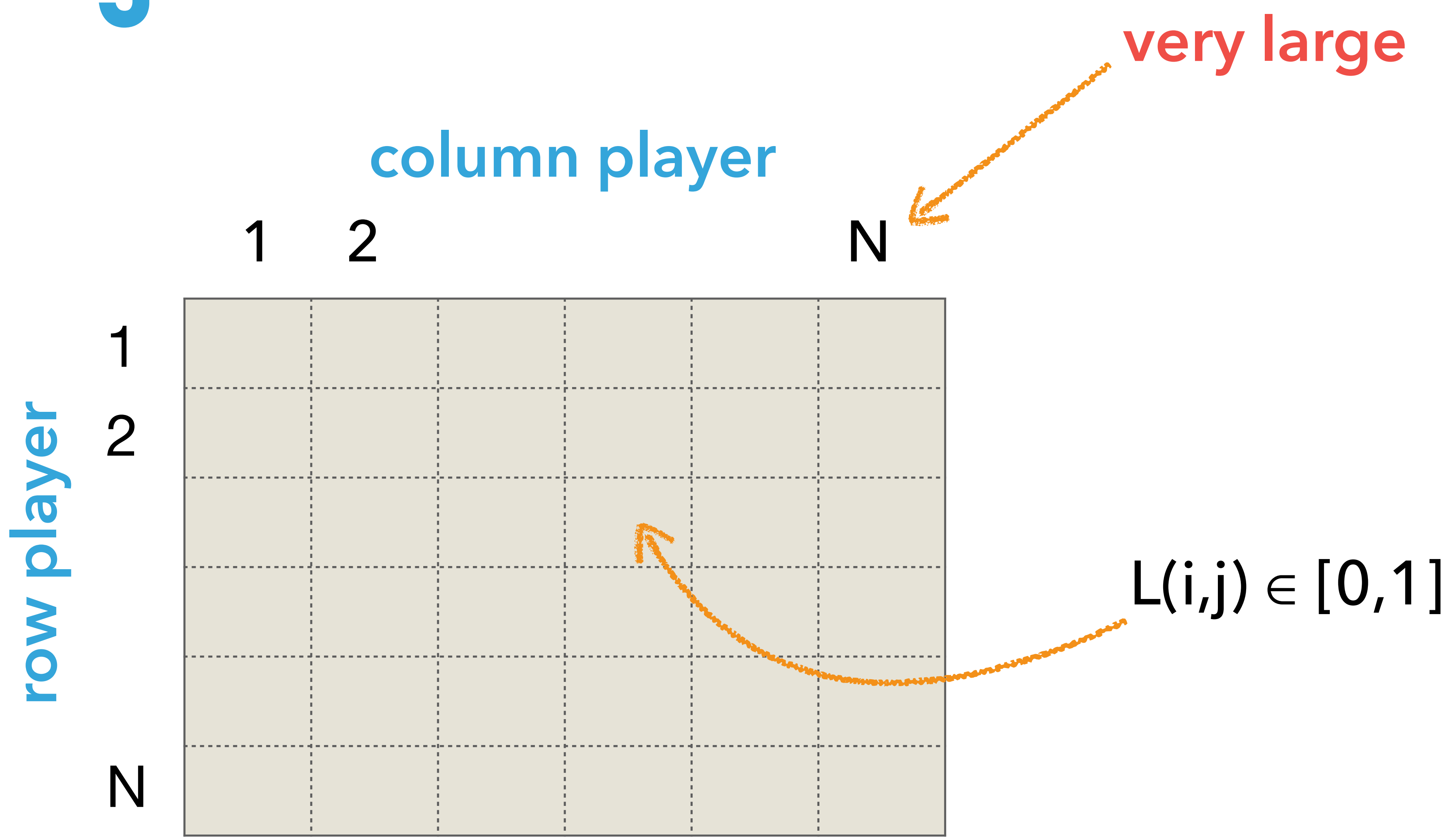
Learning in Games with Best-Response Oracles

Tomer Koren (Google Research)
joint work with
Elad Hazan (Princeton)

Zero-sum games



TIME TO COMPUTE / APPROX?

von Neumann equilibrium:

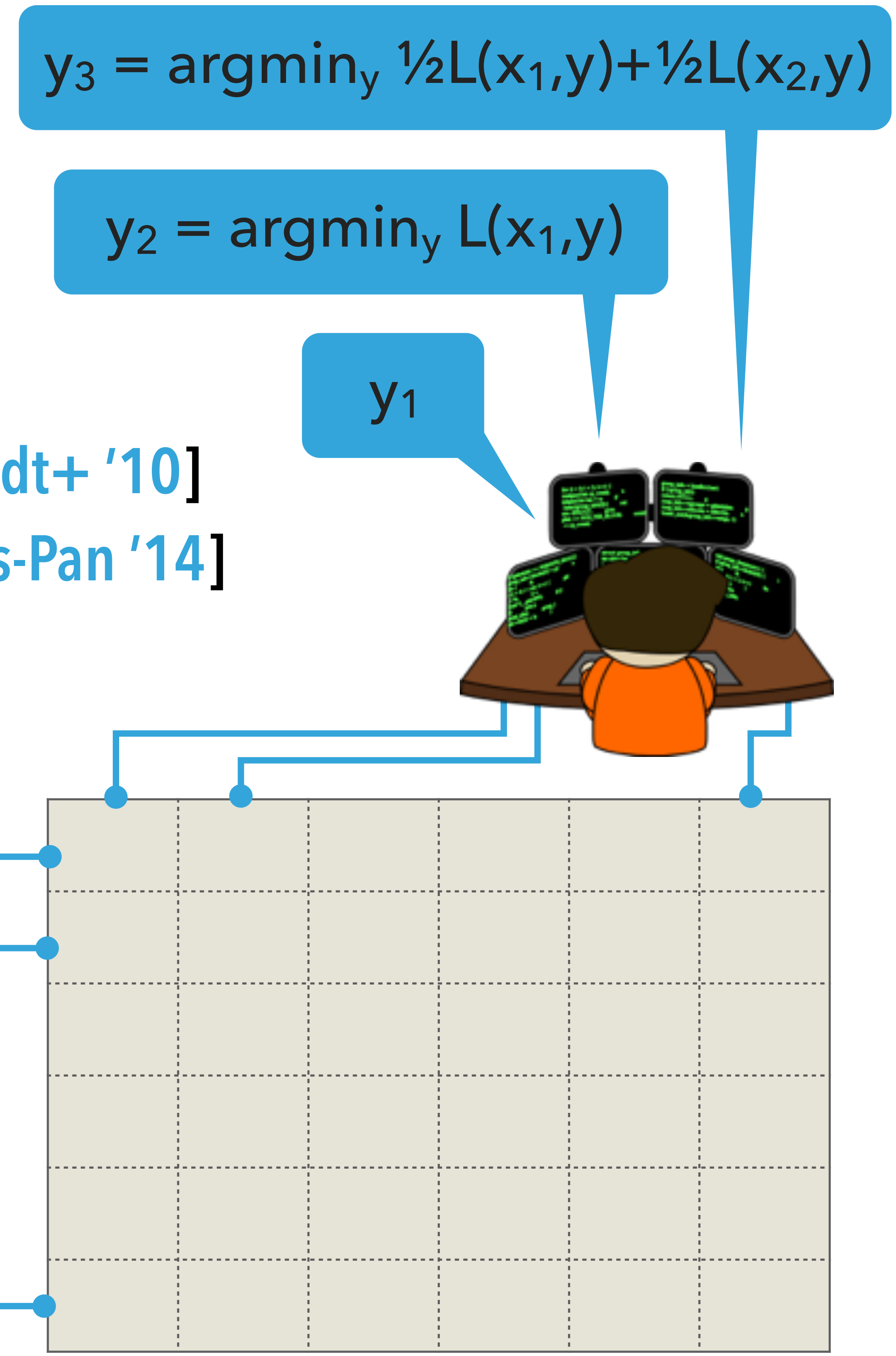
$$\lambda = \min_{p \in \Delta} \max_{q \in \Delta} \mathop{\mathbb{E}}_{\substack{i \sim p \\ j \sim q}} [L(i,j)]$$

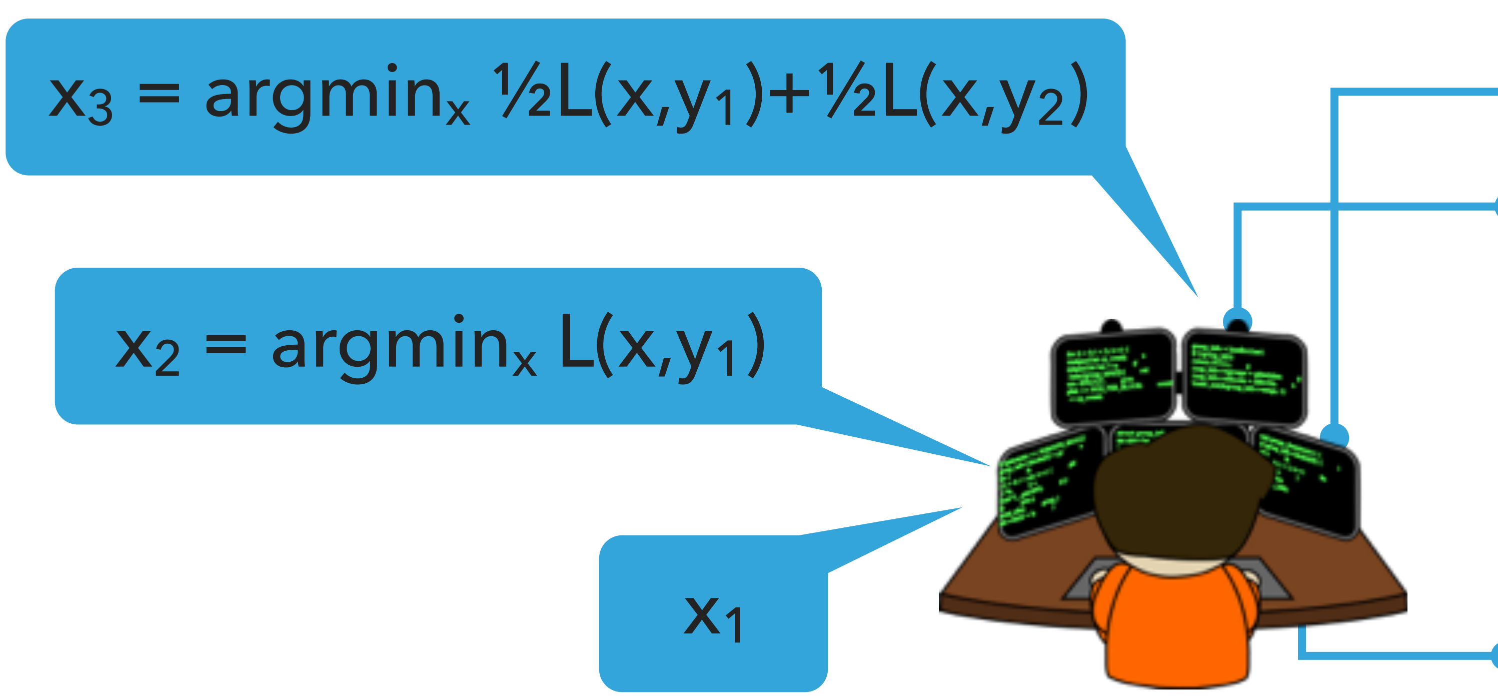
EFFICIENT STRATEGIES FOR PLAYERS ?

Fictitious Play [Brown '49]

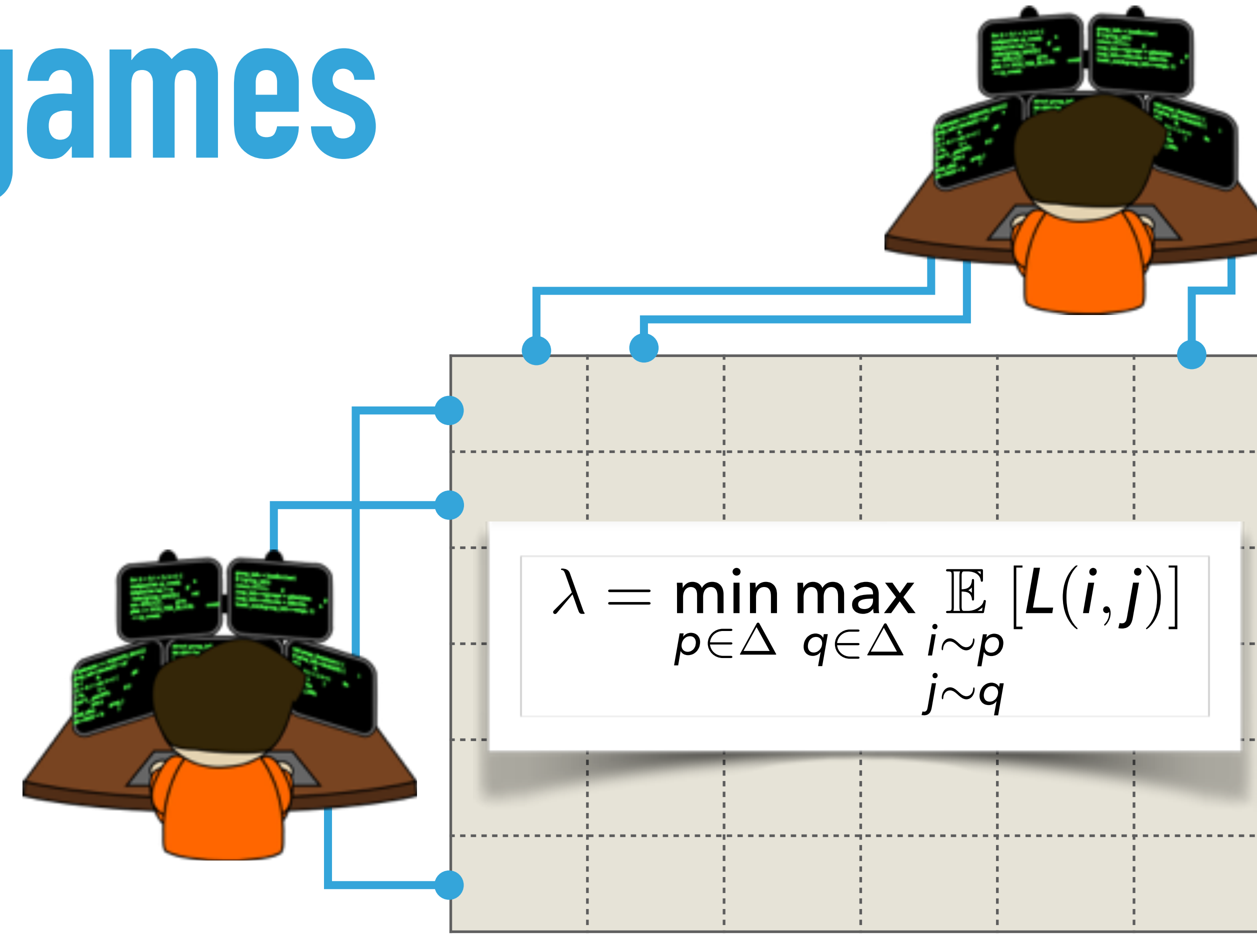
players plays their **best-response** to empirical dist. of opponent's past plays

- FP converge to minimax [Robinson '51]
- but, might need $\Omega(2^N)$ iterations [Brandt+ '10]
- convergence rate is $\Omega(T^{-1/N})$ [Daskalakis-Pan '14] (refutes Karlin's conjecture [Karlin '59])





Solving zero-sum games



- Poly-time since the 70's... (equivalent to LP)
- state of the art: Õ(N) time algorithm, tight [Grigoriadis-Khachiyan '95]
- Õ(N) time via regret minimization [Freund-Schapire '99]

SUBLINEAR IN INPUT SIZE

- Õ(N) time for generalized minimax problems [Clarkson+ '10]
- More recent results: poly(N) / T convergence rates [Daskalakis+ '11, Rakhlin-Sridharan '13]

this talk: focus on N

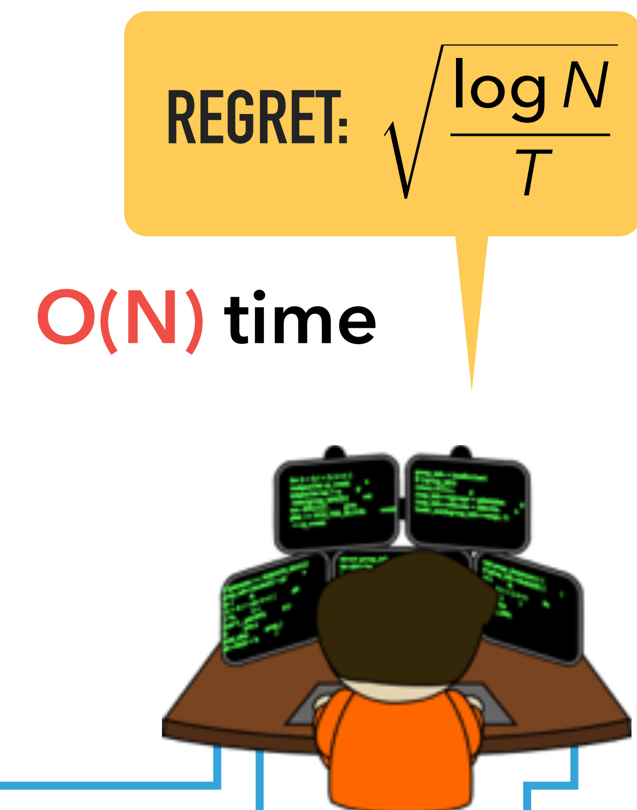
Learning in zero-sum games

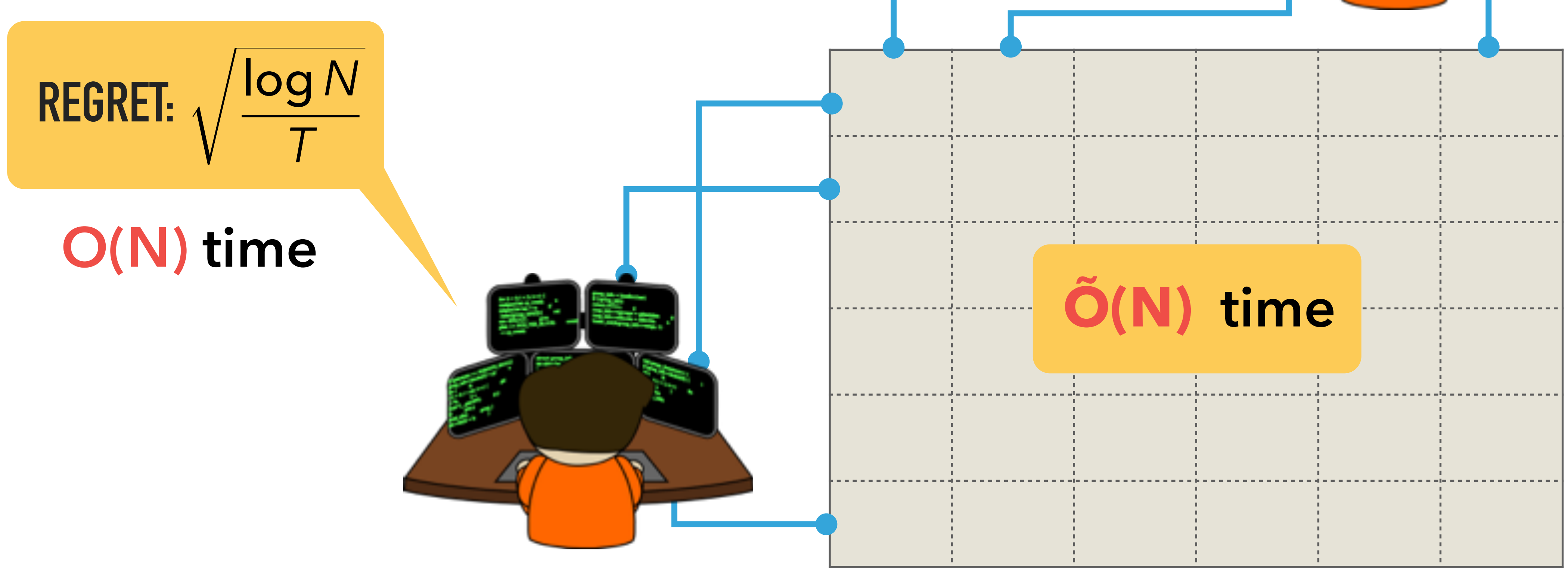
[Freund–Schapire '99]

Players use online learning algos
(e.g., Multiplicative Weights)

- $\log(N) / \varepsilon^2$ iterations for regret $< \varepsilon$
- $O(N)$ time per iteration

→ $O(N / \varepsilon^2)$ time for ε-approximation





Can we do better?

Games are often exponentially large

- X = { all (s,t)-paths in a given graph }
  Y = { costs on edges }

- X = { all permutations over [n] }
  Y = { value assignments to items }

- X = { subsets of [n] }
  Y = { submodular evaluation functions }

But best-response / optimization is **poly-time** = poly(log N)

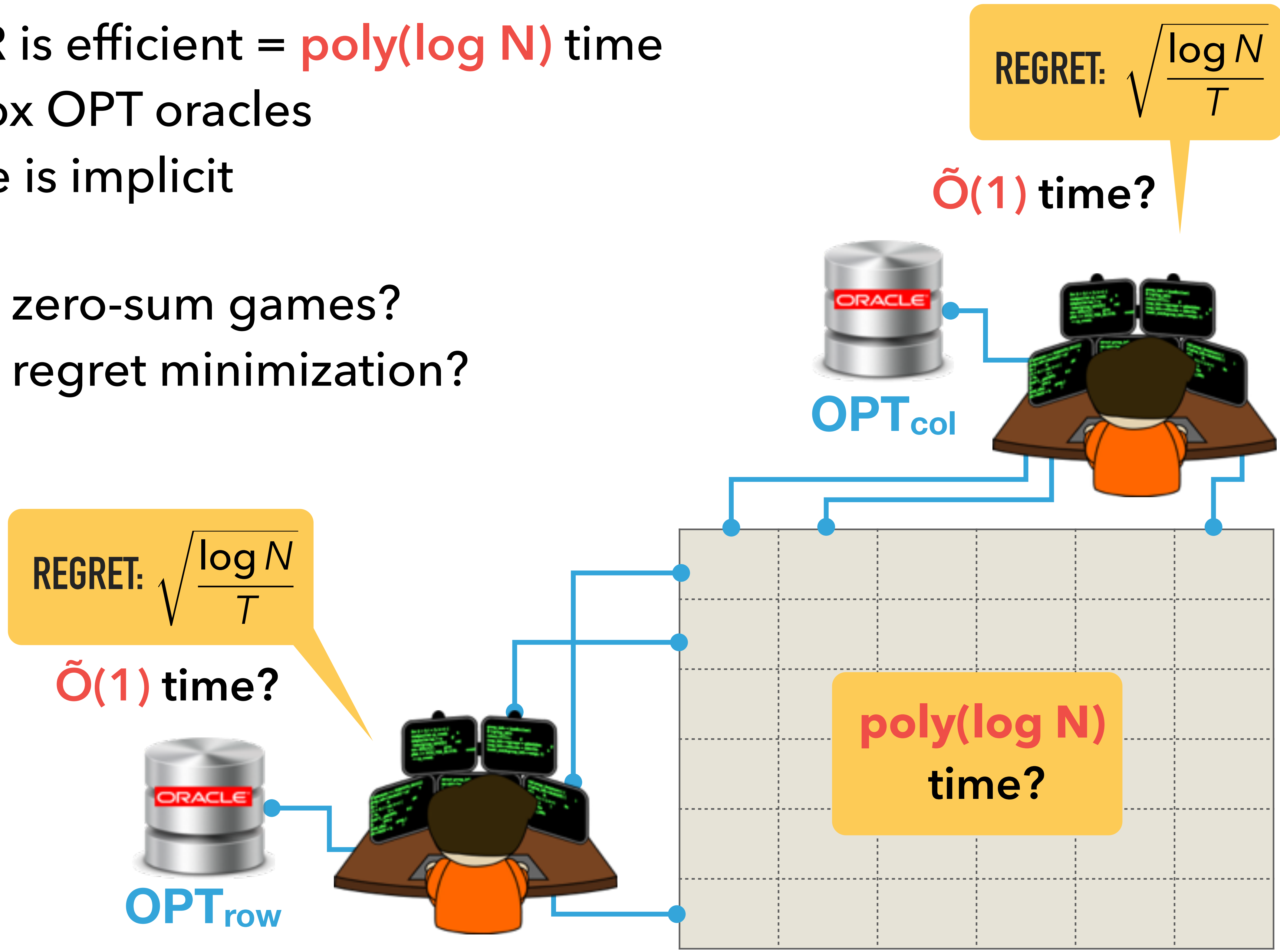
Best response oracles

assume BR is efficient = poly(log N) time

- black-box OPT oracles
- structure is implicit

→ efficient zero-sum games?

→ efficient regret minimization?



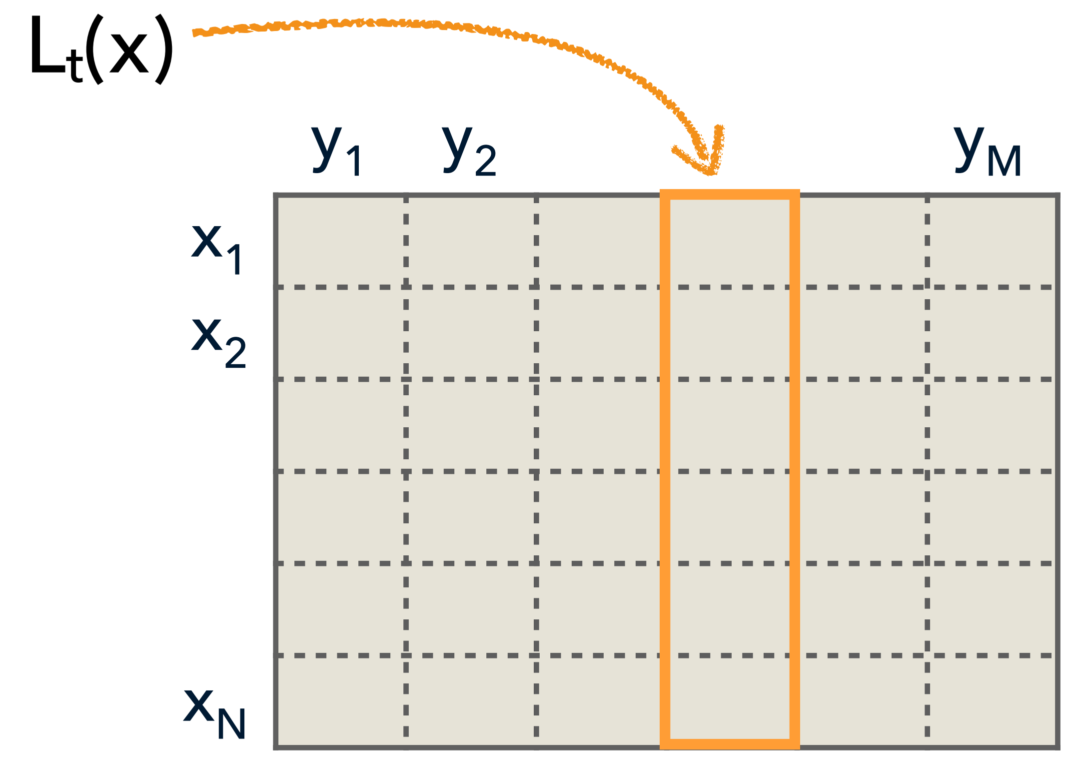
Online Learning

Iteratively, for t=1,2,…,T:

(1) player: $x_t \in X$ ("expert")

(2) adversary: $y_t \in Y$

(3) player's loss = $L(x_t,y_t) = L_t(x_t)$



- Goal: minimize **regret**: (average, expected)

$$\frac{1}{T}\sum_{t=1}^{T} L_t(x_t) - \frac{1}{T}\min_{x^*}\sum_{t=1}^{T} L_t(x^*) \to 0$$

- **Value access** to matrix: VAL(x,y) = L(x,y)

Õ(1) time

- **Best Response oracle**:

$$\mathrm{OPT}(S \subseteq Y) = \operatorname{argmin}_{x} \sum_{y \in S} L(x,y)$$

Learn efficiently? Regret < ¼ in poly(log N) time?

Learning-theoretic motivation

- **Fundamental question in learning theory:**
  generic & efficient reduction of online learning to optimization?
  (analogous to fundamental theorem of statistical learning)

- **Many specialized online algorithms for optimizable settings:**
  submodular opt., network routing, online PCA, contextual bandits, online ranking,…

- **Practical – numerous previous attempts:**
  Online convex optimization [Zinkevich '03, Hazan+ '06],
  Follow the Perturbed Leader (FPL) [Hannan '57, Kalai-Vempala '06],
  Dropout perturbation [vanErven-Kotlowski-Warmuth '14],
  Contextual bandits [Agarwal+ '14], …
  ☛ **typically poly(log N) computation, but need explicit structure**

Results

In **OPT oracle** model:

- **Thm 1.** Any algo that approximates $N \times N$ zero-sum games to within $\varepsilon = 1/4$ runs in total time $\tilde{\Omega}(\sqrt{N})$
  ☛ $\tilde{\Omega}(N)$ time needed to minimize regret

- **Thm 2.** There exists (new) online learning algo that attains regret $< \varepsilon$ in total time $\tilde{O}(\sqrt{N} / \varepsilon^2)$, **tight**
  ☛ vs. $\Theta(N/\varepsilon^2)$ time w/o OPT oracle

- **Corr.** There exists (new) algo that approximates $N \times N$ zero-sum games in total time $\tilde{O}(\sqrt{N})$, **tight**
  ☛ vs. $\Theta(N)$ time without oracles

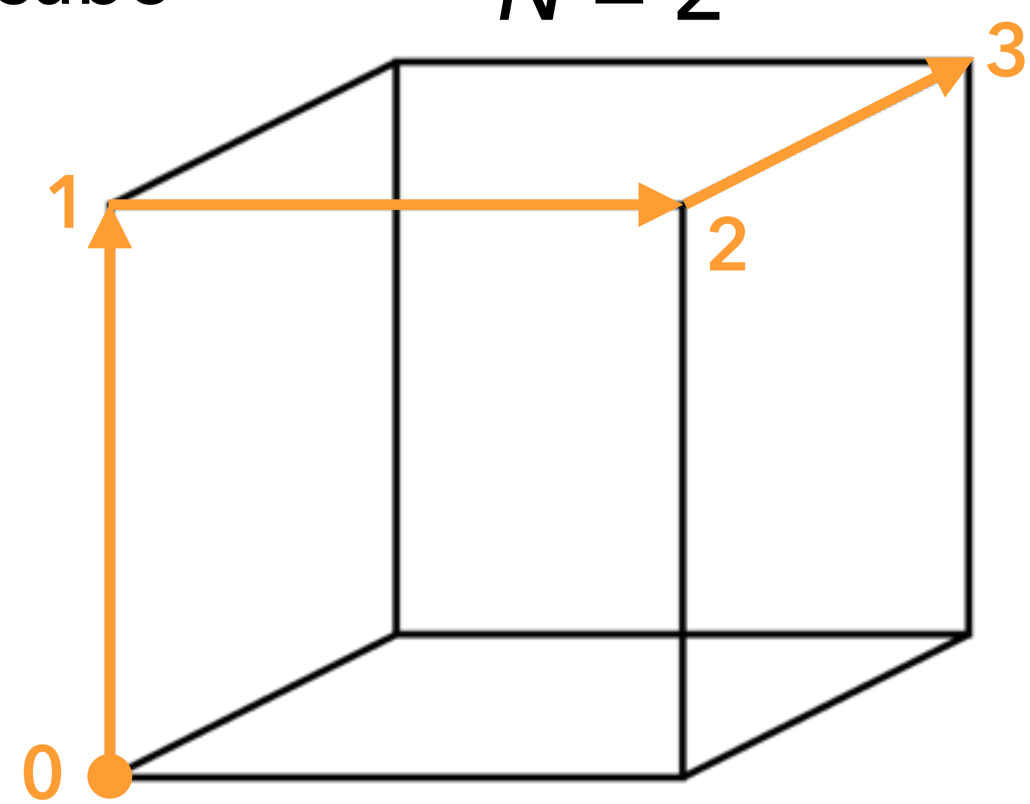
4TH ROOT OF INPUT SIZE

Aldous' random walk function

[Aldous '83, Aaronson '06]

Dist. over functions that have a single local min.
(1) start RW from uniform vertex of d-dim cube
(2) $f(i)$ = time to hit vertex i

$V = \{0,1\}^d$
$N = 2^d$



- Any such f has a single local min?
- #queries to find minimum?

$$O(\sqrt{N}) = O(2^{d/2})$$

LOCAL SEARCH…

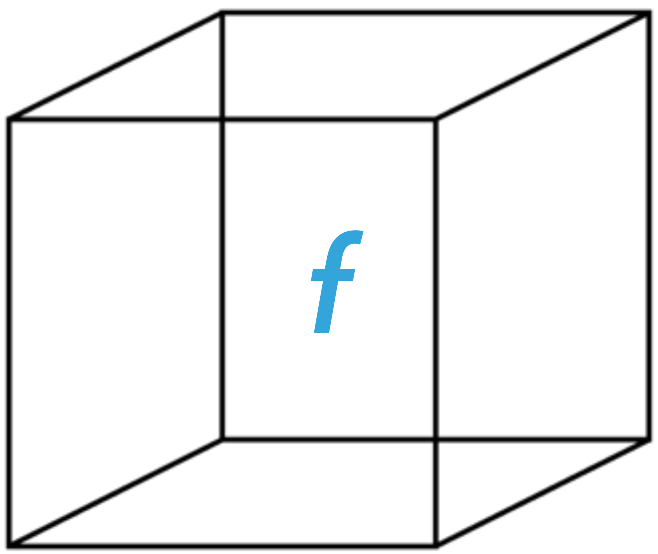
Thm [Aldous, Aaronson]: this is tight

Any algo that tells whether argmin vertex is odd/even w.p. > 2/3 would need $\tilde{\Omega}(\sqrt{N})$ queries to f

Local search → Learning in games

Reduction: oracle access to f ➔ VAL + OPT oracles for game

argmin f is odd? even?



$N = 2^d$

1	1	1	1	1	1
0	1	1	1	1	1
0	0	1	1	1	1
0	0	0	1	1	1
0	0	0	0	1	1
0	0	0	0	0	¼? ¾?

$N = 2^d$

game equilibrium

$$L(i,j) = \begin{cases} 0 & \text{if } f(i) < f(j) \\ 1 & \text{otherwise} \end{cases}$$

Local search → Learning in games

Reduction: oracle access to *f* ➔ VAL + OPT oracles for game

VAL oracle?

two queries to compute L(i,j)

OPT oracle?

$$\mathrm{OPT}(S) = \arg\min_i \left\{ \sum_{j \in S} L(i,j) \right\}$$

- find i s.t.: f(i) < min(f(S))
- **local search:**

  ≤ log(N) queries

  per each j ∈ S

N = 2^d

1	1	1	1	1	1
0	1	1	1	1	1
0	0	1	1	1	1
0	0	0	1	1	1
0	0	0	0	1	1
0	0	0	0	0	¼? ¾?

N = 2^d

$$L(i,j) = \begin{cases} 0 & \text{if } f(i) < f(j) \\ 1 & \text{otherwise} \end{cases}$$

game equilibrium

Results

In **OPT oracle** model:

- **Thm 1.** Any algo that approximates $N \times N$ zero-sum games to within $\varepsilon = 1/4$ runs in total time $\tilde{\Omega}(\sqrt{N})$
  ☛ $\tilde{\Omega}(N)$ time needed to minimize regret

- **Thm 2.** There exists (new) online learning algo that attains regret $< \varepsilon$ in total time $\tilde{O}(\sqrt{N} / \varepsilon^2)$, **tight**
  ☛ vs. $\Theta(N/\varepsilon^2)$ time w/o OPT oracle

- **Corr.** There exists (new) algo that approximates $N \times N$ zero-sum games in total time $\tilde{O}(\sqrt{N})$, **tight**
  ☛ vs. $\Theta(N)$ time without oracles

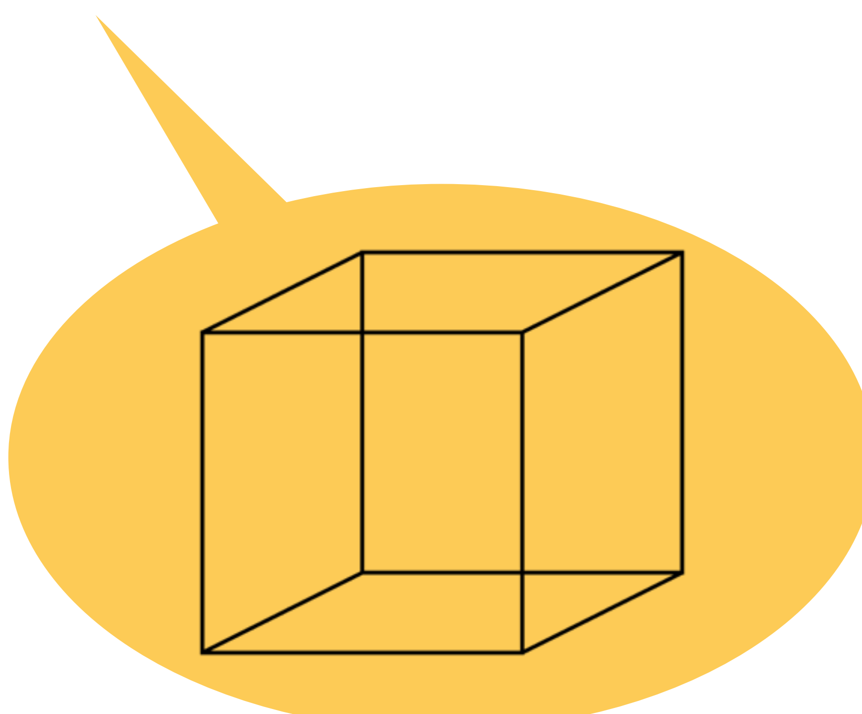
Intuition

Idea: reduce effective #experts **from N to √N**

Interpolate two extreme cases:

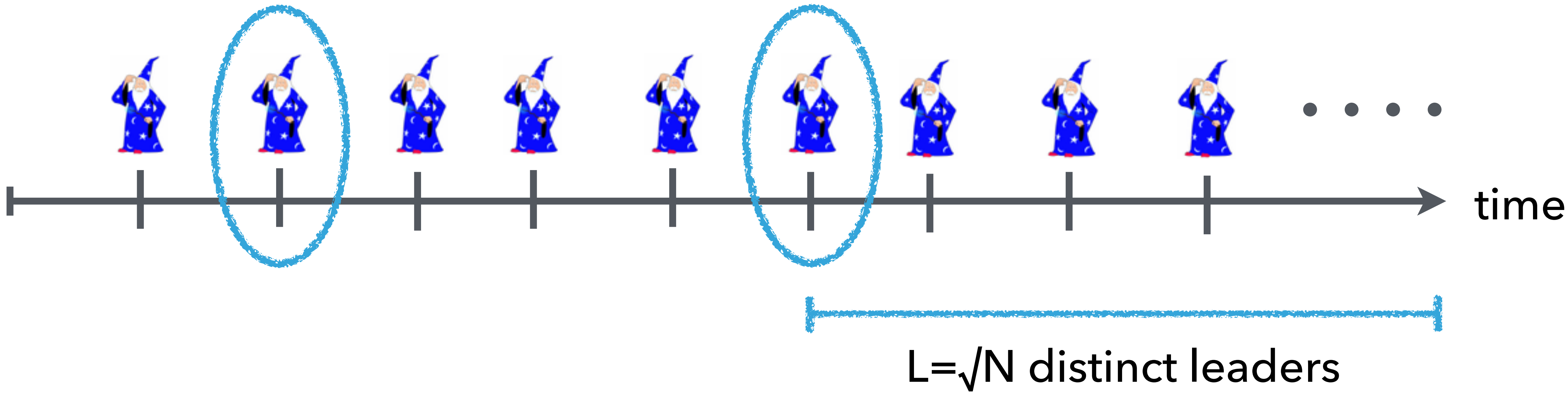
1. **There are few leaders:**
   ➔ OPT oracle is useful
2. **Leader keeps changing:**
   ➔ sampling ~√N experts will get us into √N "finalists"

$$x_t^* = \mathrm{OPT}(y_1, \ldots, y_{t-1})$$



Stream of leaders

- Sort leaders by "death time" = last time ever to appear as leader



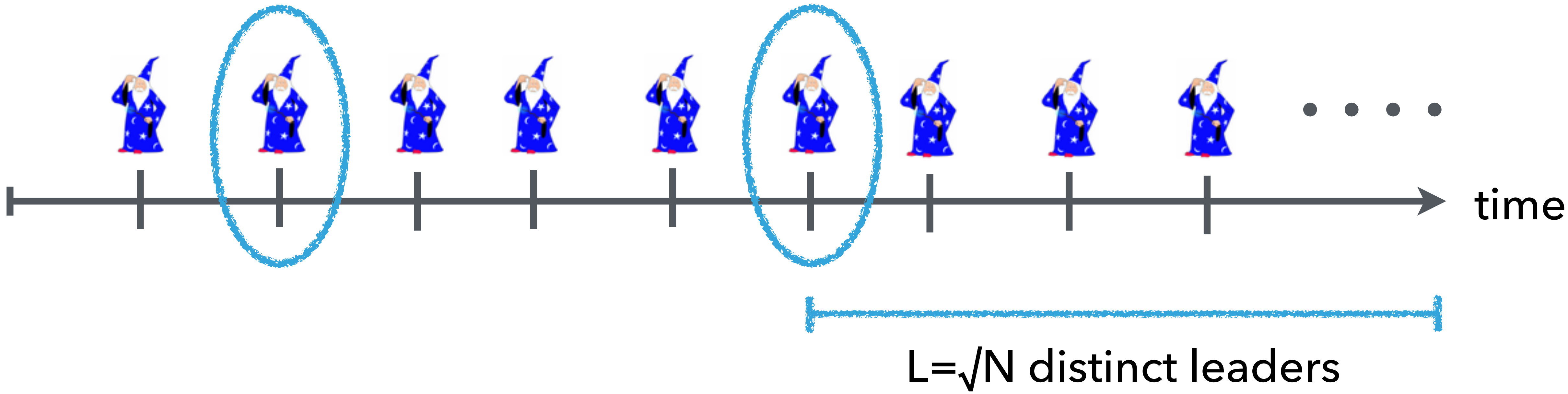
- **Sampling √N experts:**
  1. w.h.p. gets us to last √N leaders
  2. Exp3 regret vs √N sampled experts $\leq \sqrt{\frac{\sqrt{N}}{T}}$

- "Only" need to get low-regret in last time interval: $\sqrt{\frac{L}{T}}$

  # LEADERS

Stream of leaders

- Sort leaders by “death time” = last time ever to appear as leader



Combine two algorithms:

(1) Bandit algo over random sample of √N experts

(2) “Leaders” algorithm

- **Leaders:** for any sequence with **at most L distinct** leaders: $\frac{1}{T}\sum_{t=1}^{T} L_t(x_t) - \frac{1}{T}\sum_{t=1}^{T} L_t(x^*) \lesssim \sqrt{\frac{L}{T}}$

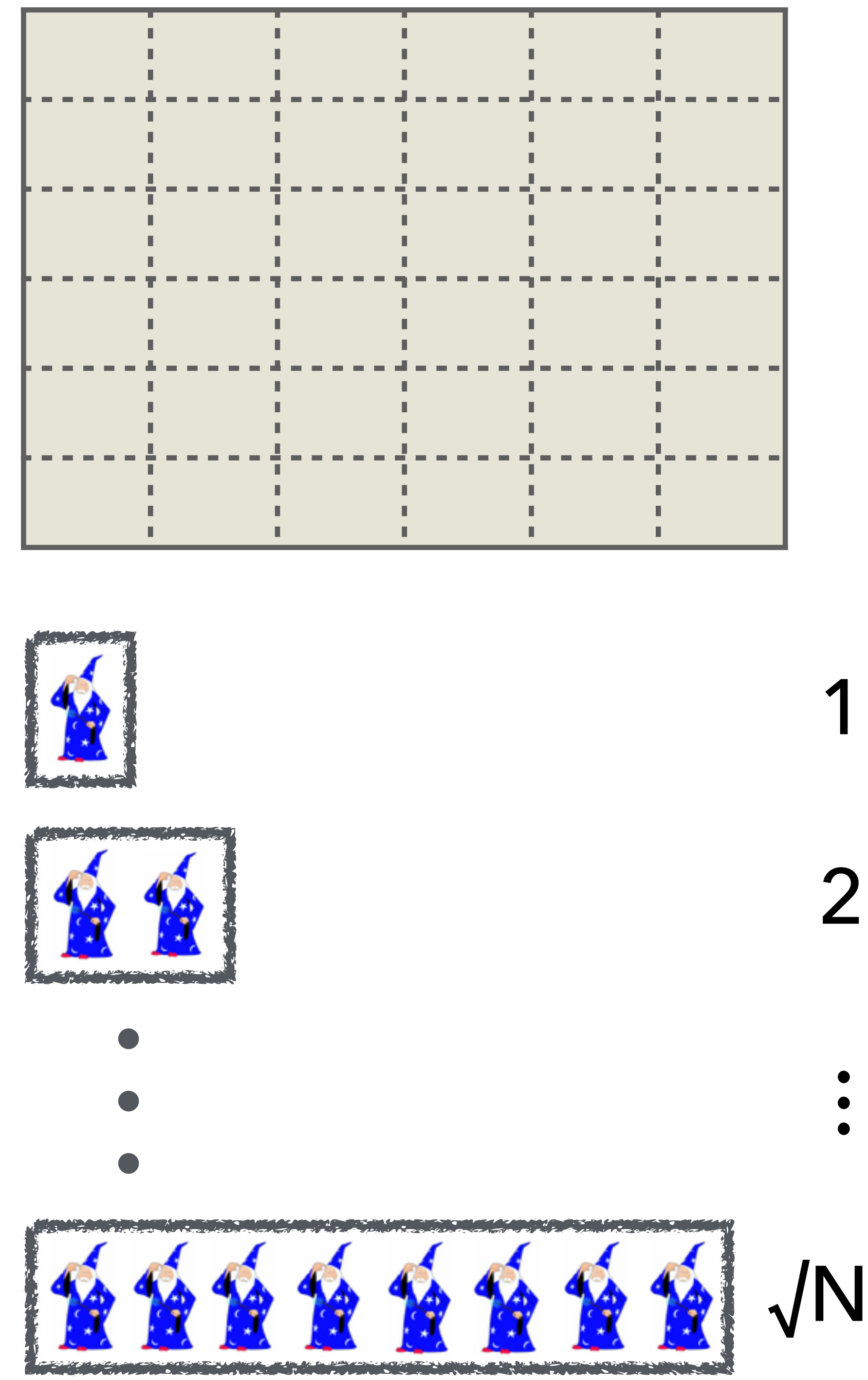
  ☛ needs **Õ(1) time per round**

Final algorithm

bandit on
$\sqrt{N} + \log(N)$
"meta-arms"

$|S| = \sqrt{N}$
random

log(N)
"sliding
windows"



Thm. for any sequence $y_1, \ldots, y_T$,
w.p. $\geq 1-\delta$:

$$\frac{1}{T}\sum_{t=1}^{T} L_t(x_t) - \frac{1}{T}\sum_{t=1}^{T} L_t(x^*) \lesssim \sqrt{\frac{\sqrt{N}}{T}\log\frac{1}{\delta}}$$

Bottom line

- efficient OPT $\not\Rightarrow$ efficient online learning
- but it helps, quadratically
- intriguing connections to local search

Many questions:

- stronger positive results? what assumptions? (e.g., [Daskalakis-Syrgkanis '16])
- what about oracle complexity? (lower b.)
- approximate optimization? (upper b.)
- ...