X \subset [0,1]^n$ achieving ϵM regret after M steps (e.g., with gradient descent, $M = \text{poly}(d)/\epsilon^2$)

Goal: Build a **swap regret minimizer** on X

Idea:



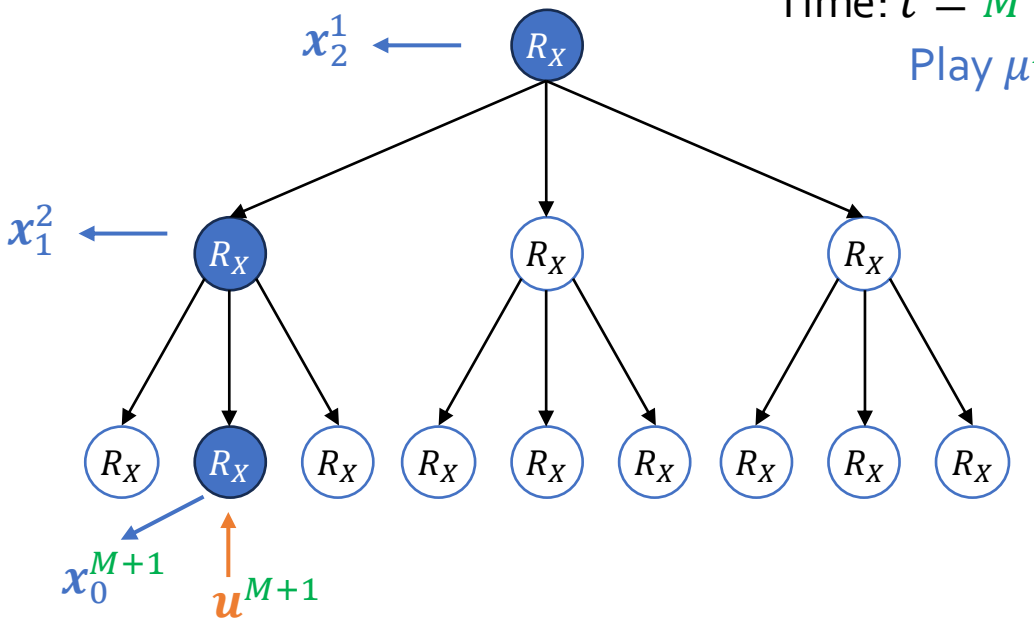
TreeSwap

Peng & Rubinstein (*STOC'24*); Dagan, Daskalakis, Fishelson, Golowich (*STOC'24*)

Given: External regret minimizer R_X on $X \subset [0,1]^n$ achieving ϵM regret after M steps (e.g., with gradient descent, $M = \text{poly}(d)/\epsilon^2$)

Goal: Build a **swap regret minimizer** on X

Idea:



Time: $t = M + 1$

Play $\mu^{M+1} := \text{Unif}\{x_0^{M+1}, x_1^1 \ldots, , x_{K-1}^1\}$

(Mixed strategy!)

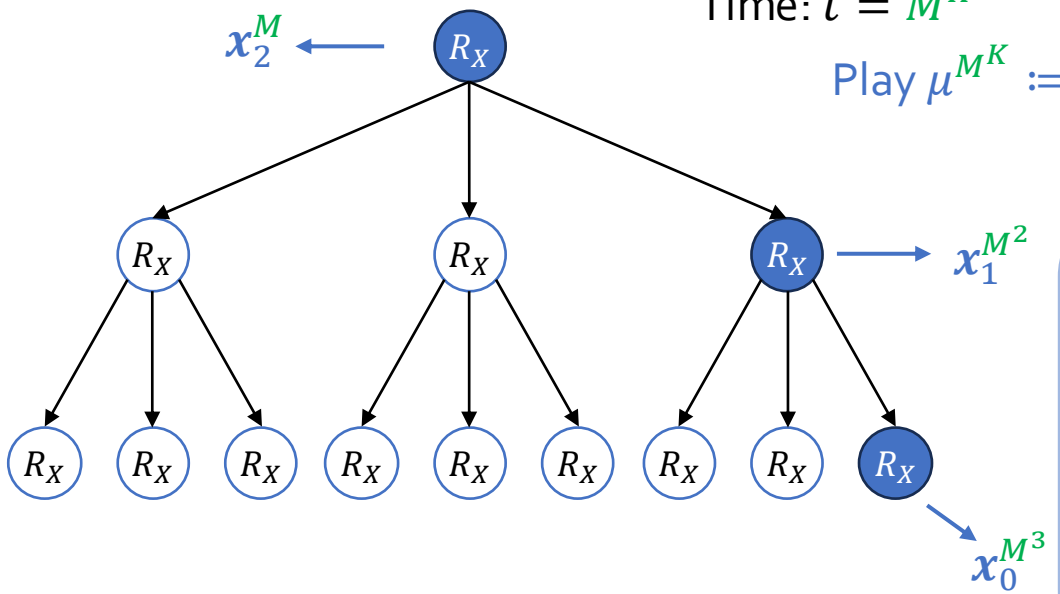
TreeSwap

Peng & Rubinstein (*STOC*'24); Dagan, Daskalakis, Fishelson, Golowich (*STOC*'24)

Given: External regret minimizer R_X on $X \subset [0,1]^n$ achieving ϵM regret after M steps (e.g., with gradient descent, $M = \text{poly}(d)/\epsilon^2$)

Goal: Build a **swap regret minimizer** on X

Idea:



Intuition: In the GGM framework, if $\mu^t = \text{Unif}\{x_1, \dots, x_K\}$ let ϕ^t be the "map" that takes $x_{K-1} \mapsto x_{K-2} \mapsto \cdots \mapsto x_0$

- μ^t is an expected fixed point of ϕ^t
- each value of ϕ^t is being picked by regret minimizer ⇒ Φ-regret is small!

TreeSwap

Peng & Rubinstein (*STOC*'24); Dagan, Daskalakis, Fishelson, Golowich (*STOC*'24)

Given: External regret minimizer R_X on $X \subset [0,1]^n$ achieving ϵM regret after M steps (e.g., with gradient descent, $M = \text{poly}(d)/\epsilon^2$)

Goal: Build a **swap regret minimizer** on X

Theorem: After M^K iterations with $K = 1/\epsilon$, (= $d^{\tilde{O}(1/\epsilon)}$ for all typical choices of R_X),

$$R_X^{\text{Swap}}(T) \leq T\left(\epsilon + \frac{1}{K}\right) \leq 2\epsilon T$$

from regret of each R_X

from expected fixed point error

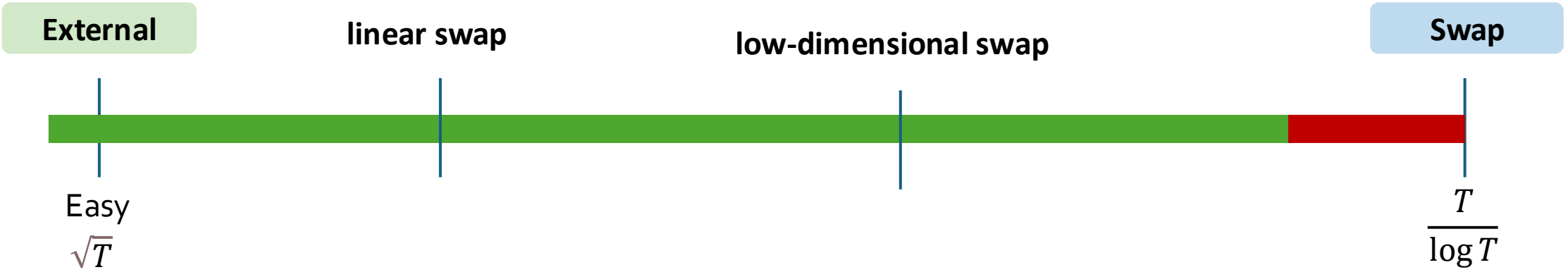
Nearly-matching lower bound:
This is basically tight! [DFGSZ, 2024]

Intuition: In the GGM framework, if $\mu^t = \text{Unif}\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_K\}$ let ϕ^t be the "map" that takes $\boldsymbol{x}_{K-1} \mapsto \boldsymbol{x}_{K-2} \mapsto \cdots \mapsto \boldsymbol{x}_0$

- $\boldsymbol{\mu}^t$ is an expected fixed point of ϕ^t
- each value of ϕ^t is being picked by regret minimizer $\Rightarrow \Phi$-regret is small!

Equilibrium computation

So far: Online learning



Is the picture the same for equilibrium *computation*?

Ellipsoid Against Hope

Goal: find $\mu \in \Delta(X)$ where $X = X_1 \times \cdots \times X_n$ s.t.

$$\mathop{\mathbb{E}}_{\boldsymbol{x}\sim\mu} \sum_i [u_i(\phi_i(\boldsymbol{x}_i), \boldsymbol{x}_{-i}) - u_i(\boldsymbol{x}_i, \boldsymbol{x}_{-i})] \leq \epsilon$$

for all $(\phi_1, \ldots, \phi_n) \in \Phi := \Phi_1 \times \cdots \times \Phi_n$

Originally proposed by Papadimitriou & Roughgarden [JACM'08], since significantly generalized by papers we will cite

Ellipsoid Against Hope

Goal: find $\mu \in \Delta(X)$ where $X = X_1 \times \cdots \times X_n$ s.t.

$$\mathop{\mathbb{E}}_{x \sim \mu} \sum_i [u_i(\boldsymbol{x}_{-i})^\top \mathbf{M}_i q_i(\boldsymbol{x}_i) - u_i(\boldsymbol{x}_i, \boldsymbol{x}_{-i})] \leq \epsilon$$

for all $(\mathbf{M}_1, \ldots \mathbf{M}_n) \in \Phi^q \coloneqq \Phi_1^{q_1} \times \cdots \times \Phi_n^{q_n}$

Ellipsoid Against Hope

Goal:

$$\text{find } \mu \in \Delta(X) \quad \text{s.t.} \quad \mathop{\mathbb{E}}_{\boldsymbol{x}\sim\mu} \langle \mathbf{M}, G(\boldsymbol{x}) \rangle \leq \epsilon \quad \text{for all } \mathbf{M} \in \Phi^q$$

Ellipsoid Against Hope (General form)

Primal:

$$\text{find } \mu \in \Delta(X) \quad \text{s.t.} \quad \mathop{\mathbb{E}}_{x\sim\mu} \langle \boldsymbol{y}, G(\boldsymbol{x}) \rangle \leq \epsilon \quad \text{for all } \boldsymbol{y} \in Y$$

Dual:

$$\text{find } \boldsymbol{y} \in Y \quad \text{s.t.} \quad \langle \boldsymbol{y}, G(\boldsymbol{x}) \rangle \geq \epsilon \quad \forall \boldsymbol{x} \in X$$

Dual is **infeasible** (because primal is feasible with $\epsilon = 0$)
Certificate of infeasibility of dual = primal solution
Idea: Run ellipsoid on the dual!

Recall: Semi-separation:
Given a transformation ϕ, either:

- find an expected fixed point of ϕ, or
- find $\boldsymbol{x} \in X$ with $\phi(\boldsymbol{x}) \notin X$

Ellipsoid Against Hope (General form)

Primal:

$$\text{find } \mu \in \Delta(X) \quad \text{s.t.} \quad \mathop{\mathbb{E}}_{x\sim\mu} \langle y, G(x)\rangle \le \epsilon \quad \text{for all } y \in Y$$

Dual:

$$\text{find } y \in Y \quad \text{s.t.} \quad \langle y, G(x)\rangle \ge \epsilon \quad \forall x \in X$$

Dual is **infeasible** (because primal is feasible with $\epsilon = 0$)
Certificate of infeasibility of dual = primal solution
Idea: Run ellipsoid on the dual!

is a separation oracle for

Semi-separation (**General form**):
Given y, either:

- find $\mu \in \Delta(X)$ with $\mathop{\mathbb{E}}_{x\sim\mu} \langle y, G(x)\rangle \le \epsilon$
- return a direction separating y from Y

Theorem [DFFPS STOC'25, ZATBFCS EC'25]:
$\log(1/\epsilon)$ semi-separation oracle
$\Rightarrow \log(1/\epsilon)$ computation of a primal solution

"Wait, the EFP oracle wasn't $\log(1/\epsilon)$-time!"

Ellipsoid Against Hope (General form)

Primal:

$$\text{find } \mu \in \Delta(X) \quad \text{s.t.} \quad \mathop{\mathbb{E}}_{x\sim\mu} \langle \boldsymbol{y}, G(\boldsymbol{x}) \rangle \leq \epsilon \quad \text{for all } \boldsymbol{y} \in Y$$

Dual:

$$\text{find } \boldsymbol{y} \in Y \quad \text{s.t.} \quad \langle \boldsymbol{y}, G(\boldsymbol{x}) \rangle \geq \epsilon \quad \forall \boldsymbol{x} \in X$$

Dual is **infeasible** (because primal is feasible with $\epsilon = 0$)
Certificate of infeasibility of dual = primal solution
Idea: Run ellipsoid on the dual!

is a separation oracle for

Semi-separation (**General form**):
Given $\boldsymbol{y}$, either:

- find $\mu \in \Delta(X)$ with $\mathop{\mathbb{E}}_{x\sim\mu} \langle \boldsymbol{y}, G(\boldsymbol{x}) \rangle \leq \epsilon$
- return a direction separating $\boldsymbol{y}$ from Y

Theorem [**FP** STOC'25, **ZA**TB**F**CS EC'25]:
$\log(1/\epsilon)$ semi-separation oracle
$\Rightarrow \log(1/\epsilon)$ computation of a primal solution

"Wait, the EFP oracle wasn't $\log(1/\epsilon)$-time!"

$\log(1/\epsilon)$-time Expected Fixed Points

Expected fixed point problem: given function $\phi : X \to X$,

find $\mu \in \Delta(X)$ s.t. $\left\| \mathbb{E}_{x \sim \mu} [\phi(x) - x] \right\|_2 \leq \epsilon$

$\log(1/\epsilon)$-time Expected Fixed Points

Expected fixed point problem: given function $\phi : X \to X$,

$$\text{find} \quad \mu \in \Delta(X) \quad \text{s.t.} \quad \mathop{\mathbb{E}}_{x\sim\mu} \langle y, \underbrace{\phi(x) - x}_{G(x)} \rangle \leq \epsilon \quad \forall y \in Y = B_1(0)$$

This problem can be solved with EAH, if we can semi-separate:

Given y, either:

- find $\mu \in \Delta(X)$ with $\mathop{\mathbb{E}}_{x\sim\mu} \langle y, \phi(x) - x \rangle \leq \epsilon$, or
- return a direction separating y from Y, or → easy: $Y = B_1(0)$
- return a direction separating ϕ from Φ

$\log(1/\epsilon)$-time Expected Fixed Points

Expected fixed point problem: given function $\phi : X \to X$,

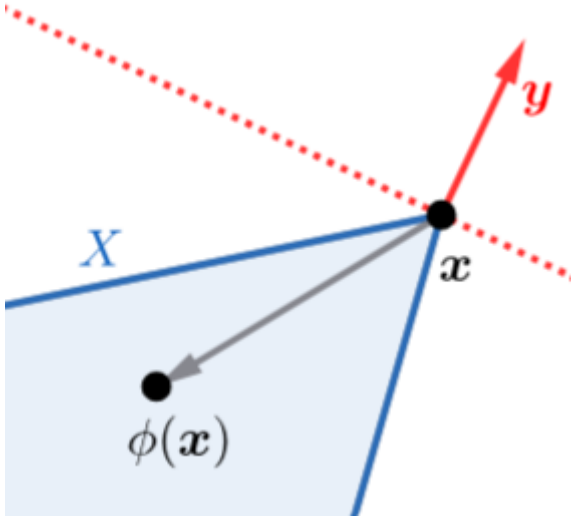
$$\text{find} \quad \mu \in \Delta(X) \quad \text{s.t.} \quad \mathop{\mathbb{E}}_{x\sim\mu} \langle y, \phi(x) - x \rangle \leq \epsilon \quad \forall y \in Y = B_1(0)$$

This problem can be solved with EAH, if we can semi-separate:

Given $y \in B_1(0)$, either:

- find $x \in X$ with $\langle y, \phi(x) - x \rangle \leq \epsilon$, or
- return a direction separating ϕ from Φ

Easy! Choose x to maximize $\langle y, x \rangle$



$\log(1/\epsilon)$-time Expected Fixed Points

Expected fixed point problem: given function $\phi : X \to X$,

$$\text{find} \quad \mu \in \Delta(X) \quad \text{s.t.} \quad \mathop{\mathbb{E}}_{\boldsymbol{x}\sim\mu} \langle \boldsymbol{y}, \phi(\boldsymbol{x}) - \boldsymbol{x} \rangle \leq \epsilon \quad \forall \boldsymbol{y} \in Y = B_1(0)$$

Caveat: Only works for **linear utilities!** Concave (nonlinear) games are at least as hard as fixed points of contractions [**A**D**F**GS**Z** preprint'26]

Theorem [**ZA**TB**F**CS EC'25]:
Semi-separation with EFPs can be done in $\mathrm{poly}(d, \log(1/\epsilon))$ time

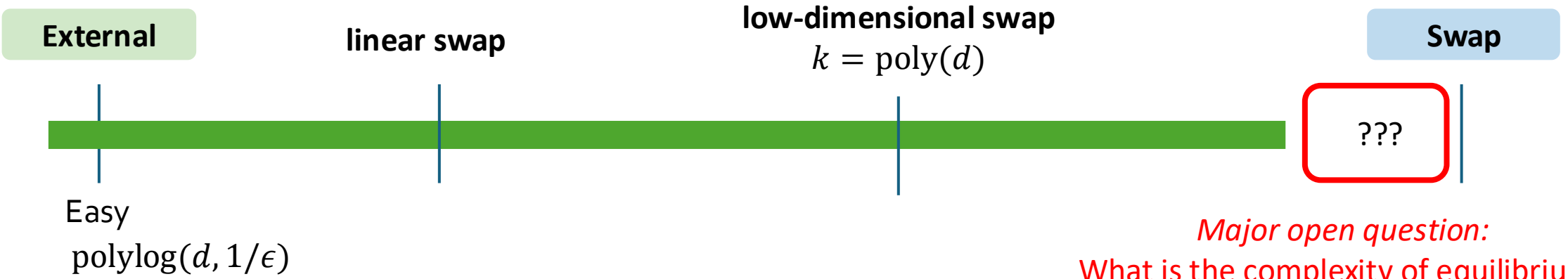
Theorem [**ZA**TB**F**CS EC'25]:
Φ^q-equilibria with $q_i : X_i \to \mathbb{R}^k$ can be computed in $\mathrm{poly}(d, k, \log(1/\epsilon))$ time

Φ-Equilibrium Computation

Is the picture the same for equilibrium *computation*?

Yes!

…as far as we know



Major open question:
What is the complexity of equilibrium computation with swap deviations ("normal-form correlated equilibria")?

The connection with **multicalibration**

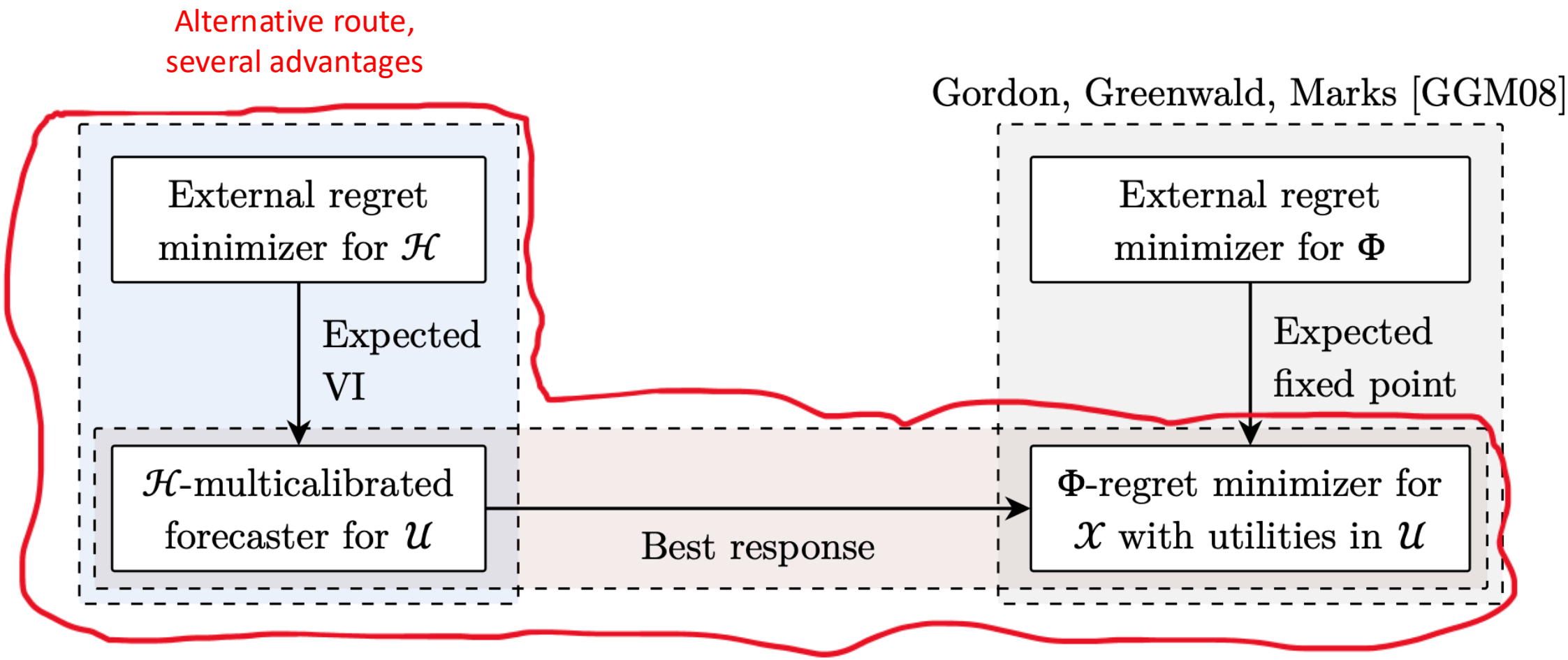
Multicalibration and Phi-Regret

- Gordon, Greenwald and Marks [GGM08] centers around the algorithmic primitive of **fixed points**
- There exists a different route to the same destination that passes through **forecasting** [Farina and Perdomo, 2026]

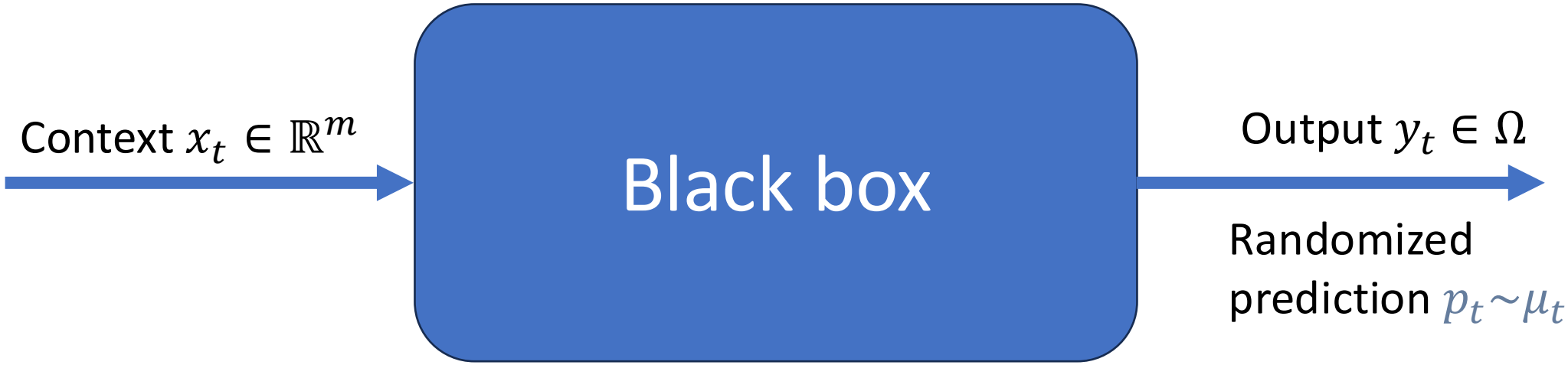
One can construct a Φ-regret minimizer starting from a powerful enough multicalibrated forecaster of the upcoming utility, and best respond to such forecast

In turn, the task of constructing such a forecaster admits a reduction to online learning in the style of Gordon-Greenwald-Marks, with **expected variational inequalities** replacing expected fixed points as the nonlinear optimization primitive

Multicalibration and Phi-Regret



Multicalibration



Natural regret-like quantity

$$\sup_{h \in H} \left| \sum_t \mathbb{E}_{p_t \sim \mu_t} h(x_t, p_t)^T (y_t - p_t) \right|$$

Set of **tests**

Adversary tries to tell apart real y_t from prediction p_t

From Multicalibration to Online Learning

At every step t:

1. Nature reveals context x_t
2. Regret minimizer picks the next test $h_t \in H$
3. Using h_t the learner produces μ_t **such that no matter y**

Learner solves an **Expected variational inequality** (EVI)

$$\mathbb{E}_{p_t \sim \mu_t}[h_t(x_t, p_t)^T (y_t - p_t)] \leq 0$$

4. Nature reveals the true outcome y_t, regret minimizer incurs linear loss

$$f_t(h) := -\mathbb{E}_{p_t \sim \mu_t}[h(x_t, p_t)^T (y_t - p_t)]$$

Proof Sketch

- For any test $h \in H$, we want to bound

$$\text{Gap} := \sum_t \mathbb{E}_{p_t \sim \mu_t}[h(x_t, p_t)^T (y_t - p_t)]$$

- Observe:

$$\text{Gap} \le \sum_t \mathbb{E}_{p_t \sim \mu_t}[h(x_t, p_t)^T (y_t - p_t)] - \mathbb{E}_{p_t \sim \mu_t}[h_t(x_t, p_t)^T (y_t - p_t)]$$

≤ 0 by **EVI**

$$= \sum_t f_t(h) - f_t(h_t) = \text{Regret over } H$$

Special case I: Multiplicative Weights $\Longrightarrow$ Regret $\le \sqrt{T \log |H|}$

Special case II: OGD on RKHS ball $\Longrightarrow$ Regret $\le \sqrt{T}\, \text{poly}(d, r)$

Proof Sketch

- For any test $h \in H$, we want to bound

$$\text{Gap} := \sum_t \mathbb{E}_{p_t \sim \mu_t}[h(x_t, p_t)^T (y_t - p_t)]$$

- Observe:

$$\text{Gap} \leq \sum_t \mathbb{E}_{p_t \sim \mu_t}[h(x_t, p_t)^T (y_t - p_t)] - \mathbb{E}_{p_t \sim \mu_t}[h_t(x_t, p_t)^T (y_t - p_t)]$$

≤ 0 by **EVI**

$$= \sum_t f_t(h) - f_t(h_t) = \text{Regret over } H$$

Fact: EVIs can always be solved efficiently provided oracle access to the domain Ω and to evaluation of h_t. In fact, we can ϵ-solve them in $O(\text{poly}(\text{d})\log 1/\epsilon)$ time!

Proof Sketch

- For any test $h \in H$, we want to bound

$$\text{Gap} := \sum_t \mathbb{E}_{p_t \sim \mu_t}[h(x_t, p_t)^T (y_t - p_t)]$$

- Observe:

$$\text{Gap} \leq \sum_t \mathbb{E}_{p_t \sim \mu_t}[h(x_t, p_t)^T (y_t - p_t)] - \mathbb{E}_{p_t \sim \mu_t}[h_t(x_t, p_t)^T (y_t - p_t)]$$

≤ 0 by **EVI**

$$= \sum_t f_t(h) - f_t(h_t) = \text{Regret over } H$$

$\Longrightarrow$ **Theorem:** As long as we know how to do regret minimization on H and optimize over Ω, we can efficiently achieve outcome indistinguishability!

From Phi-Regret to Multicalibration

- Forecaster forecasts the next utility and best-responds to it
  - Let $\sigma(p_t) \in \arg\max p_t^T x$ be a best-response to the prediction
  - x_t is the pushforward of the prediction distribution μ_t under σ
- Main observation:

$$\sum_t \mathbb{E}_{x_t}[u_t^T(\phi(x_t) - x_t)] = \sum_t \mathbb{E}_{p_t \sim \mu_t}[u_t^T(\phi(\sigma(p_t)) - \sigma(p_t))]$$

$$\leq \sum_t \mathbb{E}_{p_t \sim \mu_t}[(u_t - p_t)^T(\phi(\sigma(p_t)) - \sigma(p_t))]$$

Multicalibration tests $h(p_t)$

From Phi-Regret to Multicalibration

$\Longrightarrow$ **Theorem:** Best responding to a forecaster that is multicalibrated with respect to all tests of the form

$$p_t \mapsto \phi\big(\sigma(p_t)\big) - \sigma(p_t) \text{ for } \phi \in \Phi$$

guarantees Φ-regret minimization

Can think of this as a generalization of Foster and Vohra [1997]

$\Longrightarrow$ Whenever ϕ spans a subset of a low-dimensional ball of functions, then so does the set of multicalibration tests we care about

Example: Linear Swap Regret via Multicalibration

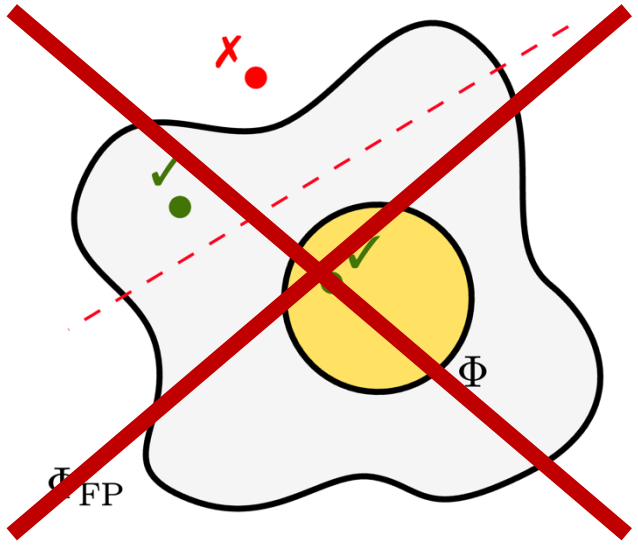
- Consider again the task of constructing a no-linear-swap-regret algorithm

Need a forecaster multicalibrated against tests
$p \mapsto A\sigma(p) - \sigma(p)$
where A spans the set of all linear endomorphism

Idea: A is bounded in norm. Might as well construct a forecaster multicalibrated wrt
$p \mapsto B\sigma(p)$
where B spans an appropriately sized **matrix ball**

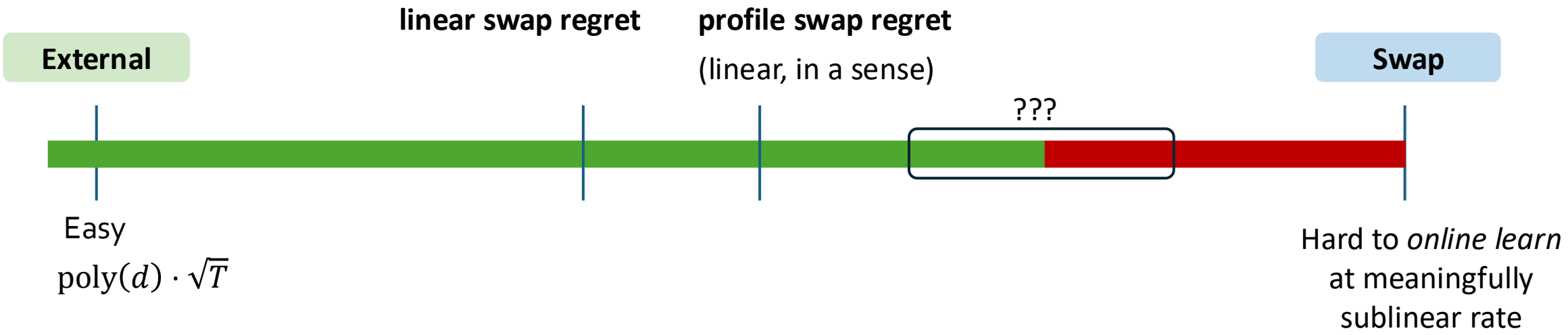
We can do so by doing online learning over a ball (e.g., gradient descent)!

No need for semi-separation machinery



Conclusions

- Important connections with different areas (dynamics, games, complexity, online learning, economics, forecasting)
- We tried to systematize the main trends we observe and paint open questions / areas of opportunity
- Many open questions remain. What's the strongest notion of rationality that can be guaranteed efficiently?

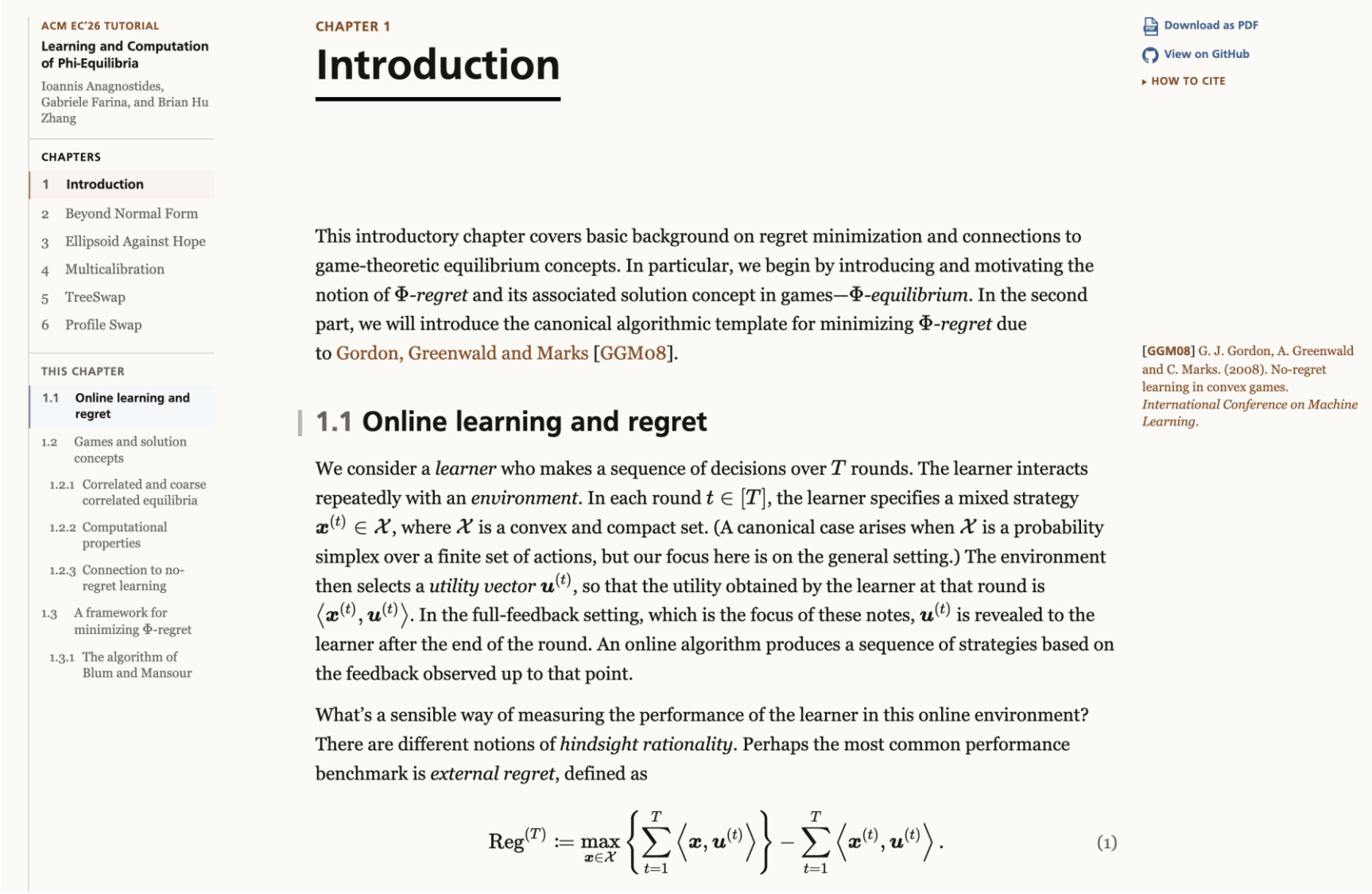


Thank you!

https://ec26-phi-regret-tutorial.github.io/

Any questions?

- Feel free to report issues or open pull requests on GitHub!



ACM EC'26 TUTORIAL

Learning and Computation of Phi-Equilibria

Ioannis Anagnostides, Gabriele Farina, and Brian Hu Zhang

CHAPTERS

1 Introduction
2 Beyond Normal Form
3 Ellipsoid Against Hope
4 Multicalibration
5 TreeSwap
6 Profile Swap

THIS CHAPTER

1.1 Online learning and regret
1.2 Games and solution concepts
1.2.1 Correlated and coarse correlated equilibria
1.2.2 Computational properties
1.2.3 Connection to no-regret learning
1.3 A framework for minimizing Φ-regret
1.3.1 The algorithm of Blum and Mansour

CHAPTER 1

Introduction

Download as PDF
View on GitHub
HOW TO CITE

This introductory chapter covers basic background on regret minimization and connections to game-theoretic equilibrium concepts. In particular, we begin by introducing and motivating the notion of Φ-*regret* and its associated solution concept in games—Φ-*equilibrium*. In the second part, we will introduce the canonical algorithmic template for minimizing Φ-*regret* due to Gordon, Greenwald and Marks [GGM08].

[GGM08] G. J. Gordon, A. Greenwald and C. Marks. (2008). No-regret learning in convex games. *International Conference on Machine Learning*.

1.1 Online learning and regret

We consider a *learner* who makes a sequence of decisions over T rounds. The learner interacts repeatedly with an *environment*. In each round $t \in [T]$, the learner specifies a mixed strategy $\boldsymbol{x}^{(t)} \in \mathcal{X}$, where $\mathcal{X}$ is a convex and compact set. (A canonical case arises when $\mathcal{X}$ is a probability simplex over a finite set of actions, but our focus here is on the general setting.) The environment then selects a *utility vector* $\boldsymbol{u}^{(t)}$, so that the utility obtained by the learner at that round is $\left\langle \boldsymbol{x}^{(t)}, \boldsymbol{u}^{(t)} \right\rangle$. In the full-feedback setting, which is the focus of these notes, $\boldsymbol{u}^{(t)}$ is revealed to the learner after the end of the round. An online algorithm produces a sequence of strategies based on the feedback observed up to that point.

What's a sensible way of measuring the performance of the learner in this online environment? There are different notions of *hindsight rationality*. Perhaps the most common performance benchmark is *external regret*, defined as

$$\mathrm{Reg}^{(T)} := \max_{\boldsymbol{x} \in \mathcal{X}} \left\{ \sum_{t=1}^{T} \left\langle \boldsymbol{x}, \boldsymbol{u}^{(t)} \right\rangle \right\} - \sum_{t=1}^{T} \left\langle \boldsymbol{x}^{(t)}, \boldsymbol{u}^{(t)} \right\rangle. \tag{1}$$