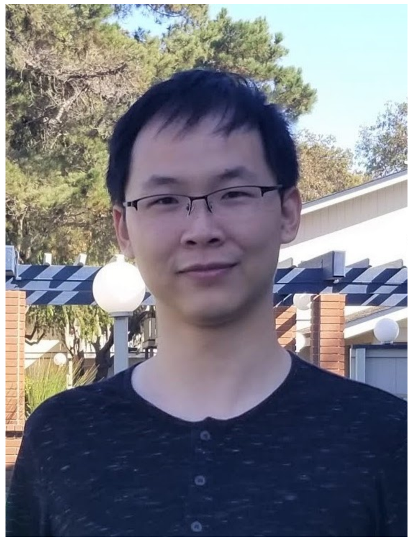
# Near-Optimal Learning of Extensive-Form Games with Imperfect Information

**Yu Bai**
Salesforce Research

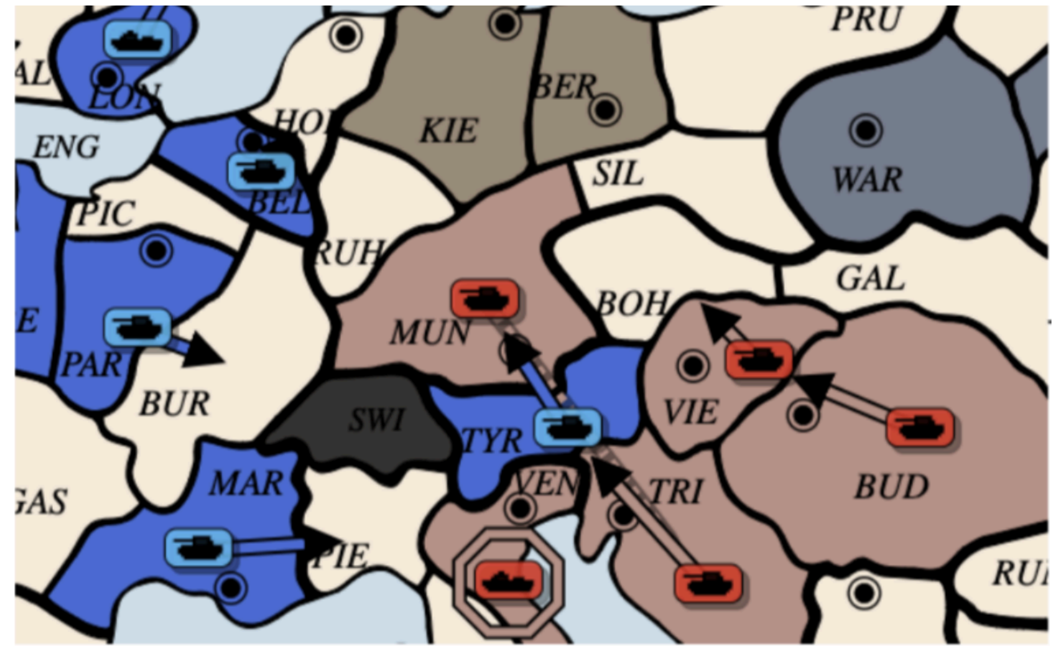


Chi Jin (Princeton)

Song Mei (UC Berkeley)

Tiancheng Yu (MIT)

# Multi-Agent RL / Games with Imperfect Information




**Imperfect Information:**
Players can only observe *partial information* about the true underlying game state

Recent advances in Poker [Moravcik et al. 2017, Brown & Sandholm 2018, 2019], Bridge [Tian et al. 2020], Diplomacy [Bakhtin et al. 2021], …

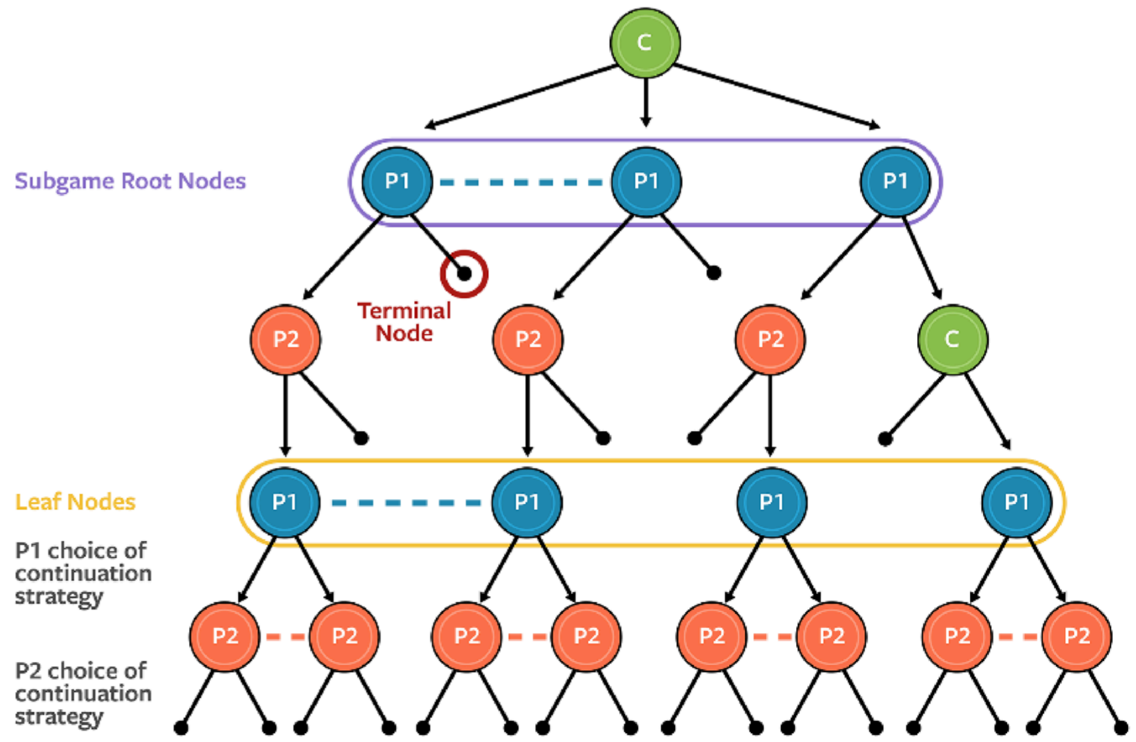
*Image source (right):*
*No-Press Diplomacy from Scratch, Bakhtin et al. 2021.*

# Outline

- Formulation: Imperfect-Information Extensive-Form Games (IIEFGs)
- Game structure
  - Bilinear structure, sequence-form policies
  - Formulation as online linear regret minimization
- Online Mirror Descent
  - IXOMD algorithm
  - Balanced OMD (our algorithm)
- Counterfactual Regret Minimization
  - MCCFR framework
  - Balanced CFR (our algorithm)
- Implications in multi-player general-sum games

# Imperfect-Information Extensive-Form Games (IIEFGs)

[Kuhn 1953]



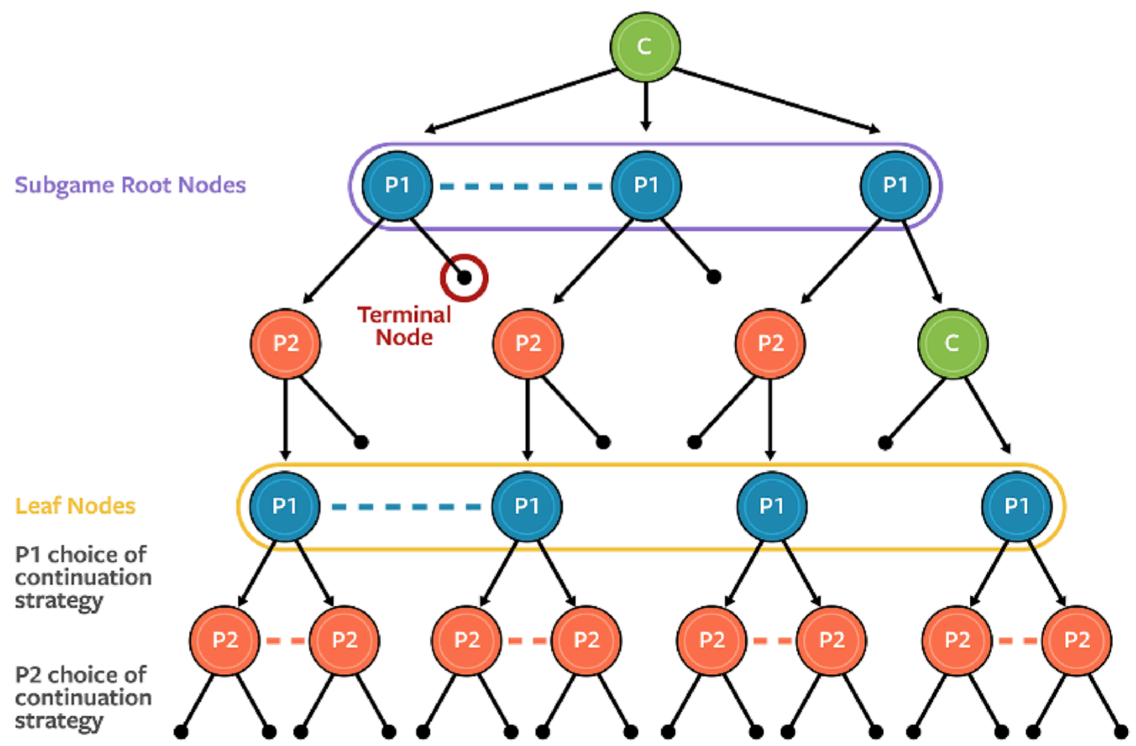

**A commonly used formulation of games involving**

- Multi-agent
- Sequential plays
- Imperfect information

*Image source: Superhuman AI for Multiplayer Poker, Brown & Sandholm 2019.*

# Imperfect-Information Extensive-Form Games (IIEFGs)

[Kuhn 1953]




**A commonly used formulation of games involving**

- Multi-agent
- Sequential plays
- Imperfect information

💡 We formulate IIEFGs as *Partially Observable Markov Games* (POMGs) with *tree structure* + *perfect recall* [Kovarik et al. 2019, Kozuno et al. 2021]

*Image source: Superhuman AI for Multiplayer Poker, Brown & Sandholm 2019.*

# Definition of IIEFGs


Two-player zero-sum IIEFG

- $\mu \in \Pi_{\max}$: max-player
- $\nu \in \Pi_{\min}$: min-player

State, action, reward, transition

$(s_h, a_h, b_h) \longrightarrow (r_h, s_{h+1})$

$r_h = r_h(s_h, a_h, b_h)$

$s_{h+1} \sim p_h(\cdot \mid s_h, a_h, b_h)$

$A = |\mathcal{A}|$, $B = |\mathcal{B}|$

Information sets

$$x_h = x(s_h), \quad y_h = y(s_h)$$

$X$ = # infosets for max-player, $Y$

Policy

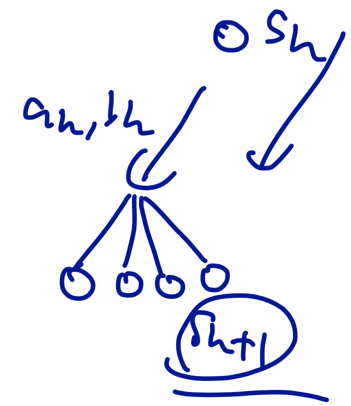
$a_h \sim \{\mu_h(\cdot \mid x_h)\}$

$b_h \sim \{\nu_h(\cdot \mid s_h)\}$

# Definition of IIEFGs

Tree structure:

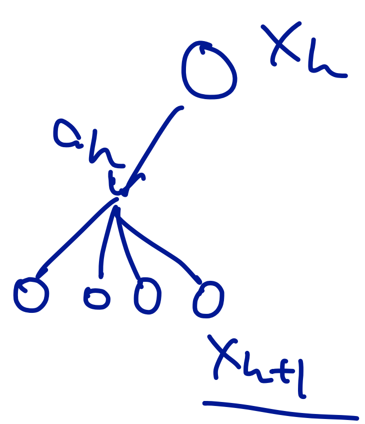
At state $s_h$, history $(s_1, a_1, b_1, \ldots, s_{h-1}, a_{h-1}, b_{h-1})$ is unique




game tree

Perfect recall assumption

At infoset $x_h$, history $(x_1, a_1, \ldots, x_{h-1}, a_{h-1})$ is unique




game tree for max-player

# Learning goals in IIEFGs

**Game value** (expected cumulative reward):

$$V^{\mu,\nu} := \mathbb{E}\Big[\sum_{h=1}^{H} r_h(s_h, a_h, b_h) \mid a_h \sim \mu_h(\cdot \mid x_h), b_h \sim \nu_h(\cdot \mid y_h)\Big]$$

# Learning goals in IIEFGs

**Game value** (expected cumulative reward):

$$V^{\mu,\nu} := \mathbb{E}\Big[\sum_{h=1}^{H} r_h(s_h, a_h, b_h) \mid a_h \sim \mu_h(\,\cdot\,|x_h), b_h \sim \nu_h(\,\cdot\,|y_h)\Big]$$

**Goal: Approximate Nash Equilibrium**

$$\mathrm{NEGap}(\mu, \nu) := \max_{\mu^\dagger} V^{\mu^\dagger,\nu} - \min_{\nu^\dagger} V^{\mu,\nu^\dagger} \leq \varepsilon$$

# Learning goals in IIEFGs

**Game value** (expected cumulative reward):

$$V^{\mu,\nu} := \mathbb{E}\Big[\sum_{h=1}^{H} r_h(s_h, a_h, b_h) \mid a_h \sim \mu_h(\cdot \mid x_h), b_h \sim \nu_h(\cdot \mid y_h)\Big]$$

**Goal: Approximate Nash Equilibrium**

$$\mathrm{NEGap}(\mu, \nu) := \max_{\mu^\dagger} V^{\mu^\dagger,\nu} - \min_{\nu^\dagger} V^{\mu,\nu^\dagger} \le \varepsilon$$

**Goal': No-regret** (only control max player)

$$\mathrm{Reg}(T) := \max_{\mu^\dagger} \sum_{t=1}^{T} V^{\mu^\dagger,\nu^t} - V^{\mu^t,\nu^t} = o(T)$$

# Learning goals in IIEFGs

**Game value** (expected cumulative reward):

$$V^{\mu,\nu} := \mathbb{E}\left[\sum_{h=1}^{H} r_h(s_h, a_h, b_h) \mid a_h \sim \mu_h(\cdot \mid x_h), b_h \sim \nu_h(\cdot \mid y_h)\right]$$

**Goal: Approximate Nash Equilibrium**

$$\mathrm{NEGap}(\mu, \nu) := \max_{\mu^\dagger} V^{\mu^\dagger,\nu} - \min_{\nu^\dagger} V^{\mu,\nu^\dagger} \le \varepsilon$$

**Goal': No-regret** (only control max player)

$$\mathrm{Reg}(T) := \max_{\mu^\dagger} \sum_{t=1}^{T} V^{\mu^\dagger,\nu^t} - V^{\mu^t,\nu^t} = o(T)$$

$\mathrm{Reg}_\mu$, $\mathrm{Reg}_\nu(T)$ $\{\mu^t\}_{t=1}^T$ $\{\nu^t\}_{t=1}^T$ $\bar{\mu}^T, \bar{\nu}^T$

**Online-to-batch conversion** (e.g. [Zinkevich et al. 2007])

Play 2 no-regret algs against each other => Average policies* are approximate Nash

$$\mathrm{NEGap}(\bar{\mu}^T, \bar{\nu}^T) \le \frac{\mathrm{Reg}_\mu(T) + \mathrm{Reg}_\nu(T)}{T}$$

# Bilinear structure, sequence-form policy

[Romanovskii 1962, Koller et al. 1996, Von Stengel 1996, …]

Reaching probability

$$p^{\mu,\nu}_{1:h}(s_h, a_h, b_h) = p_0(s_1)\,\mu_1(a_1|x_1)\,\nu_1(b_1|y_1) \times \cdots \times p_{h-1}(s_h|s_{h-1},a_{h-1},b_{h-1})\,\mu_h(a_h|x_h)\,\nu_h(b_h|y_h)$$

$$= \prod_{h'=1}^{h} \mu_{h'}(a_{h'}|x_{h'}) \times \prod_{h'=1}^{h} p_{h'-1}(s_{h'}|s_{h'-1},a_{h'-1},b_{h'-1}) \cdot \nu_{h'}(b_{h'}|y_{h'})$$

$(x_h, a_h)$

$\mu_{1:h}(x_h, a_h)$ sequence-form policy

Decompose game value

$$H - V^{\mu,\nu} = \sum_{h=1}^{H} \sum_{s_h,a_h,b_h} p^{\mu,\nu}_{1:h}(s_h, a_h, b_h)(1 - r_h(s_h, a_h, b_h))$$

$$= \sum_{h=1}^{H} \sum_{x_h,a_h} \mu_{1:h}(x_h,a_h) \cdot \sum_{\substack{s_h:\,x(s_h)=x_h \\ b_h\in\mathcal{B}}} p_{h'-1}(s_{h'}|s_{h'-1},a_{h'-1},b_{h'-1}) \cdot \nu_{h'}(b_{h'}|y_{h'}) \cdot (1-r_h(s_h,a_h,b_h))$$

$$:= \ell^{\nu}_h(x_h,a_h)$$

# Online linear regret minimization

Opponent $\{\nu^t\}_{t=1}^T$, loss function $\{\ell^t := \ell^{\nu^t}\}_{t=1}^T$

$$H - V^{\mu,\nu^t} = \langle \mu, \ell^t \rangle$$

$$\langle \mu, \ell \rangle := \sum_{h=1}^{H} \sum_{x_h, a_h} \mu_{1:h}(x_h, a_h)\, \ell_h(x_h, a_h)$$

Regret

$$\mathrm{Reg}(T) = \max_{\mu^\dagger \in \Pi_{\max}} \sum_{t=1}^T \left( V^{\mu^\dagger,\nu^t} - V^{\mu^t,\nu^t} \right)$$

$$= \max_{\mu^\dagger \in \Pi_{\max}} \sum_{t=1}^{T} \langle \mu^\dagger - \mu^t, \ell^t \rangle .$$

# Existing algorithms

# Existing algorithms

Full feedback / known game:

- Formulation as a linear program [von Stengel 1996, Koller et al. 1996, …]
- First-order optimization / online mirror descent (OMD) over sequence-form strategy space [Gilpin et al. 2008, Hoda et al. 2010, Kroer et al. 2015, Lee et al. 2021, …]
- Counterfactual regret minimization (CFR) [Zinkevich et al. 2007, Lanctot et al. 2009, Tammelin 2014, Burch et al. 2019, Farina et al. 2020b, …]

# Existing algorithms

Full feedback / known game:

- Formulation as a linear program [von Stengel 1996, Koller et al. 1996, …]
- First-order optimization / online mirror descent (OMD) over sequence-form strategy space [Gilpin et al. 2008, Hoda et al. 2010, Kroer et al. 2015, Lee et al. 2021, …]
- Counterfactual regret minimization (CFR) [Zinkevich et al. 2007, Lanctot et al. 2009, Tammelin 2014, Burch et al. 2019, Farina et al. 2020b, …]

**Bandit feedback** (only observe trajectories from playing):

- Model-based approaches [Zhou et al. 2019, Zhang & Sandholm 2021]
- Monte-Carlo CFR (MCCFR) [Farina et al. 2020c, Farina & Sandholm 2021, …]
- Implicit-Exploration Online Mirror Descent (IXOMD) [Kozuno et al. 2021]
  - Learns an $\varepsilon$-Nash within $\widetilde{O}((X^2A + Y^2B)/\varepsilon^2)$ episodes (prior best; *ignoring* $\mathrm{poly}(H)$)
  - $X, Y$: number of information sets; $A, B$: number of actions
  - Lower bound is $\Omega((XA + YB)/\varepsilon^2)$, still $\max\{X, Y\}$ factor away

# Existing algorithms

Full feedback / known game:

- Formulation as a linear program [von Stengel 1996, Koller et al. 1996, …]
- First-order optimization / online mirror descent (OMD) over sequence-form strategy space [Gilpin et al. 2008, Hoda et al. 2010, Kroer et al. 2015, Lee et al. 2021, …]
- Counterfactual regret minimization (CFR) [Zinkevich et al. 2007, Lanctot et al. 2009, Tammelin 2014, Burch et al. 2019, Farina et al. 2020b, …]

**Bandit feedback** (only observe trajectories from playing):

- Model-based approaches [Zhou et al. 2019, Zhang & Sandholm 2021]
- Monte-Carlo CFR (MCCFR) [Farina et al. 2020c, Farina & Sandholm 2021, …]
- Implicit-Exploration Online Mirror Descent (IXOMD) [Kozuno et al. 2021]
    - Learns an $\varepsilon$-Nash within $\widetilde{O}((X^2A + Y^2B)/\varepsilon^2)$ episodes (current best; *ignoring* $\mathrm{poly}(H)$)
    - $X, Y$: number of information sets; $A, B$: number of actions
    - Lower bound is $\Omega((XA + YB)/\varepsilon^2)$, still $\max\{X, Y\}$ factor away

**Question:** How to design algorithms for learning Nash in two-player zero-sum IIEFGs from *bandit feedback* with *near-optimal sample complexity*?

# Online Mirror Descent (OMD)

[Gilpin et al. 2008, Hoda et al. 2010, Kroer et al. 2015, …]

Recall the regret

$$\mathrm{Reg}(T) = \max_{\mu^\dagger \in \Pi_{\max}} \sum_{t=1}^{T} \langle \mu^t - \mu^\dagger, \ell^t \rangle$$

**Algorithm** (OMD, sketch):

For $t = 1, \ldots, T$:

$$\mu^{t+1} = \underset{\mu \in \Pi_{\max}}{\operatorname{argmin}} \{ \langle \mu, \ell^t \rangle + D(\mu \| \mu^t)$$

# Online Mirror Descent (OMD)

**Algorithm** (OMD, sketch):

For $t = 1, \ldots, T$:

$$\mu^{t+1} = \operatorname*{argmin}_{\mu \in \Pi_{\max}} \eta \langle \mu, \widetilde{\ell}^{\,t} \rangle + D(\mu \| \mu^t)$$

(i) Dilated KL distance [Hoda et. al. 2010, Kroer et al. 2015]:

$$D(\mu \| \mu') := \sum_{h=1}^{H} \sum_{x_h, a_h} \mu_{1:h}(x_h, a_h) \cdot \log \frac{\mu_h(a_h | x_h)}{\mu'_h(a_h | x_h)}.$$

# Online Mirror Descent (OMD)

**Algorithm** (OMD, sketch):

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

$$\mu^{t+1} = \operatorname*{argmin}_{\mu \in \Pi_{\max}} \eta \langle \mu, \tilde{\ell}^t \rangle + D(\mu \| \mu^t)$$

(ii) Loss vector

Full feedback: Set $\tilde{\ell}^t := \ell^t$

# Online Mirror Descent (OMD)

**Algorithm** (OMD, sketch):

For $t = 1, \ldots, T$:

$$\mu^{t+1} = \operatorname*{argmin}_{\mu \in \Pi_{\max}} \eta \langle \mu, \widetilde{\ell}^{\,t} \rangle + D(\mu \| \mu^t)$$

(ii) Loss vector

Full feedback: Set $\widetilde{\ell}^{\,t} := \ell^t$

**Bandit feedback:** Importance weighted loss estimator (like EXP3)

1. Play one episode with $\mu^t$ (opponent plays $\nu^t$), observe trajectory

$$(x_1^t, a_1^t, r_1^t, \ldots, x_H^t, a_H^t, r_H^t)$$

2. Unbiased loss estimator

$$\mathbb{E}[\widetilde{\ell}_h^{\,t}(x_h, a_h)] = \ell_h^t(x_h, a_h).$$

$$\widetilde{\ell}_h^{\,t}(x_h, a_h) = \frac{\mathbb{1}\{x_h^t, a_h^t = x_h, a_h\}}{\mu_{1:h}^t(x_h, a_h)} \cdot (1 - r_h^t)$$

# Implicit-Exploration Online Mirror Descent (IXOMD)

[Kozuno et al. 2021]

**Algorithm** (IXOMD):

1. Play an episode with policy $\mu^t$, construct loss estimator

$$\widetilde{\ell}^t_h(x_h, a_h) := \frac{\mathbf{1}\{(x_h^t, a_h^t) = (x_h, a_h)\} \cdot (1 - r_h^t)}{\mu_{1:h}^t(x_h, a_h) + \gamma}.$$

IX bonus

2. Update policy

$$\mu^{t+1} = \operatorname*{argmin}_{\mu \in \Pi_{\max}} \eta \langle \mu, \widetilde{\ell}^t \rangle + D(\mu \| \mu^t),$$

(with efficient implementation)

**Theorem** [Kozuno, Menard, Munos, Valko, 2021]:

IXOMD achieves $\widetilde{O}(\sqrt{X^2AT})$ regret (against adversarial opponents), and learns $\epsilon$-Nash within $\widetilde{O}((X^2A + Y^2B)/\varepsilon^2)$ episodes of self-play.

# Balanced OMD

**Algorithm** (Balanced OMD, max-player):

1. Play an episode with policy $\mu^t$, construct loss estimator
$$\widetilde{\ell}_h^t(x_h, a_h) := \frac{\mathbf{1}\{(x_h^t, a_h^t) = (x_h, a_h)\} \cdot (1 - r_h^t)}{\mu_{1:h}^t(x_h, a_h) + \gamma \mu_{1:h}^{\star,h}(x_h, a_h)}.$$
2. Update policy
$$\mu^{t+1} = \operatorname*{argmin}_{\mu \in \Pi_{\max}} \eta \langle \widetilde{\ell}^t, \mu \rangle + D^{\mathrm{bal}}(\mu \| \mu^t),$$
(with efficient implementation)

# Balanced OMD

**Algorithm** (Balanced OMD, max-player):

1. Play an episode with policy $\mu^t$, construct loss estimator

$$\widetilde{\ell}^{\,t}_h(x_h, a_h) := \frac{\mathbf{1}\{(x_h^t, a_h^t) = (x_h, a_h)\} \cdot (1 - r_h^t)}{\mu_{1:h}^t(x_h, a_h) + \gamma \mu_{1:h}^{\star,h}(x_h, a_h)}.$$

2. Update policy

$$\mu^{t+1} = \operatorname*{argmin}_{\mu \in \Pi_{\max}} \eta \langle \widetilde{\ell}^{\,t}, \mu \rangle + D^{\mathrm{bal}}(\mu \| \mu^t),$$

(with efficient implementation)

Main new ingredient: **Balanced dilated KL distance**

$$D^{\mathrm{bal}}(\mu \| \mu') := \sum_{h, x_h, a_h} \frac{\mu_{1:h}(x_h, a_h)}{\mu_{1:h}^{\star,h}(x_h, a_h)} \log \frac{\mu_h(a_h \mid x_h)}{\mu'_h(a_h \mid x_h)},$$

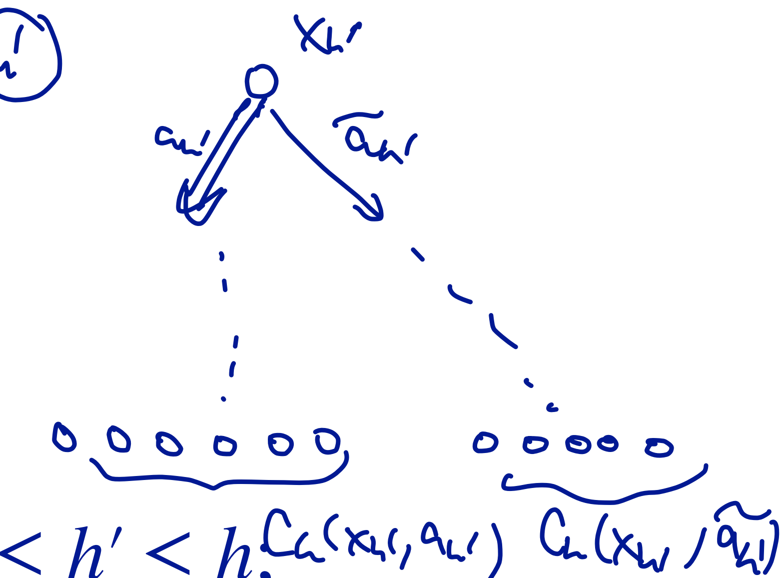
= Dilated KL + reweighting by **Balanced exploration policies** $\{\mu^{\star,h}\}_{h=1}^H$

$$\mu_{1:h}^{\star,h}(x_h, a_h) = \prod_{h'=1}^{h} \frac{|C_h(x_{h'}, a_{h'})|}{|C_h(x_{h'})|}$$

Number of descendants of $(x_{h'}, a_{h'})$ within h-th layer

(extension of [Farina et al. 2020c]).

# Balanced exploration policies




Sequence-form (till step $h$): $\mu^{\star,h}_{1:h}(x_h, a_h) = \prod_{h'=1}^{h} \frac{|C_h(x_{h'}, a_{h'})|}{|C_h(x_{h'})|}$

Conditional-form: $\mu^{\star,h}_{h'}(a_{h'} | x_{h'}) = \begin{cases} \frac{|C_h(x_{h'}, a_{h'})|}{|C_h(x_{h'})|}, & \text{for } 1 \le h' \le h, \\ 1/A, & \text{for } h+1 \le h' \le H. \end{cases}$

**Intuition:** Visit "larger subtrees" more often, balanced by # descendants in layer h

"Balancing property": For any $\mu \in \Pi_{\max}$,

$$\sum_{x_h, a_h} \frac{\mu_{1:h}(x_h, a_h)}{\mu^{\star,h}_{1:h}(x_h, a_h)} = X_h A.$$

# Balanced OMD

**Algorithm** (Balanced OMD, max-player):

1. Play an episode with policy $\mu^t$, construct loss estimator

$$\widetilde{\ell}^{\,t}_h(x_h, a_h) := \frac{\mathbf{1}\{(x^t_h, a^t_h) = (x_h, a_h)\} \cdot (1 - r^t_h)}{\mu^t_{1:h}(x_h, a_h) + \gamma \mu^{\star,h}_{1:h}(x_h, a_h)}.$$

2. Update policy

$$\mu^{t+1} = \operatorname*{argmin}_{\mu \in \Pi_{\max}} \eta \langle \widetilde{\ell}^{\,t}, \mu \rangle + D^{\mathrm{bal}}(\mu \| \mu^t),$$

(with efficient implementation)

**Theorem** [**Bai**, Jin, Mei, Yu, 2022]:
~~IX~~ Balanced OMD achieves $\widetilde{O}(\sqrt{XAT})$ regret (against adversarial opponents), and learns $\varepsilon$-Nash within $\widetilde{O}((XA + YB)/\varepsilon^2)$ episodes of self-play.

# Balanced OMD

**Algorithm** (Balanced OMD, max-player):

1. Play an episode with policy $\mu^t$, construct loss estimator

$$\widetilde{\ell}_h^{\,t}(x_h, a_h) := \frac{\mathbf{1}\{(x_h^t, a_h^t) = (x_h, a_h)\} \cdot (1 - r_h^t)}{\mu_{1:h}^t(x_h, a_h) + \gamma \mu_{1:h}^{\star,h}(x_h, a_h)}.$$

2. Update policy

$$\mu^{t+1} = \arg\min_{\mu \in \Pi_{\max}} \eta \langle \widetilde{\ell}^{\,t}, \mu \rangle + D^{\mathrm{bal}}(\mu \| \mu^t),$$

(with efficient implementation)

**Main technical highlight:**

"Balancing effect" introduced by $D^{\mathrm{bal}}$ (adapts to geometry of policy space)

==> better stability bound than existing OMD analyses (e.g. [Kozuno et al. 2021]) ,

by bounding a certain *log-partition function* via *2nd order Taylor expansion*

# Counterfactual Regret Minimization

[Zinkevich et al. 2007]

**Idea:** Counterfactual Regret Decomposition (≈ performance difference lemma)

$\to \langle \mu^t - \mu^\dagger, \ell^t \rangle$

$$= \sum_{h=1}^{H} \mathbb{E}_{\mu^\dagger_{1:h-1}\mu^t_{h:H}}\left[\sum_{h'=h}^{H} r_h\right] - \mathbb{E}_{\mu^\dagger_{1:h}\mu^t_{h+1:H}}\left[\sum_{h'=h}^{H} r_h\right]$$

$$= \sum_{h=1}^{H} \sum_{x_h, a_h} \mu^\dagger_{1:h-1}(x_{h-1}, a_{h-1}) \cdot \left(\mu^t_h(a_h | x_h) - \mu^\dagger_h(a_h | x_h)\right) \cdot L^t_h(x_h, a_h).$$

$$= \sum_{h=1}^{H} \sum_{x_h} \mu^\dagger_{1:h-1}(x_{h-1}, a_{h-1}) \cdot \langle \mu^t_h(\cdot | x_h) - \mu^\dagger_h(\cdot | x_h), L^t_h(x_h, \cdot) \rangle_{a_h}$$

Above, $L^t_h(x_h, a_h)$ is the *counterfactual loss function* (≈ Q function x "probabilities")

$$L^t_h(x_h, a_h) := \ell^t_h(x_h, a_h) + \sum_{h'=h+1}^{H} \sum_{(x_{h'}, a_{h'}) \in (x_h, a_h)} \mu_{h+1:h'}(x_{h'}, a_{h'}) \cdot \ell^t_{h'}(x_{h'}, a_{h'}).$$

# Counterfactual Regret Minimization

[Zinkevich et al. 2007]

Counterfactual regret decomposition:

$$\mathrm{Reg}(T) = \max_{\mu^\dagger \in \Pi_{\max}} \sum_{t=1}^{T} \langle \mu^t - \mu^\dagger, \ell^t \rangle$$

(i)

$$\leq \sum_{h=1}^{H} \max_{\mu^\dagger_{1:h-1}} \sum_{x_h, a_h} \mu^\dagger_{1:h-1}(x_{h-1}, a_{h-1}) \underbrace{\max_{\mu^\dagger(\cdot|x_h)} \sum_{t=1}^{T} \langle \mu^t(\cdot|x_h) - \mu^\dagger(\cdot|x_h), L^t_h(x_h, \cdot) \rangle_{a_h}}_{:= R^{\mathrm{imm},T}_h(x_h)}$$

$\leq 1$

(ii)

$$\leq \sum_{h=1}^{H} \sum_{x_h, a_h} R^{\mathrm{imm},T}_h(x_h).$$

**Algorithm** (CFR, sketch):

For $t = 1, \ldots, T$, all $(h, x_h, a_h)$:

$$\mu^{t+1}(\cdot|x_h) = R_{x_h}.\mathrm{Update}(\{\widetilde{L}^t_h(x_h, a)\}_{a \in \mathcal{A}})$$

Regret minimization subroutine on simplex (e.g. Hedge)

Loss estimator for counterfactual losses

# Monte-Carlo Counterfactual Regret Minimization (MCCFR)

[Lanctot et al. 2009]

**Algorithm** (MCCFR framework, bandit feedback case):

For $t = 1,\ldots,T$:

Not necessarily $\mu^t$

1. Play **one** episode with some sampling policy $\widetilde{\mu}^t$, observe trajectory

$$(x_1^t, a_1^t, r_1^t, \ldots, x_H^t, a_H^t, r_H^t)$$

2. Construct unbiased counterfactual loss estimator

e.g. from $\{\widetilde{\ell}_h^t(x_h, a_h)\}$

$$\widetilde{L}_h^t(x_h, a_h): \quad \mathbb{E}[\widetilde{L}_h^t(x_h, a_h)] = L_h^t(x_h, a_h).$$

3. Update policy at each information set

$$\mu^{t+1}(\cdot \mid x_h) = R_{x_h}.\mathrm{Update}(\{\widetilde{L}_h^t(x_h, a)\}_{a \in \mathscr{A}}).$$

# Monte-Carlo Counterfactual Regret Minimization (MCCFR)

[Lanctot et al. 2009]

**Algorithm** (MCCFR framework, bandit feedback case):

For $t = 1, \ldots, T$:

Not necessarily $\mu^t$

1. Play **one** episode with some sampling policy $\widetilde{\mu}^t$, observe trajectory
$$(x_1^t, a_1^t, r_1^t, \ldots, x_H^t, a_H^t, r_H^t)$$
2. Construct unbiased counterfactual loss estimator
$$\widetilde{L}_h^t(x_h, a_h): \quad \mathbb{E}[\widetilde{L}_h^t(x_h, a_h)] = L_h^t(x_h, a_h).$$
e.g. from $\{\widetilde{\ell}_h^t(x_h, a_h)\}$
3. Update policy at each information set
$$\mu^{t+1}(\cdot \mid x_h) = R_{x_h}.\mathrm{Update}(\{\widetilde{L}_h^t(x_h, a)\}_{a \in \mathscr{A}}).$$

Many design choices:

- Sampling policy $\widetilde{\mu}^t$
- Loss estimator
- Regret minimization algorithm $R_{x_h}$ (e.g. Hedge, Regret Matching, …)
- Bandit feedback / general stochastic feedback (>1 episodes per iteration)

# MCCFR framework

[Lanctot et al. 2009]

**Algorithm** (MCCFR framework, bandit feedback case):

For $t = 1,\ldots,T$:

1. Play **one** episode with some sampling policy $\widetilde{\mu}^t$, observe trajectory
$$(x_1^t, a_1^t, r_1^t, \ldots, x_H^t, a_H^t, r_H^t)$$
2. Construct unbiased counterfactual loss estimator
$$\widetilde{L}_h^t(x_h, a_h) : \quad \mathbb{E}[\widetilde{L}_h^t(x_h, a_h)] = L_h^t(x_h, a_h).$$
3. Update policy at each information set
$$\mu^{t+1}(\cdot \mid x_h) = R_{x_h}.\mathrm{Update}(\{\widetilde{L}_h^t(x_h, a)\}_{a \in \mathscr{A}}).$$

- An initial regret concentration analysis is given in [Farina et al. 2020c]
- Later instantiated by [Farina & Sandholm 2021] => $\widetilde{O}(\mathrm{poly}(X, Y, A, B)/\epsilon^4)$ rate for learning NE from bandit feedback.

# Balanced CFR

Mixture of $\mu^{\star,h}$ and $\mu^t$

**Algorithm** (Balanced CFR, max-player):

1. Play **H** episodes with policy $\mu^{\star,h}_{1:h}\mu^t_{h+1:H}$, observe trajectory
$$(x_1^{t,(h)}, a_1^{t,(h)}, r_1^{t,(h)}, \ldots, x_H^{t,(h)}, a_H^{t,(h)}, r_H^{t,(h)})$$
2. Construct counterfactual loss estimator
$$\widetilde{L}_h^t(x_h, a_h) := \frac{\mathbf{1}\{(x_h^{t,(h)}, a_h^{t,(h)}) = (x_h, a_h)\}}{\mu^{\star,h}_{1:h}(x_h, a_h)} \cdot \sum_{h'=h}^{H} (1 - r_{h'}^{t,(h)}).$$
3. Update policy at each information set via **Hedge**
$$\mu_h^{t+1}(a \,|\, x_h) \propto_a \mu_h^t(a \,|\, x_h) \cdot \exp\Big( - \eta \mu^{\star,h}_{1:h}(x_h, a) \widetilde{L}_h^t(x_h, a) \Big).$$
(can also use Regret Matching [Zinkevich et al. 2007].)

# Balanced CFR

Mixture of $\mu^{\star,h}$ and $\mu^t$

**Algorithm** (Balanced CFR, max-player):

1. Play **H** episodes with policy $\mu^{\star,h}_{1:h}\mu^t_{h+1:H}$, observe trajectory
$$(x_1^{t,(h)}, a_1^{t,(h)}, r_1^{t,(h)}, \ldots, x_H^{t,(h)}, a_H^{t,(h)}, r_H^{t,(h)})$$
2. Construct counterfactual loss estimator
$$\widetilde{L}_h^t(x_h, a_h) := \frac{\mathbf{1}\{(x_h^{t,(h)}, a_h^{t,(h)}) = (x_h, a_h)\}}{\mu^{\star,h}_{1:h}(x_h, a_h)} \cdot \sum_{h'=h}^{H} (1 - r_{h'}^{t,(h)}).$$
3. Update policy at each information set via **Hedge**
$$\mu_h^{t+1}(a \mid x_h) \propto_a \mu_h^t(a \mid x_h) \cdot \exp\Big( -\eta \mu^{\star,h}_{1:h}(x_h, a) \widetilde{L}_h^t(x_h, a) \Big).$$
(can also use Regret Matching [Zinkevich et al. 2007].)

Our Balanced CFR Algorithm = MCCFR framework

+ balanced exploration policy $\{\mu^{\star,h}\}$

+ sampling by **mixing importance weighting** (using $\mu^{\star,h}$) **and Monte Carlo** (using $\mu^t$)

+ "adaptive" learning rate $\mu^{\star,h}_{1:h}(x_h, a_h)$ at each infoset

# Balanced CFR

**Algorithm** (Balanced CFR, max-player):

1. Play **H** episodes with policy $\mu_{1:h}^{\star,h}\mu_{h+1:H}^{t}$, observe trajectory
$$(x_1^{t,(h)}, a_1^{t,(h)}, r_1^{t,(h)}, \dots, x_H^{t,(h)}, a_H^{t,(h)}, r_H^{t,(h)})$$
2. Construct counterfactual loss estimator
$$\widetilde{L}_h^t(x_h, a_h) := \frac{\mathbf{1}\{(x_h^{t,(h)}, a_h^{t,(h)}) = (x_h, a_h)\}}{\mu_{1:h}^{\star,h}(x_h, a_h)} \cdot \sum_{h'=h}^{H} (1 - r_{h'}^{t,(h)}).$$
3. Update policy at each information set via **Hedge**
$$\mu_h^{t+1}(a \mid x_h) \propto_a \mu_h^t(a \mid x_h) \cdot \exp\Big( - \eta \mu_{1:h}^{\star,h}(x_h, a) \widetilde{L}_h^t(x_h, a) \Big).$$

**Theorem** [**Bai**, Jin, Mei, Yu, 2022]:

Balanced CFR learns $\varepsilon$-Nash within $\widetilde{O}((XA + YB)/\varepsilon^2)$ episodes of self-play.

🤔 $\{\mu^t\}_{t=1}^T$ also achieves $\mathrm{Reg}(T) \le \widetilde{O}(\sqrt{XAT})$, but ≠ actual played policies.

# Balanced CFR

**Algorithm** (Balanced CFR, max-player):

1. Play **H** episodes with policy $\mu^{\star,h}_{1:h}\mu^{t}_{h+1:H}$, observe trajectory
$$(x_1^{t,(h)}, a_1^{t,(h)}, r_1^{t,(h)}, \ldots, x_H^{t,(h)}, a_H^{t,(h)}, r_H^{t,(h)})$$
2. Construct counterfactual loss estimator
$$\widetilde{L}_h^t(x_h, a_h) := \frac{\mathbf{1}\{(x_h^{t,(h)}, a_h^{t,(h)}) = (x_h, a_h)\}}{\mu^{\star,h}_{1:h}(x_h, a_h)} \cdot \sum_{h'=h}^{H} (1 - r_{h'}^{t,(h)}).$$
3. Update policy at each information set via **Hedge**
$$\mu_h^{t+1}(a \mid x_h) \propto_a \mu_h^t(a \mid x_h) \cdot \exp\Big( - \eta \mu^{\star,h}_{1:h}(x_h, a) \widetilde{L}_h^t(x_h, a) \Big).$$

**Main technical highlight:**

Sharp counterfactual regret decomposition involving coefficient $\mu^{\dagger}_{1:h-1}(x_{h-1}, a_{h-1})$

"balanced" with Hedge's regret bound $\dfrac{\log A}{\eta\, \mu^{\star,h}_{1:h}(x_h, a)} + \eta \sum_{a,t} \mu^{\star,h}_{1:h}(x_h, a) \cdot \widetilde{L}_h^t(x_h, a)^2$

$$\frac{\Delta A \cdot XA}{\eta} + \eta T.$$

# Comparison against existing results

| Algorithm | OMD | CFR | Sample Complexity |
|---|---|---|---|
| Zhang and Sandholm (2021) | - (model-based) | | $\widetilde{\mathcal{O}}\left(S^2AB/\varepsilon^2\right)$ |
| Farina and Sandholm (2021) | | ✓ | $\widetilde{\mathcal{O}}(\mathrm{poly}\left(X, Y, A, B\right)/\varepsilon^4)$ |
| Farina et al. (2021) | ✓ | | $\widetilde{\mathcal{O}}\left(\left(X^4A^3 + Y^4B^3\right)/\varepsilon^2\right)$ |
| Kozuno et al. (2021) | ✓ | | $\widetilde{\mathcal{O}}\left(\left(X^2A + Y^2B\right)/\varepsilon^2\right)$ |
| Balanced OMD (Algorithm 1) | ✓ | | $\widetilde{\mathcal{O}}\left(\left(XA + YB\right)/\varepsilon^2\right)$ |
| Balanced CFR (Algorithm 2) | | ✓ | $\widetilde{\mathcal{O}}\left(\left(XA + YB\right)/\varepsilon^2\right)$ |
| Lower bound (Theorem 6) | - | - | $\Omega\left(\left(XA + YB\right)/\varepsilon^2\right)$ |

# Coarse Correlated Equilibria (CCEs) in multi-player IIEFGs

## Normal-Form Coarse Correlated Equilibrium

$$\mathrm{CCEGap}(\pi) := \max_{i \in [m]} \left( \max_{\pi_i^\dagger} V^{\pi_i^\dagger, \pi_{-i}} - V^{\pi} \right) \le \varepsilon$$

No gains in deviating from *correlated policy* $\boldsymbol{\pi}$

# Coarse Correlated Equilibria (CCEs) in multi-player IIEFGs

**Normal-Form Coarse Correlated Equilibrium**

$$\mathrm{CCEGap}(\pi) := \max_{i \in [m]} \left( \max_{\pi_i^\dagger} V^{\pi_i^\dagger, \pi_{-i}} - V^{\pi} \right) \le \varepsilon$$

No gains in deviating from *correlated policy* $\boldsymbol{\pi}$

**Corollary:** Run Balanced OMD or Balanced CFR on all players ==> $\varepsilon$-NFCCE of multi-player general-sum IIEFGs within $\widetilde{O}((\max_i X_i A_i)/\varepsilon^2)$ episodes of play.

Proof follows directly by known connection between NFCCE and no-regret learning in multi-player general-sum IIEFGs [Celli et al. 2019].

# Summary

First line of near-optimal algorithms for learning IIEFGs from bandit feedback

Crucial use of **balanced exploration policies**

- distance functions in OMD
- sampling policies in CFR

# Summary

First line of near-optimal algorithms for learning IIEFGs from bandit feedback

Crucial use of **balanced exploration policies**

- distance functions in OMD
- sampling policies in CFR

**Future directions**

- Further understandings of OMD/CFR type algorithms
- Sample-efficient learning of other equilibria (e.g. correlated equilibria)
- Relationship between Markov Games and Extensive-Form Games
- Empirical investigations

**Thank you!**

https://arxiv.org/abs/2202.01752