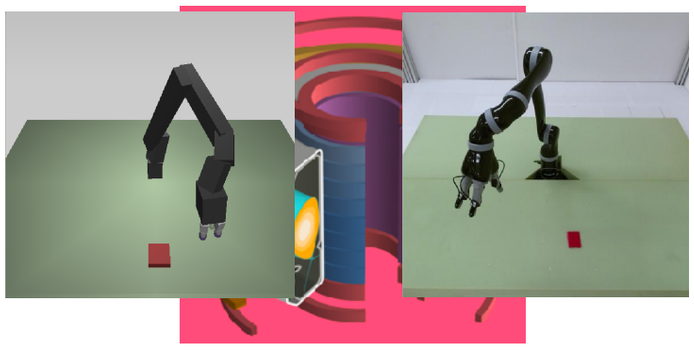
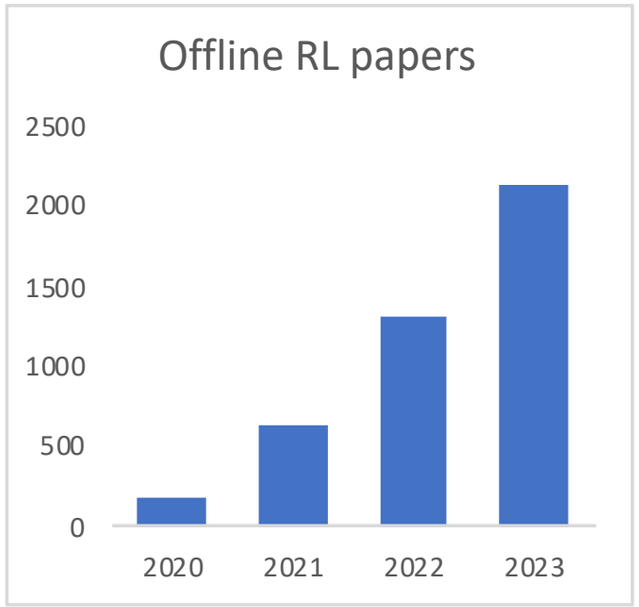
Rethinking the theoretical foundation of reinforcement learning

Nan Jiang
University of Illinois at Urbana-Champaign
April 24, 2025
@Shariff University of Technology

- (Offline) RL in real life



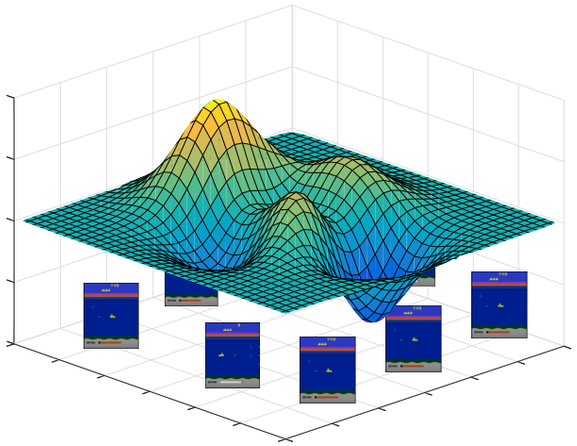


Key ingredient: **simulator**

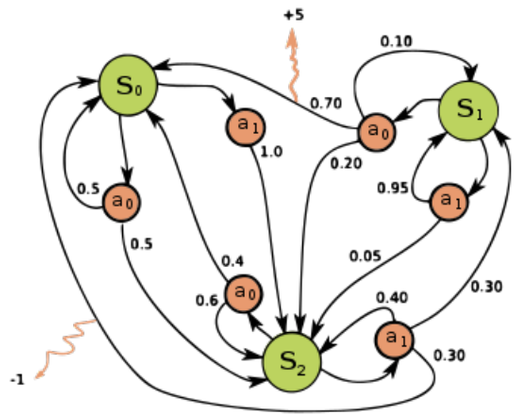
- Unlimited data **X**
- Decision w/o real consequences **X**
- Can easily evaluate new strategy **X**

Why are we not seeing (offline) RL deployed everywhere already?

~2015



Bellman rank, Eluder dimension, Concentrability, …

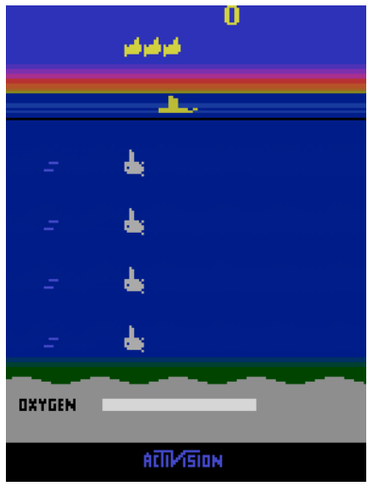


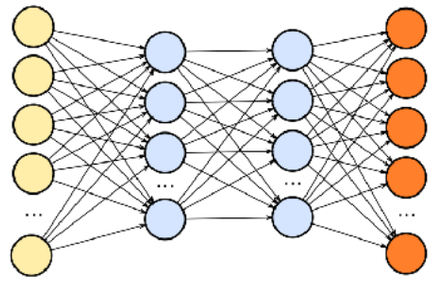
$\sqrt{HSAT}$ *regret,* SAH^2/ϵ^2 *sample complexity, …*

~2000

*

- (Offline) RL in real life
- Role of theory in modern RL





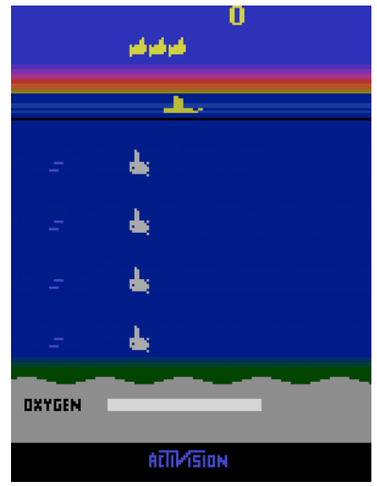
Empirical: *Atari, Mujoco, OpenAI Gym, target network, architecture, …*

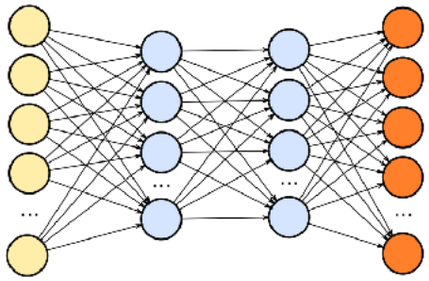
* finite-sample analysis of ADP & MCTS 00~10

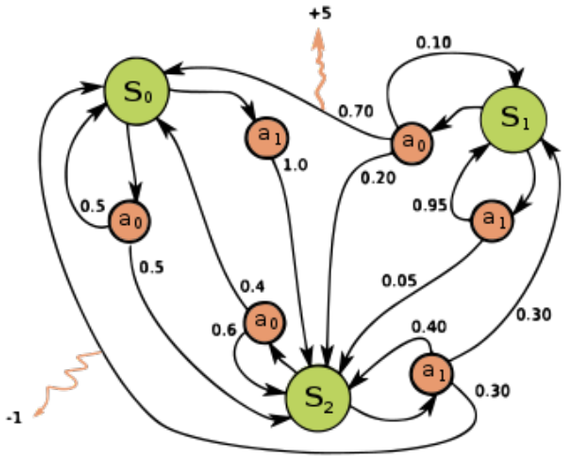
- (Offline) RL in real life
- Role of theory in modern RL
- Theoretical foundation

simplify

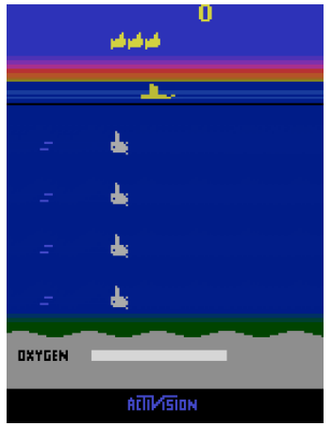
extend

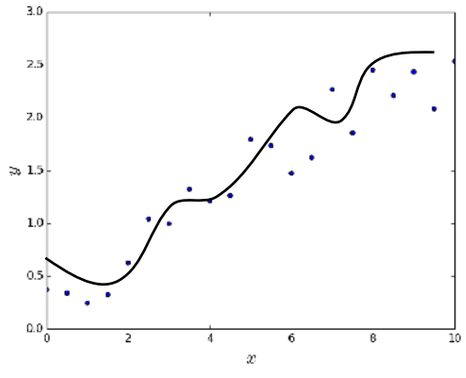


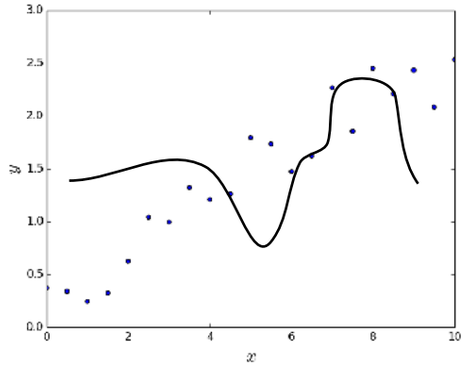




	$Q^*(s, a)$
(s_1, a_1)	…
(s_1, a_2)	…
(s_2, a_1)	…







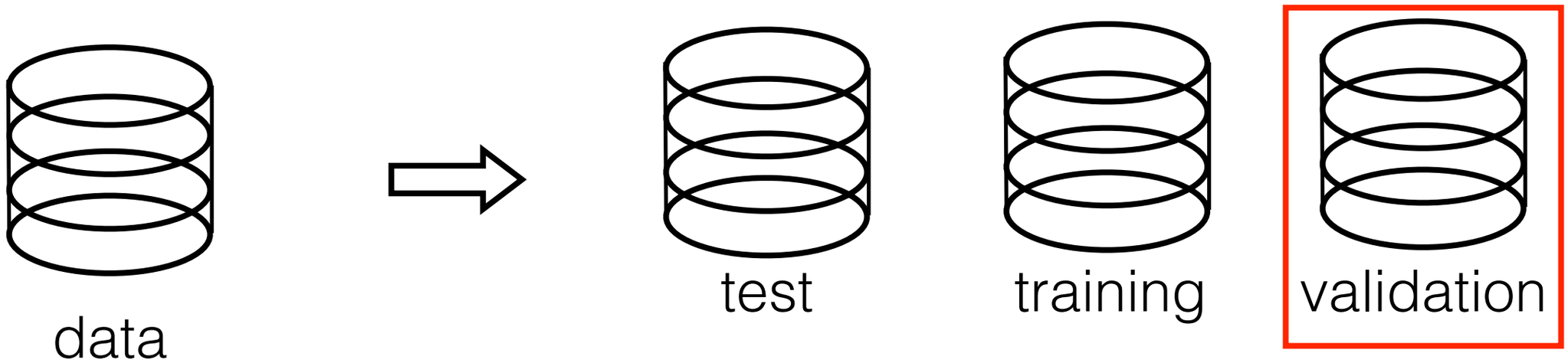
“tabular” RL

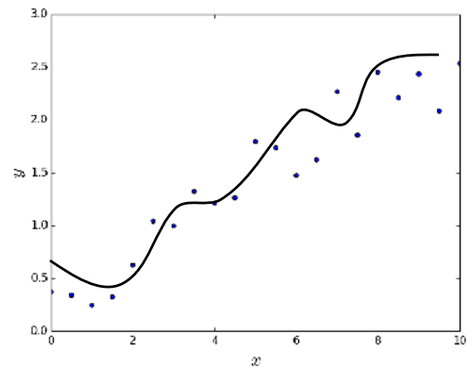
- (Offline) RL in real life
- Role of theory in modern RL
- Theoretical foundation

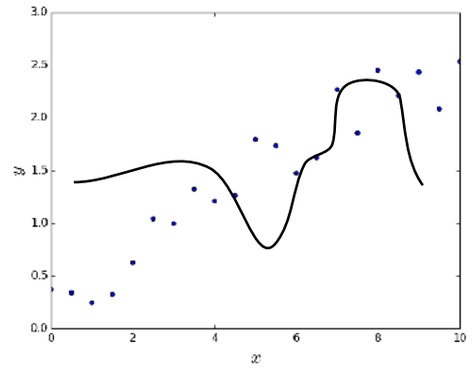
modern RL ⟹ real life

Rethinking
theoretical foundation

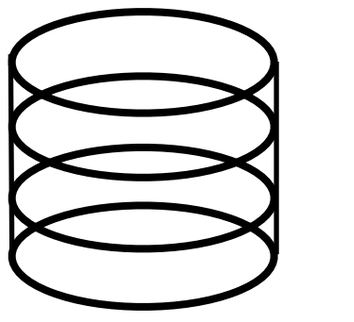
Supervised learning pipeline



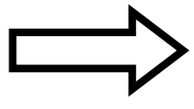


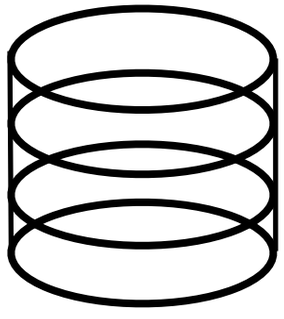


Offline RL pipeline

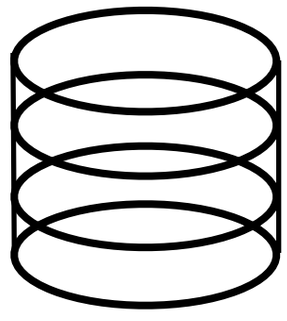


data

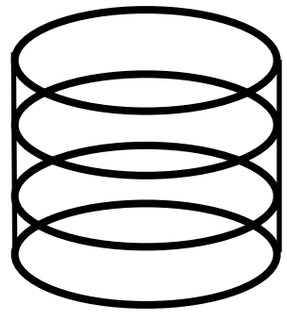




test



training



validation

$\{ (s, a, r, s') \}$

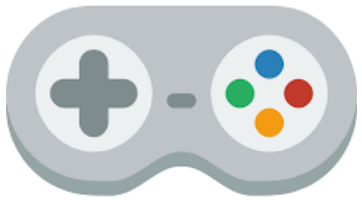
Off-policy Evaluation (OPE)

reward $r_t = R\,(s_t\,,\,a_t)$

state $s_t \in S$

+20

next-state $s_{t+1} \in S$



transition dynamics $P\,(\,\cdot \mid s_t\,,\,a_t)$

policy $\pi\colon S \to \Delta(A)$

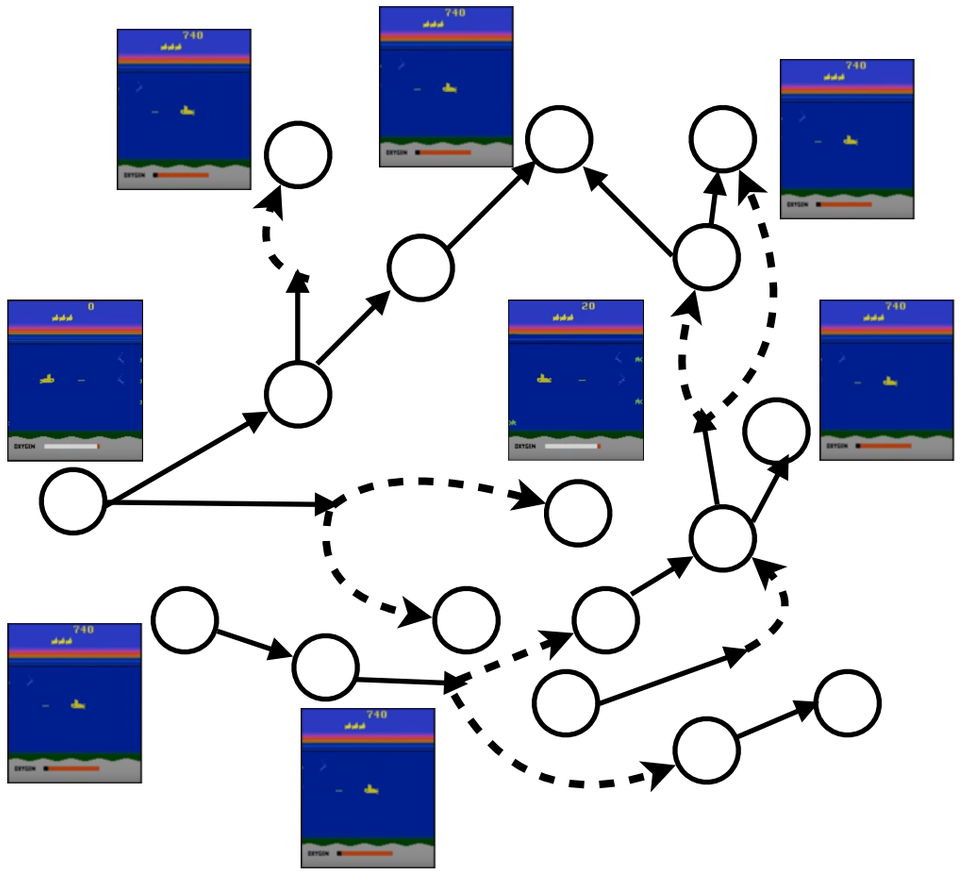
action $a_t \in A$

Policy evaluation: estimate $J(\pi) := \mathbb{E}_\pi[\sum_{t=0}^{\infty} \gamma^t r_t \mid s_0]$ given π

Policy optimization: $\max_\pi J(\pi) \quad = Q^\pi(s_0, \pi)$

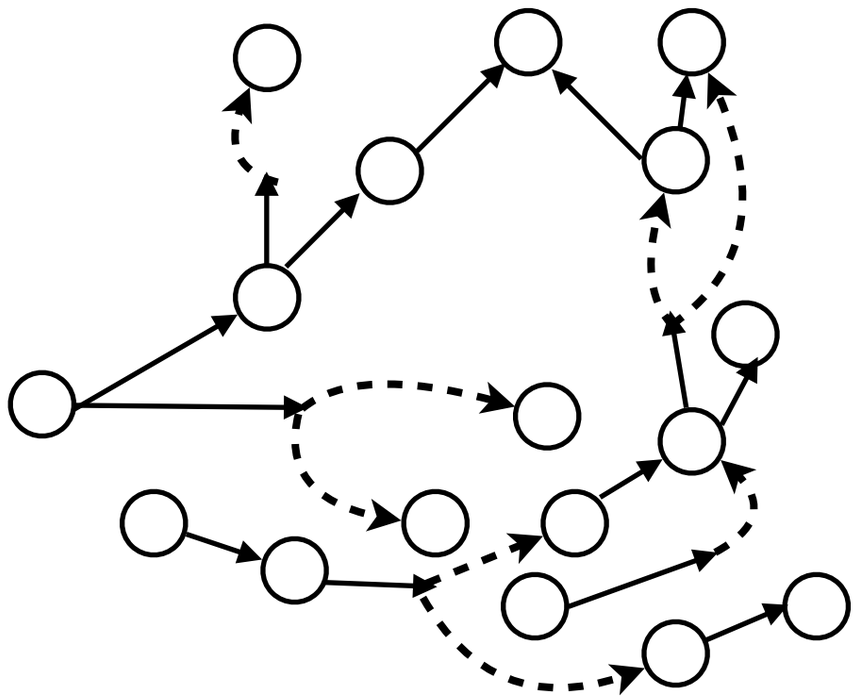
How to find Q^π?

$Q^\pi = \mathcal{T}^\pi \mathcal{Q}^\pi \;\rightarrow\;$ |SxA| equations



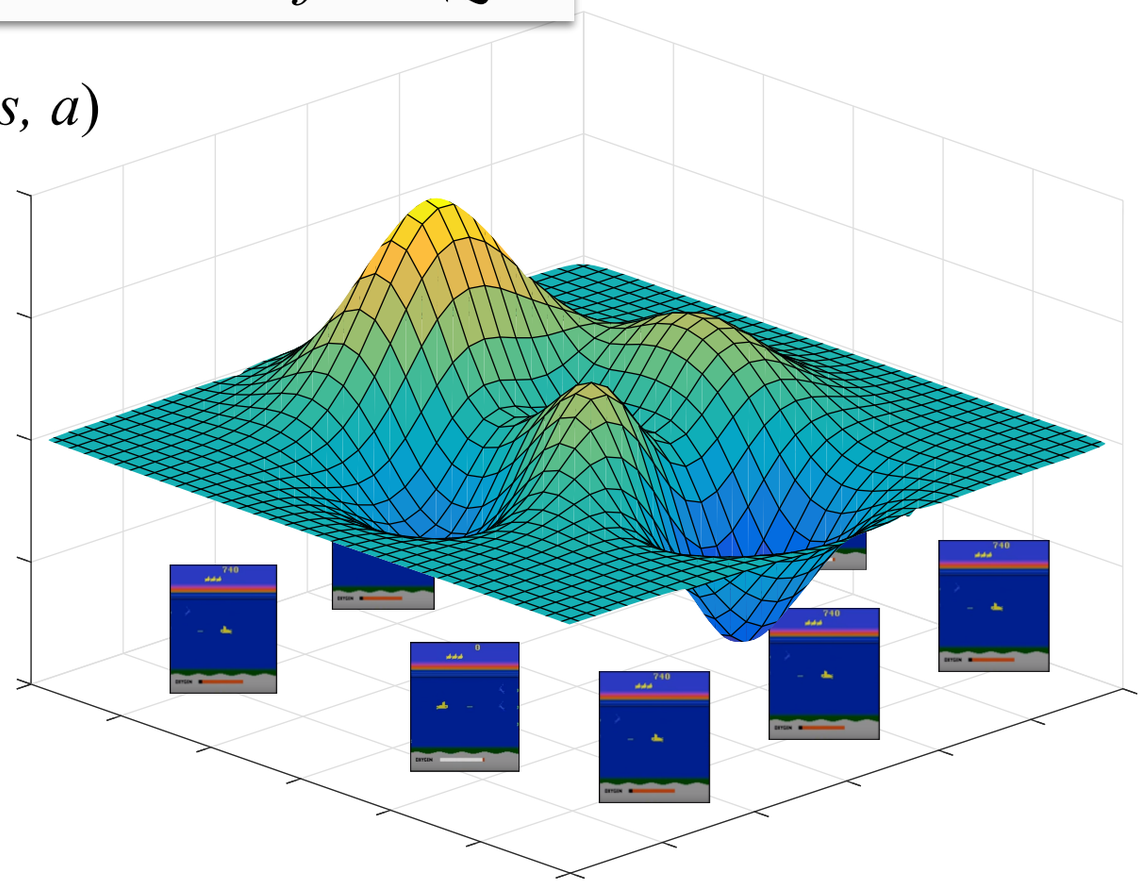
How to find Q^π?

$$Q^\pi = \mathcal{T}^\pi \mathcal{Q}^\pi \rightarrow \quad |S \text{x} A| \text{ equations } \textbf{X}$$



Find θ s.t. $f_\theta \approx Q^\pi$

$f_\theta(s, a)$



s, a

$$s_0, a_0, r_0, s_1, a_1, r_1, s_2, a_2, r_2, s_3, \ldots$$

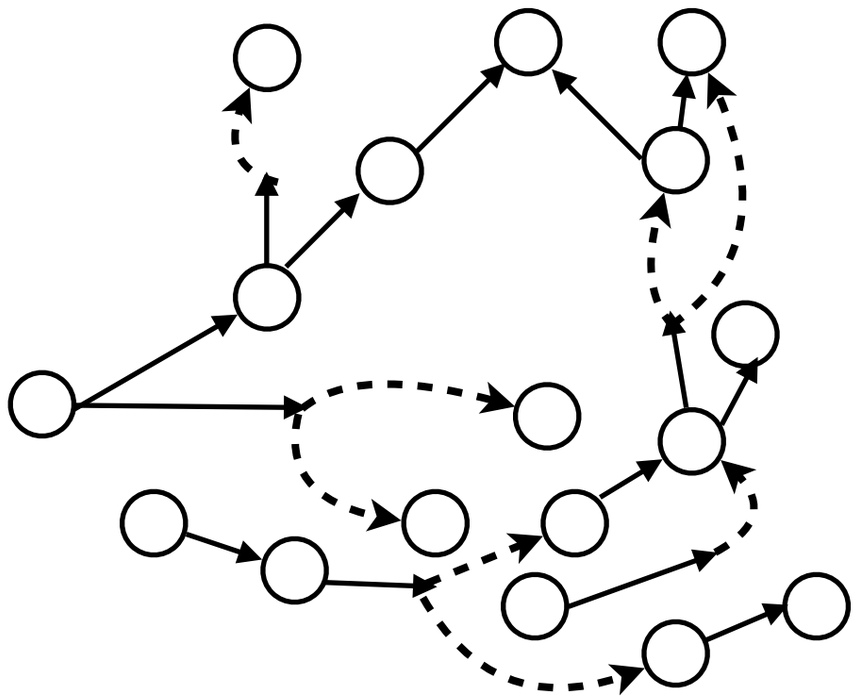
$$(s, a, r, s') \sim D$$

Validation:

(FQE: learn Q^π)

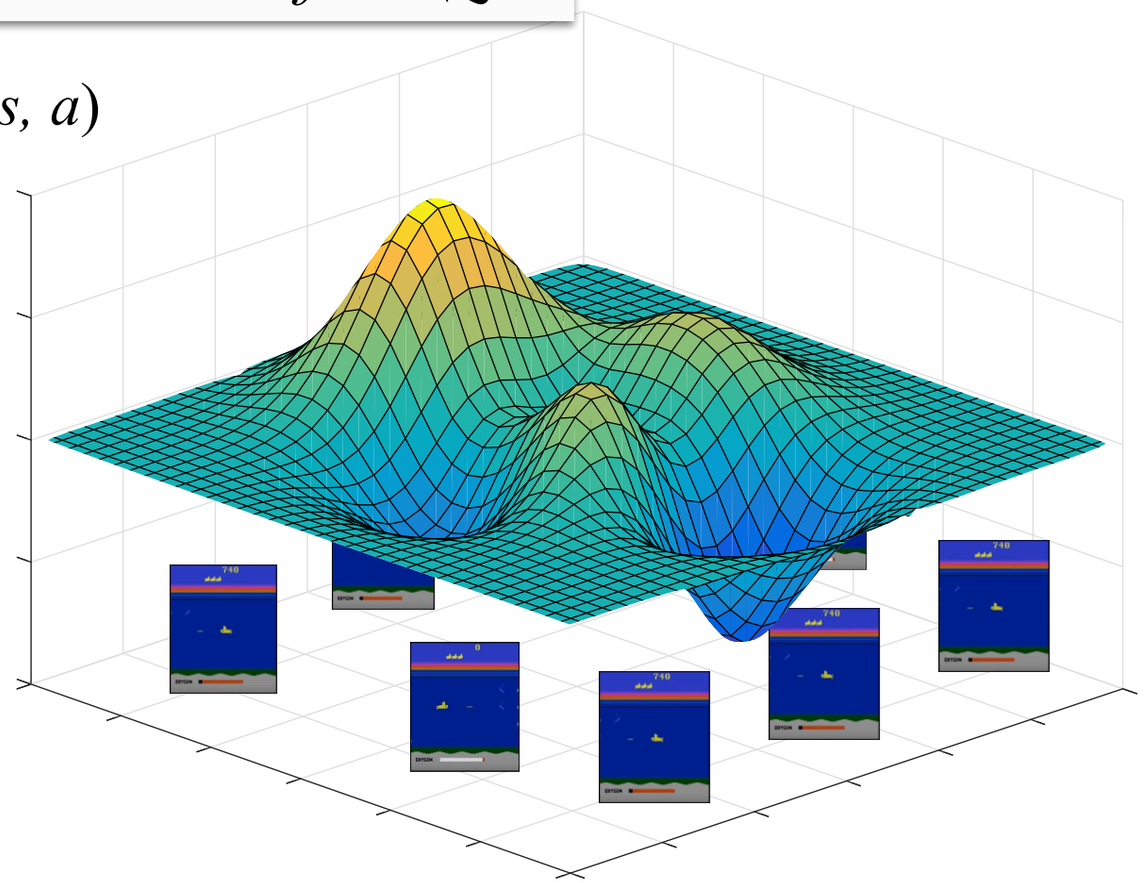
iterative

$$f_k \leftarrow \arg\min_{f_\theta} \mathbb{E}_D [(f_\theta(s, a) - r - \gamma f_{k-1}(s', \pi))^2]$$



Find θ s.t. $f_\theta \approx Q^\pi$

$f_\theta(s,\, a)$



$s,\, a$

Validation:

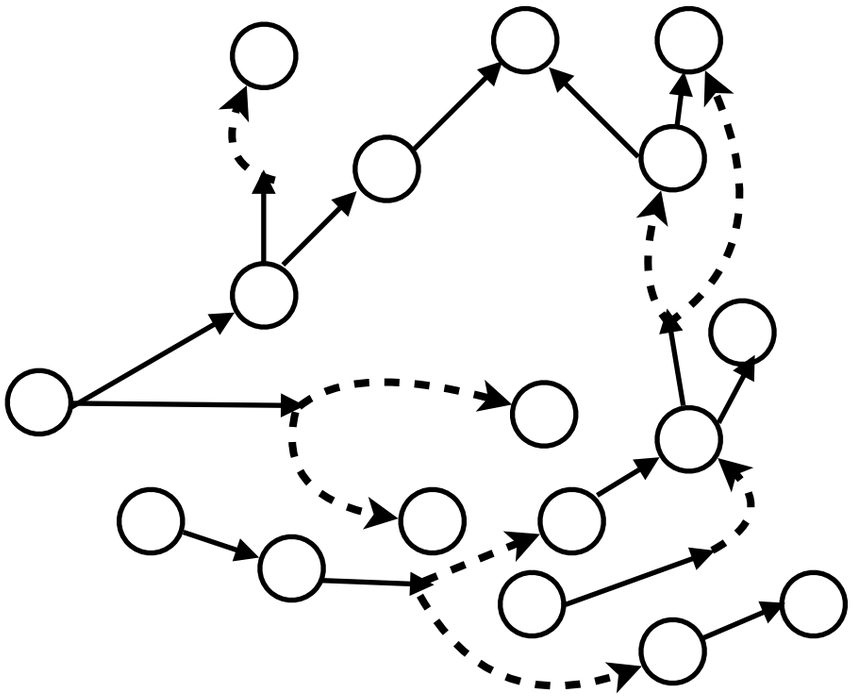
(FQE: learn Q^π)

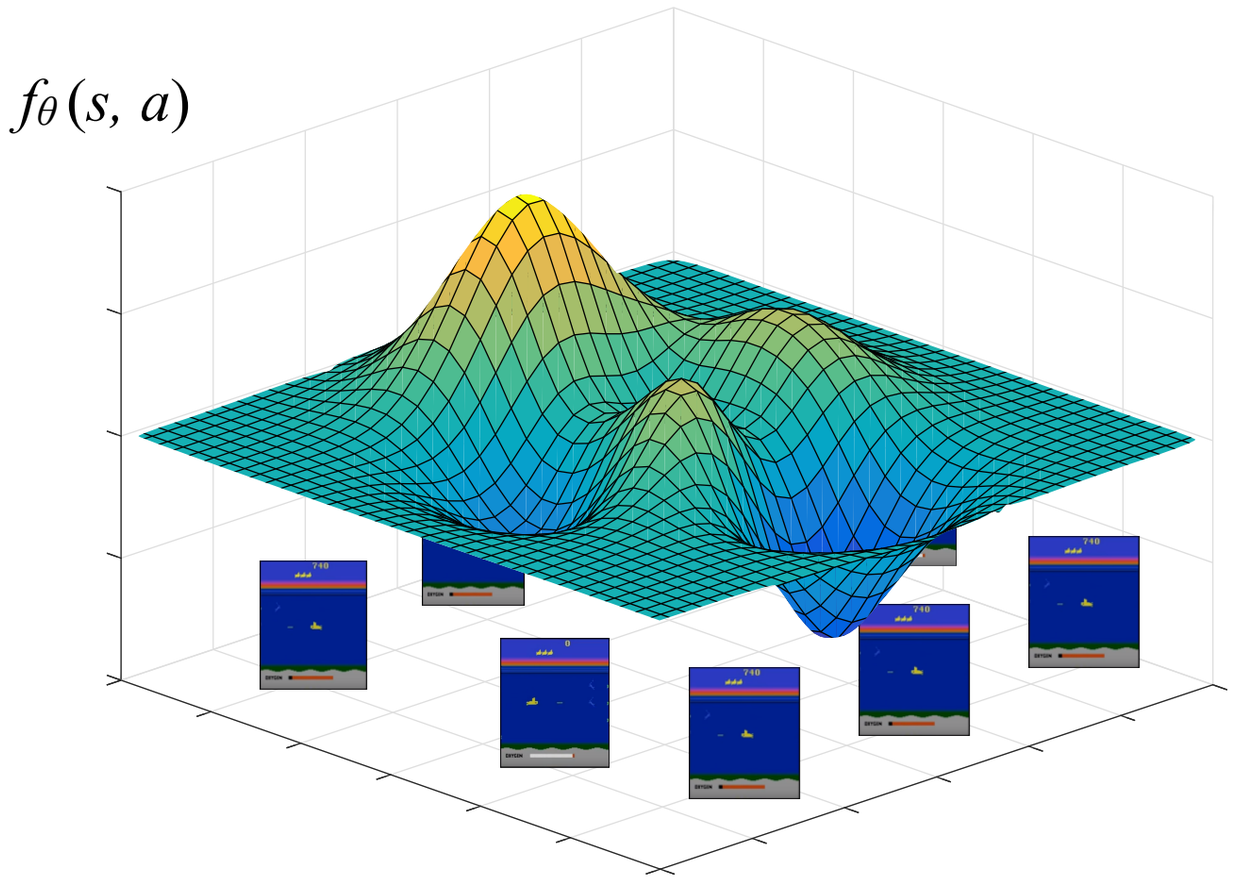
$$f_k \leftarrow \arg\min_{f_\theta} \mathbb{E}_D \left[\left(f_\theta(s,a) - r - \gamma f_{k-1}(s', \pi) \right)^2 \right]$$

iterative

$$\approx \mathcal{T}^\pi f_{k-1}$$

$\mathrm{E}[\ \cdot\ \mid s, a]$





$s,\ a$

Training: $\hat{f} = f_k$ where

(FQI: learn Q^*)

$$f_k \leftarrow \arg\min_{f_\theta} \mathbb{E}_D \left[(f_\theta(s,a) - r - \gamma \max_{a'} f_{k-1}(s',a'))^2 \right]$$

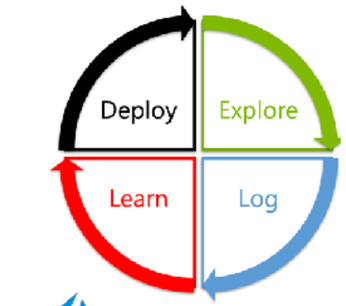
$$\Downarrow \pi = \text{greedy w.r.t. } \hat{f}$$

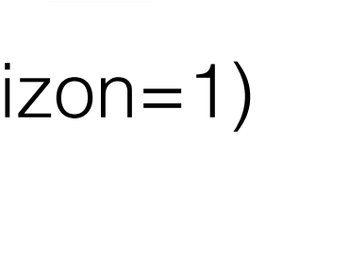
Validation:

(FQE: learn Q^π)

$$f_k \leftarrow \arg\min_{f_\theta} \mathbb{E}_D \left[(f_\theta(s,a) - r - \gamma f_{k-1}(s',\pi))^2 \right]$$

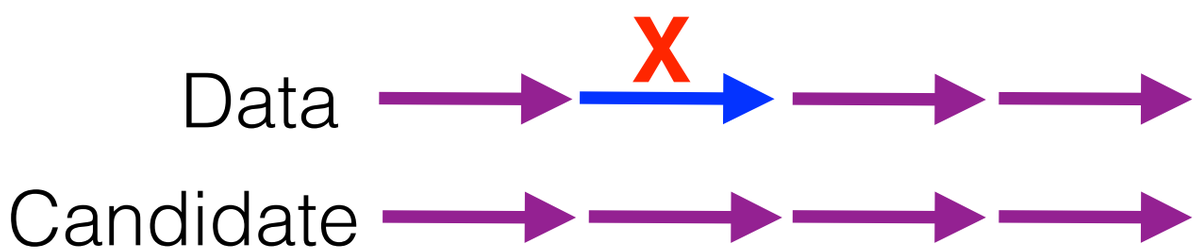
Hyperparameter-free methods?



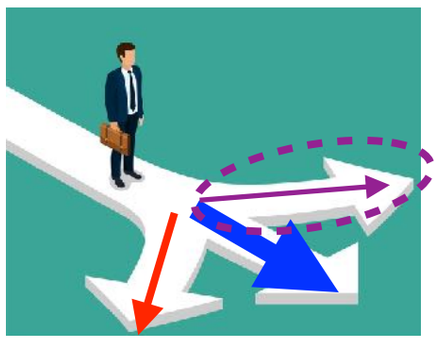


Importance sampling [Precup'00]

- Hyperparameter-free ✓
- No Markovianity required ✓
- Industry deployment (ctx. bandit, horizon=1)
- Exponential-in-horizon variance!
- Variance reduction?



Data



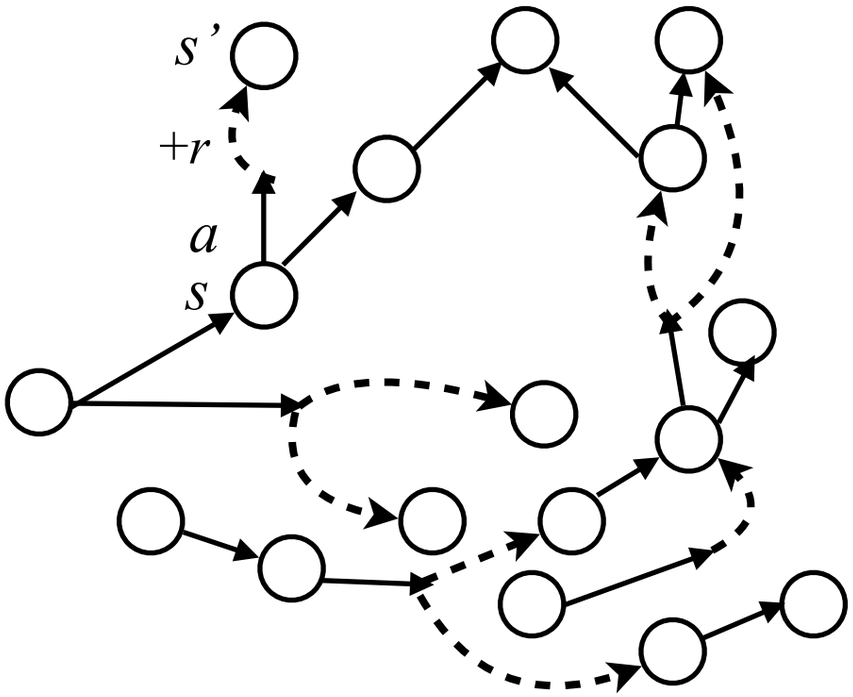
Candidate

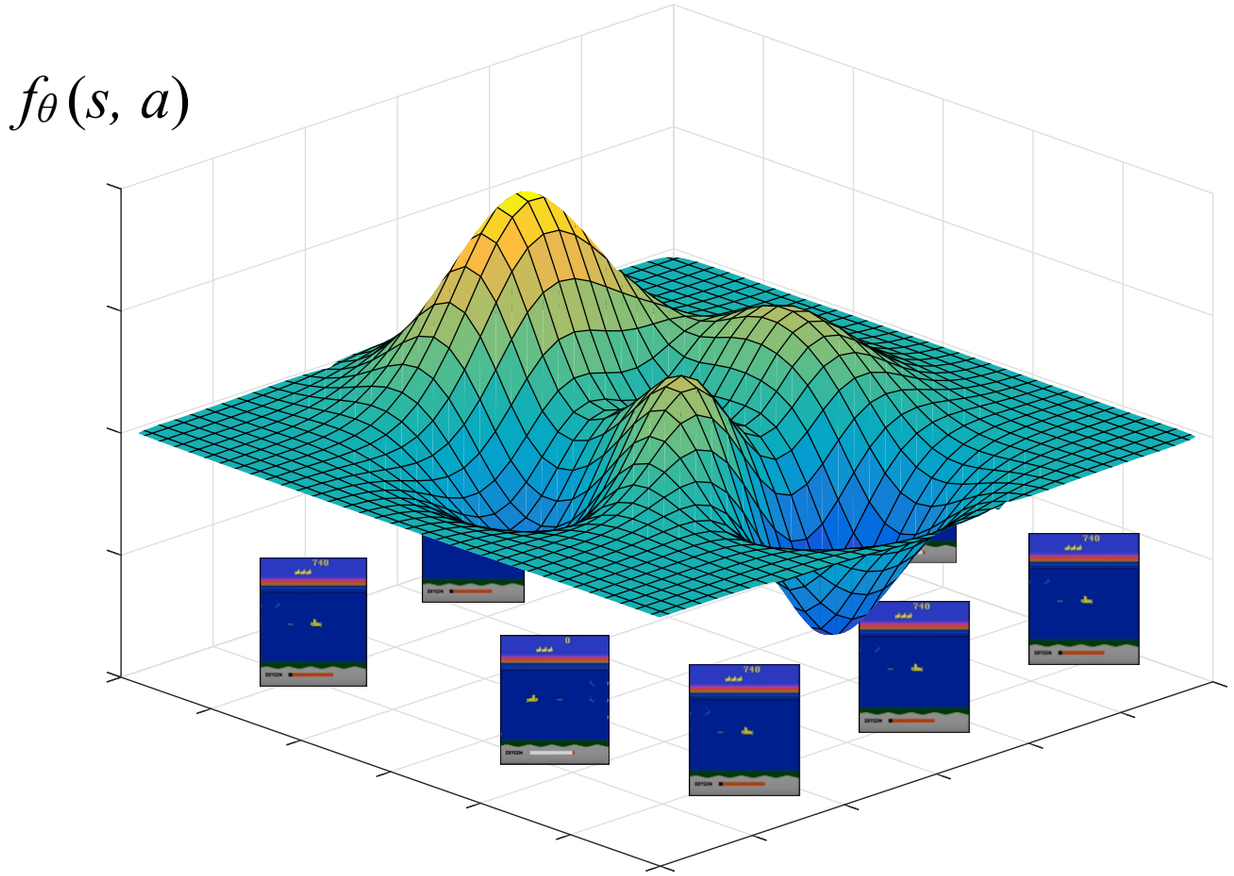
Doubly robust [JL'16]

- Even perfect control variate cannot eliminate exponential variance!

Precup. 2000. Eligibility traces for off-policy policy evaluation.

Nan Jiang, Lihong Li. ICML-16. Doubly Robust Off-policy Value Evaluation for Reinforcement Learning.





Training: $\hat{f} = f_k$ where

(FQI: learn Q*)

$$f_k \leftarrow \arg\min_{f_\theta} \mathbb{E}_{s,\ a \sim D} \left[(f_\theta(s,a) - r - \gamma \max_{a'} f_{k-1}(s',a'))^2 \right]$$

π = greedy w.r.t. $\hat{f}$

Reformulation: Value-function Selection

Simple(?) Problem

- Run different training algorithms
- Get candidate value functions f_1, f_2, …
- Holdout data $\{(s, a, r, s')\}$
- Select a good approx of Q^* w/ a "small" holdout dataset?
  - "small" = no |S| or exponential-in-horizon
  - & no further function approximation!
- Simpler: identify Q^* out of f_1, f_2

The training perspective

- Baird'95: design L s.t. $Q^\star \overset{?}{=} \arg\min_{f \in \mathcal{F}} L(f)$
- RL doesn't work like that!

$\mathcal{F} = \{f_1, f_2, \ldots\}$

Training: $\hat{f} = f_k$ where

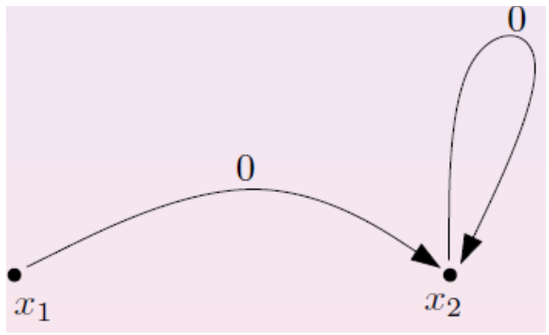
(FQI: learn Q*)

optimization

$$f_k \leftarrow \arg\min_{f_\theta} \mathbb{E}_D \left[(f_\theta(s,a) - r - \gamma \max_{a'} f_{k-1}(s',a'))^2 \right]$$

$\approx \mathcal{T} f_{k-1}$

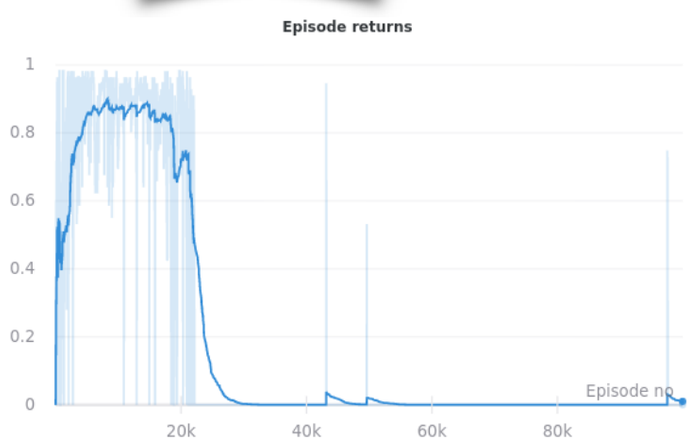
iterative

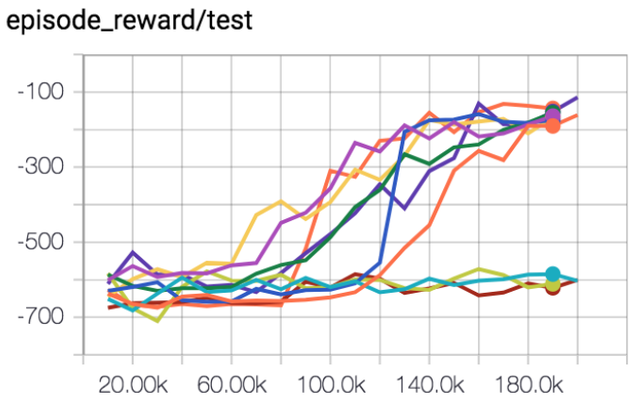


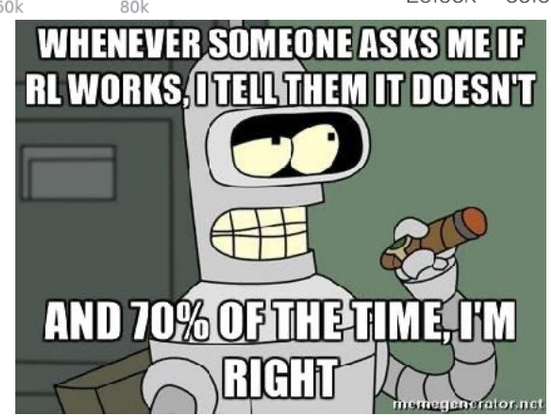
Divergence under 1-d linear

[TvR'96]

"Deadly triad"







 Baird. 1995. Residual algorithms: Reinforcement learning with function approximation. https://www.alexirpan.com/2018/02/14/rl-hard-well.html

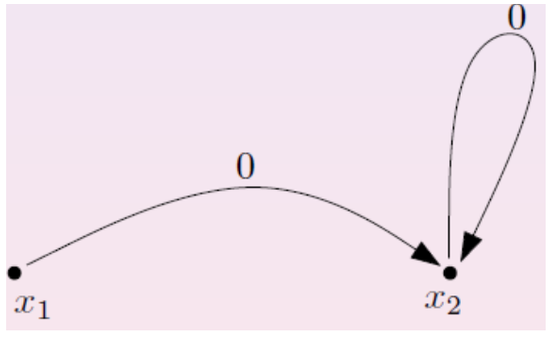
The training perspective

- Baird'95: design L s.t. $Q^\star \overset{?}{=} \arg\min_{f \in \mathcal{F}} L(f)$
- RL doesn't work like that!

Training: $\hat{f} = f_k$ where

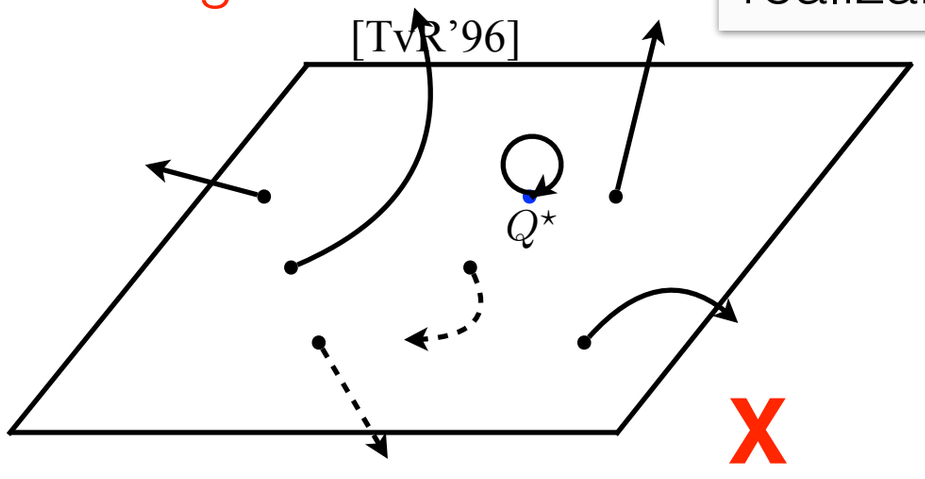
(FQI: learn Q*)

$$f_k \leftarrow \arg\min_{f_\theta} \mathbb{E}_D \left[(f_\theta(s,a) - r - \gamma \max_{a'} f_{k-1}(s',a'))^2 \right] \approx \mathcal{T} f_{k-1}$$



Divergence under 1-d lin realizability (of Q^*)

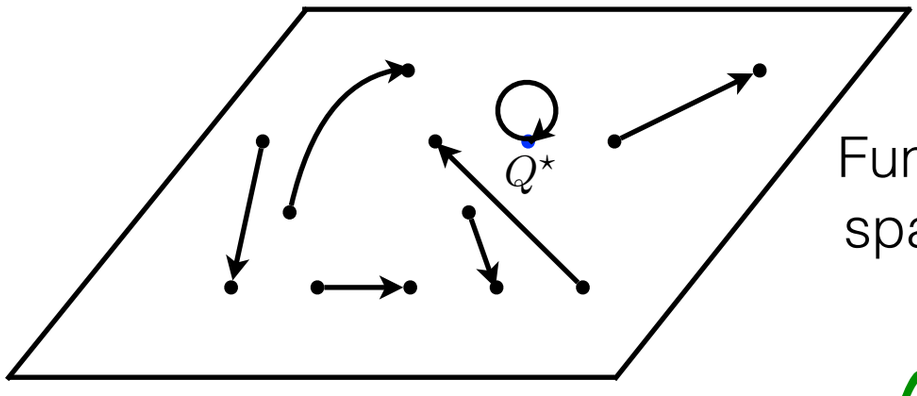
[TvR'96]



X

"Bellman-completeness"

$\mathcal{T} f \in \mathcal{F}, \forall f \in \mathcal{F}$



Function space F

✓

• Function in F → Bellman operator $\mathcal{T}$

The training perspective

- Baird'95: design L s.t. $Q^\star \overset{?}{=} \arg\min_{f \in \mathcal{F}} L(f)$
- $f = Q^\star \Leftrightarrow f = \mathcal{T} f$, so how about

$$f - \mathcal{T} f$$

$$= \mathbb{E}_D \left[(f(s,a) - \mathbb{E}[r + \gamma \max_{a'} f(s',a') | s, a])^2 \right]$$

$$\neq$$

$$\mathbb{E}_D \left[(f(s,a) - \quad (r + \gamma \max_{a'} f(s',a')) \qquad)^2 \right]$$

- Naive "1-sample" estimator is biased
  - debasing requires simulator ("*double sampling*" [Baird'95])
  - or, helper class $\mathcal{F}' \ni \mathcal{T} f$ [ASM'08, FS'10]

* over-estimate by a Bayes-error-like term: $\mathbb{E}_{d^D} \left[\mathbb{V}_{s'|s,a} [r + \gamma \max_{a'} f(s',a')] \right]$

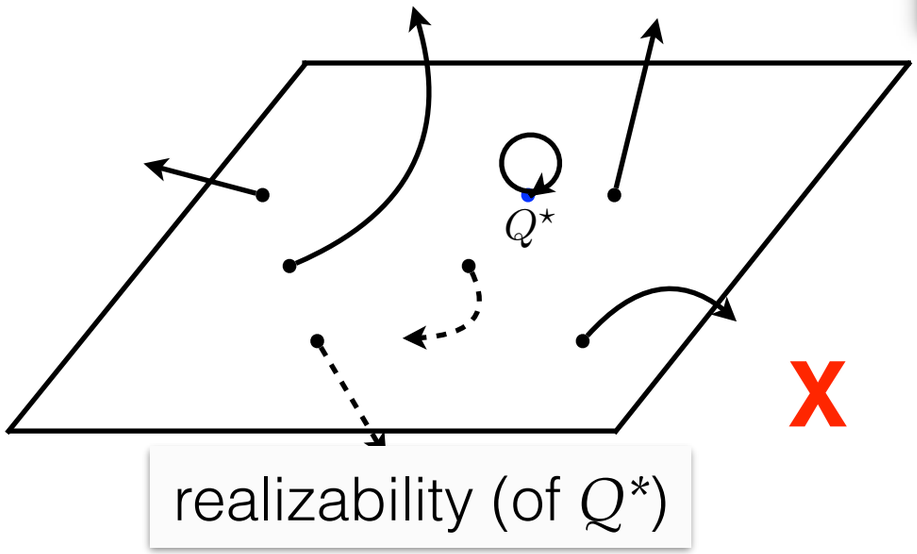
Basis of resolution

Training: $\hat{f} = f_k$ where

(FQI: learn Q*)

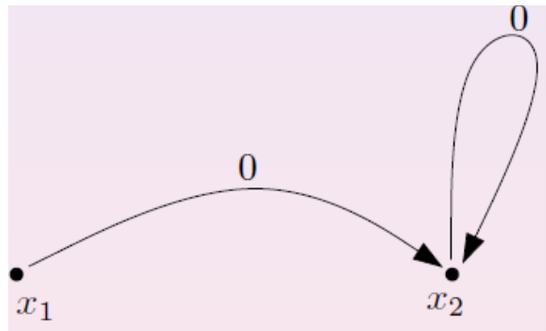
$$f_k \leftarrow \arg\min_{f_\theta \in \mathcal{F}} \mathbb{E}_D \left[(f_\theta(s,a) - r - \gamma \max_{a'} f_{k-1}(s',a'))^2 \right]$$

iterative



realizability (of Q^*)

X



Divergence under 1-d linear $\mathcal{F}$

[TvR'96]

To select b/t f_1, f_2

Gordan. 1995. Stable Function Approximation in Dynamic Programming.

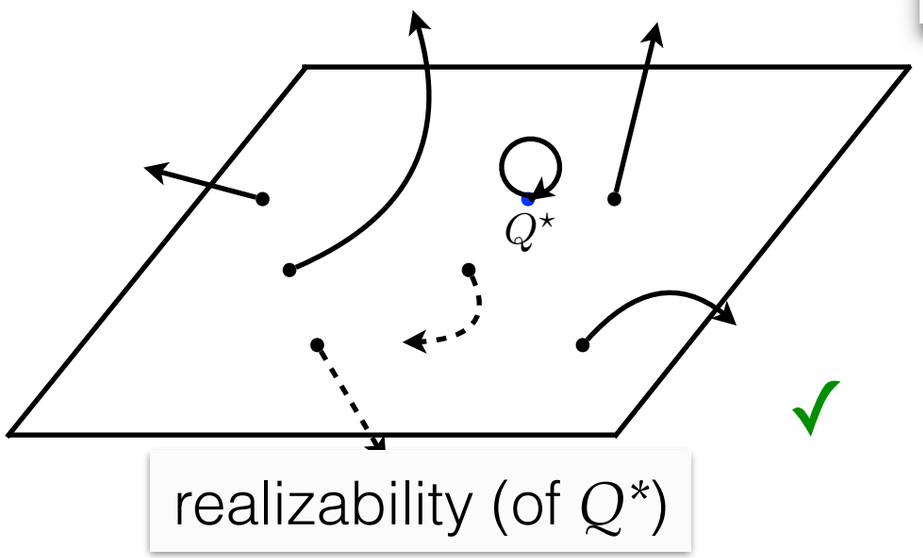
Basis of resolution

Training: $\hat{f} = f_k$ where

(FQI: learn Q*)

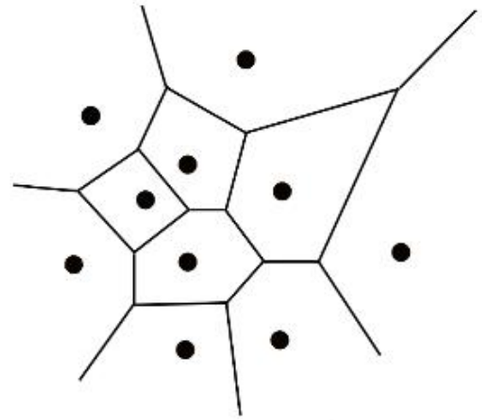
$$f_k \leftarrow \arg\min_{f_\theta \in \mathcal{F}} \mathbb{E}_D \left[(f_\theta(s,a) - r - \gamma \max_{a'} f_{k-1}(s',a'))^2 \right]$$

iterative



realizability (of Q^*)

✓



Convergence under piecewise constant $\mathcal{F}$! [Gordon'95]

same

To select b/t f_1, f_2, suffices to have class G s.t.

- piecewise constant
- can express Q^*
- small # partitions (bounded complexity)

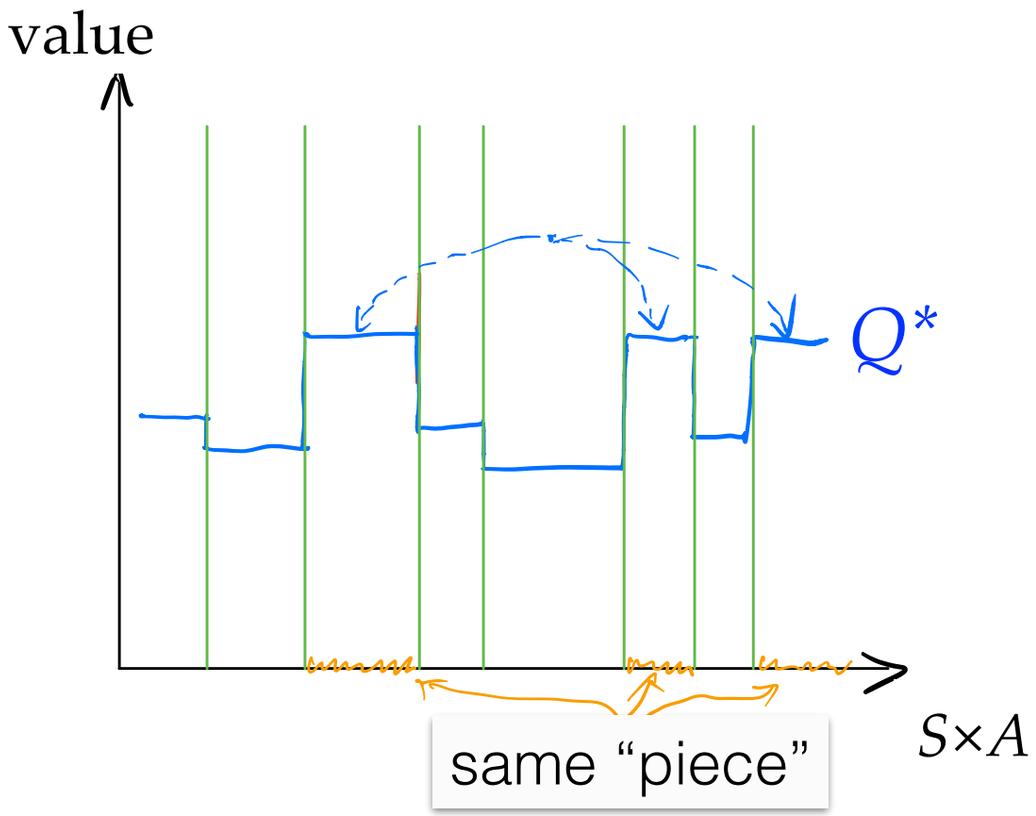
Then: minimize $\|f - \text{Proj}_{\mathcal{G}}(\mathcal{T}f)\|_{2,D}$

Our method: create such a magical G "out of nothing"!

Gordan. 1995. Stable Function Approximation in Dynamic Programming.

Does a magical G always exist?

- YES! Just partition SxA according to **output** of Q^*

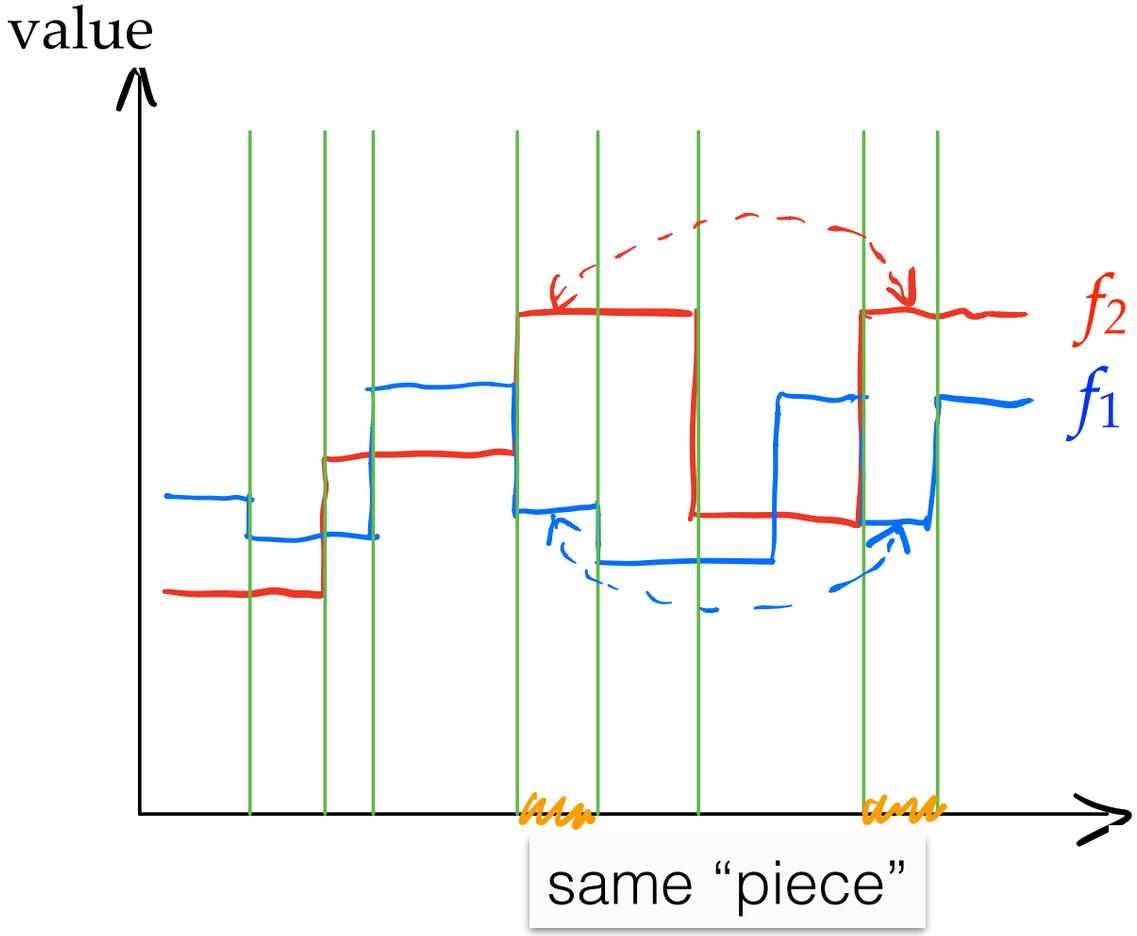


To select b/t f_1, f_2, suffices to have class G s.t.

- piecewise constant ✓
- can express Q^* ✓
- $O(1/\varepsilon)$ partitions (bounded complexity) ✓

Then: minimize $\|f - \mathrm{Proj}_{\mathcal{G}}(\mathcal{T}f)\|_{2,D}$

Batch Value-Function Tournament [XJ, ICML-21]

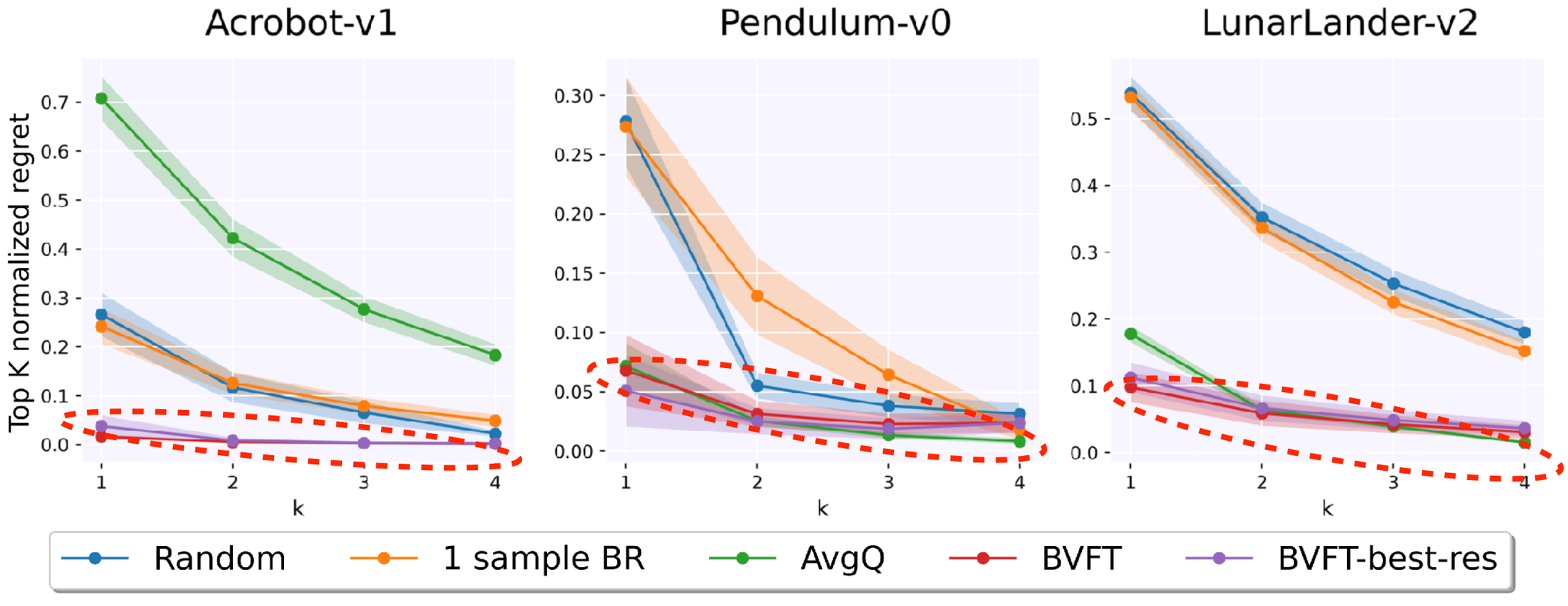


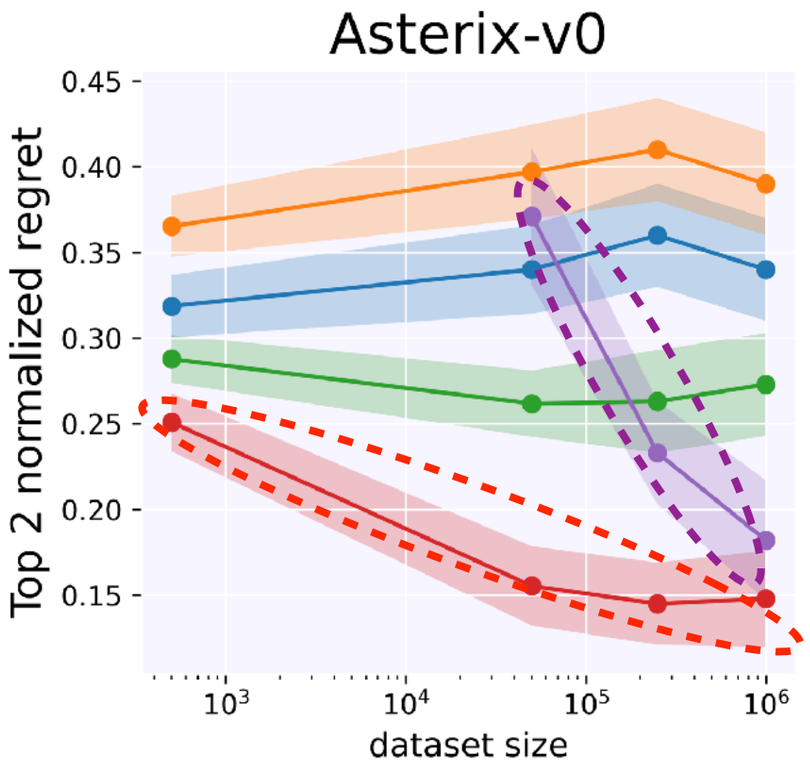
- Algorithm: **BVFT**
  $\arg\min_i \max_j \|f_i - \mathrm{Proj}_{\mathcal{G}_{i,j}}(\mathcal{T} f_i)\|_{2,D}$
- Sample complexity poly in horizon, $1/\varepsilon$, log(#candidates), and C (data coverage)
- Computation: #data points * $|F|^2$

- (Simplified) problem: identify Q^* out of $F = \{ f_1, f_2 \}$
- Partition SxA according to both functions simultaneously!
  - Pw-const class $G_{1,2}$ w/ size $\boldsymbol{O(1/\varepsilon^2)}$ **!!**
- Naive extension to >2 functions in F: $O(1/\varepsilon^{|F|})$
  - Pairwise comparison + tournament

Formal guarantee in backup slide

Xie & **Jiang**. ICML-21. Batch Value-function Approximation with Only Realizability.





$$f_k \leftarrow \arg\min_{f_\theta} \mathbb{E}_D \left[(f_\theta(s,a) - r - \gamma f_{k-1}(s', \pi))^2 \right]$$

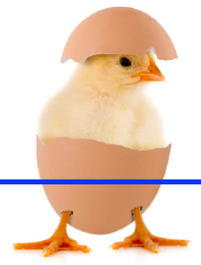
Neural architecture designed by "cheating"

Zhang & **Jiang**. NeurIPS-21. Towards Hyperparameter-free Policy Selection for Offline Reinforcement Learning.

Training: $\hat{f} = f_k$ where

(FQI: learn Q^*)

$$f_k \leftarrow \arg\min_{f_\theta} \mathbb{E}_D \left[(f_\theta(s,a) - r - \gamma \max_{a'} f_{k-1}(s',a'))^2 \right]$$



$$\Downarrow \quad \pi = \text{greedy w.r.t. } \hat{f}$$

Validation:

(FQE: learn Q^π)

$$f_k \leftarrow \arg\min_{f_\theta} \mathbb{E}_D \left[(f_\theta(s,a) - r - \gamma f_{k-1}(s',\pi))^2 \right]$$

- BVFT: H-P free solution for value-function selection
- Many open problems in validation
  - Data coverage issues (see lower bound [FKSX'22])
  - Combine with different OPE methods
    e.g., marginalized importance sampling
    [LLTZ'18, NCDL'19, UHJ'20, JH'20, VJY'21, HJ'22]
  - Practical toolkit (cf. for OPE [VLJY'20])

Foster et al. COLT-22. Offline reinforcement learning: Fundamental barriers for value function approximation.
Voloshin et al. NeurIPS-21. Empirical Study of Off-Policy Policy Evaluation for Reinforcement Learning.