Provably Efficient Reinforcement Learning for Mean-Field Leader-Follower Games

Sebastian Jaimungal, U. Toronto & Oxford Man Institute

Wenlong Mou and **Weizheng Zhang**, U. Toronto

Statistical Sciences
UNIVERSITY OF TORONTO

Big Picture

Question

How can a **leader learn** an optimal policy when sampling from **finitely many boundedly rational followers** from a population?

Big Picture

Question

How can a **leader learn** an optimal policy when sampling from **finitely many boundedly rational followers** from a population?

- **Episodic Stackelberg Mean Field Game**

Big Picture

Question

How can a **leader learn** an optimal policy when sampling from **finitely many boundedly rational followers** from a population?

- **Episodic Stackelberg Mean Field Game**
- Followers are **myopic** and **boundedly rational** hence use a **quantal-response**

Big Picture

Question

How can a **leader learn** an optimal policy when sampling from **finitely many boundedly rational followers** from a population?

- **Episodic Stackelberg Mean Field Game**
- Followers are **myopic** and **boundedly rational** hence use a **quantal-response**
- Leader:
  - is a value-maximizing planner

Big Picture

Question

How can a **leader learn** an optimal policy when sampling from **finitely many boundedly rational followers** from a population?

- **Episodic Stackelberg Mean Field Game**
- Followers are **myopic** and **boundedly rational** hence use a **quantal-response**
- Leader:
  - is a value-maximizing planner
  - does **not know**

Big Picture

Question

How can a **leader learn** an optimal policy when sampling from **finitely many boundedly rational followers** from a population?

- **Episodic Stackelberg Mean Field Game**
- Followers are **myopic** and **boundedly rational** hence use a **quantal-response**
- Leader:
  - is a value-maximizing planner
  - does **not know**
    - transition model

Big Picture

Question

How can a **leader learn** an optimal policy when sampling from **finitely many boundedly rational followers** from a population?

- **Episodic Stackelberg Mean Field Game**
- Followers are **myopic** and **boundedly rational** hence use a **quantal-response**
- Leader:
  - is a value-maximizing planner
  - does **not know**
    - transition model
    - follower/leader reward parameters

Big Picture

Question

How can a **leader learn** an optimal policy when sampling from **finitely many boundedly rational followers** from a population?

- **Episodic Stackelberg Mean Field Game**
- Followers are **myopic** and **boundedly rational** hence use a **quantal-response**
- Leader:
  - is a value-maximizing planner
  - does **not know**
    - transition model
    - follower/leader reward parameters
- Main algorithm: optimistic value iteration method with finite sample observation (**OVI-FSO**).

Big Picture

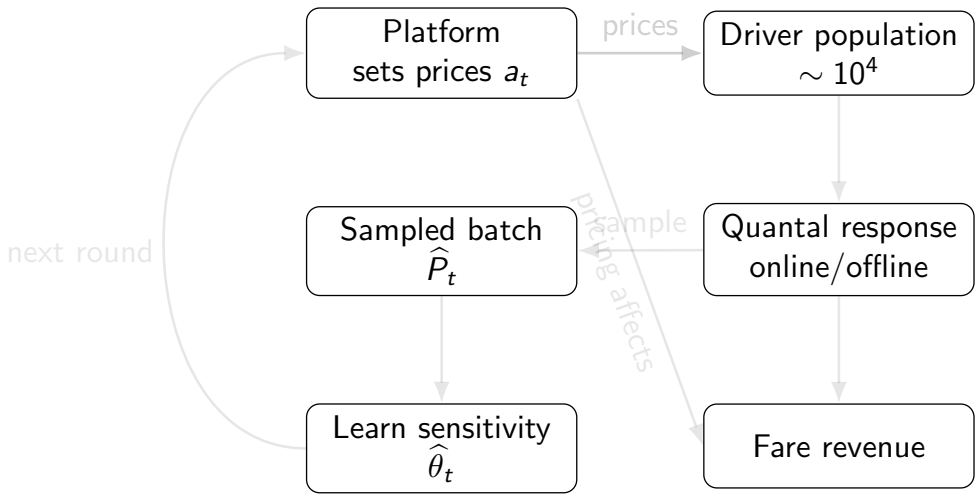
Question

How can a **leader learn** an optimal policy when sampling from **finitely many boundedly rational followers** from a population?

- **Episodic Stackelberg Mean Field Game**
- Followers are **myopic** and **boundedly rational** hence use a **quantal-response**
- Leader:
  - is a value-maximizing planner
  - does **not know**
    - transition model
    - follower/leader reward parameters
- Main algorithm: optimistic value iteration method with finite sample observation (**OVI-FSO**).
- Goal: **sub-linear regret**

Stylized Example: Learning from Sampled Driver Responses

Ride-sharing platform

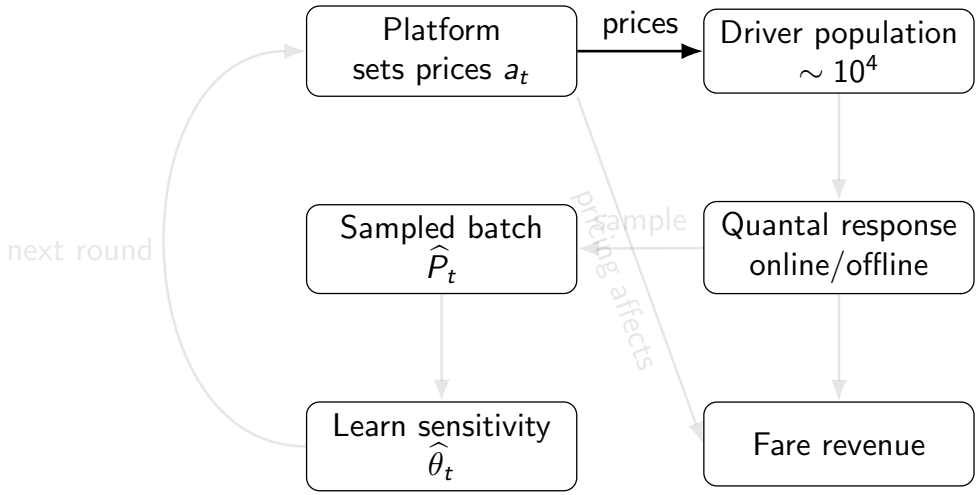


a_1 from prior info

Stylized Example: Learning from Sampled Driver Responses

Ride-sharing platform

- Sets regional surge prices

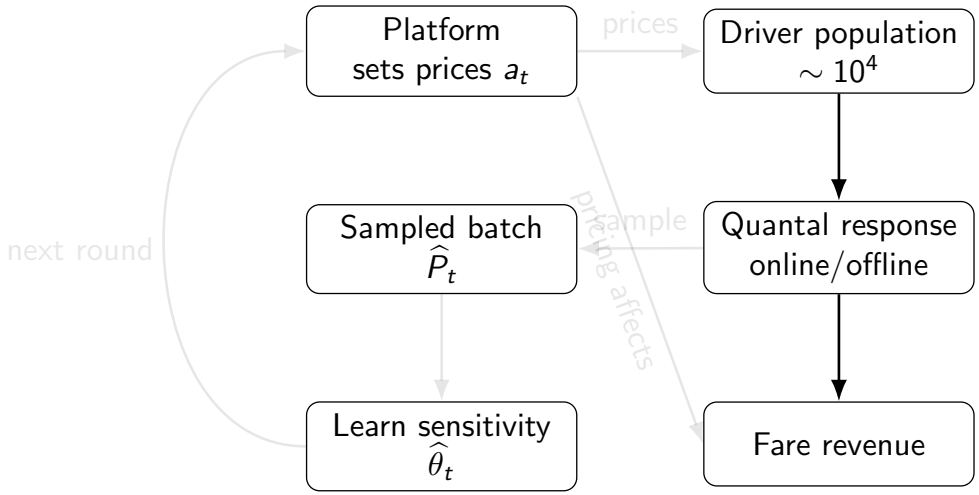


$$\nu^{a_1,\theta^*}(\text{online} \mid x)$$

Stylized Example: Learning from Sampled Driver Responses

Ride-sharing platform

- Sets regional surge prices
- Drivers respond stochastically

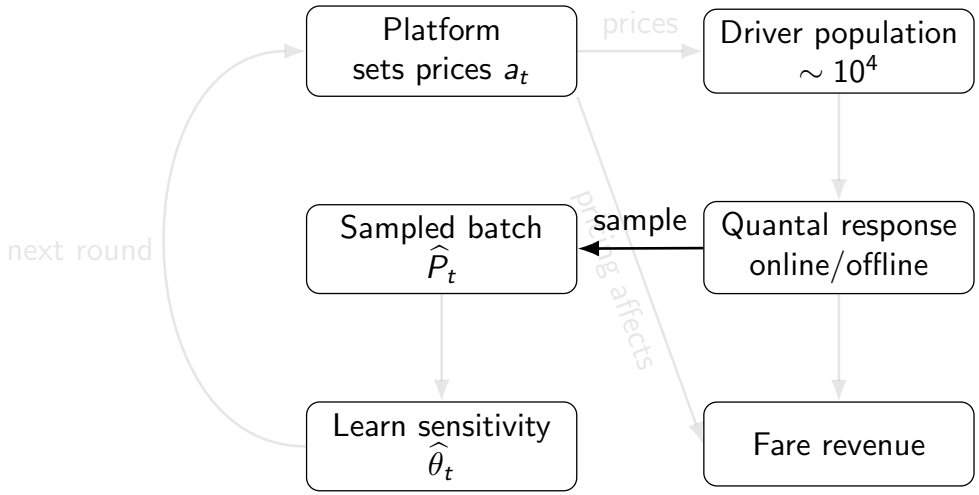


$$\widehat{P}_1 \approx P_1$$

Stylized Example: Learning from Sampled Driver Responses

Ride-sharing platform

- Sets regional surge prices
- Drivers respond stochastically
- Platform observes only a sampled batch

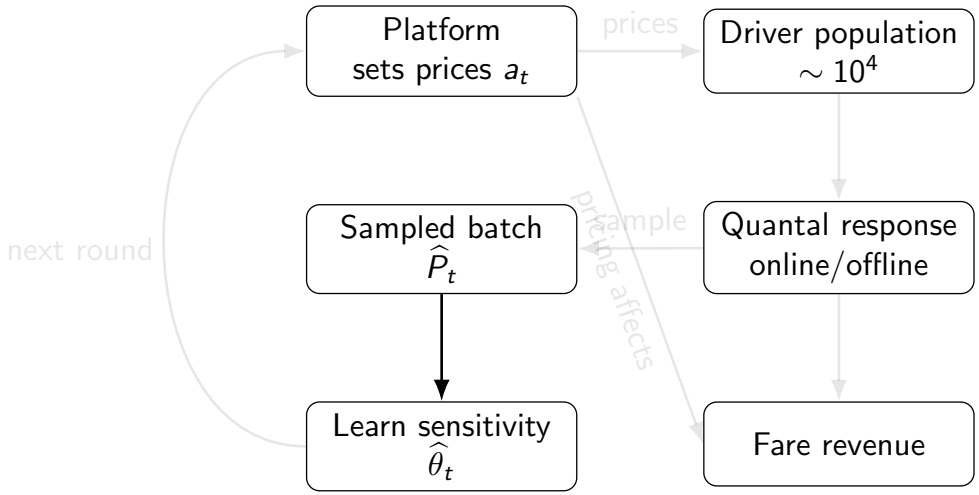


$\widehat{\theta}_1 \leftarrow$ MLE(sampled actions)

Stylized Example: Learning from Sampled Driver Responses

Ride-sharing platform

- Sets regional surge prices
- Drivers respond stochastically
- Platform observes only a sampled batch
- Updates estimate of price sensitivity

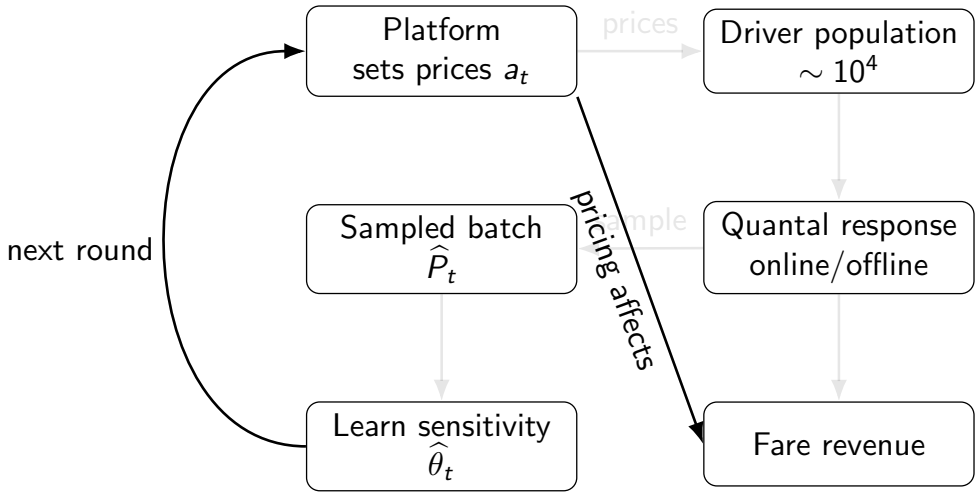


a_2 chosen using $\widehat{\theta}_1$

Stylized Example: Learning from Sampled Driver Responses

Ride-sharing platform

- Sets regional surge prices
- Drivers respond stochastically
- Platform observes only a sampled batch
- Updates estimate of price sensitivity
- Uses estimate to improve next prices



a_2 chosen using $\widehat{\theta}_1$

Game Model

Game Model

- Leader state/action: $s \in \mathcal{S}$, $a \in \mathcal{A}$

Game Model

- Leader state/action: $s \in \mathcal{S}$, $a \in \mathcal{A}$
- Follower state/action: $x \in \mathcal{X}$, $b \in \mathcal{B}$

Game Model

- Leader state/action: $s \in \mathcal{S}$, $a \in \mathcal{A}$
- Follower state/action: $x \in \mathcal{X}$, $b \in \mathcal{B}$
- Mean field: $\mu \in \Delta(\mathcal{X})$

Game Model

- Leader state/action: $s \in \mathcal{S}$, $a \in \mathcal{A}$
- Follower state/action: $x \in \mathcal{X}$, $b \in \mathcal{B}$
- Mean field: $\mu \in \Delta(\mathcal{X})$
- Joint follower state-action distribution:

$$P \in \Delta(\mathcal{Z}), \qquad \mathcal{Z} := \mathcal{X} \times \mathcal{B}.$$

Game Model

- Leader state/action: $s \in \mathcal{S}$, $a \in \mathcal{A}$
- Follower state/action: $x \in \mathcal{X}$, $b \in \mathcal{B}$
- Mean field: $\mu \in \Delta(\mathcal{X})$
- Joint follower state-action distribution:

$$P \in \Delta(\mathcal{Z}), \qquad \mathcal{Z} := \mathcal{X} \times \mathcal{B}.$$

- Leader policy: $\pi : \mathcal{S} \times \Delta(\mathcal{X}) \to \Delta(\mathcal{A})$.

Feature Embeddings

We assume we have fixed feature embeddings as follows:

$$\phi(s, a, P) \qquad \text{and} \qquad \varphi(s, a, x, b)$$

Feature Embeddings

We assume we have fixed feature embeddings as follows:

$$\phi(s, a, P) \qquad \text{and} \qquad \varphi(s, a, x, b)$$

We assume the one-step structure (at stage-h):

Feature Embeddings

We assume we have fixed feature embeddings as follows:

$$\phi(s, a, P) \qquad \text{and} \qquad \varphi(s, a, x, b)$$

We assume the one-step structure (at stage-h):

$$u_h(s, a, P) = \langle \phi(s, a, P)\,,\, \xi^*_{\mathbf{h}} \rangle\,, \qquad \text{leader reward}$$

Feature Embeddings

We assume we have fixed feature embeddings as follows:

$$\phi(s, a, P) \qquad \text{and} \qquad \varphi(s, a, x, b)$$

We assume the one-step structure (at stage-h):

$$\begin{aligned} u_h(s, a, P) &= \langle \phi(s, a, P)\,,\, \xi_{\mathbf{h}}^* \rangle\,, && \text{leader reward} \\ r_h(x, b; s, a) &= \langle \varphi(s, a, x, b)\,,\, \theta_{\mathbf{h}}^* \rangle\,, && \text{follower reward} \end{aligned}$$

Feature Embeddings

We assume we have fixed feature embeddings as follows:

$$\phi(s, a, P) \qquad \text{and} \qquad \varphi(s, a, x, b)$$

We assume the one-step structure (at stage-h):

$$\begin{aligned}
u_h(s, a, P) &= \langle \phi(s, a, P)\,,\, \xi^*_{\mathbf{h}} \rangle\,, && \text{leader reward} \\
r_h(x, b; s, a) &= \langle \varphi(s, a, x, b)\,,\, \theta^*_{\mathbf{h}} \rangle\,, && \text{follower reward} \\
P_h(s', \mu' \mid s, a, P) &= \left\langle \phi(s, a, P)\,,\, \omega^*_{\mathbf{h}}(s', \mu') \right\rangle\,, && \text{transition kernel}
\end{aligned}$$

Concrete Example: Feature Embeddings

Ride-sharing platform

Leader-side features

$$\phi_L(s, a, P) = \begin{bmatrix} 1 \\ a \\ \mathbb{E}_P[b] \\ \mathrm{Var}_P(b) \end{bmatrix}$$

- a: surge price
- P: driver state-action distribution
- $\mathbb{E}_P[b]$: fraction online

Follower-side features

$$\phi_F(s, a, x, b) = \begin{bmatrix} 1 \\ a \\ \mathrm{dist}(x, \mathrm{downtown}) \\ \mathbf{1}\{b = \mathrm{online}\} \end{bmatrix}$$

- x: driver location/state
- b: online/offline choice
- Captures price sensitivity

$$u_h(s, a, P) = \langle \phi_L(s, a, P), \xi_h^* \rangle, \qquad r_h(x, b; s, a) = \langle \phi_F(s, a, x, b), \theta_h^* \rangle.$$

Boundedly Rational Followers

- For a given **leader action** $p \in \Delta(\mathcal{A})$, define

$$\varphi^p(s, x, b) := \sum_{a \in \mathcal{A}} p(a)\, \varphi(s, a, x, b).$$

[followers respond to the **distribution** of leader actions]

Boundedly Rational Followers

- For a given **leader action** $p \in \Delta(\mathcal{A})$, define

$$\varphi^p(s, x, b) := \sum_{a \in \mathcal{A}} p(a)\, \varphi(s, a, x, b).$$

  [followers respond to the **distribution** of leader actions]

- Followers are **myopic** and wish to choose action b to optimize $\varphi^p(s, x, b)$.

Boundedly Rational Followers

- For a given **leader action** $p \in \Delta(\mathcal{A})$, define

$$\varphi^p(s, x, b) := \sum_{a \in \mathcal{A}} p(a)\, \varphi(s, a, x, b).$$

  [followers respond to the **distribution** of leader actions]

- Followers are **myopic** and wish to choose action b to optimize $\varphi^p(s, x, b)$.
- However, they are **bounded rational** and use **quantal response**...

Boundedly Rational Followers

- For a given **leader action** $p \in \Delta(\mathcal{A})$, define

$$\varphi^p(s, x, b) := \sum_{a \in \mathcal{A}} p(a)\, \varphi(s, a, x, b).$$

  [followers respond to the **distribution** of leader actions]

- Followers are **myopic** and wish to choose action b to optimize $\varphi^p(s, x, b)$.
- However, they are **bounded rational** and use **quantal response**...
- At stage-h, the **follower action's** drawn from

$$\nu_h^{p,\theta^*}(b \mid x; s) = \frac{\exp\{\eta \left\langle \varphi^p(s, x, b)\,,\, \theta^*_{\mathbf{h}} \right\rangle\}}{\sum_{b' \in \mathcal{B}} \exp\{\eta \left\langle \varphi^p(s, x, b')\,,\, \theta^*_{\mathbf{h}} \right\rangle\}}.$$

Boundedly Rational Followers

- For a given **leader action** $p \in \Delta(\mathcal{A})$, define

$$\varphi^p(s, x, b) := \sum_{a \in \mathcal{A}} p(a)\, \varphi(s, a, x, b).$$

  [followers respond to the **distribution** of leader actions]

- Followers are **myopic** and wish to choose action b to optimize $\varphi^p(s, x, b)$.
- However, they are **bounded rational** and use **quantal response**...
- At stage-h, the **follower action's** drawn from

$$\nu_h^{p,\theta^*}(b \mid x; s) = \frac{\exp\{\eta \left\langle \varphi^p(s, x, b)\,,\, \theta^*_{\mathbf{h}} \right\rangle\}}{\sum_{b' \in \mathcal{B}} \exp\{\eta \left\langle \varphi^p(s, x, b')\,,\, \theta^*_{\mathbf{h}} \right\rangle\}}.$$

  - $\eta > 0$: inverse temperature / rationality parameter.

Boundedly Rational Followers

- For a given **leader action** $p \in \Delta(\mathcal{A})$, define

$$\varphi^p(s, x, b) := \sum_{a \in \mathcal{A}} p(a)\, \varphi(s, a, x, b).$$

  [followers respond to the **distribution** of leader actions]

- Followers are **myopic** and wish to choose action b to optimize $\varphi^p(s, x, b)$.
- However, they are **bounded rational** and use **quantal response**...
- At stage-h, the **follower action's** drawn from

$$\nu_h^{p,\theta^*}(b \mid x; s) = \frac{\exp\{\eta \left\langle \varphi^p(s, x, b)\,,\, \theta^*_{\mathbf{h}} \right\rangle\}}{\sum_{b' \in \mathcal{B}} \exp\{\eta \left\langle \varphi^p(s, x, b')\,,\, \theta^*_{\mathbf{h}} \right\rangle\}}.$$

  - $\eta > 0$: inverse temperature / rationality parameter.
  - Large η: closer to best response.

Boundedly Rational Followers

- For a given **leader action** $p \in \Delta(\mathcal{A})$, define

$$\varphi^p(s, x, b) := \sum_{a \in \mathcal{A}} p(a)\, \varphi(s, a, x, b).$$

  [followers respond to the **distribution** of leader actions]

- Followers are **myopic** and wish to choose action b to optimize $\varphi^p(s, x, b)$.
- However, they are **bounded rational** and use **quantal response**...
- At stage-h, the **follower action's** drawn from

$$\nu_h^{p,\theta^*}(b \mid x; s) = \frac{\exp\{\eta \left\langle \varphi^p(s, x, b)\,,\, \theta^*_{\mathbf{h}} \right\rangle\}}{\sum_{b' \in \mathcal{B}} \exp\{\eta \left\langle \varphi^p(s, x, b')\,,\, \theta^*_{\mathbf{h}} \right\rangle\}}.$$

  - $\eta > 0$: inverse temperature / rationality parameter.
  - Large η: closer to best response.
  - Small η: more random behavior.

Leader's Objective

- Choose a policy π to maximize **expected cumulative reward**

$$J(\pi) = \mathbb{E}^{\pi}\left[\sum_{h=1}^{H} u_h\big(s_h, a_h, \mu_h \otimes \nu_h^{\pi}\big)\right]$$

Leader's Objective

- Choose a policy π to maximize **expected cumulative reward**

$$J(\pi) = \mathbb{E}^{\pi}\left[\sum_{h=1}^{H} u_h\big(s_h, a_h, \mu_h \otimes \nu_h^{\pi}\big)\right]$$

  - $a_h \sim \pi(\cdot \mid s_h, \mu_h)$: leader chooses actions

Leader's Objective

- Choose a policy π to maximize **expected cumulative reward**

$$J(\pi) = \mathbb{E}^{\pi}\left[\sum_{h=1}^{H} u_h\big(s_h, a_h, \mu_h \otimes \nu_h^{\pi}\big)\right]$$

  - $a_h \sim \pi(\cdot \mid s_h, \mu_h)$: leader chooses actions
  - ν_h^{π}: followers respond (quantal response)

Leader's Objective

- Choose a policy π to maximize **expected cumulative reward**

$$J(\pi) = \mathbb{E}^{\pi}\left[\sum_{h=1}^{H} u_h\big(s_h, a_h, \mu_h \otimes \nu_h^{\pi}\big)\right]$$

  - $a_h \sim \pi(\cdot \mid s_h, \mu_h)$: leader chooses actions
  - ν_h^{π}: followers respond (quantal response)
  - $\mu_h \otimes \nu_h^{\pi}$: population state-action distribution

Leader's Objective

- Choose a policy π to maximize **expected cumulative reward**

$$J(\pi) = \mathbb{E}^{\pi}\left[\sum_{h=1}^{H} u_h\big(s_h, a_h, \mu_h \otimes \nu_h^{\pi}\big)\right]$$

  - $a_h \sim \pi(\cdot \mid s_h, \mu_h)$: leader chooses actions
  - ν_h^{π}: followers respond (quantal response)
  - $\mu_h \otimes \nu_h^{\pi}$: population state-action distribution
  - $u_h(s, a, P)$: reward depends on **aggregate behavior**

Leader's Objective

- Choose a policy π to maximize **expected cumulative reward**

$$J(\pi) = \mathbb{E}^{\pi}\left[\sum_{h=1}^{H} u_h\big(s_h, a_h, \mu_h \otimes \nu_h^{\pi}\big)\right]$$

  - $a_h \sim \pi(\cdot \mid s_h, \mu_h)$: leader chooses actions
  - ν_h^{π}: followers respond (quantal response)
  - $\mu_h \otimes \nu_h^{\pi}$: population state-action distribution
  - $u_h(s, a, P)$: reward depends on **aggregate behavior**
- $J(\pi)$ is a **true expected performance** (not empirical)

Leader's Objective

- Choose a policy π to maximize **expected cumulative reward**

$$J(\pi) = \mathbb{E}^{\pi}\left[\sum_{h=1}^{H} u_h\big(s_h, a_h, \mu_h \otimes \nu_h^{\pi}\big)\right]$$

  - $a_h \sim \pi(\cdot \mid s_h, \mu_h)$: leader chooses actions
  - ν_h^{π}: followers respond (quantal response)
  - $\mu_h \otimes \nu_h^{\pi}$: population state-action distribution
  - $u_h(s, a, P)$: reward depends on **aggregate behavior**
- $J(\pi)$ is a **true expected performance** (not empirical)
- Depends on **unknown dynamics** and follower/leader parameters

Leader's Objective

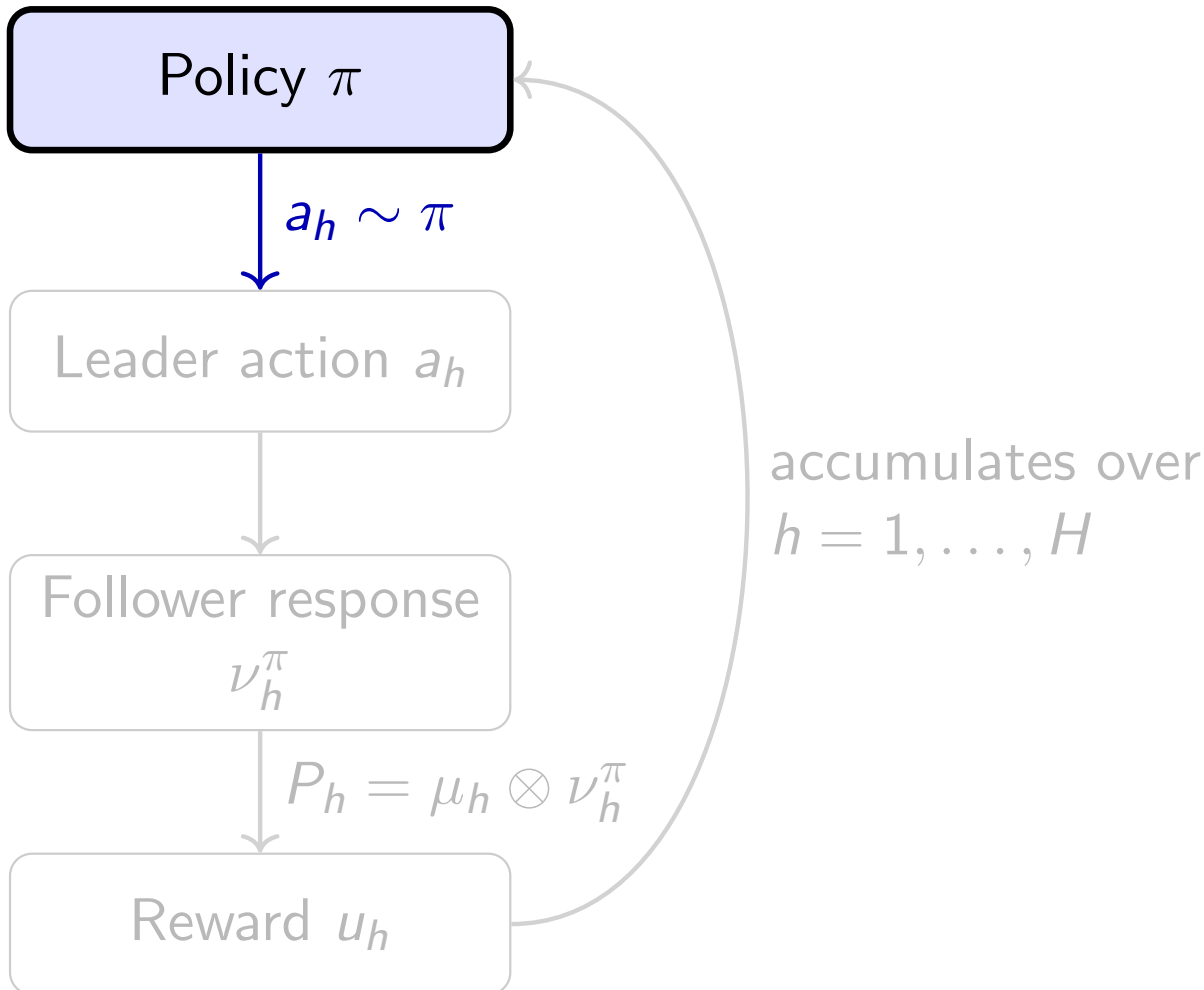
- Choose a policy π to maximize **expected cumulative reward**

$$J(\pi) = \mathbb{E}^{\pi}\left[\sum_{h=1}^{H} u_h\big(s_h, a_h, \mu_h \otimes \nu_h^{\pi}\big)\right]$$

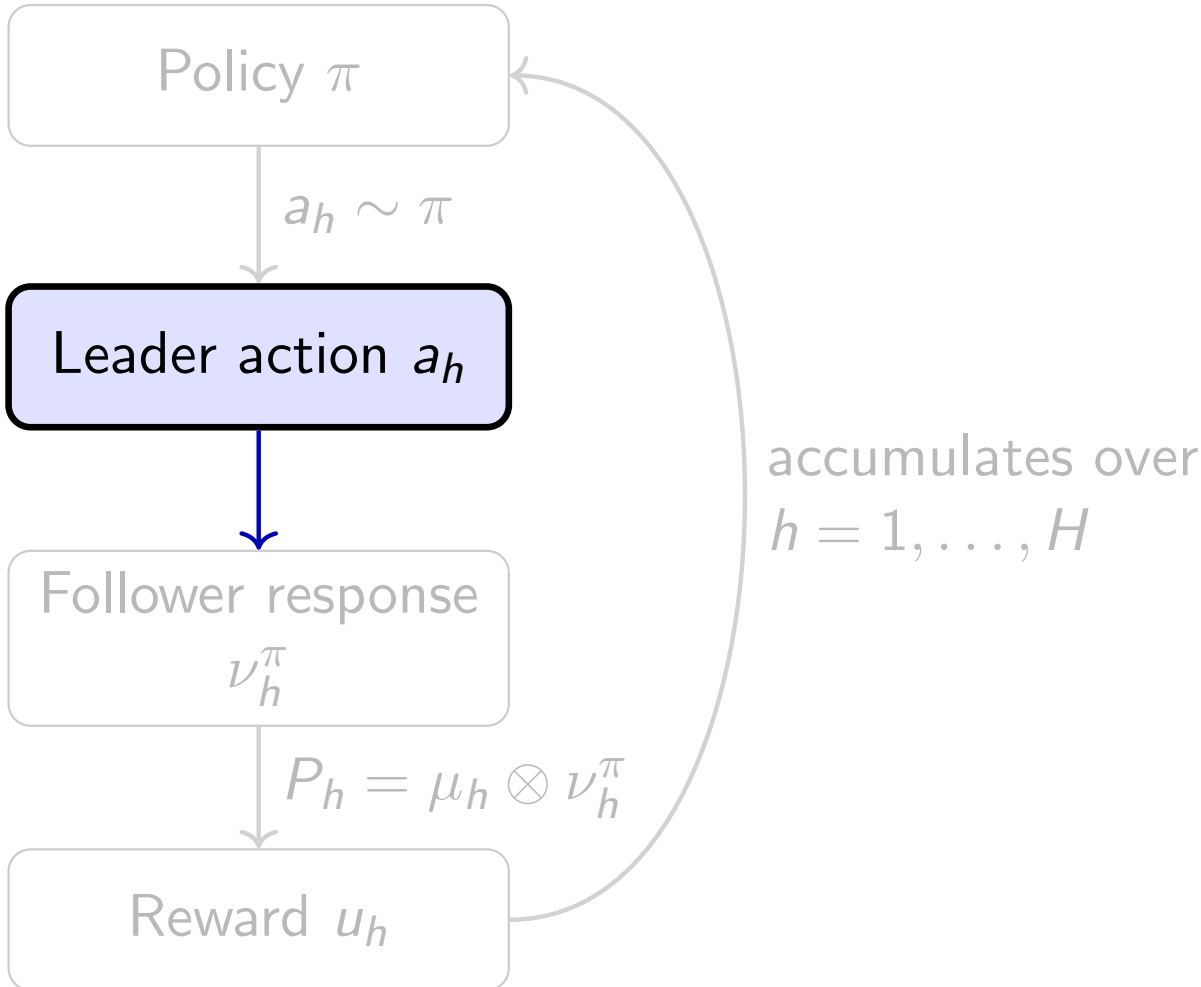
  - $a_h \sim \pi(\cdot \mid s_h, \mu_h)$: leader chooses actions
  - ν_h^{π}: followers respond (quantal response)
  - $\mu_h \otimes \nu_h^{\pi}$: population state-action distribution
  - $u_h(s, a, P)$: reward depends on **aggregate behavior**
- $J(\pi)$ is a **true expected performance** (not empirical)
- Depends on **unknown dynamics** and follower/leader parameters
- Optimal strategy is

$$\pi^* = \operatorname*{argmax}_{\pi} J(\pi)$$

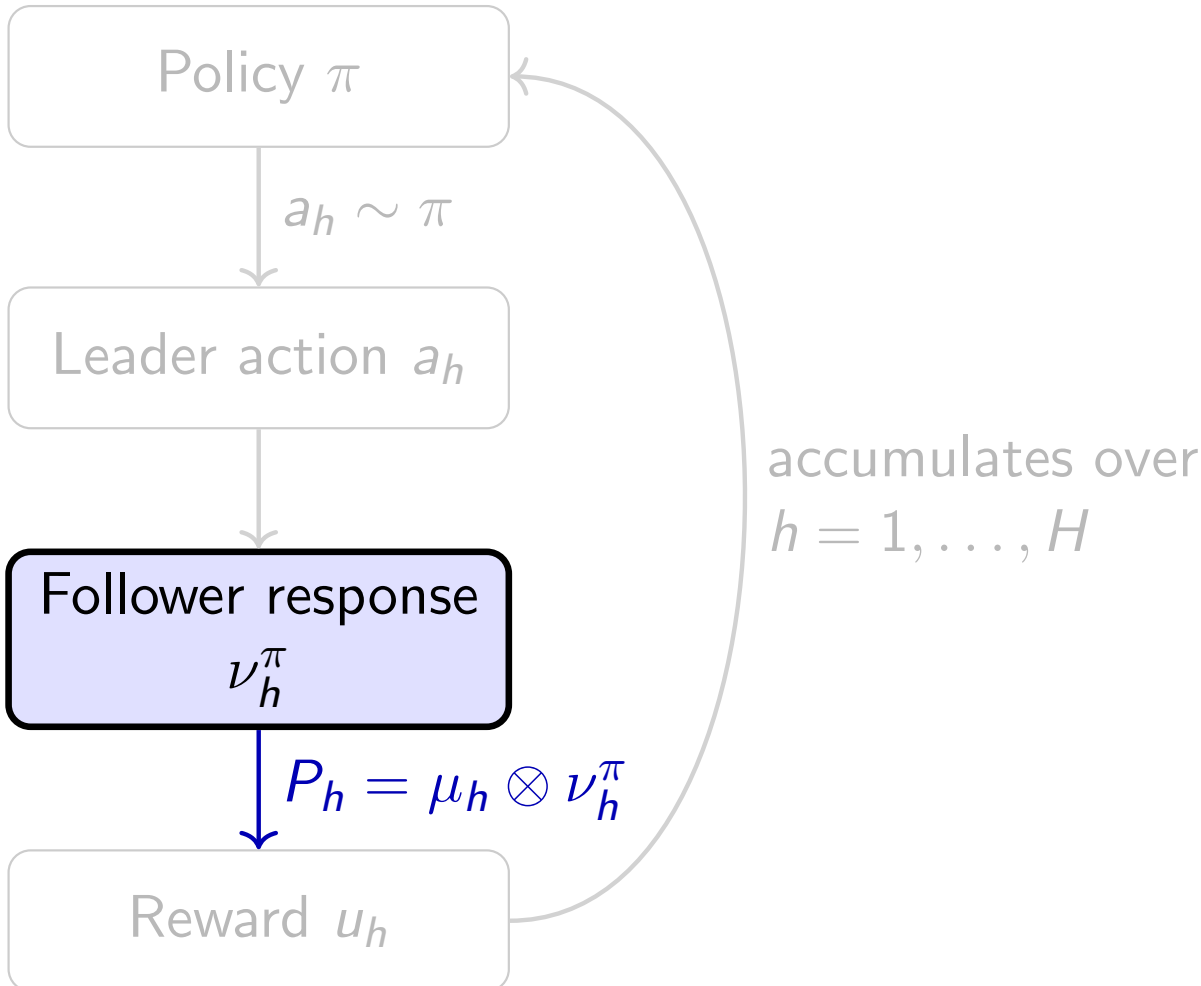
Leader's Objective: Interaction Flow



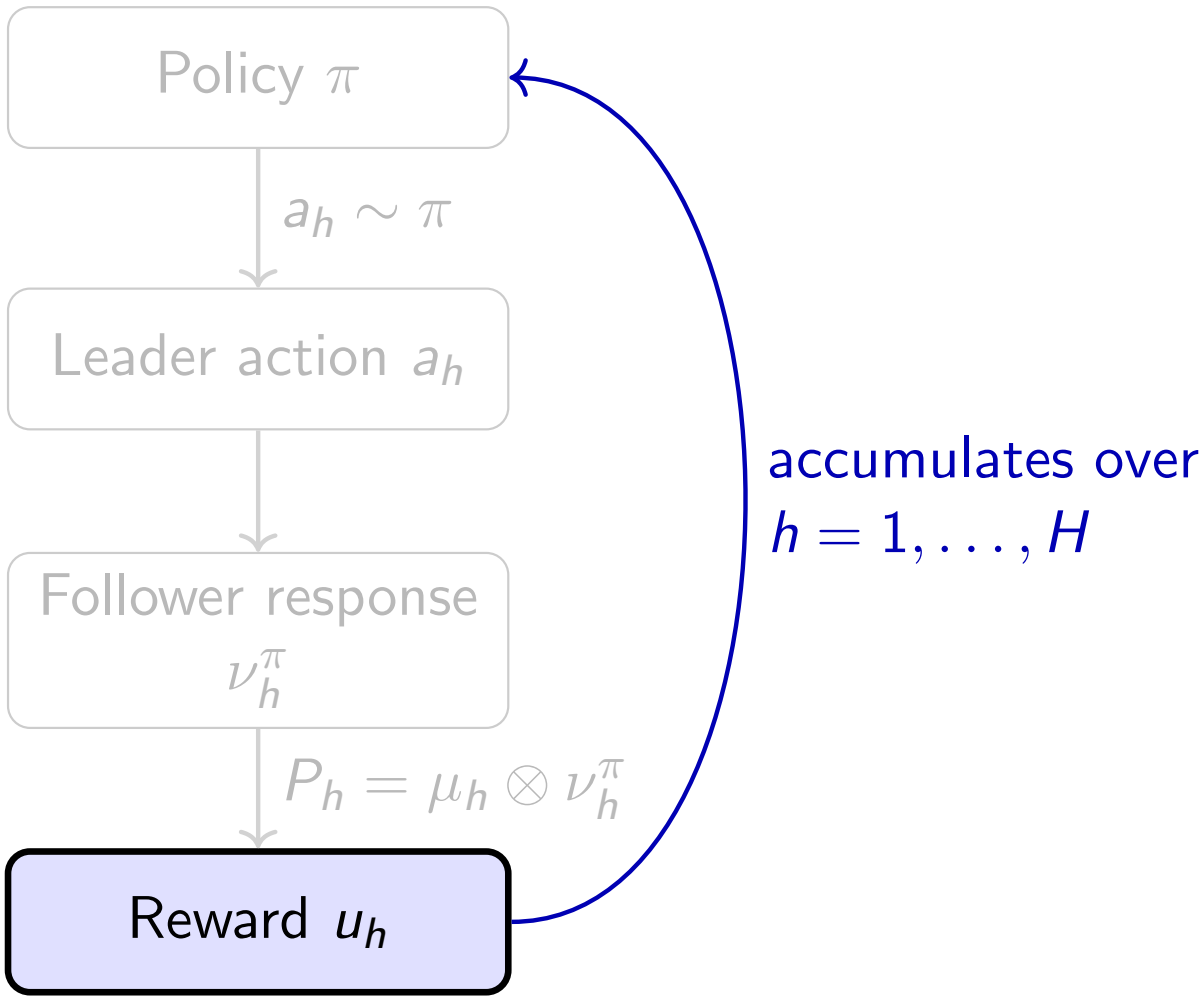
Leader's Objective: Interaction Flow



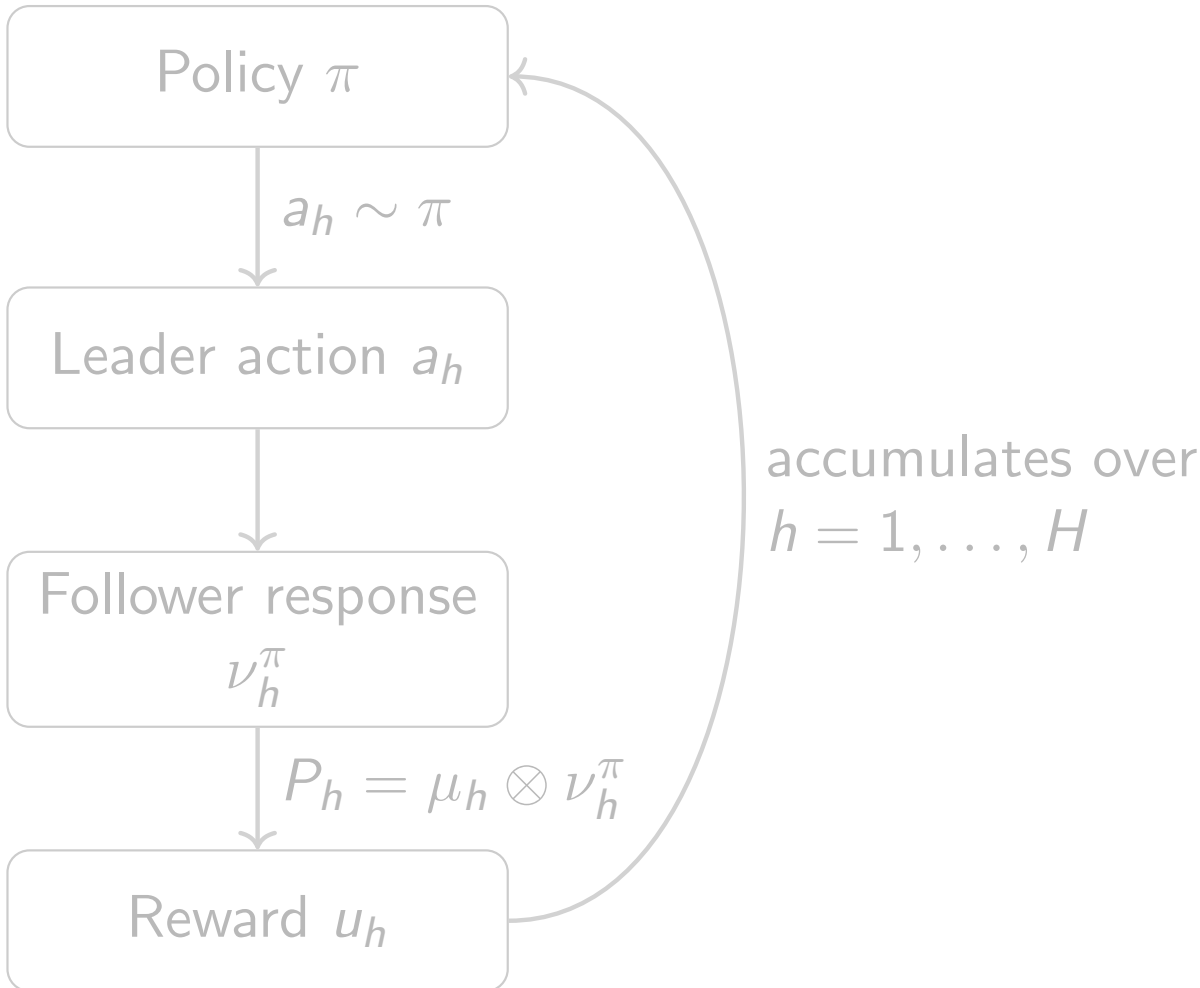
Leader's Objective: Interaction Flow



Leader's Objective: Interaction Flow



Leader's Objective: Interaction Flow



Quantal Stackelberg Equilibrium and Regret

- A **QSE** is a leader policy π^* that maximizes return subject to quantal follower response:

$$\pi^* \in \operatorname*{argmax}_{\pi} J(\pi) \qquad \text{with} \qquad \nu_h = \nu_h^{\pi(\cdot|s,\mu),\theta_h^*}.$$

Quantal Stackelberg Equilibrium and Regret

- A **QSE** is a leader policy π^* that maximizes return subject to quantal follower response:

$$\pi^* \in \operatorname*{argmax}_{\pi} J(\pi) \qquad \text{with} \qquad \nu_h = \nu_h^{\pi(\cdot|s,\mu),\theta_h^*}.$$

- The online learning objective is **cumulative regret** over multiple episodes:

$$\mathrm{Reg}(T) = \sum_{t=1}^{T} \left(J(\pi^*) - J(\pi^t) \right)$$

Quantal Stackelberg Equilibrium and Regret

- A **QSE** is a leader policy π^* that maximizes return subject to quantal follower response:

$$\pi^* \in \operatorname*{argmax}_{\pi} J(\pi) \qquad \text{with} \qquad \nu_h = \nu_h^{\pi(\cdot|s,\mu),\theta_h^*}.$$

- The online learning objective is **cumulative regret** over multiple episodes:

$$\mathrm{Reg}(T) = \sum_{t=1}^{T} \left(J(\pi^*) - J(\pi^t) \right)$$

- Goal: obtain **sub-linear regret** (in terms of T)

Quantal Stackelberg Equilibrium and Regret

- A **QSE** is a leader policy π^* that maximizes return subject to quantal follower response:

$$\pi^* \in \operatorname*{argmax}_{\pi} J(\pi) \qquad \text{with} \qquad \nu_h = \nu_h^{\pi(\cdot|s,\mu),\theta_h^*}.$$

- The online learning objective is **cumulative regret** over multiple episodes:

$$\mathrm{Reg}(T) = \sum_{t=1}^{T} \left(J(\pi^*) - J(\pi^t) \right)$$

- Goal: obtain **sub-linear regret** (in terms of T)
- **Core challenge**: The leader must learn the environment and the followers' response model while choosing policies that affect the data distribution.

Two Observation Regimes

Complete Observation (CO)

The leader observes the **exact population distribution**:

$$P_h^t = \mu_h^t \otimes \nu_h^{p_h^t, \theta_h^*}.$$

- Benchmark
- No finite-sample plug-in error.

Two Observation Regimes

Complete Observation (CO)

The leader observes the **exact population distribution**:

$$P_h^t = \mu_h^t \otimes \nu_h^{p_h^t, \theta_h^*}.$$

- Benchmark
- No finite-sample plug-in error.

Finite-Sample Observation (FSO)

The leader obeserves N **followers** per step:

$$\widehat{\mu}_h^t = \frac{1}{N} \sum_{n=1}^{N} \delta_{X_h^{t,n}},$$

$$\widehat{P}_h^t = \frac{1}{N} \sum_{n=1}^{N} \delta_{(X_h^{t,n}, B_h^{t,n})}.$$

- Adds residual $\sqrt{|\mathcal{Z}|/N}$

Leader-Side Function Approximation

The leader's **value function** is **estimated** in an **RKHS** $\mathcal{H}$ over $\mathcal{S} \times \mathcal{A} \times \Delta(\mathcal{Z})$.

Leader-Side Function Approximation

The leader's **value function** is **estimated** in an **RKHS** $\mathcal{H}$ over $\mathcal{S} \times \mathcal{A} \times \Delta(\mathcal{Z})$.
Complexity is controlled by the **maximal information gain**:

$$G_K(T, \lambda) := \sup_{D \subseteq \mathcal{Z}_{\mathrm{lf}},\ |D| \leq T} \frac{1}{2} \log \det \left(I + K_D/\lambda\right).$$

Leader-Side Function Approximation

The leader's **value function** is **estimated** in an **RKHS** $\mathcal{H}$ over $\mathcal{S} \times \mathcal{A} \times \Delta(\mathcal{Z})$.
Complexity is controlled by the **maximal information gain**:

$$G_K(T, \lambda) := \sup_{D \subseteq \mathcal{Z}_{\mathrm{lf}},\ |D| \leq T} \frac{1}{2} \log \det \left(I + K_D/\lambda\right).$$

- Finite-rank case: $G_K(T, \lambda) = \widetilde{\mathcal{O}}(d_L)$

Leader-Side Function Approximation

The leader's **value function** is **estimated** in an **RKHS** $\mathcal{H}$ over $\mathcal{S} \times \mathcal{A} \times \Delta(\mathcal{Z})$.
Complexity is controlled by the **maximal information gain**:

$$G_K(T, \lambda) := \sup_{D \subseteq \mathcal{Z}_{\mathrm{lf}},\ |D| \leq T} \frac{1}{2} \log \det \left(I + K_D/\lambda\right).$$

- Finite-rank case: $G_K(T, \lambda) = \widetilde{\mathcal{O}}(d_L)$
- General kernels: regret depends on information gain not directly on $|\mathcal{S}|$

Optimistic Value Iteration (OVI)

- **Goal:** Learn the value of actions while exploring uncertain regions

Optimistic Value Iteration (OVI)

- **Goal:** Learn the value of actions while exploring uncertain regions
- **Key idea:**
  - Estimate value using data (regression)

Optimistic Value Iteration (OVI)

- **Goal:** Learn the value of actions while exploring uncertain regions
- **Key idea:**
  - Estimate value using data (regression)
  - Add a **bonus** for uncertainty

Optimistic Value Iteration (OVI)

- **Goal:** Learn the value of actions while exploring uncertain regions
- **Key idea:**
  - Estimate value using data (regression)
  - Add a **bonus** for uncertainty
  - Act optimistically: value estimate + uncertainty bonus

Optimistic Value Iteration (OVI)

- **Goal:** Learn the value of actions while exploring uncertain regions
- **Key idea:**
  - Estimate value using data (regression)
  - Add a **bonus** for uncertainty
  - Act optimistically: value estimate + uncertainty bonus
- **At each episode:**
  1. Fit value functions from past data

Optimistic Value Iteration (OVI)

- **Goal:** Learn the value of actions while exploring uncertain regions
- **Key idea:**
  - Estimate value using data (regression)
  - Add a **bonus** for uncertainty
  - Act optimistically: value estimate + uncertainty bonus
- **At each episode:**
  1. Fit value functions from past data
  2. Add optimism bonus

Optimistic Value Iteration (OVI)

- **Goal:** Learn the value of actions while exploring uncertain regions
- **Key idea:**
  - Estimate value using data (regression)
  - Add a **bonus** for uncertainty
  - Act optimistically: value estimate + uncertainty bonus
- **At each episode:**
  1. Fit value functions from past data
  2. Add optimism bonus
  3. Choose actions that maximize optimistic value

Optimistic Value Iteration (OVI)

- **Goal:** Learn the value of actions while exploring uncertain regions
- **Key idea:**
  - Estimate value using data (regression)
  - Add a **bonus** for uncertainty
  - Act optimistically: value estimate + uncertainty bonus
- **At each episode:**
  1. Fit value functions from past data
  2. Add optimism bonus
  3. Choose actions that maximize optimistic value
- **Effectively it:**
  - Encourages exploration of uncertain states

Optimistic Value Iteration (OVI)

- **Goal:** Learn the value of actions while exploring uncertain regions
- **Key idea:**
  - Estimate value using data (regression)
  - Add a **bonus** for uncertainty
  - Act optimistically: value estimate + uncertainty bonus
- **At each episode:**
  1. Fit value functions from past data
  2. Add optimism bonus
  3. Choose actions that maximize optimistic value
- **Effectively it:**
  - Encourages exploration of uncertain states
  - Leads to efficient learning (low regret)

OVI: Data and Bellman Targets

- At past episode $\tau < t$, step h, the **leader observes**:

OVI: Data and Bellman Targets

- At past episode $\tau < t$, step h, the **leader observes**:
  - State: s_h^τ

OVI: Data and Bellman Targets

- At past episode $\tau < t$, step h, the **leader observes**:
  - State: s_h^τ
  - Action: a_h^τ

OVI: Data and Bellman Targets

- At past episode $\tau < t$, step h, the **leader observes**:
  - State: s_h^τ
  - Action: a_h^τ
  - Population (empirical): $\widehat{P}_h^\tau$

OVI: Data and Bellman Targets

- At past episode $\tau < t$, step h, the **leader observes**:
  - State: s_h^τ
  - Action: a_h^τ
  - Population (empirical): $\widehat{P}_h^\tau$
  - Reward: u_h^τ

OVI: Data and Bellman Targets

- At past episode $\tau < t$, step h, the **leader observes**:
  - State: s_h^τ
  - Action: a_h^τ
  - Population (empirical): $\widehat{P}_h^\tau$
  - Reward: u_h^τ
  - Next state: $(s_{h+1}^\tau, \mu_{h+1}^\tau)$

OVI: Data and Bellman Targets

- At past episode $\tau < t$, step h, the **leader observes**:
  - State: s_h^τ
  - Action: a_h^τ
  - Population (empirical): $\widehat{P}_h^\tau$
  - Reward: u_h^τ
  - Next state: $(s_{h+1}^\tau, \mu_{h+1}^\tau)$
- Regression input:

$$z_h^\tau = (s_h^\tau, a_h^\tau, \widehat{P}_h^\tau)$$

OVI: Data and Bellman Targets

- At past episode $\tau < t$, step h, the **leader observes**:
  - State: s_h^τ
  - Action: a_h^τ
  - Population (empirical): $\widehat{P}_h^\tau$
  - Reward: u_h^τ
  - Next state: $(s_{h+1}^\tau, \mu_{h+1}^\tau)$
- Regression input:

$$z_h^\tau = (s_h^\tau, a_h^\tau, \widehat{P}_h^\tau)$$

- Regression target (**Bellman backup**):

$$y_h^\tau = u_h^\tau + \widehat{W}_{h+1}^t(s_{h+1}^\tau, \mu_{h+1}^\tau)$$

$\widehat{W}^t$ is the current estimate of the **value function**

Timeline

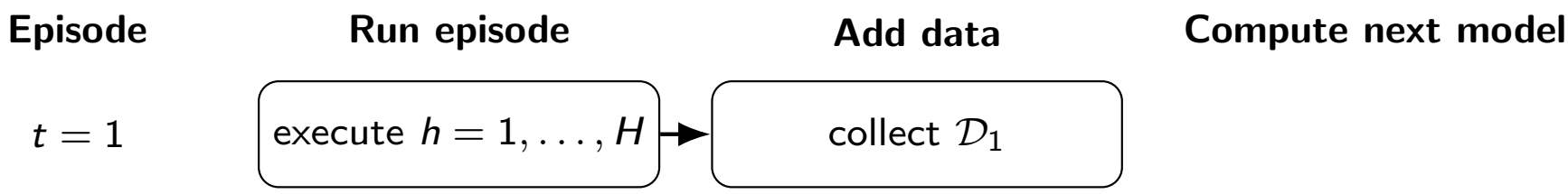
Episode	Run episode	Add data	Compute next model
$t = 1$	execute $h = 1, \ldots, H$		

At the start of episode t, the value functions $\widehat{W}_h^t$ are computed using only past data

$$\mathcal{D}_{t-1} = \{(z_h^\tau, y_h^\tau) : \tau < t,\ h = 1, \ldots, H\}.$$

In episode 1, use an optimistic initialization $\widehat{W}_h^1(s, \mu) = H - h + 1$.

Timeline

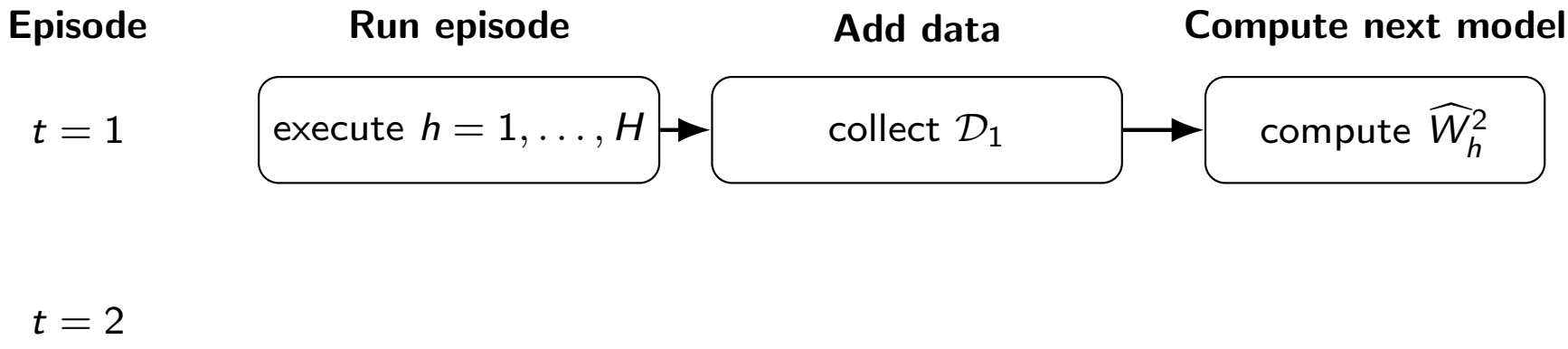


At the start of episode t, the value functions $\widehat{W}_h^t$ are computed using only past data

$$\mathcal{D}_{t-1} = \{(z_h^\tau, y_h^\tau) : \tau < t,\ h = 1, \ldots, H\}.$$

In episode 1, use an optimistic initialization $\widehat{W}_h^1(s, \mu) = H - h + 1$.

Timeline

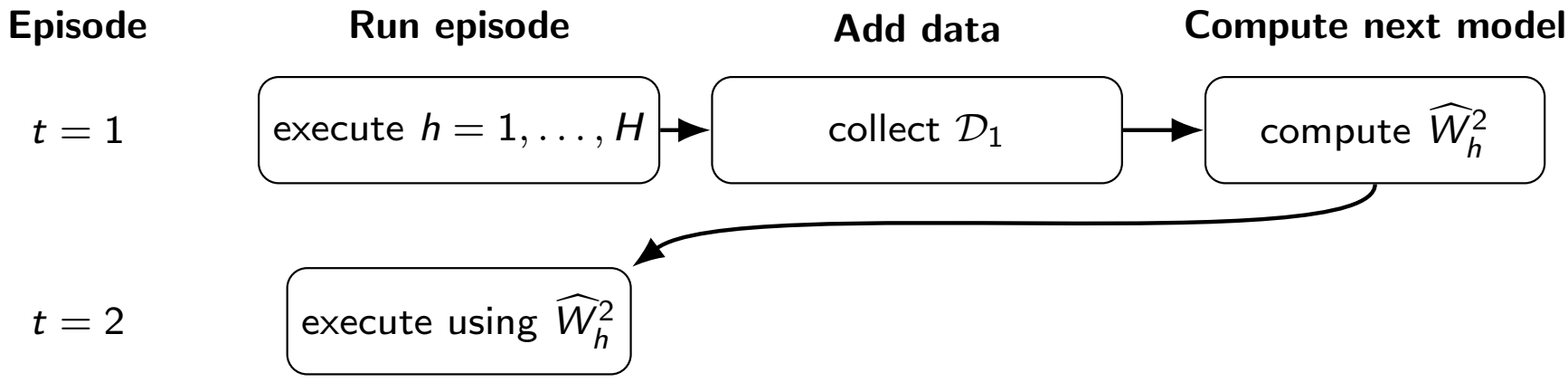


At the start of episode t, the value functions $\widehat{W}_h^t$ are computed using only past data

$$\mathcal{D}_{t-1} = \{(z_h^\tau, y_h^\tau) : \tau < t,\ h = 1, \ldots, H\}.$$

In episode 1, use an optimistic initialization $\widehat{W}_h^1(s, \mu) = H - h + 1$.

Timeline

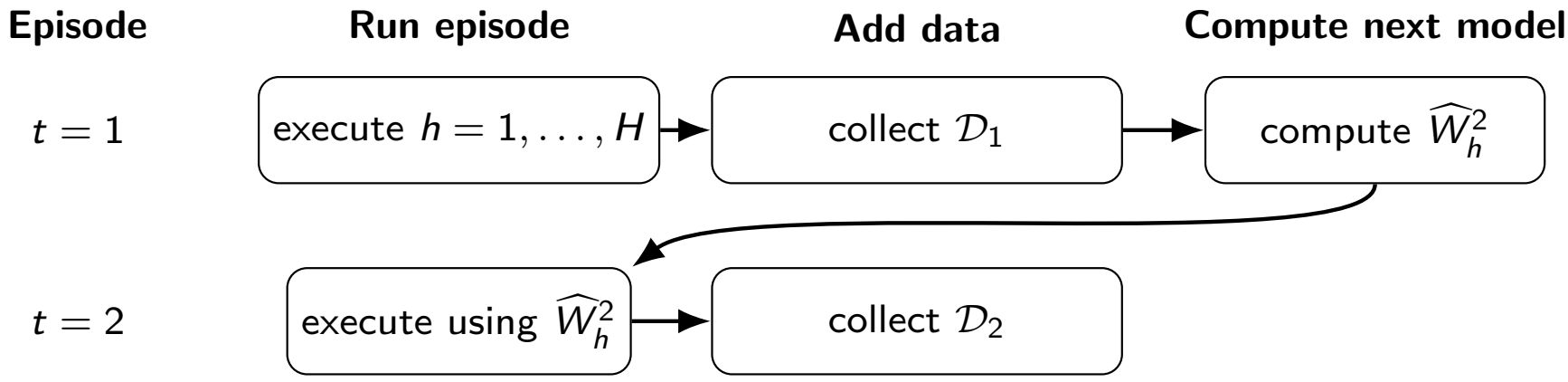


At the start of episode t, the value functions $\widehat{W}_h^t$ are computed using only past data

$$\mathcal{D}_{t-1} = \{(z_h^\tau, y_h^\tau) : \tau < t,\ h = 1, \ldots, H\}.$$

In episode 1, use an optimistic initialization $\widehat{W}_h^1(s, \mu) = H - h + 1$.

Timeline

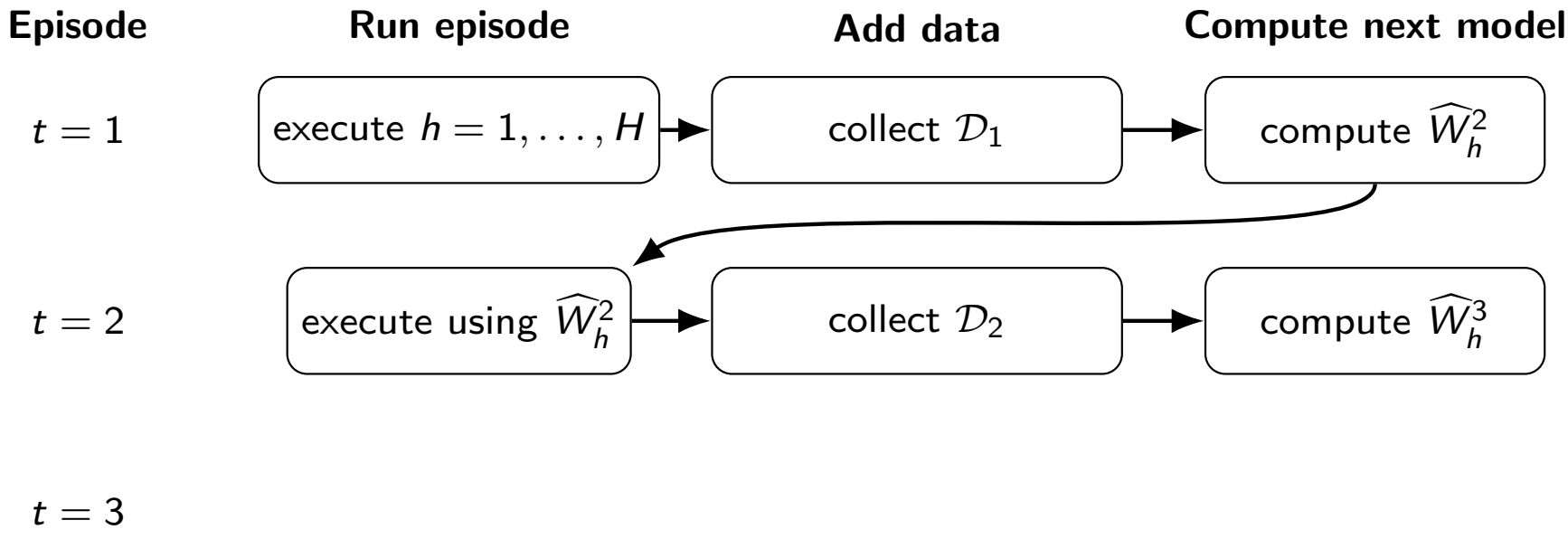


At the start of episode t, the value functions $\widehat{W}_h^t$ are computed using only past data

$$\mathcal{D}_{t-1} = \{(z_h^\tau, y_h^\tau) : \tau < t,\ h = 1, \ldots, H\}.$$

In episode 1, use an optimistic initialization $\widehat{W}_h^1(s, \mu) = H - h + 1$.

Timeline

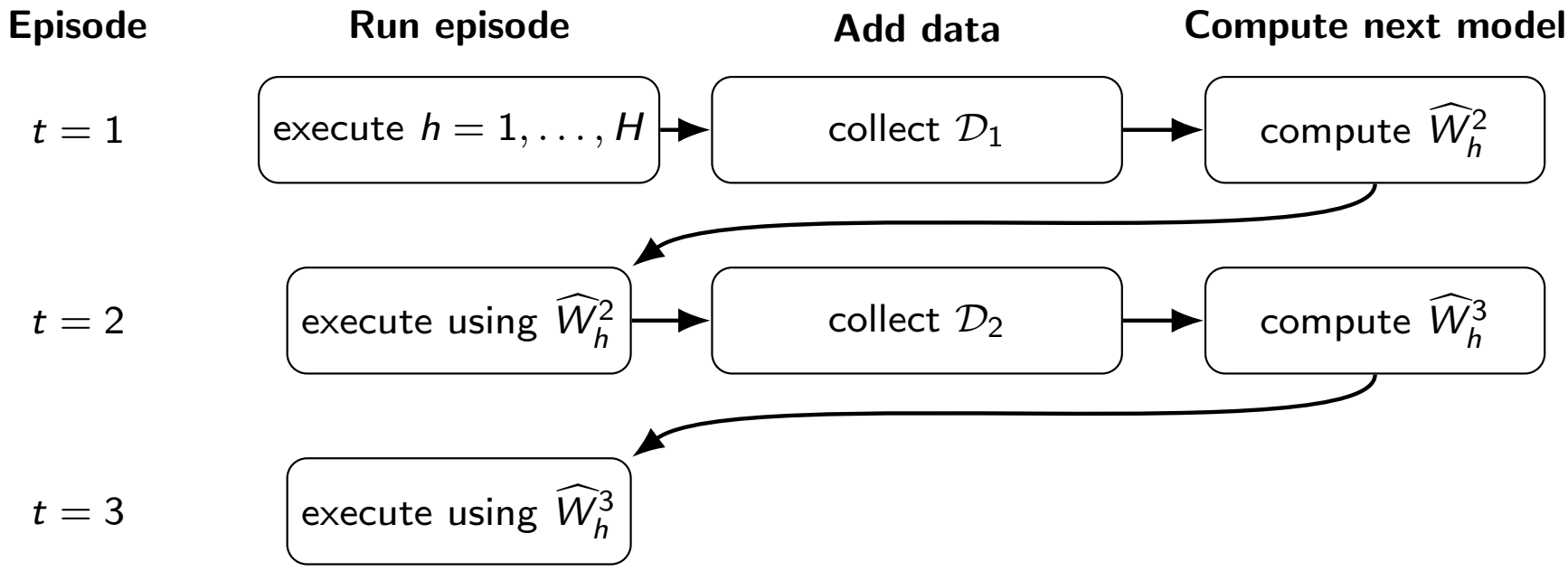


At the start of episode t, the value functions $\widehat{W}_h^t$ are computed using only past data

$$\mathcal{D}_{t-1} = \{(z_h^\tau, y_h^\tau) : \tau < t,\ h = 1, \ldots, H\}.$$

In episode 1, use an optimistic initialization $\widehat{W}_h^1(s, \mu) = H - h + 1$.

Timeline

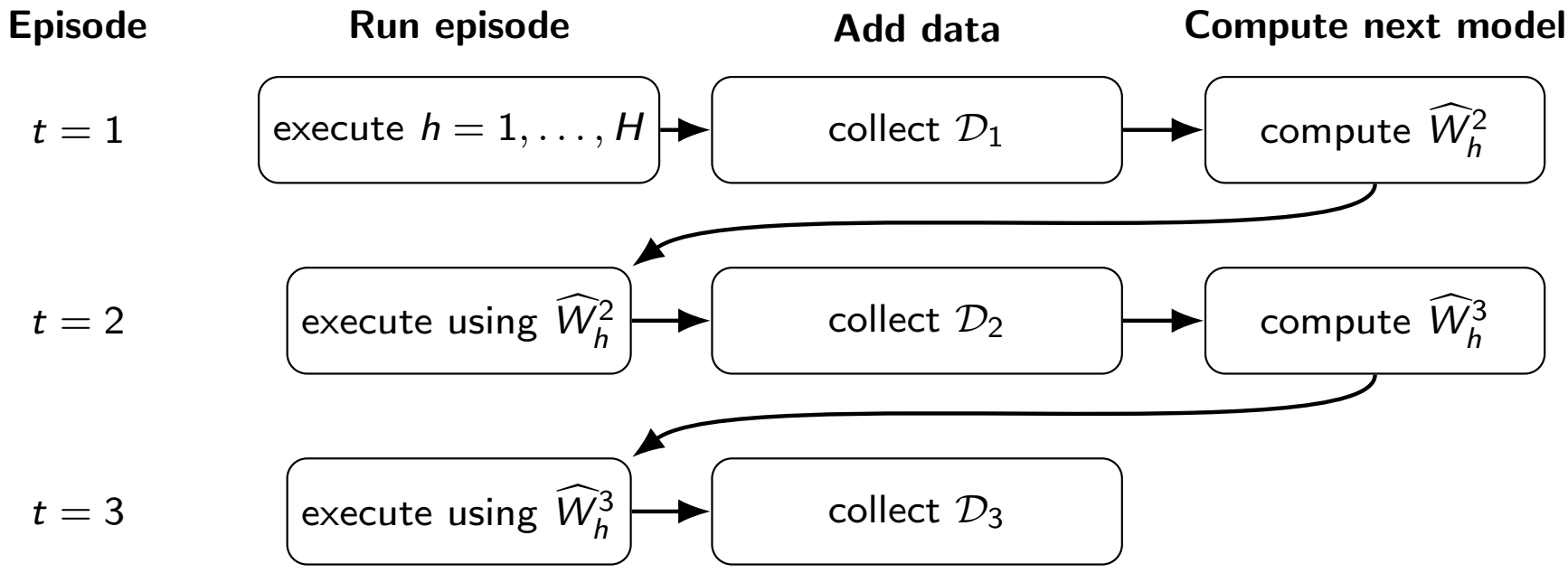


At the start of episode t, the value functions $\widehat{W}_h^t$ are computed using only past data

$$\mathcal{D}_{t-1} = \{(z_h^\tau, y_h^\tau) : \tau < t,\ h = 1, \ldots, H\}.$$

In episode 1, use an optimistic initialization $\widehat{W}_h^1(s, \mu) = H - h + 1$.

Timeline

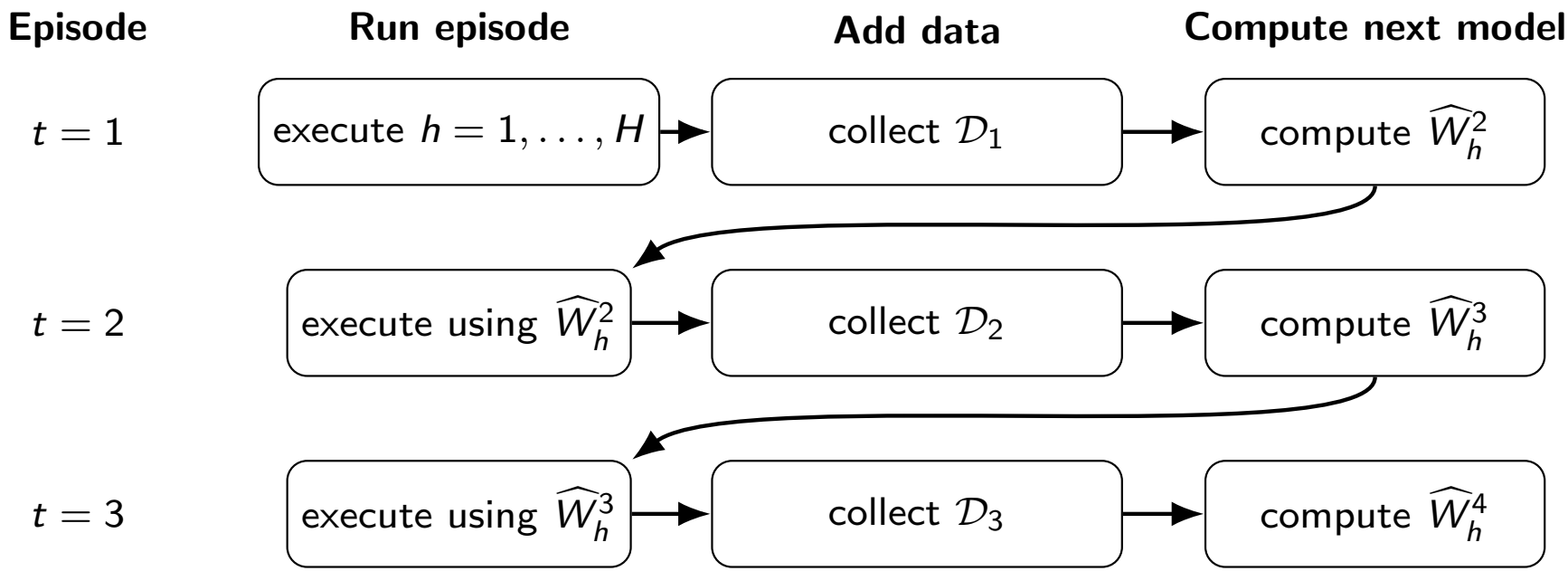


At the start of episode t, the value functions $\widehat{W}_h^t$ are computed using only past data

$$\mathcal{D}_{t-1} = \{(z_h^\tau, y_h^\tau) : \tau < t,\ h = 1, \ldots, H\}.$$

In episode 1, use an optimistic initialization $\widehat{W}_h^1(s, \mu) = H - h + 1$.

Timeline



At the start of episode t, the value functions $\widehat{W}_h^t$ are computed using only past data

$$\mathcal{D}_{t-1} = \{(z_h^\tau, y_h^\tau) : \tau < t,\ h = 1, \ldots, H\}.$$

In episode 1, use an optimistic initialization $\widehat{W}_h^1(s, \mu) = H - h + 1$.

OVI: Learning Value Functions

- Step 1: Fit Bellman backup (**kernel regression**)

$$\widehat{f}_h^t \in \arg\min_{f \in \mathcal{H}} \sum_{\tau < t} \left(y_h^\tau - f(z_h^\tau)\right)^2 + \lambda \|f\|_{\mathcal{H}}^2$$

OVI: Learning Value Functions

- Step 1: Fit Bellman backup (**kernel regression**)

$$\widehat{f}_h^t \in \arg\min_{f \in \mathcal{H}} \sum_{\tau < t} \left(y_h^\tau - f(z_h^\tau)\right)^2 + \lambda \|f\|_{\mathcal{H}}^2$$

- Step 2: Add optimism (**uncertainty bonus**)

$$\widehat{U}_h^t(z) = \text{clip}_{[0, H-h+1]} \left(\widehat{f}_h^t(z) + \Gamma_h^t(z) \right)$$

$$\Gamma_h^t(z) = \beta \, \|\phi(z)\|_{(\Lambda_{L,h}^t)^{-1}}$$

OVI: Learning Value Functions

- Step 1: Fit Bellman backup (**kernel regression**)

$$\widehat{f}_h^t \in \arg\min_{f \in \mathcal{H}} \sum_{\tau < t} (y_h^\tau - f(z_h^\tau))^2 + \lambda \|f\|_{\mathcal{H}}^2$$

- Step 2: Add optimism (**uncertainty bonus**)

$$\widehat{U}_h^t(z) = \text{clip}_{[0, H-h+1]} \left(\widehat{f}_h^t(z) + \Gamma_h^t(z) \right)$$

$$\Gamma_h^t(z) = \beta \, \|\phi(z)\|_{(\Lambda_{L,h}^t)^{-1}}$$

where the **design matrix** (from past data) is

$$\Lambda_{L,h}^t = \lambda I + \sum_{\tau < t} \phi(z_h^\tau) \phi(z_h^\tau)^\top$$

OVI: Learning Value Functions

- Step 1: Fit Bellman backup (**kernel regression**)

$$\widehat{f}_h^t \in \arg\min_{f \in \mathcal{H}} \sum_{\tau < t} (y_h^\tau - f(z_h^\tau))^2 + \lambda \|f\|_{\mathcal{H}}^2$$

- Step 2: Add optimism (**uncertainty bonus**)

$$\widehat{U}_h^t(z) = \text{clip}_{[0, H-h+1]} \left(\widehat{f}_h^t(z) + \Gamma_h^t(z) \right)$$

$$\Gamma_h^t(z) = \beta \, \|\phi(z)\|_{(\Lambda_{L,h}^t)^{-1}}$$

where the **design matrix** (from past data) is

$$\Lambda_{L,h}^t = \lambda I + \sum_{\tau < t} \phi(z_h^\tau) \phi(z_h^\tau)^\top$$

- Step 3: update the **value function**

$$\widehat{W}_h^t(s, \mu) = \max_{p \in \Delta(A)} \mathbb{E}_{a \sim p} \left[\widehat{U}_h^t(s, a, \mu \otimes \nu^p) \right]$$

OVI-Finite Sample Observation Algorithm

1. Initialize dataset $\mathcal{D}_0 = \varnothing$ and MLEs $\widehat{\theta}_h^0$.

OVI-Finite Sample Observation Algorithm

1. Initialize dataset $\mathcal{D}_0 = \varnothing$ and MLEs $\widehat{\theta}_h^0$.
2. For each episode $t = 1, \ldots, T$:

OVI-Finite Sample Observation Algorithm

1. Initialize dataset $\mathcal{D}_0 = \varnothing$ and MLEs $\widehat{\theta}_h^0$.
2. For each episode $t = 1, \ldots, T$:
   2.1 Backward planning: compute optimistic value functions.

OVI-Finite Sample Observation Algorithm

1. Initialize dataset $\mathcal{D}_0 = \varnothing$ and MLEs $\widehat{\theta}_h^0$.
2. For each episode $t = 1, \ldots, T$:
   - 2.1 Backward planning: compute optimistic value functions.
   - 2.2 Forward execution: sample followers, act, observe responses, update data.

OVI-Finite Sample Observation Algorithm

1. Initialize dataset $\mathcal{D}_0 = \varnothing$ and MLEs $\widehat{\theta}_h^0$.
2. For each episode $t = 1, \ldots, T$:
   - 2.1 Backward planning: compute optimistic value functions.
   - 2.2 Forward execution: sample followers, act, observe responses, update data.
   - 2.3 Refit follower MLE and confidence ellipsoid.

OVI-FSO: Backward Planning Phase

For $(h = H, H-1, \ldots, 1)$:

1. Estimate Bellman backup (leader-side)

OVI-FSO: Backward Planning Phase

For $(h = H, H-1, \ldots, 1)$:

1. Estimate Bellman backup (leader-side)

   $\widehat{f}_h^t$ via kernel regression on past data $\mathcal{D}^{t-1}$.

OVI-FSO: Backward Planning Phase

For $(h = H, H-1, \ldots, 1)$:

1. Estimate Bellman backup (leader-side)

   $\widehat{f}_h^t$ via kernel regression on past data $\mathcal{D}^{t-1}$.

2. Add optimism (exploration)

OVI-FSO: Backward Planning Phase

For $(h = H, H-1, \ldots, 1)$:

1. Estimate Bellman backup (leader-side)

$$\widehat{f}_h^t \text{ via kernel regression on past data } \mathcal{D}^{t-1}.$$

2. Add optimism (exploration)

$$\Gamma_h^t(z) = \beta \|\phi(z)\|_{(\Lambda_{L,h}^t)^{-1}}.$$

OVI-FSO: Backward Planning Phase

For $(h = H, H-1, \ldots, 1)$:

1. Estimate Bellman backup (leader-side)

$$\widehat{f}_h^t \text{ via kernel regression on past data } \mathcal{D}^{t-1}.$$

2. Add optimism (exploration)

$$\Gamma_h^t(z) = \beta \|\phi(z)\|_{(\Lambda_{L,h}^t)^{-1}}.$$

3. Account for sampling noise (FSO setting) by adding an extra term

OVI-FSO: Backward Planning Phase

For $(h = H, H-1, \ldots, 1)$:

1. Estimate Bellman backup (leader-side)

$$\widehat{f}_h^t \text{ via kernel regression on past data } \mathcal{D}^{t-1}.$$

2. Add optimism (exploration)

$$\Gamma_h^t(z) = \beta \|\phi(z)\|_{(\Lambda_{L,h}^t)^{-1}}.$$

3. Account for sampling noise (FSO setting) by adding an extra term

$$\Gamma_{h,\mathrm{FSO}}^t := C_{\widehat{V}}(H-h+1)\epsilon_N, \qquad \epsilon_N = \widetilde{\mathcal{O}}\left(\sqrt{\frac{|\mathcal{Z}|}{N}}\right).$$

OVI-FSO: Backward Planning Phase

For $(h = H, H-1, \ldots, 1)$:

1. Estimate Bellman backup (leader-side)

$$\widehat{f}_h^t \text{ via kernel regression on past data } \mathcal{D}^{t-1}.$$

2. Add optimism (exploration)

$$\Gamma_h^t(z) = \beta \|\phi(z)\|_{(\Lambda_{L,h}^t)^{-1}}.$$

3. Account for sampling noise (FSO setting) by adding an extra term

$$\Gamma_{h,\mathrm{FSO}}^t := C_{\widehat{V}}(H-h+1)\epsilon_N, \qquad \epsilon_N = \widetilde{\mathcal{O}}\left(\sqrt{\frac{|\mathcal{Z}|}{N}}\right).$$

so that

$$\widehat{U}_h^t(z) = \mathrm{clip}_{[0,H-h+1]}\left(\widehat{f}_h^t(z) + \Gamma_h^t(z) + \Gamma_{h,\mathrm{FSO}}^t\right)$$

OVI-FSO: Forward Execution Phase

At stage h of episode t:

OVI-FSO: Forward Execution Phase

At stage h of episode t:

1. Observe N follower states and form

$$\widehat{\mu}_h^t = \frac{1}{N} \sum_{n=1}^{N} \delta_{X_h^{t,n}}.$$

OVI-FSO: Forward Execution Phase

At stage h of episode t:

1. Observe N follower states and form

$$\widehat{\mu}_h^t = \frac{1}{N} \sum_{n=1}^{N} \delta_{X_h^{t,n}}.$$

2. Choose optimistic local policy and follower witness:

$$(p_h^t, \theta_h^t) \in \underset{p \in \mathcal{P}_A,\ \theta \in \Theta_h^{t-1}}{\operatorname{argmax}} \left\langle \widehat{U}_h^t\big(s_h^t, \cdot, \widehat{\mu}_h^t \otimes \nu_h^{p,\theta}\big), p \right\rangle.$$

OVI-FSO: Forward Execution Phase

At stage h of episode t:

1. Observe N follower states and form

$$\widehat{\mu}_h^t = \frac{1}{N} \sum_{n=1}^{N} \delta_{X_h^{t,n}}.$$

2. Choose optimistic local policy and follower witness:

$$(p_h^t, \theta_h^t) \in \underset{p \in \mathcal{P}_A,\ \theta \in \Theta_h^{t-1}}{\operatorname{argmax}} \left\langle \widehat{U}_h^t\big(s_h^t, \cdot, \widehat{\mu}_h^t \otimes \nu_h^{p,\theta}\big), p \right\rangle.$$

3. Followers respond; construct paired empirical joint distribution:

$$\widehat{P}_h^t = \frac{1}{N} \sum_{n=1}^{N} \delta_{(X_h^{t,n}, B_h^{t,n})}.$$

OVI-FSO: Forward Execution Phase

At stage h of episode t:

1. Observe N follower states and form

$$\widehat{\mu}_h^t = \frac{1}{N} \sum_{n=1}^{N} \delta_{X_h^{t,n}}.$$

2. Choose optimistic local policy and follower witness:

$$(p_h^t, \theta_h^t) \in \operatorname*{argmax}_{p \in \mathcal{P}_A,\ \theta \in \Theta_h^{t-1}} \left\langle \widehat{U}_h^t\big(s_h^t, \cdot, \widehat{\mu}_h^t \otimes \nu_h^{p,\theta}\big), p \right\rangle.$$

3. Followers respond; construct paired empirical joint distribution:

$$\widehat{P}_h^t = \frac{1}{N} \sum_{n=1}^{N} \delta_{(X_h^{t,n}, B_h^{t,n})}.$$

4. Store transition and response samples in $\mathcal{D}_t$.

Follower MLE in OVI-FSO

- The follower parameter is estimated from observed pairs $(X_h^{\tau,n}, B_h^{\tau,n})$

Follower MLE in OVI-FSO

- The follower parameter is estimated from observed pairs $(X_h^{\tau,n}, B_h^{\tau,n})$
- Using the likelihood

$$\mathcal{L}_h^t(\theta) = -\sum_{\tau<t}\sum_{n=1}^{N} \log \nu_h^{p_h^\tau,\theta}\left(B_h^{\tau,n} \mid X_h^{\tau,n}; s_h^\tau\right)$$

Follower MLE in OVI-FSO

- The follower parameter is estimated from observed pairs $(X_h^{\tau,n}, B_h^{\tau,n})$
- Using the likelihood

$$\mathcal{L}_h^t(\theta) = -\sum_{\tau<t}\sum_{n=1}^{N} \log \nu_h^{p_h^\tau,\theta}\left(B_h^{\tau,n} \mid X_h^{\tau,n}; s_h^\tau\right)$$

- The MLE is

$$\widehat{\theta}_h^t \in \operatorname*{argmin}_{\theta\in\Theta} \mathcal{L}_h^t(\theta)$$

Follower MLE in OVI-FSO

- The follower parameter is estimated from observed pairs $(X_h^{\tau,n}, B_h^{\tau,n})$
- Using the likelihood

$$\mathcal{L}_h^t(\theta) = -\sum_{\tau<t}\sum_{n=1}^{N} \log \nu_h^{p_h^\tau,\theta}\left(B_h^{\tau,n} \mid X_h^{\tau,n}; s_h^\tau\right)$$

- The MLE is

$$\widehat{\theta}_h^t \in \operatorname*{argmin}_{\theta\in\Theta} \mathcal{L}_h^t(\theta)$$

- The resulting empirical Fisher matrix is

$$\widehat{\Lambda}_h^t = \lambda_\theta I_d + \sum_{\tau<t}\sum_{n=1}^{N} \Sigma^{p_h^\tau,\widehat{\theta}_h^t}_{X_h^{\tau,n},s_h^\tau}$$

Follower MLE in OVI-FSO

- The follower parameter is estimated from observed pairs $(X_h^{\tau,n}, B_h^{\tau,n})$
- Using the likelihood

$$\mathcal{L}_h^t(\theta) = -\sum_{\tau<t}\sum_{n=1}^{N} \log \nu_h^{p_h^\tau,\theta}\left(B_h^{\tau,n} \mid X_h^{\tau,n}; s_h^\tau\right)$$

- The MLE is

$$\widehat{\theta}_h^t \in \operatorname*{argmin}_{\theta\in\Theta} \mathcal{L}_h^t(\theta)$$

- The resulting empirical Fisher matrix is

$$\widehat{\Lambda}_h^t = \lambda_\theta I_d + \sum_{\tau<t}\sum_{n=1}^{N} \Sigma^{p_h^\tau,\widehat{\theta}_h^t}_{X_h^{\tau,n},s_h^\tau}$$

(The MLE is multinomial logistic regression applied to the followers' observed actions under the quantal-response model)

Empirical Concentration in FSO

Under paired adaptive sampling, with high probability,

$$\left\|\widehat{\mu}_h^t - \mu_h^t\right\|_1 \leq \epsilon_{\mu,N}, \qquad \left\|\widehat{P}_h^t - P_h^t\right\|_1 \leq \epsilon_N$$

for all (t, h), where

$$\epsilon_N = \epsilon_{\mu,N} + \epsilon_{\mathrm{act},N} = \widetilde{\mathcal{O}}\left(\sqrt{\frac{|\mathcal{Z}|}{N}}\right).$$

The empirical concentration inequality itself is standard; however, we handle adaptive paired sampling and using a "TV bridge" argument to prevent finite-sample observation noise from degrading the regret rate.

Regret Decomposition for OVI-FSO

- Main **regret bound**

$$\mathrm{Reg}_{\mathrm{FSO}}(T) \leq \widetilde{\mathcal{O}}\left(\left(G_K + d\sqrt{G_K}\right) H^2\sqrt{T} + G_K H^3 T\sqrt{\frac{|\mathcal{Z}|}{N}}\right).$$

Regret Decomposition for OVI-FSO

- Main **regret bound**

$$\mathrm{Reg}_{\mathrm{FSO}}(T) \leq \widetilde{\mathcal{O}}\left(\left(G_K + d\sqrt{G_K}\right) H^2\sqrt{T} + G_K H^3 T\sqrt{\frac{|\mathcal{Z}|}{N}}\right).$$

- Interpretation

$$\underbrace{G_K H^2\sqrt{T}}$$

Leader-side RKHS learning

  - Learning rewards and dynamics
  - Controlled by kernel information gain G_K

Regret Decomposition for OVI-FSO

- Main **regret bound**

$$\mathrm{Reg}_{\mathrm{FSO}}(T) \leq \widetilde{\mathcal{O}}\left(\left(G_K + d\sqrt{G_K}\right) H^2\sqrt{T} + G_K H^3 T\sqrt{\frac{|\mathcal{Z}|}{N}}\right).$$

- Interpretation

$$\underbrace{d\sqrt{G_K}\, H^2\sqrt{T}}_{\textbf{Follower-model learning}}$$

  - Estimating the quantal-response parameter θ^*
  - d: follower feature dimension

Regret Decomposition for OVI-FSO

- Main **regret bound**

$$\mathrm{Reg}_{\mathrm{FSO}}(T) \leq \widetilde{\mathcal{O}}\left(\left(G_K + d\sqrt{G_K}\right) H^2\sqrt{T} + G_K H^3 T\sqrt{\frac{|\mathcal{Z}|}{N}}\right).$$

- Interpretation

$$\underbrace{G_K H^3 T\sqrt{\frac{|\mathcal{Z}|}{N}}}_{\textbf{Finite-sample observation penalty}}$$

  - Error from observing only sampled followers
  - Vanishes as $N \uparrow$

Regret Decomposition for OVI-FSO

- Main **regret bound**

$$\mathrm{Reg}_{\mathrm{FSO}}(T) \leq \widetilde{\mathcal{O}}\left(\left(G_K + d\sqrt{G_K}\right) H^2\sqrt{T} + G_K H^3 T\sqrt{\frac{|\mathcal{Z}|}{N}}\right).$$

- Key implication: **data rich leads to sub-linear regret**

$$N \gtrsim \widetilde{\Omega}(H^2 T|\mathcal{Z}|) \quad \Longrightarrow \quad \mathrm{Reg}_{\mathrm{FSO}}(T) \approx \widetilde{\mathcal{O}}(\sqrt{T}).$$

Minimax Lower Bound (Complete Observation)

$$\inf_{\mathcal{A}} \sup_{\mathcal{M}} \mathbb{E}_{\mathcal{M}}[\mathrm{Reg}_{\mathcal{A}}(T)] \geq c\, d_L H \sqrt{T}.$$

Minimax Lower Bound (Complete Observation)

$$\inf_{\mathcal{A}} \sup_{\mathcal{M}} \mathbb{E}_{\mathcal{M}}[\mathrm{Reg}_{\mathcal{A}}(T)] \geq c\, d_L H \sqrt{T}.$$

$$\inf_{\mathcal{A}} \quad \Rightarrow \quad \text{best possible learning algorithm}$$

Minimax Lower Bound (Complete Observation)

$$\inf_{\mathcal{A}} \sup_{\mathcal{M}} \mathbb{E}_{\mathcal{M}}[\mathrm{Reg}_{\mathcal{A}}(T)] \geq c\, d_L H \sqrt{T}.$$

$\inf_{\mathcal{A}}$ $\Rightarrow$ best possible learning algorithm

$\sup_{\mathcal{M}}$ $\Rightarrow$ hardest environment / game instance

Minimax Lower Bound (Complete Observation)

$$\inf_{\mathcal{A}} \sup_{\mathcal{M}} \mathbb{E}_{\mathcal{M}}[\mathrm{Reg}_{\mathcal{A}}(T)] \geq c\, d_L H \sqrt{T}.$$

$$\inf_{\mathcal{A}} \quad \Rightarrow \quad \text{best possible learning algorithm}$$

$$\sup_{\mathcal{M}} \quad \Rightarrow \quad \text{hardest environment / game instance}$$

So: *Even the best algorithm suffers at least order $d_L H \sqrt{T}$ regret on some difficult problem instance.*

Minimax Lower Bound (Complete Observation)

$$\inf_{\mathcal{A}} \sup_{\mathcal{M}} \mathbb{E}_{\mathcal{M}}[\mathrm{Reg}_{\mathcal{A}}(T)] \geq c\, d_L H\sqrt{T}.$$

$$\inf_{\mathcal{A}} \quad \Rightarrow \quad \text{best possible learning algorithm}$$

$$\sup_{\mathcal{M}} \quad \Rightarrow \quad \text{hardest environment / game instance}$$

So: *Even the best algorithm suffers at least order $d_L H\sqrt{T}$ regret on some difficult problem instance.*

Thus OVI-FSO is *nearly minimax optimal*.

Numerical Experiment Setup

- Environment:

Numerical Experiment Setup

- Environment:
  - Finite-rank Linear Mean-Field Game

Numerical Experiment Setup

- Environment:
  - Finite-rank Linear Mean-Field Game
  - Synthetic ride-sharing style setting

Numerical Experiment Setup

- Environment:
  - Finite-rank Linear Mean-Field Game
  - Synthetic ride-sharing style setting
  - Leader learns from sampled follower responses

Numerical Experiment Setup

- Environment:
  - Finite-rank Linear Mean-Field Game
  - Synthetic ride-sharing style setting
  - Leader learns from sampled follower responses
- Model dimensions

$$|S| = 4, \qquad |A| = 3, \qquad |X| = 3, \qquad |B| = 2$$

Numerical Experiment Setup

- Environment:
  - Finite-rank Linear Mean-Field Game
  - Synthetic ride-sharing style setting
  - Leader learns from sampled follower responses
- Model dimensions

$$|S| = 4, \quad |A| = 3, \quad |X| = 3, \quad |B| = 2$$
$$H = 3, \quad d_L = 12, \quad d_F = 4, \quad \eta = 0.4$$

Numerical Experiment Setup

- Environment:
  - Finite-rank Linear Mean-Field Game
  - Synthetic ride-sharing style setting
  - Leader learns from sampled follower responses
- Model dimensions

$$|S| = 4, \quad |A| = 3, \quad |X| = 3, \quad |B| = 2$$
$$H = 3, \quad d_L = 12, \quad d_F = 4, \quad \eta = 0.4$$

- Finite-sample observation levels

$$N \in \{20,\ 60,\ 200\}$$

- Evaluation

Numerical Experiment Setup

- Environment:
  - Finite-rank Linear Mean-Field Game
  - Synthetic ride-sharing style setting
  - Leader learns from sampled follower responses
- Model dimensions

$$|S| = 4, \quad |A| = 3, \quad |X| = 3, \quad |B| = 2$$
$$H = 3, \quad d_L = 12, \quad d_F = 4, \quad \eta = 0.4$$

- Finite-sample observation levels

$$N \in \{20,\ 60,\ 200\}$$

- Evaluation
  - Run OVI-FSO over multiple episodes

Numerical Experiment Setup

- Environment:
  - Finite-rank Linear Mean-Field Game
  - Synthetic ride-sharing style setting
  - Leader learns from sampled follower responses
- Model dimensions

$$|S| = 4, \quad |A| = 3, \quad |X| = 3, \quad |B| = 2$$
$$H = 3, \quad d_L = 12, \quad d_F = 4, \quad \eta = 0.4$$

- Finite-sample observation levels

$$N \in \{20,\ 60,\ 200\}$$

- Evaluation
  - Run OVI-FSO over multiple episodes
  - Average cumulative regret over 8 random seeds

Numerical Results

- Main empirical findings

Numerical Results

- Main empirical findings
  1. Regret grows roughly as $\widetilde{\mathcal{O}}(\sqrt{T})$... matching the theory

Numerical Results

- Main empirical findings
  1. Regret grows roughly as $\widetilde{\mathcal{O}}(\sqrt{T})$... matching the theory
  2. Larger sample sizes N reduce regret

Numerical Results

- Main empirical findings
  1. Regret grows roughly as $\widetilde{\mathcal{O}}(\sqrt{T})$... matching the theory
  2. Larger sample sizes N reduce regret
  3. Small N exhibits the predicted finite-sample penalty:

$$\widetilde{\mathcal{O}}\left(T\sqrt{\frac{|Z|}{N}}\right).$$

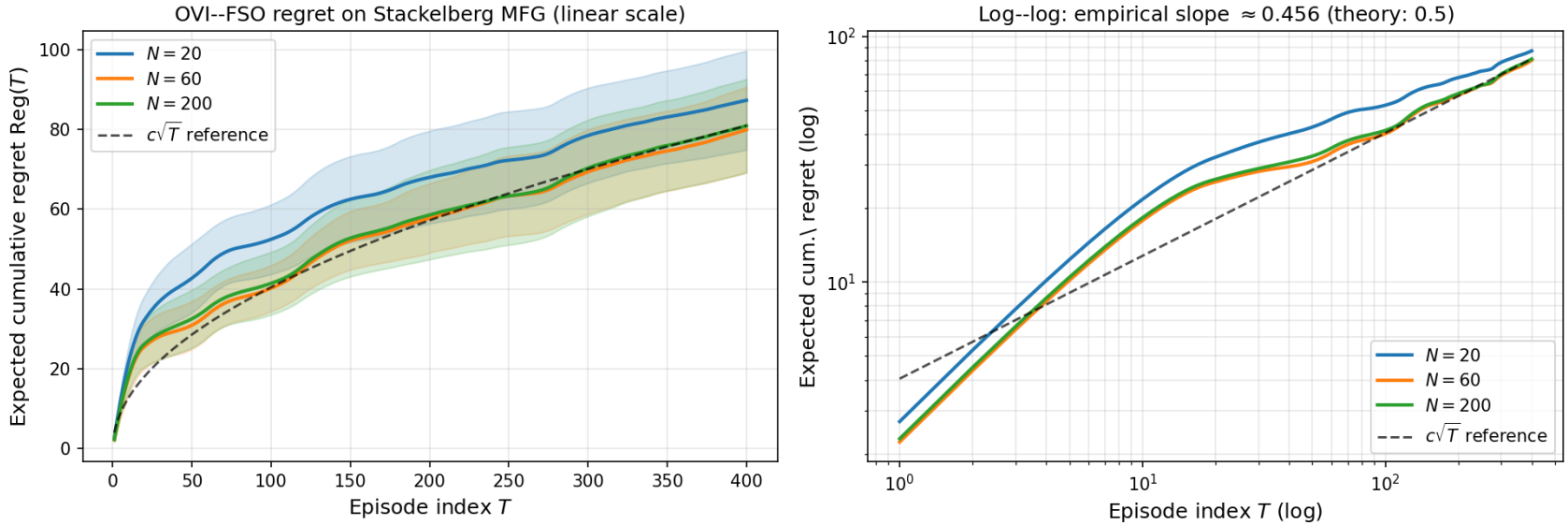
Numerical Results

- Main empirical findings
  1. Regret grows roughly as $\widetilde{\mathcal{O}}(\sqrt{T})$... matching the theory
  2. Larger sample sizes N reduce regret
  3. Small N exhibits the predicted finite-sample penalty:

$$\widetilde{\mathcal{O}}\left(T\sqrt{\frac{|Z|}{N}}\right).$$

- Conclusion

  As $N \uparrow$, OVI-FSO approaches the complete-observation rate.



$|\mathcal{S}| = 4, |\mathcal{A}| = 3, |\mathcal{X}| = 3, |\mathcal{B}| = 2, H = 3, d_L = 12, d_f = 4, \eta = 0.4; \beta = 2.0$; *8 seeds (mean ± s.e.m.); slopes by N: N=20: 0.350, N=60: 0.459, N=200: 0.456*

Thank you...

... Questions?

https://sebastian.statistics.utoronto.ca

Infinite-Dimensional Feature Embeddings (RKHS)

- Example: Gaussian (RBF) kernel. Take $z = (s, a, P)$ and $z' = (s', a', P')$. Define the kernel:

$$K(z, z') = e^{-\frac{1}{2\sigma^2}\left(d_S(s,s')^2 + d_A(a,a')^2 + \|P - P'\|_1^2\right)}$$

- Key idea:
  - The feature embedding $\phi(z)$ is now *infinite-dimensional* – but is not computed explicitly.
  - Instead, we use the kernel identity

$$K(z, z') = \langle \phi(z), \phi(z') \rangle_{\mathcal{H}}.$$

- Example reward model:

$$u_h(s, a, P) = \langle \phi(s, a, P), \xi_h^* \rangle_{\mathcal{H}}.$$

Proof Strategy for OVI-FSO

Goal: Bound the regret

$$\mathrm{Reg}_{\mathrm{FSO}}(T) = \sum_{t=1}^{T} \big(J(\pi^*) - J(\pi^t)\big).$$

Key decomposition

$$\mathrm{Reg}(T) \lesssim \underbrace{\mathrm{LBE}}_{\text{leader Bellman error}} + \underbrace{\mathrm{JDE}}_{\text{joint distribution error}} + \underbrace{\mathrm{FSO\ residual}}_{\text{sampling noise}}.$$

Interpretation

- **LBE**: error from learning rewards/dynamics
- **JDE**: error from estimating follower behavior
- **FSO residual**: error from observing only sampled followers

Main challenge

$$\widehat{P}_h^t \neq P_h^t$$

The leader plans using empirical population observations rather than the true mean field.

Key Proof Ingredients

1. **RKHS Optimism (Leader-side)**

$$\widehat{U}_h^t(z) = \widehat{f}_h^t(z) + \Gamma_h^t(z) + \Gamma_{h,\mathrm{FSO}}^t \qquad \Gamma_h^t(z) = \beta \|\phi_L(z)\|_{(\Lambda_{L,h}^t)^{-1}}$$

   - Controls uncertainty in rewards/dynamics
   - optimistic RL ingredient

2. **MLE Confidence Sets (Follower-side)**

$$\Theta_h^t = \left\{ \theta : \|\theta - \widehat{\theta}_h^t\|_{\widehat{\Lambda}_h^t}^2 \le \beta_\theta^t \right\}$$

   - Learns follower quantal-response parameter
   - Uses empirical Fisher information

3. **Finite-Sample Observation Correction**

$$\Gamma_{h,\mathrm{FSO}}^t = C_{\widehat{V}}(H - h + 1)\epsilon_N \qquad \epsilon_N = \widetilde{\mathcal{O}}\left(\sqrt{\frac{|Z|}{N}}\right)$$

   - Accounts for observing only sampled followers

What is Novel?

Main technical novelty: separating

learning uncertainty from sampling uncertainty.

Novel ingredient #1: TV-bridge argument

$$\|\widehat{P}_h^t - P_h^t\|_1 = \widetilde{\mathcal{O}}\left(\sqrt{\frac{|Z|}{N}}\right)$$

- Converts empirical-population error into value-function error
- Avoids the naive $T^{3/4}$ regret scaling

Novel ingredient #2: Coverage-free MLE analysis

- No persistent excitation assumption
- Uses softmax self-concordance + Fisher concentration

Novel ingredient #3: RKHS + Mean-field coupling

- Leader state/action spaces may be large or infinite
- Complexity controlled by information gain

$$G_K(T, \lambda)$$

Lower Bound Strategy: Hard Family Construction

Goal

$$\inf_{\mathcal{A}} \sup_{\mathcal{M}} \mathbb{E}_{\mathcal{M}}[\mathrm{Reg}_{\mathcal{A}}(T)] \geq c\, d_L H \sqrt{T}.$$

Key idea: Embed a hard linear bandit inside the Stackelberg MFG.

Construction

- Build a family of environments

$$\{\mathcal{M}_v\}_{v \in \{\pm 1\}^{d_L - 3}}$$

  indexed by a hidden binary vector v.

- Each environment changes:
  - leader rewards,
  - leader transitions,

  only slightly.

- Followers are trivialized:

$$|X| = |B| = 2, \qquad d = 1, \qquad \theta_h^* = 0.$$

Purpose

- Isolate intrinsic *leader-side* learning difficulty

Lower Bound Strategy: Information-Theoretic Confusion

Core phenomenon
Different environments:

$$\mathcal{M}_v, \qquad \mathcal{M}_{v'}$$

produce trajectories that are statistically similar.
But:

- optimal actions differ across environments
- identifying the correct environment requires exploration

Information-theoretic argument

$$\mathrm{KL}\big(\mathbb{P}_{\mathcal{M}_v}, \mathbb{P}_{\mathcal{M}_{v'}}\big) \quad \text{is small}$$

Thus:

> *No algorithm can quickly distinguish which environment it is interacting with.*

Consequence
The learner must repeatedly choose suboptimal actions while gathering information.

Lower Bound Strategy: Regret Consequence

Exploration vs. exploitation tradeoff

To identify the hidden direction v:

- the algorithm must explore many action directions
- but exploration incurs regret

Resulting lower bound

$$\mathbb{E}_{\mathcal{M}}[\mathrm{Reg}_{\mathcal{A}}(T)] \geq c\, d_L H \sqrt{T}.$$

Interpretation of the factors

$$\underbrace{d_L}_{\text{leader complexity}} \qquad \underbrace{H}_{\text{planning horizon}} \qquad \underbrace{\sqrt{T}}_{\text{fundamental RL rate}}$$

Main implication

OVI-FSO achieves:

$$\widetilde{\mathcal{O}}(d_L H^2 \sqrt{T}),$$

so the algorithm is:

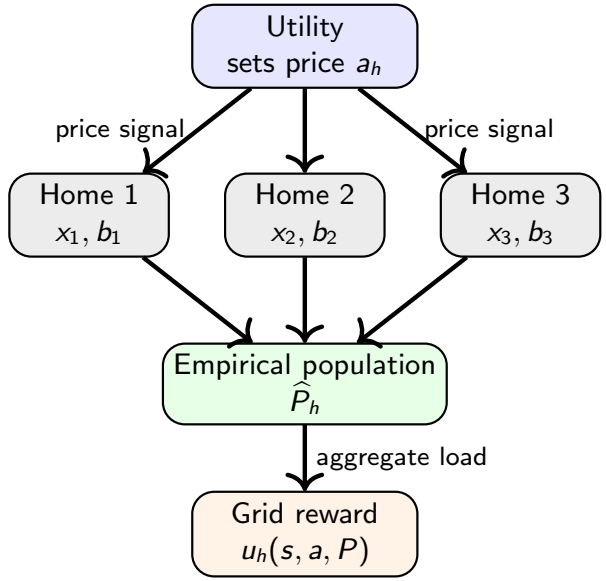
$$\boxed{\text{near minimax optimal}}$$

up to logarithmic factors and one factor of H.

Stylized Example: Smart-Grid Demand Response

Setting

- Utility company manages electricity demand
- Households decide when to consume energy
- Utility observes only sampled smart-meter data



Leader

a_h = dynamic electricity price

Followers

$b_h \in \{\text{consume now, defer}\}$ $\quad$ $x = (\text{temp., time-of-day, EV charging need})$

Goal

- reduce peak load
- maintain user participation
- learn price sensitivity θ^*