Exploiting Structure in Reinforcement Learning

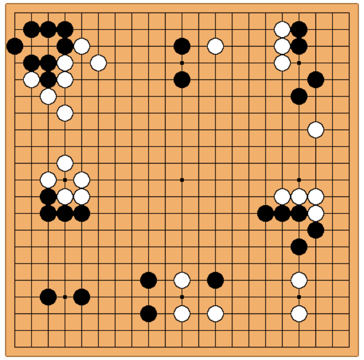
Christina Lee Yu

Cornell University

Published: 19 October 2017

Artificial intelligence

Learning to play Go from scratch



Satinder Singh, Andy Okun & Andrew Jackson

Nature **550**, 336–337 (2017) | Cite this article

23k Accesses | **41** Citations | **292** Altmetric | Metrics

An artificial-intelligence program called AlphaGo Zero has mastered the game of

NEWS | ROBOTICS

OpenAI Teaches Robot Hand to Solve Rubik's Cube

› Using reinforcement learning and randomized simulations, researchers taught this robot how to solve a Rubik's cube one-handed

BY EVAN ACKERMAN | 15 OCT 2019 | 5 MIN READ

Deep Reinforcement Learning achieves super-human performance!

Article | Open Access | Published: 05 October 2022

Discovering faster matrix multiplication algorithms with reinforcement learning

Alhussein Fawzi, Matej Balog, Aja Huang, Thomas Hubert, Bernardino Romera-Paredes, Mohammadamin Barekatain, Alexander Novikov, Francisco J. R. Ruiz, Julian Schrittwieser, Grzegorz Swirszcz, David Silver, Demis Hassabis & Pushmeet Kohli

Nature **610**, 47–53 (2022) | Cite this article

243k Accesses | **3286** Altmetric | Metrics

DeepMind's MEME Agent Achieves Human-level Atari Game Performance 200x Faster Than Agent57

In the new paper Human-level Atari 200x Faster, a DeepMind research team applies a set of diverse strategies to Agent57, with their resulting MEME (Efficient Memory-based Exploration) agent surpassing the human baseline on all 57 Atari games in just 390 million frames — two orders of magnitude faster than Agent57.

Deep Reinforcement Learning achieves super-human performance!

At what cost?

"Training AlphaGoZero to play Go took 72 hours, with over 4.9 million matches played, and with each move during self-play using about 0.4 seconds of processing time, on a single machine with 4 TPUs (Google's special-purpose Tensor Processing Unit chips), plus additional parameter updates powered by 64 GPUs and 19 CPUs." [Silver et.al. 2017]

*does not include hyperparameter tuning!

Deep Reinforcement Learning achieves super-human performance!

At what cost? High computational/storage burdens, massive training data requirements, sensitive to hyperparameter tuning

RL is not yet practical for settings where it is also critical to exhibit

- Data efficiency / efficient learning
- Low computational cost and time
- Low storage requirements / memory usage

In real-world systems, domain heuristics often outperform RL, as RL ignores the known structure of the problem.

Central Research Question

How to design RL algorithms that **provably** and **efficiently** exploit structure arising in real-world systems?

1. What types of structure are reasonable and common?

2. What type of information is commonly available?

3. How to exploit it to lead to efficient learning?

Outline – dealing with large state/action MDPs

- Part I: Exploiting smoothness in continuous state/action MDPs using adaptive discretization

Sean R. Sinclair, Siddhartha Banerjee, Christina Lee Yu. "Adaptive Discretization for Online Reinforcement Learning." *Operations Research,* 2022.

Sean R. Sinclair, Tianyu Wang, Gauri Jain, Siddhartha Banerjee, Christina Lee Yu. "Adaptive Discretization for Model-Based Reinforcement Learning." *Neurips,* 2020.

Sean R. Sinclair, Siddhartha Banerjee, Christina Lee Yu. "Adaptive Discretization for Episodic Reinforcement Learning in Metric Spaces." *POMACS + ACM SIGMETRICS*, 2019.

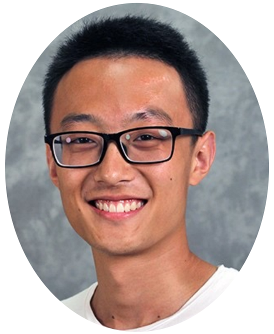
- Part II: Exploiting latent low rank structure in action-value function using matrix completion

Tyler Sam, Yudong Chen, Christina Lee Yu. "Overcoming the Long Horizon Barrier for Sample-Efficient Reinforcement Learning with Latent Low-Rank Structure." *POMACS + ACM SIGMETRICS,* 2023.

Part I: Exploiting smoothness in continuous state/action space MDPs using adaptive discretization

Joint work with Sid Banerjee, Gauri Jain, Sean Sinclair, Tianyu Wang



Episodic Reinforcement Learning

- Agent interacts with an unknown MDP over a length *H* horizon
- Agent Policy $\pi_h : \mathcal{S} \to \mathcal{A}$
- Model Parameters $r_h : \mathcal{S} \times \mathcal{A} \to [0,1]$, $T_h : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$ (bounded)
- Value Function $V_h^\pi(x) = \mathbb{E}\left[\sum_{\ell=h}^H r_\ell(x_\ell, \pi_\ell(x_\ell)) \,\middle|\, x_h = x\right]$
- Q Function $Q_h^\pi(x,a) = r_h(x,a) + \mathbb{E}\left[V_{h+1}^\pi(x_{h+1}) \mid x_h = x, a_h = a\right]$
- Goal: minimize expected regret over *K* episodes of online interaction

$$R(K) = \sum_{k=1}^K (V_1^{\pi^\star}(x_1^k) - V_1^{\pi^k}(x_1^k))$$

($\pi^\star$: optimal policy; π^k: policy played by agent in episode *k*)

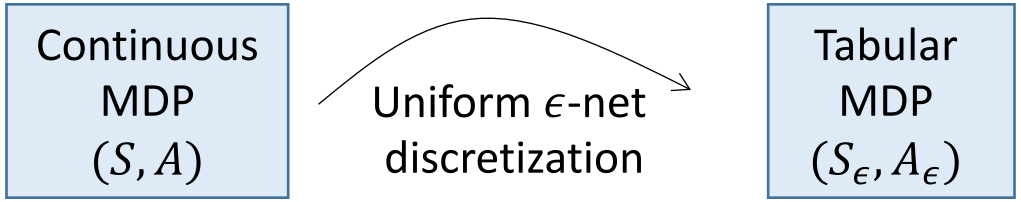
Dealing with continuous state/action spaces

1) Parametric function approximation
    - Approximate value function or policy with tractable function class
    - Leverage techniques from supervised learning
    - Sensitive to model mismatch

2) Discretization / Aggregation
    - Approximate full MDP with a smaller tabular MDP
    - Relies on smoothness assumptions with respect to known metric

Discretization for Continuous MDPs

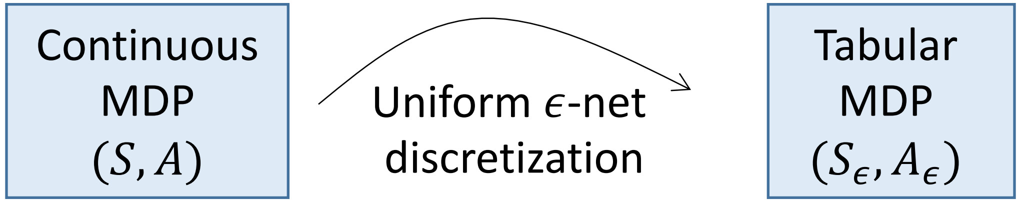
- Compact continuous state space S, A, with known metric
- Assume that MDP (Q^*, r, T) is Lipschitz continuous wrt known metric
- Naïve discretization approach



- Choose ϵ to balance approx error and regret from tabular MDP

Discretization for Continuous MDPs

- Compact continuous state space S, A, with known metric
- Assume that MDP (Q^*, r, T) is Lipschitz continuous wrt known metric
- Naïve discretization approach



- Choose ϵ to balance approx error and regret from tabular MDP

$$\text{Regret} \leq \overbrace{H^4 \sqrt{\underbrace{S_\epsilon A_\epsilon}_{\approx \epsilon^{-d}} K}}^{\text{Optimistic Q-Learning}} + \overbrace{HKL\epsilon}^{\text{Discretization Error}}$$

$\approx \epsilon^{-d}$, where d is dimension of $S \times A$

Discretization for Continuous MDPs

- Compact continuous state space S, A, with known metric
- Assume that MDP (Q^*, r, T) is Lipschitz continuous wrt known metric
- Naïve discretization approach

Continuous MDP (S, A) → Uniform ϵ-net discretization → Tabular MDP (S_ϵ, A_ϵ)

- Choose ϵ to balance approx error and regret from tabular MDP

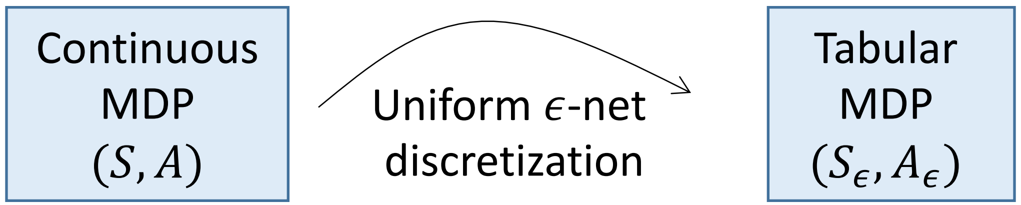
$$\text{Regret} \leq \overbrace{H^4 \sqrt{\underbrace{S_\epsilon A_\epsilon}_{\approx \epsilon^{-d}} K}}^{\text{Optimistic Q-Learning}} + \overbrace{HKL\epsilon}^{\text{Discretization Error}} \leq O(K^{(d+1)/(d+2)})$$

$\approx \epsilon^{-d}$, where d is dimension of $S \times A$

matches minimax lower bd from contextual bandits

Discretization for Continuous MDPs

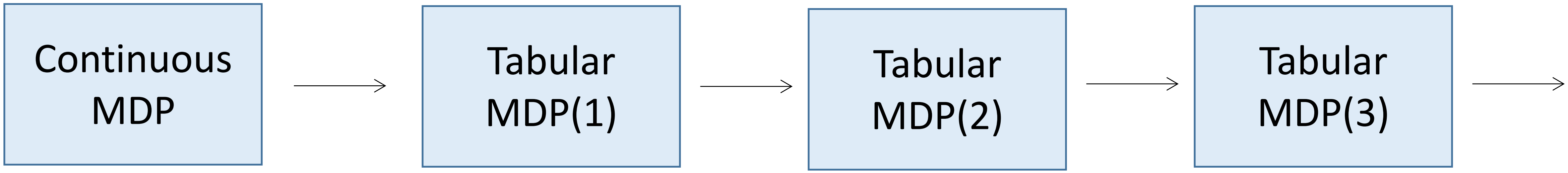
- Compact continuous state space S, A, with known metric
- Assume that MDP (Q^*, r, T) is Lipschitz continuous wrt known metric
- Naïve discretization approach



- Choose ϵ to balance approx error and regret from tabular MDP
- Could be very expensive in both memory and sample complexity
- Can we reduce memory requirements while preserving performance?

Adaptive Discretization

- Assume Lipschitz assumptions on model with respect to metric space
- Only refine discretization on an "as needed" basis



- Is there an optimal sequence of approximating MDPs?
- Overarching idea can be applied to convert any tabular RL algorithm into an algorithm for continuous spaces

Informal Theorem [SinclairBanerjeeYu2019] [SinclairWangJainBanerjeeYu2020]

We propose AdaQL (model free) and AdaMB (model based) that achieve

$$\textsc{Regret}(K) \lesssim \begin{cases} \textsc{AdaQL}: & H^{5/2} K^{\frac{z+1}{z+2}} \\ \textsc{AdaMB}: & H^{3/2} K^{\frac{z+d_S-1}{z+d_S}} \quad d_S > 2 \\ \textsc{AdaMB}: & H^{3/2} K^{\frac{z+1}{z+2}} \quad d_S \leq 2 \end{cases}$$

dependence on K matches minimax lower bound from contextual bandits

where z is zooming dim, d_S is dim of state space.

analogous to instance specific bounds in the multi-arm bandit literature

Informal Theorem [SinclairBanerjeeYu2019] [SinclairWangJainBanerjeeYu2020]

We propose AdaQL (model free) and AdaMB (model based) that achieve

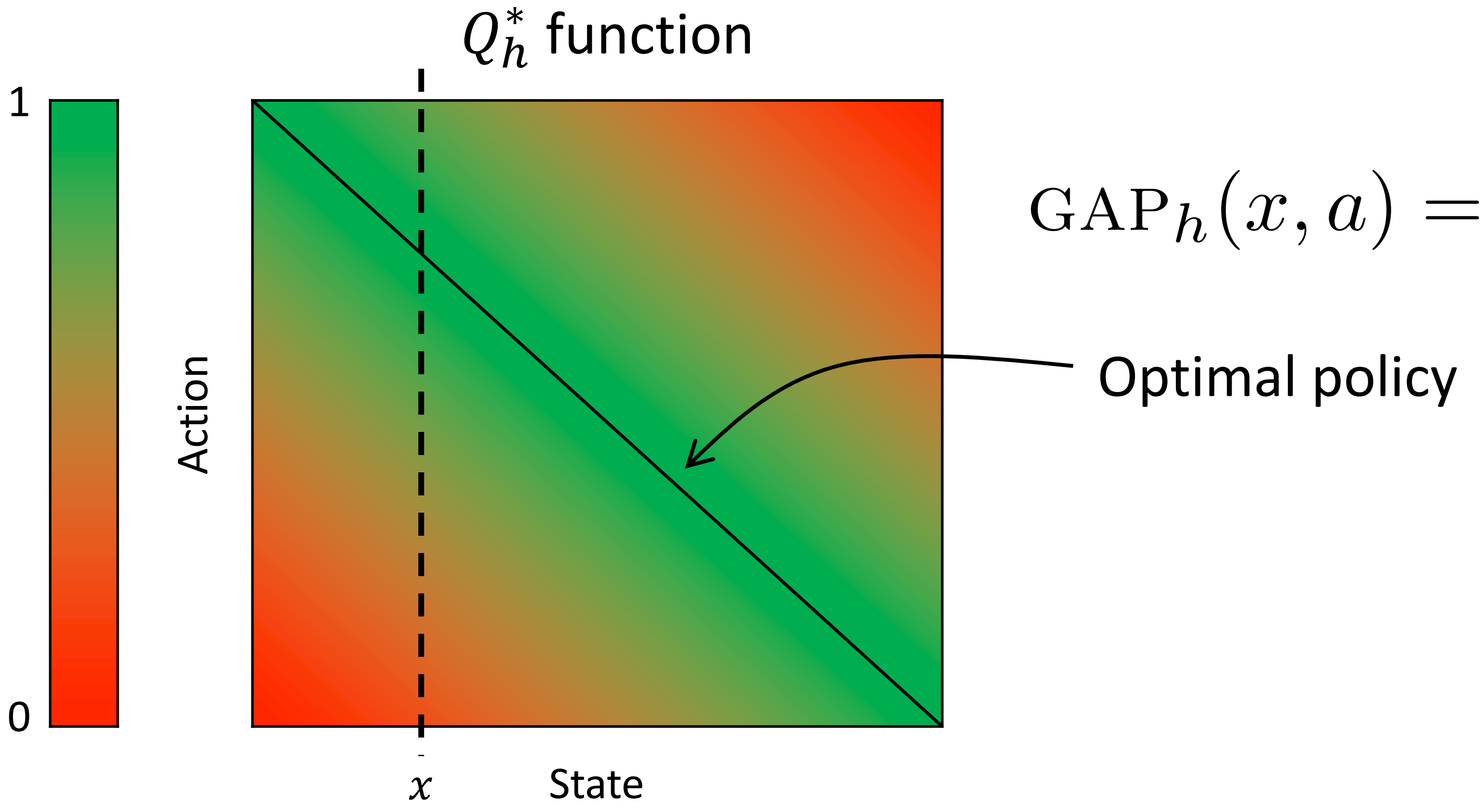
$$\textsc{Regret}(K) \lesssim \begin{cases} \textsc{AdaQL}: & H^{5/2} K^{\frac{z+1}{z+2}} \\ \textsc{AdaMB}: & H^{3/2} K^{\frac{z+d_S-1}{z+d_S}} \quad d_S > 2 \\ \textsc{AdaMB}: & H^{3/2} K^{\frac{z+1}{z+2}} \quad d_S \leq 2 \end{cases}$$

can be improved for "simple" dynamics

where z is zooming dim, d_S is dim of state space.

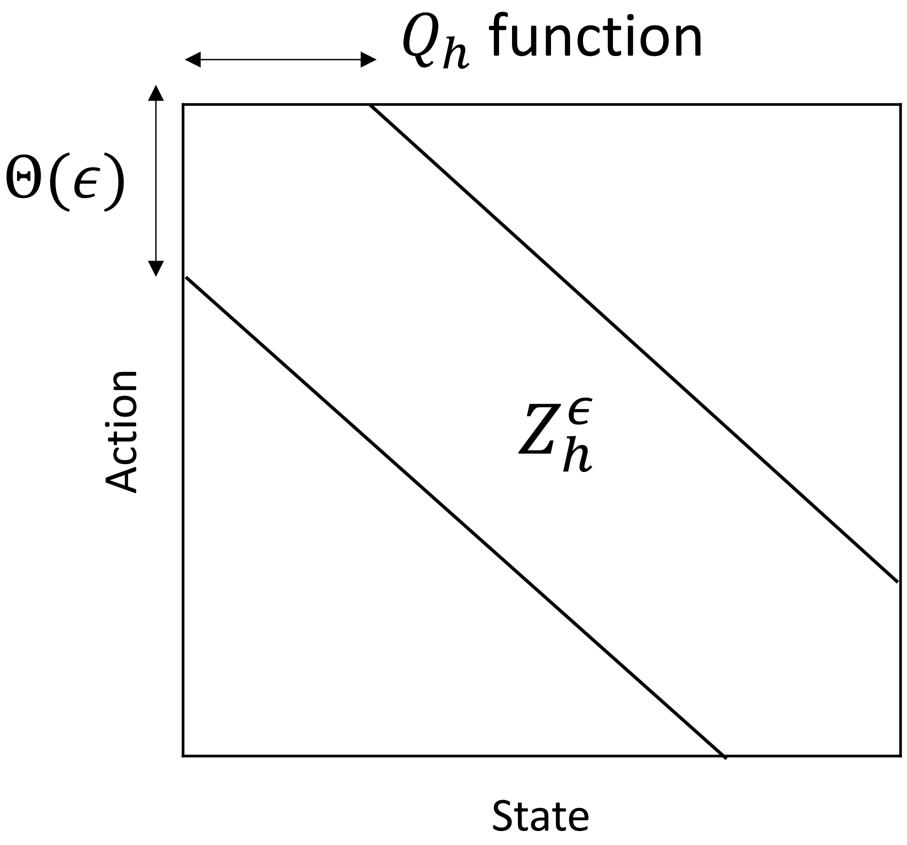
- Assume compact metric spaces S, A
- AdaQL: Lipschitz value functions Q_h^* and V_h^*
- AdaMB: Lipschitz rewards r_h and transitions T_h in the 1-Wasserstein metric

Zooming Dimension



$$\mathrm{GAP}_h(x,a) = V_h^*(x) - Q_h^*(x,a)$$

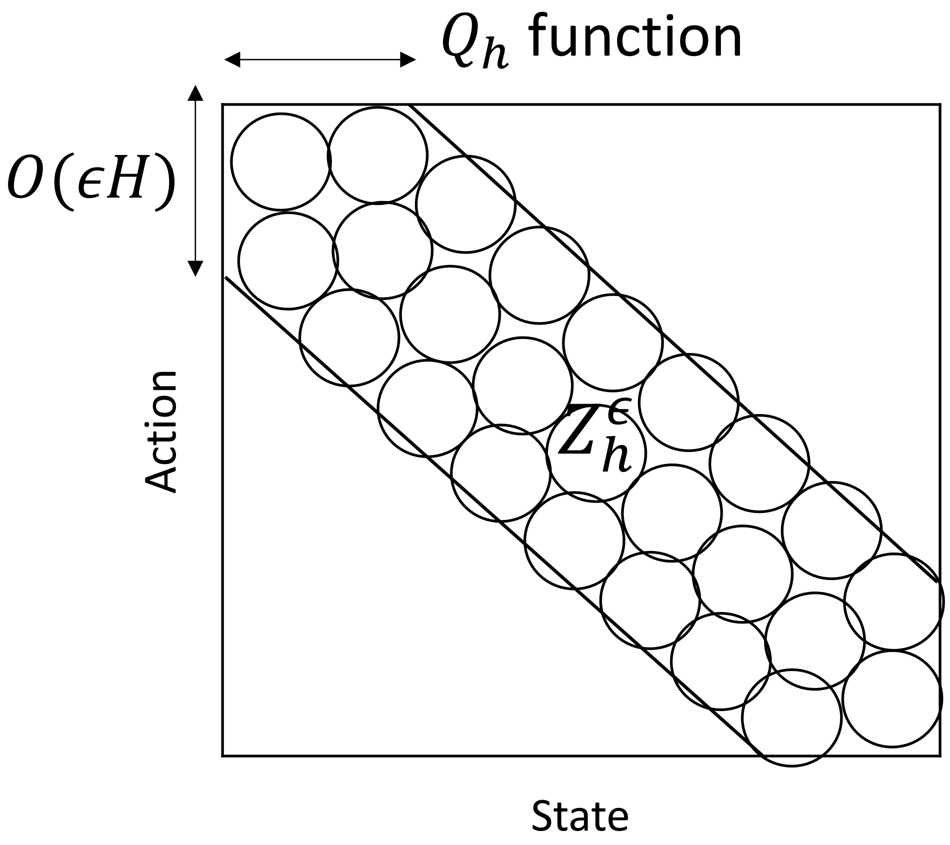
Zooming Dimension



$$\mathrm{GAP}_h(x,a) = V_h^*(x) - Q_h^*(x,a)$$

Z_h^ϵ denote ϵH-near optimal set of state-action pairs

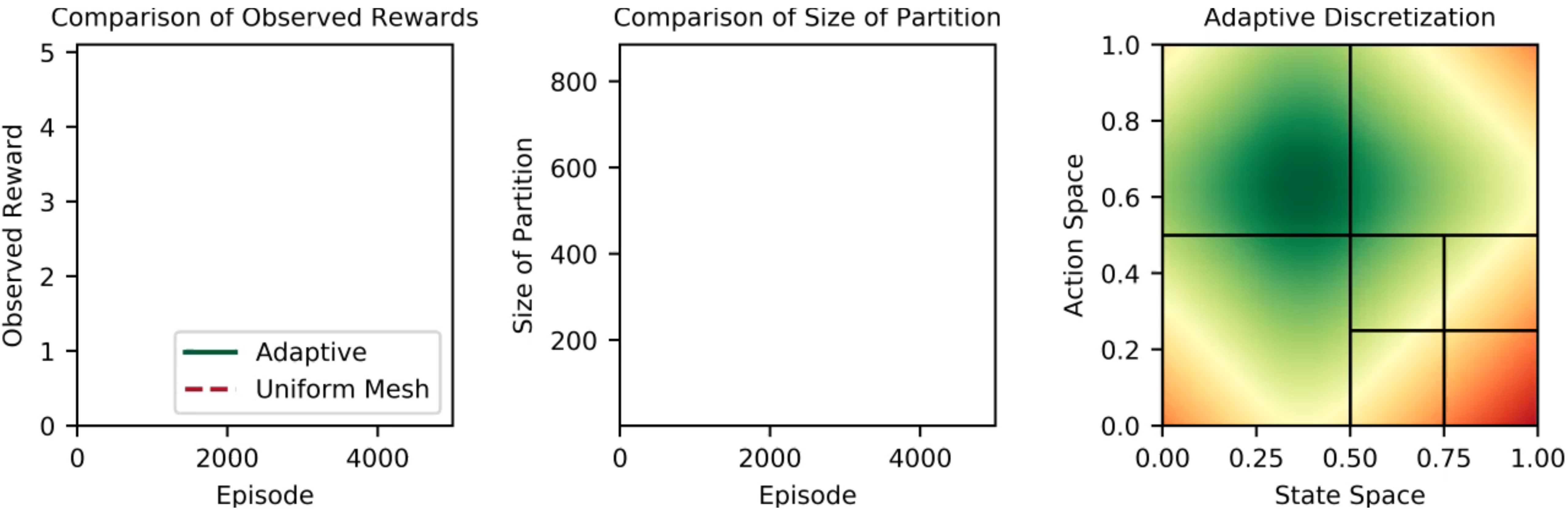
Zooming Dimension



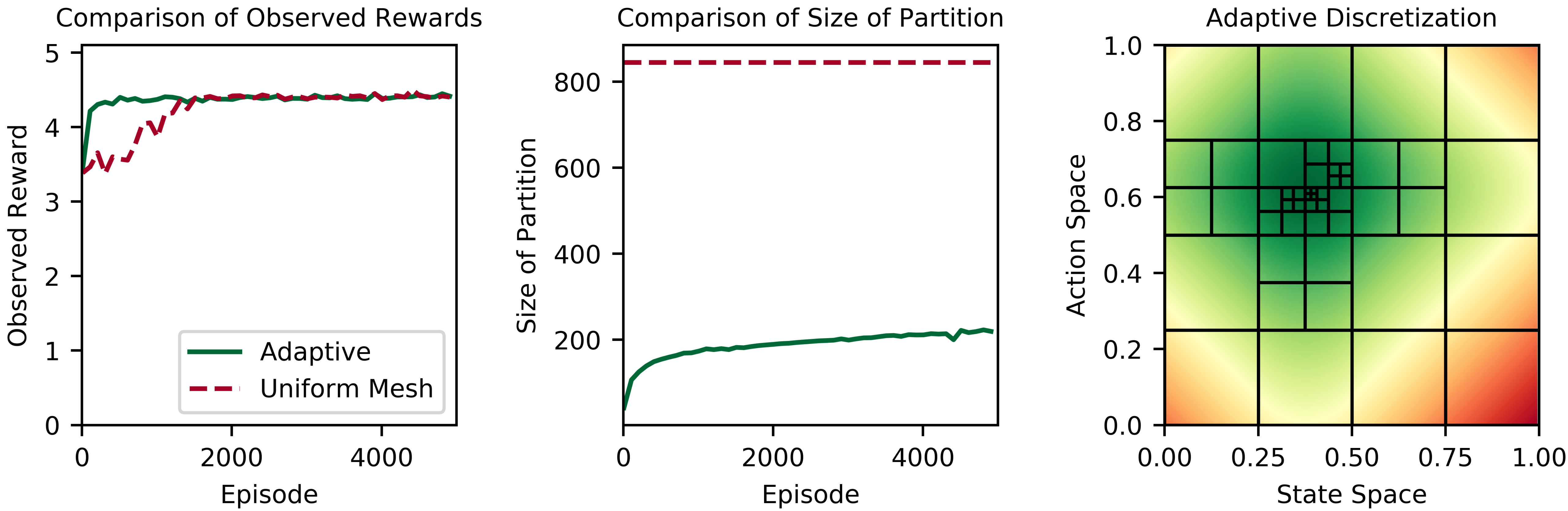
$$\text{GAP}_h(x,a) = V_h^*(x) - Q_h^*(x,a)$$

Z_h^ϵ denote ϵH-near optimal set of state-action pairs

Zooming dimension z_h is min value s.t.
ϵ-covering number of $Z_h^\epsilon = O(\epsilon^{-z_h})$



Adaptive discretization exploits structure in benign problem instances with low zooming dimension; constructing a partition that follows the contours of the value function.



Adaptive discretization exploits structure in benign problem instances with low zooming dimension; constructing a partition that follows the contours of the value function.

Main Format of Algorithm

- Maintain partition of state action space + corresponding estimates
- Given current partition, run original tabular RL algorithm
  - Greedy selection rule w.r.t. optimistic estimates, $\pi_h(x) = \arg\max_{a \in \mathcal{A}} \overline{Q}_h(x, a)$
  - Can plug in model free or model based approximations for Bellman update

$$Q_h^{\pi^*}(x, a) = r_h(x, a) + \mathbb{E}[V_{h+1}^{\pi^*}(x_{h+1}) \mid x_h = x, a_h = a]$$

- Subpartition a region B when it has been chosen "too often",

$$\textsc{bias}(B) := \texttt{diam}(B) \geq \sqrt{1/n(B)} =: \textsc{conf}(B)$$

Model Free Q Learning Algorithm → AdaQL

- Directly estimate Q function and associated value function

- Given observation $(\overbrace{x_h, a_h}^{\in B_h}, r_h, x_{h+1})$, use Q-learning update

$$\overline{Q}_h(B_h) = (1-\alpha_t)\overline{Q}_h(B_h) \; + \alpha_t \left(r_h + \overline{V}_{h+1}(x_{h+1}) + \underbrace{\textsc{bonus}}_{\textsc{bias}(B) + \textsc{conf}(B)}\right)$$

$$\overline{V}_{h+1}(x_{h+1}) = \max_{a \in \mathcal{A}} \overline{Q}_{h+1}(x_{h+1}, a)$$

where $t =$ is # of times action has been selected, $\alpha_t = (H+1)/(H+t)$

Model Based RL Algorithm → AdaMB

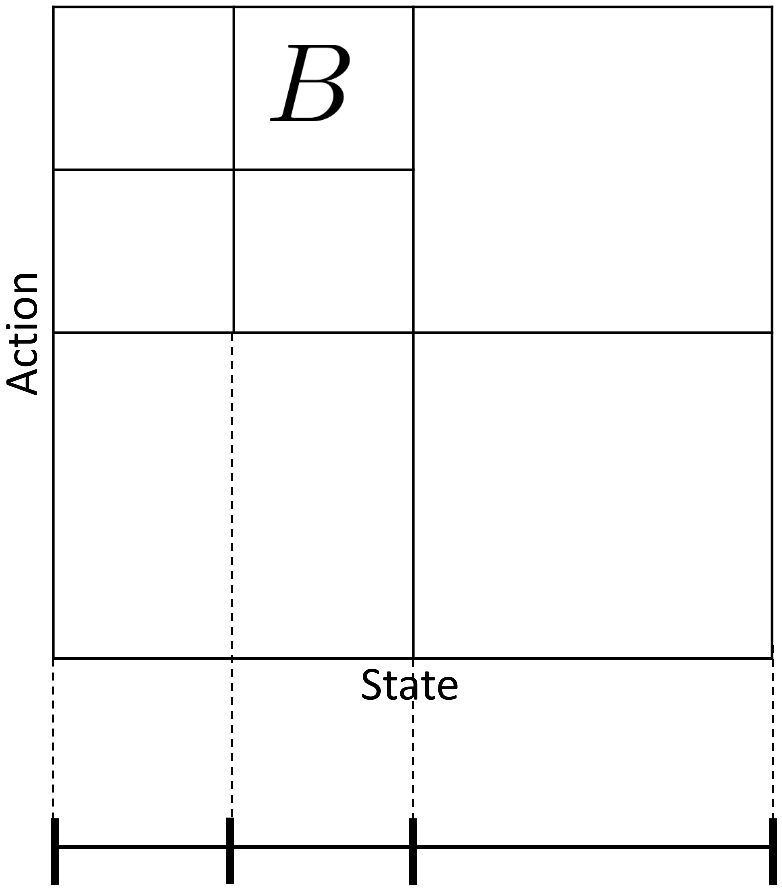
- Maintain empirical estimates for reward fn and transition kernel

- Plug in empirical estimates to the Bellman update equation

$$\overline{Q}_h(B) = \hat{r}_h(B) + \hat{\mathbb{E}}[\overline{V}_{h+1}(x) \mid B] + \text{BONUS}$$
$$\overline{V}_h(x) = \max_{a \in \mathcal{A}} \overline{Q}_h(x, a)$$

- Want to approximate $\hat{r}$ and $\widehat{\mathbb{E}}$ without needing to store all datapoints

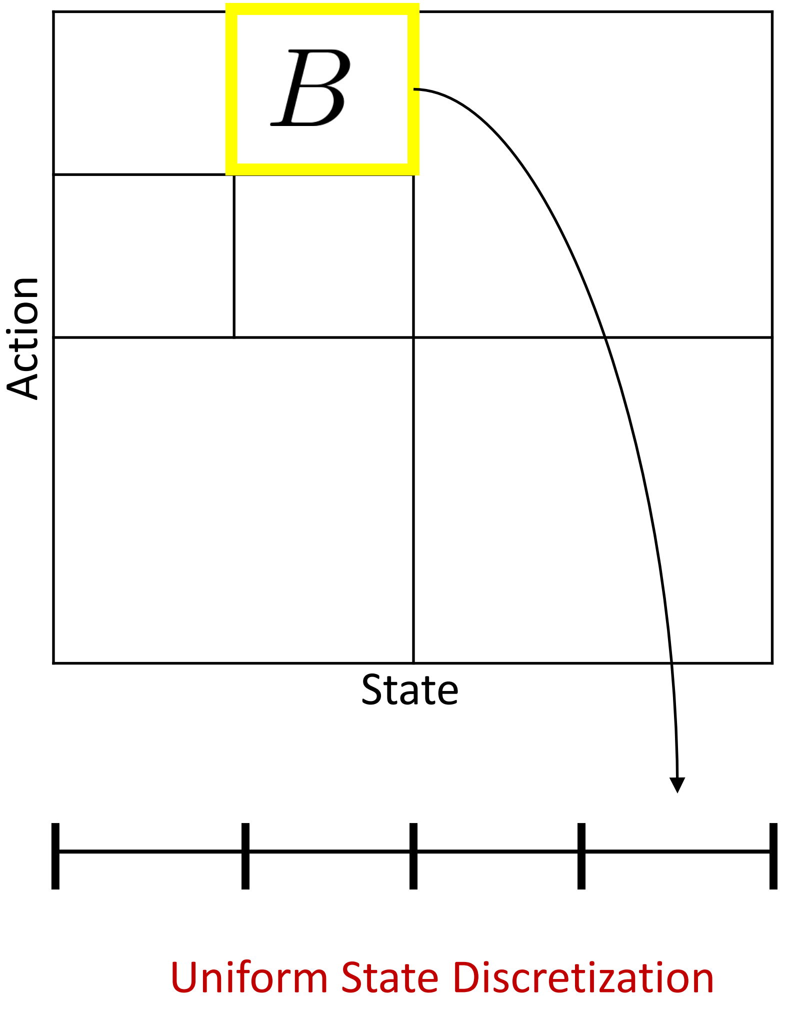
AdaMB: Model Based Adaptive Discretization



Induced State Partition

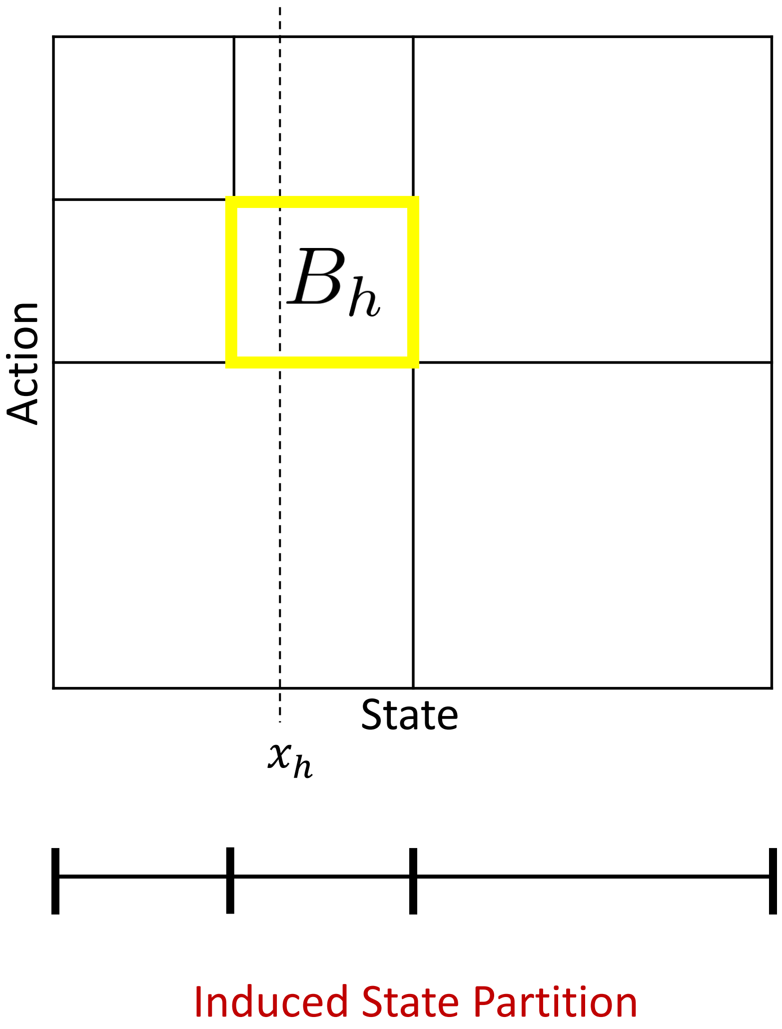
- Maintain partition of the state-action space
- Keep empirical estimates $\hat{r}_h(B), \hat{T}_h(\cdot|B)$

AdaMB: Model Based Adaptive Discretization



- Maintain partition of the state-action space
- Keep empirical estimates $\hat{r}_h(B), \hat{T}_h(\cdot|B)$
- Estimate $\hat{T}_h(\cdot|B)$ over a uniform discretization of the state space at coarseness $\texttt{diam}(B)$
  - Maintains necessary accuracy of estimate while limiting storage complexity

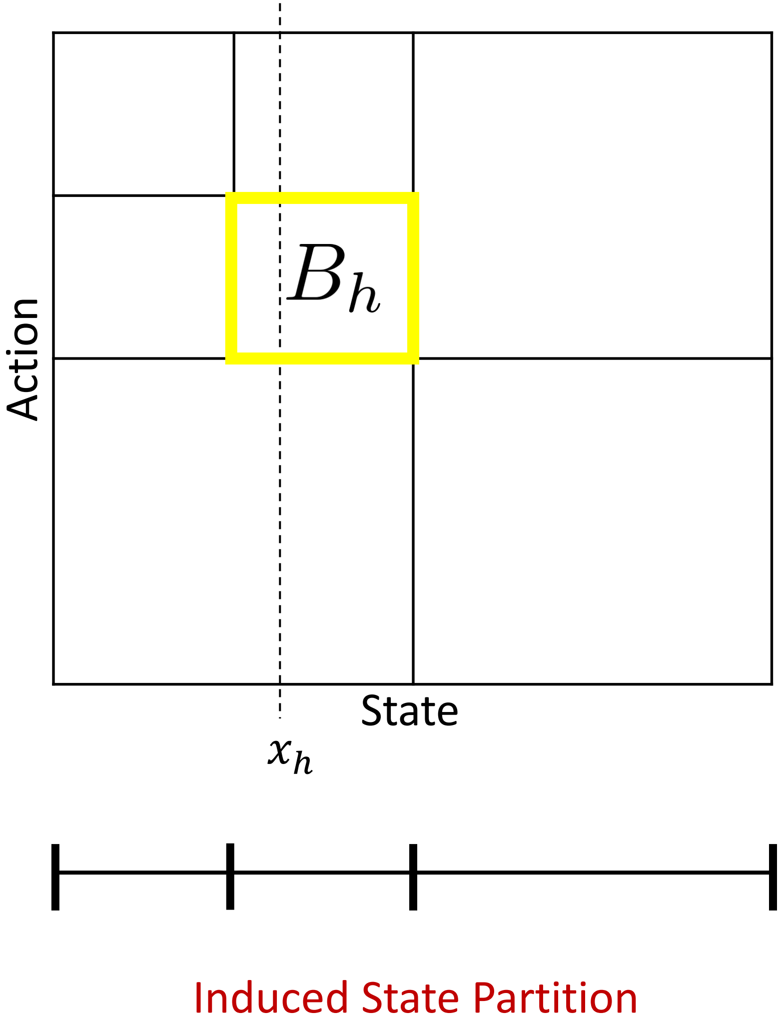
AdaMB: Model Based Adaptive Discretization



- Maintain partition of the state-action space
- Keep empirical estimates $\hat{r}_h(B), \hat{T}_h(\cdot|B)$
- Greedy Selection Rule

$$a_h = \arg\max_{a \in \mathcal{A}} \overline{Q}_h(x_h, a_h)$$

AdaMB: Model Based Adaptive Discretization



- Maintain partition of the state-action space
- Keep empirical estimates $\hat{r}_h(B), \hat{T}_h(\cdot|B)$
- Greedy Selection Rule
- Compute empirical Bellman update

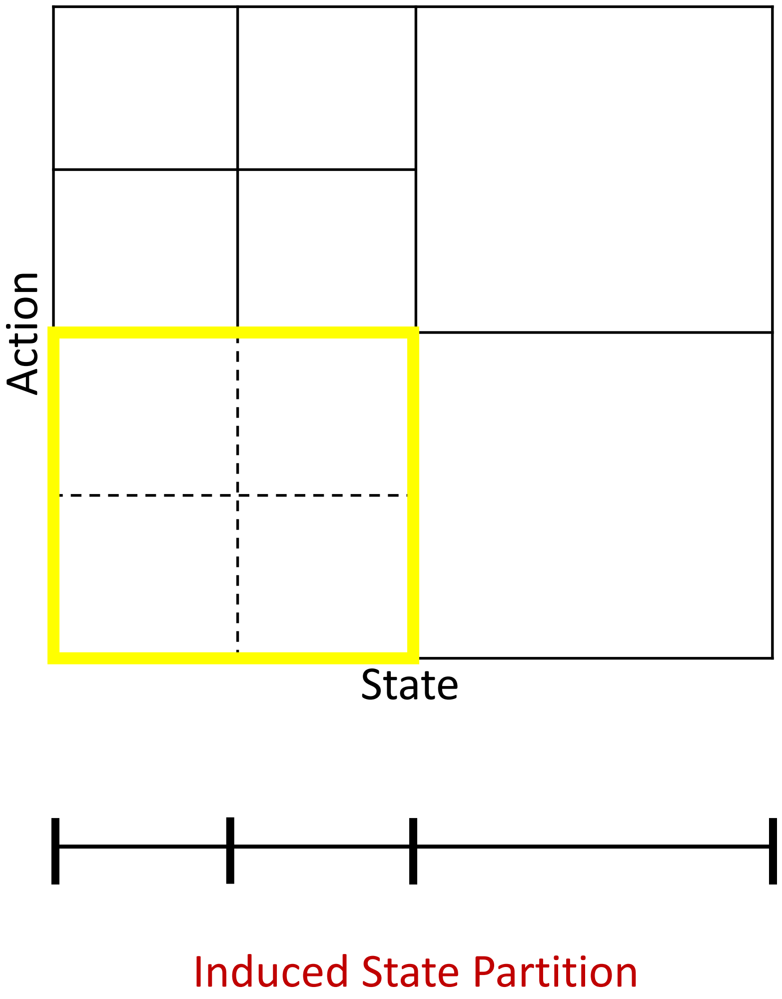
$$\overline{Q}_h(B) = \hat{r}_h(B) + \hat{\mathbb{E}}[\overline{V}_{h+1}(x) \mid B] + \text{BONUS}$$

$$\overline{V}_h(x) = \max_{a \in \mathcal{A}} \overline{Q}_h(x, a)$$

where $\text{BONUS} = \text{BIAS}(B) + \text{CONF}(B)$

concentration of $\hat{T}$ may depend on d_S

AdaMB: Model Based Adaptive Discretization



- Maintain partition of the state-action space
- Keep empirical estimates $\hat{r}_h(B), \hat{T}_h(\cdot|B)$
- Greedy Selection Rule
- Compute empirical Bellman update
- Subpartition region if bias > confidence radius
  - New regions have half diameter of parent, inherit all estimates of reward, transition, and counts

*we don't need to keep all samples; due to inherited estimates, $\hat{r}$ and $\hat{T}$ are not standard empirical estimates; we need to account for this in the analysis

Informal Theorem [SinclairBanerjeeYu2019] [SinclairWangJainBanerjeeYu2020]

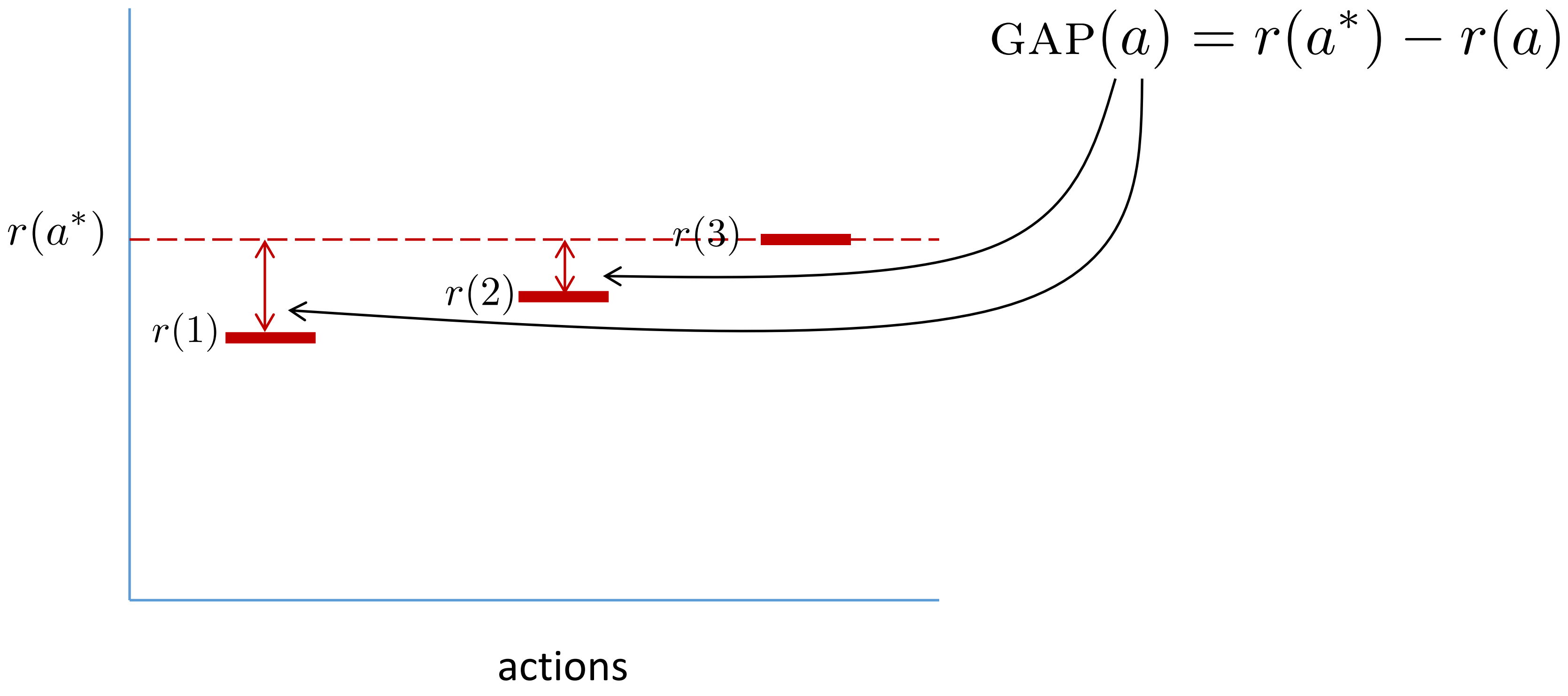
We propose AdaQL (model free) and AdaMB (model based) that achieve

$$\textsc{Regret}(K) \lesssim \begin{cases} \textsc{AdaQL}: & H^{5/2} K^{\frac{z+1}{z+2}} \\ \textsc{AdaMB}: & H^{3/2} K^{\frac{z+d_S-1}{z+d_S}} \quad d_S > 2 \\ \textsc{AdaMB}: & H^{3/2} K^{\frac{z+1}{z+2}} \quad d_S \leq 2 \end{cases}$$

where z is zooming dim, d_S is dim of state space.

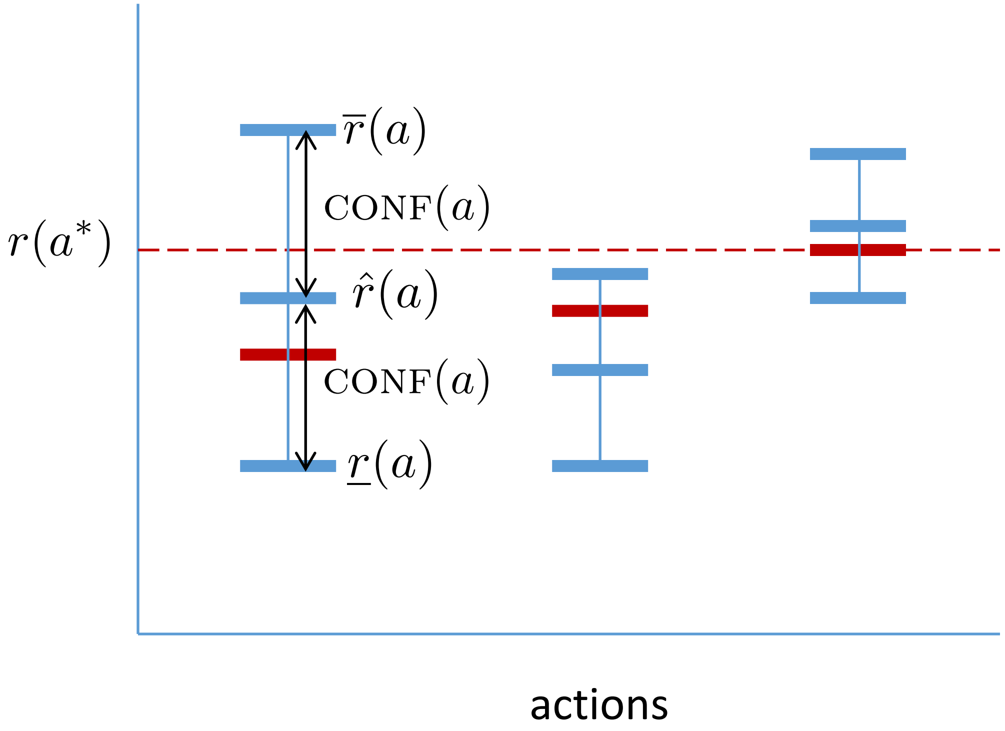
Proof Sketch – Zooming Dimension Analysis

- Instance specific analysis for finite armed bandits



Proof Sketch – Zooming Dimension Analysis

- Instance specific analysis for finite armed bandits



$$\text{GAP}(a) = r(a^*) - r(a)$$

$$\overline{r}(a) = \hat{r}(a) + \text{CONF}(a)$$

$$0 \leq \overline{r}(a) - r(a) \leq 2\text{CONF}(a) \approx \sqrt{1/n(a)}$$

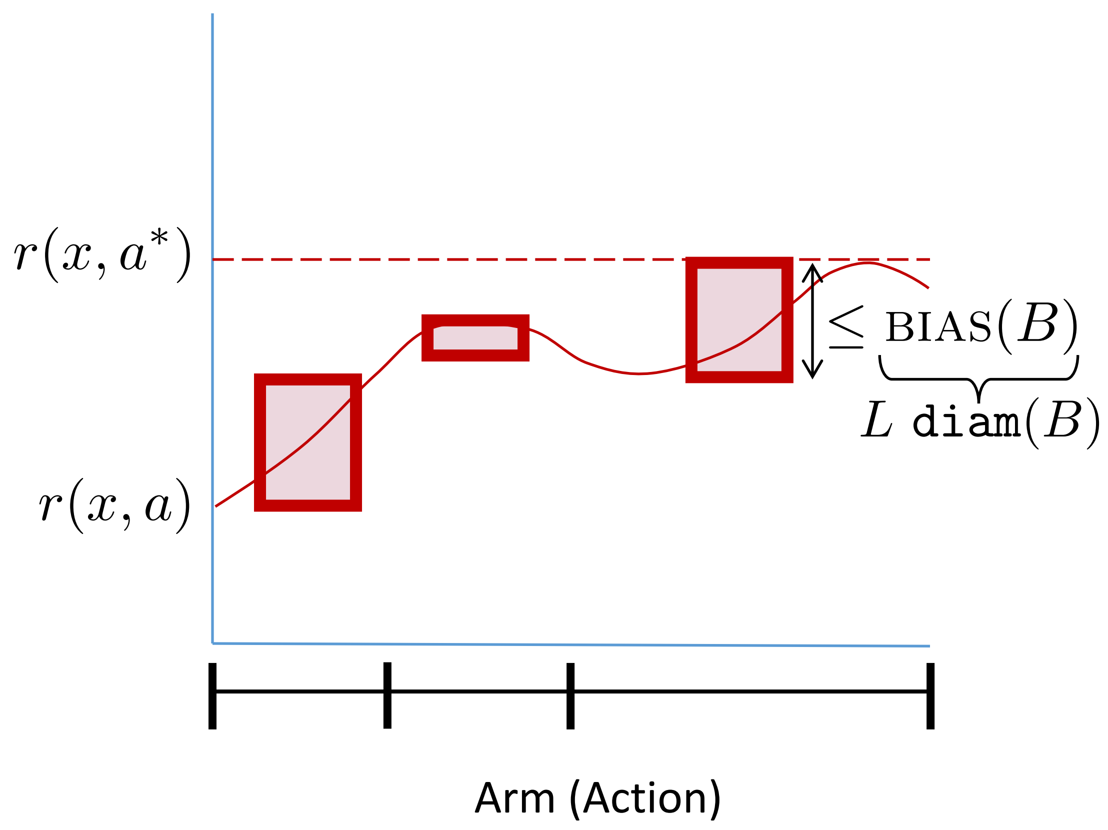
By optimistic selection, a is never chosen again once

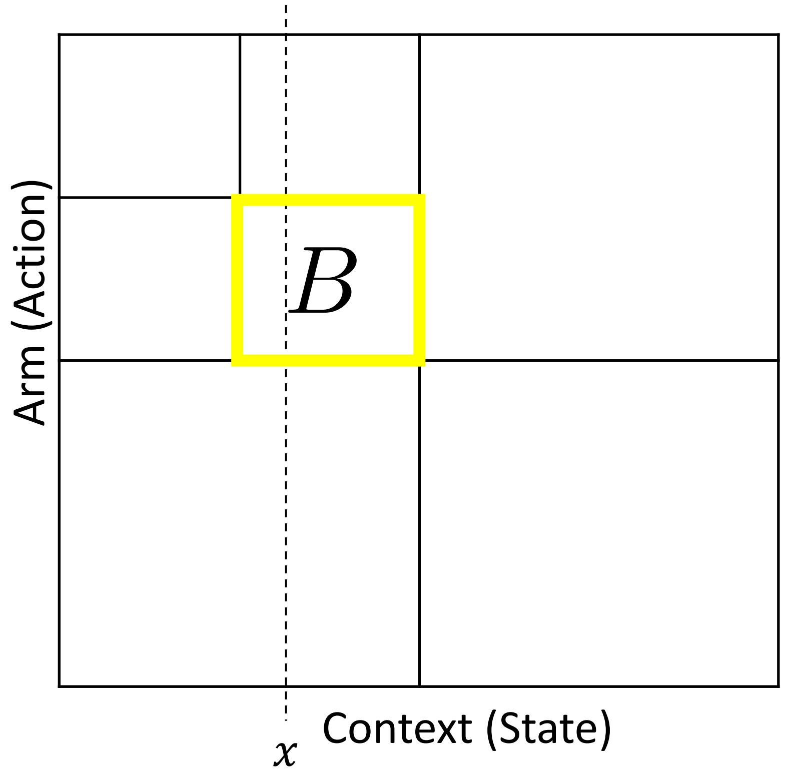
$$2\text{CONF}(a) \leq \text{GAP}(a) \implies \overline{r}(a) \leq r(a^*) \leq \overline{r}(a^*)$$

implies that $n(a) \lesssim 1/\text{GAP}(a)^2$

Proof Sketch – Zooming Dimension Analysis

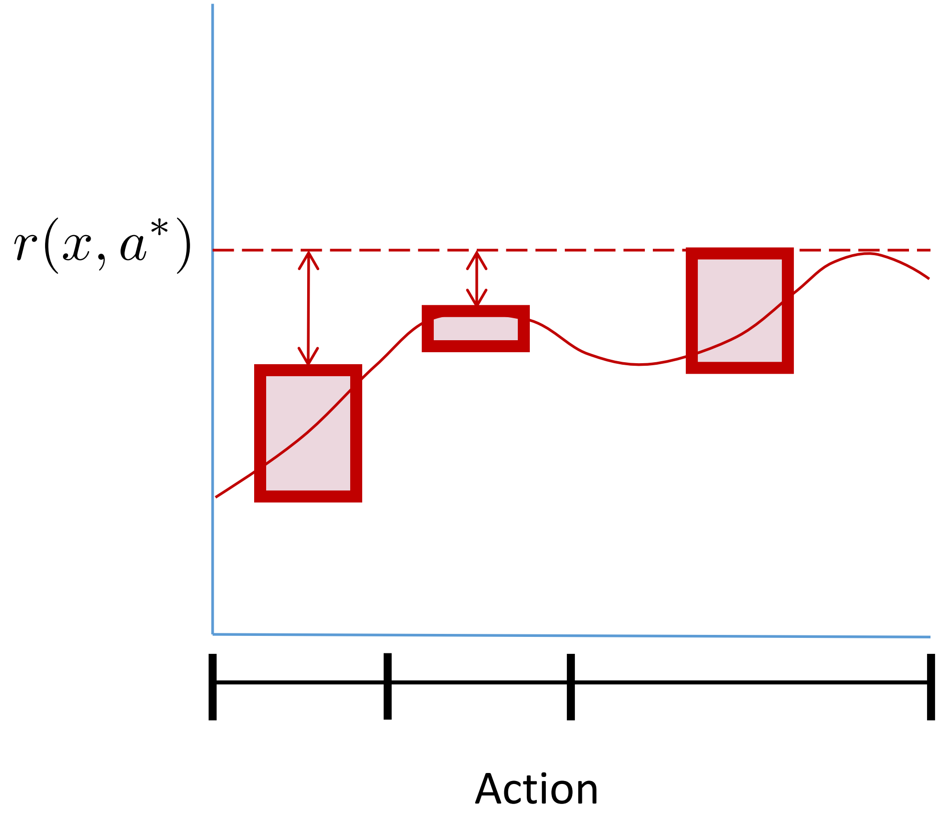
- For contextual bandits with adaptive discretization





Proof Sketch – Zooming Dimension Analysis

- For contextual bandits with adaptive discretization

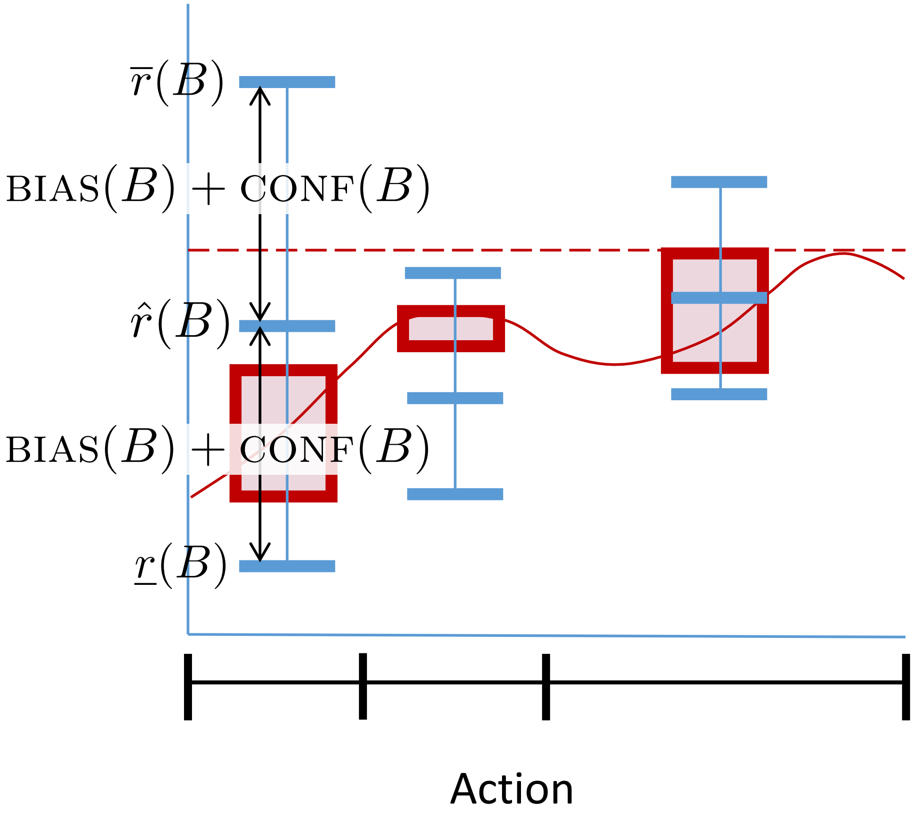


$$\mathrm{GAP}(x,a) = r(x,a^*) - r(x,a)$$

$$\mathrm{GAP}(B) = \min_{(x,a)\in B} \mathrm{GAP}(x,a)$$

Proof Sketch – Zooming Dimension Analysis

- For contextual bandits with adaptive discretization



$$\textsc{gap}(x, a) = r(x, a^*) - r(x, a)$$

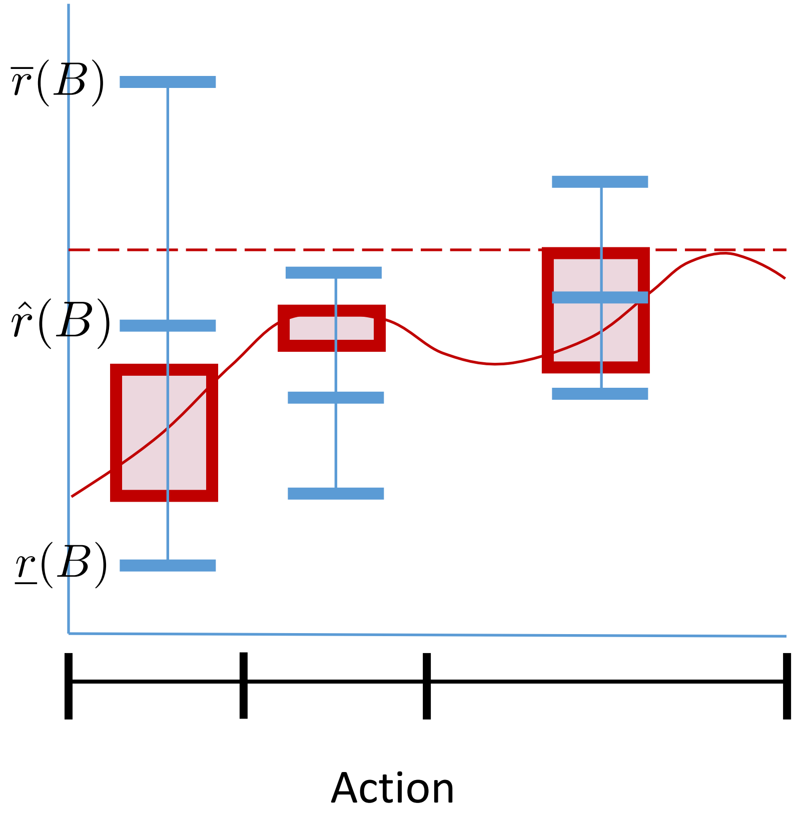
$$\textsc{gap}(B) = \min_{(x,a)\in B} \textsc{gap}(x, a)$$

$$\overline{r}(B) = \hat{r}(B) + \textsc{bias}(B) + \textsc{conf}(B)$$

$$0 \leq \overline{r}(B) - r(x, a) \leq 2\textsc{bias}(B) + 2\textsc{conf}(B)$$

Proof Sketch – Zooming Dimension Analysis

- For contextual bandits with adaptive discretization



$$0 \leq \overline{r}(B) - r(x,a) \leq 2\textsc{bias}(B) + 2\textsc{conf}(B)$$

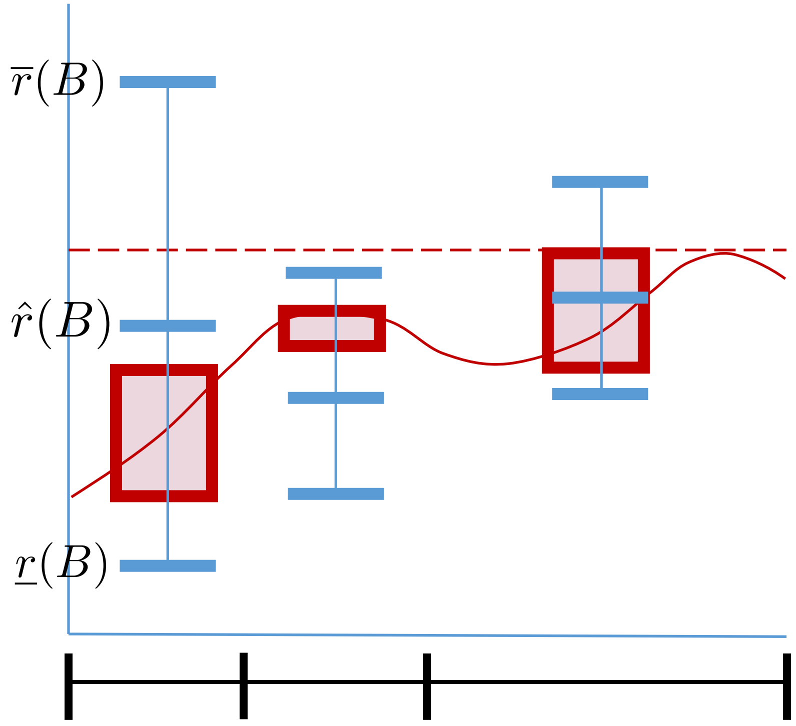
Region B is never chosen again once it is either

- Subpartitioned, i.e. $\textsc{bias}(B) \geq \textsc{conf}(B)$
- Suboptimal, i.e. $2\textsc{bias}(B) + 2\textsc{conf}(B) \leq \textsc{gap}(B)$

$$\implies \overline{r}(B) \leq r(x,a^*) \leq \overline{r}(B^*)$$

Proof Sketch – Zooming Dimension Analysis

- For contextual bandits with adaptive discretization



$$0 \leq \overline{r}(B) - r(x,a) \leq 2\textsc{bias}(B) + 2\textsc{conf}(B)$$

Region B is never chosen again once it is either

- Subpartitioned, i.e. $\textsc{bias}(B) \geq \textsc{conf}(B)$
- Suboptimal, i.e. $2\textsc{bias}(B) + 2\textsc{conf}(B) \leq \textsc{gap}(B)$

Implies that $n(B) \lesssim \min\left(1/\mathtt{diam}(B)^2, 1/\textsc{gap}(B)^2\right)$

Proof Sketch – Zooming Dimension Analysis

- Property that "suboptimal regions are not selected often" relies on

$$0 \leq \bar{r}(B) - r(x,a) \leq 2\textsc{bias}(B) + 2\textsc{conf}(B)$$

$$\implies \textsc{gap}(B_t) \leq 2\textsc{bias}(B_t) + 2\textsc{conf}(B_t) \lesssim \texttt{diam}(B)$$

- Regret is bounded by sum of gap terms over "regions"
- Number of regions is bounded by zooming dimension

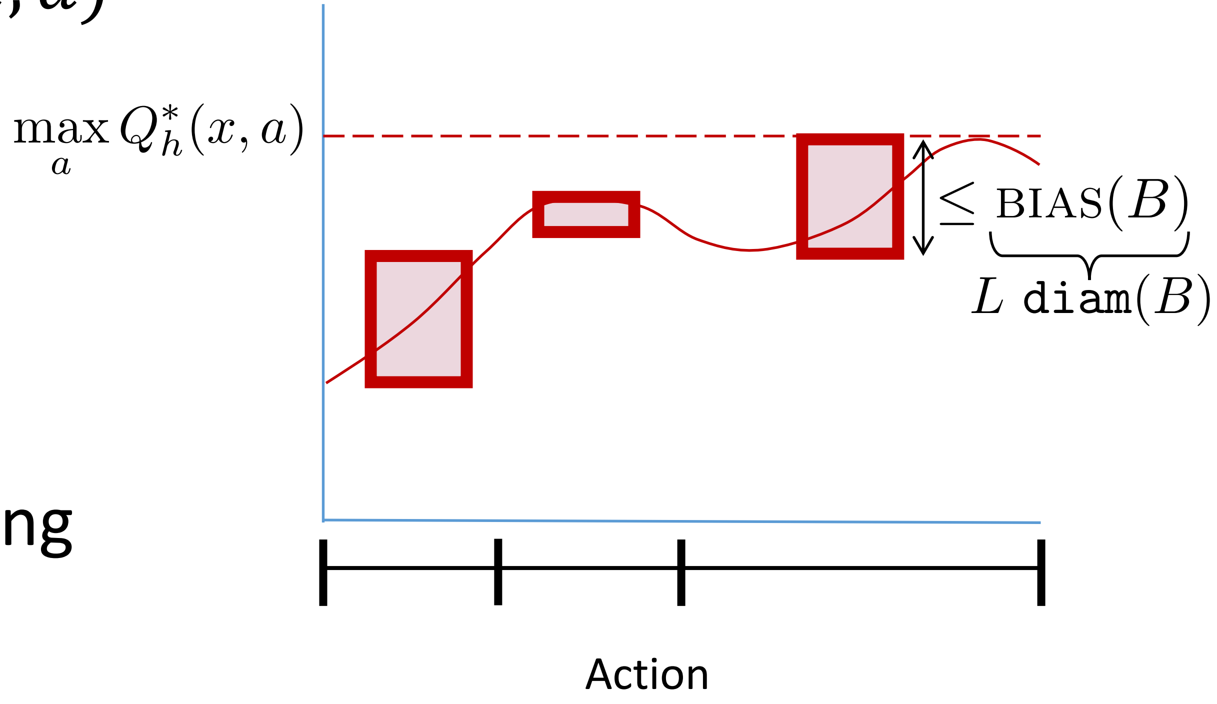
$$\textsc{regret} \leq \sum_{r \geq r_0} \sum_{B:\texttt{diam}(B)=r} \underbrace{\textsc{gap}(B) n(B)}_{\lesssim \frac{\mathbb{I}(\textsc{gap}(B) \leq \texttt{diam}(B))}{\texttt{diam}(B)}} + r_0 K$$

$$\lesssim K^{\frac{z+1}{z+2}}$$

Proof Sketch – Zooming Dimension Analysis

- In reinforcement learning we sample from $Q_h^{\widehat{\pi}}(x,a)$, which does not give an unbiased estimate for $Q_h^*(x,a)$

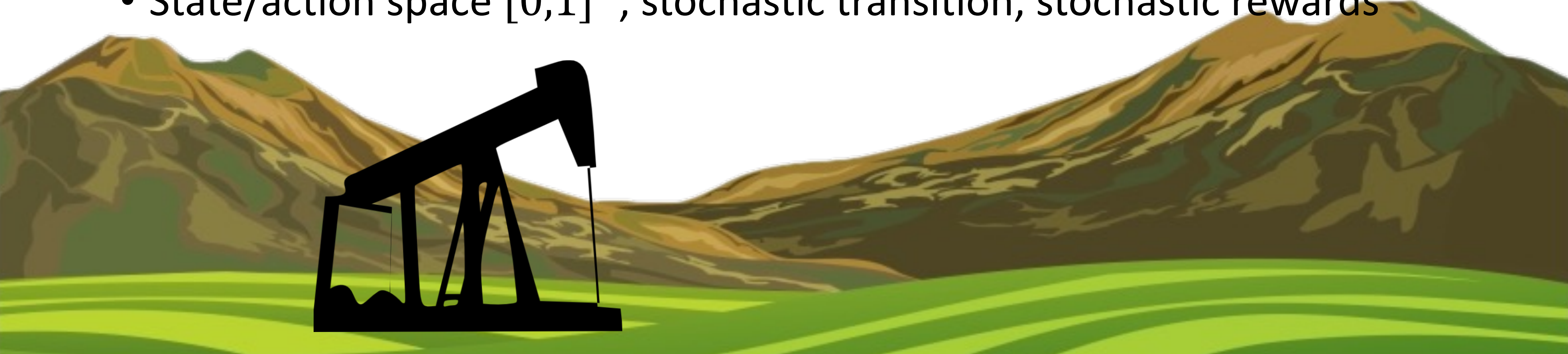
$$
\begin{aligned}
0 &\le \overline{Q}_h(B) - Q_h^*(x,a) \\
&\le 2\textsc{conf}(B) + 2\textsc{bias}(B) \\
&\quad + f\left(\overline{Q}_{h+1} - Q_{h+1}^*\right)
\end{aligned}
$$



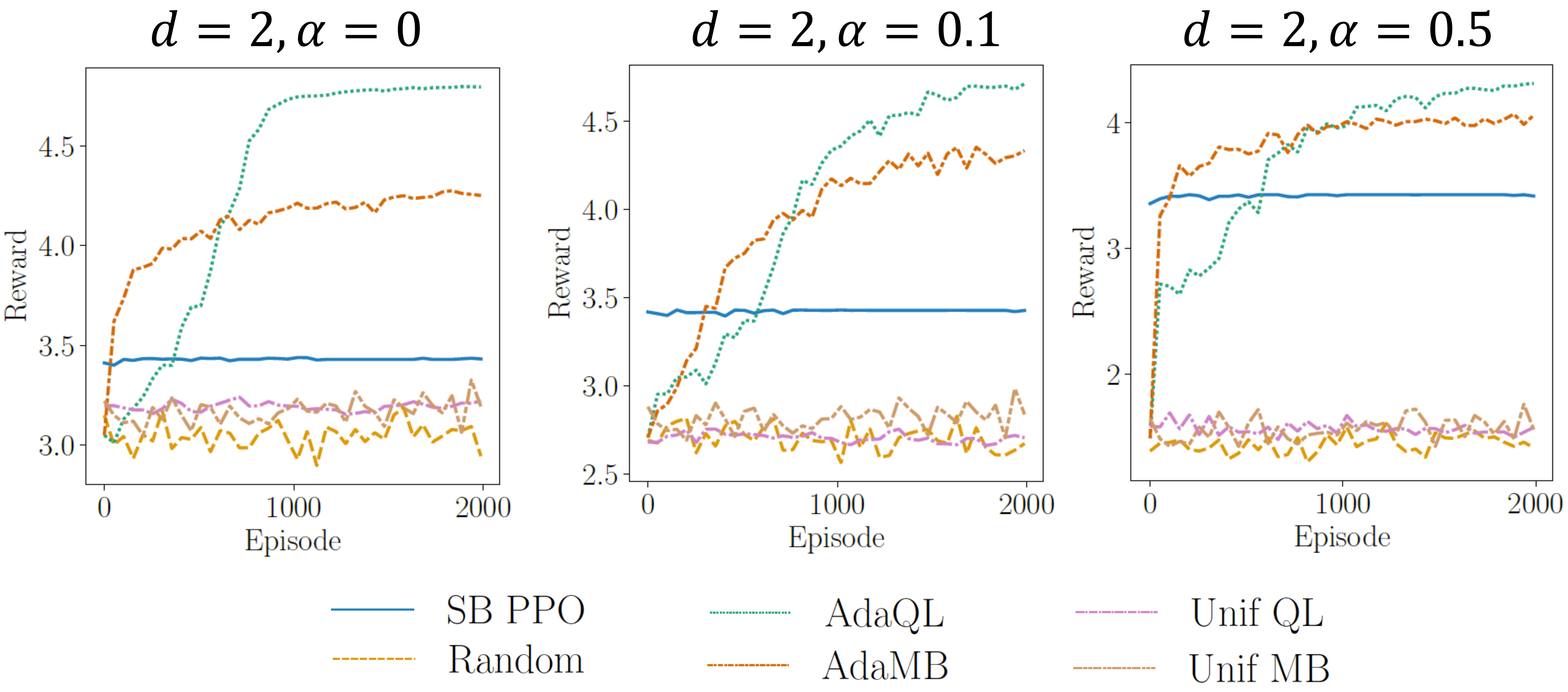
- Analysis requires carefully accounting of one-step vs. future regret

Empirical Results – Oil Discovery

- An agent surveys a (d-dim) map in search of hidden `oil deposits'
- Transportation cost proportional to distance moved, weighted by α
- Transitions perturbed by uneven land
- Surveying land produces noisy estimates of the true value
- State/action space $[0,1]^d$, stochastic transition, stochastic rewards



Empirical Results – Oil Discovery

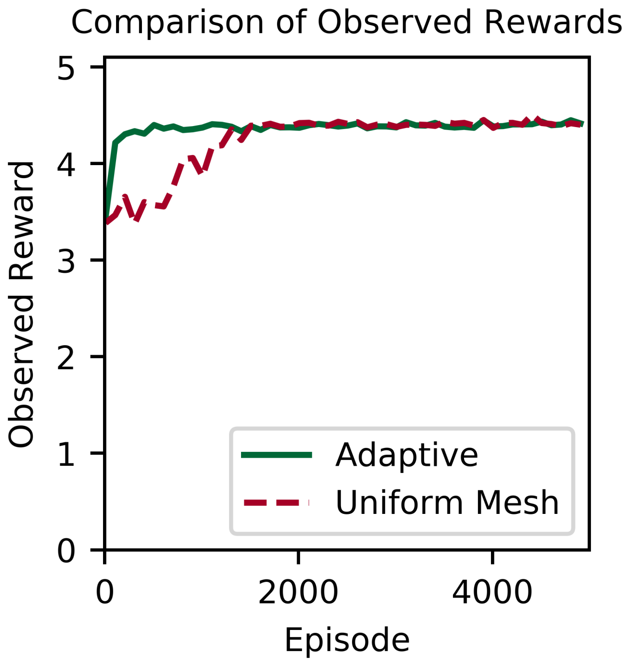


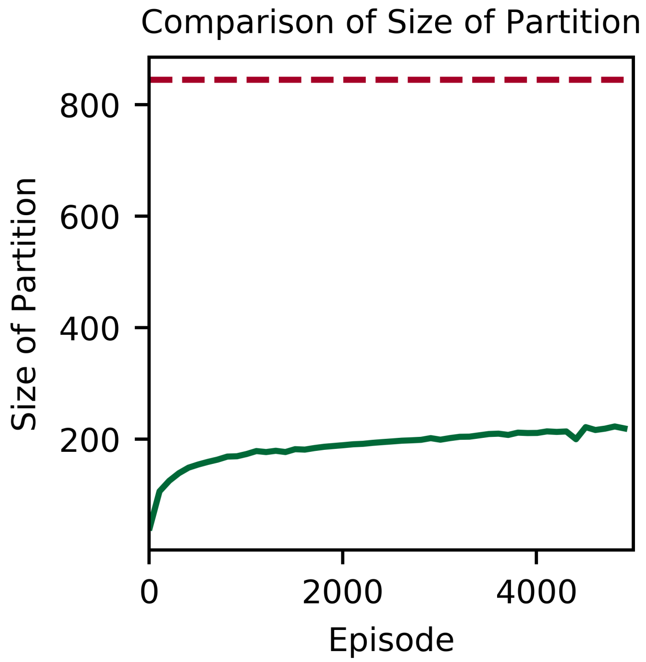
Questions?

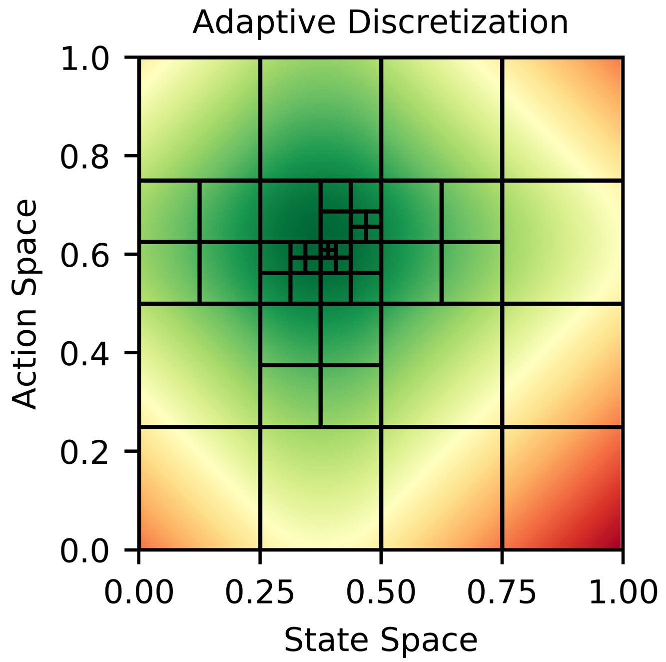
Sean R. Sinclair, Siddhartha Banerjee, Christina Lee Yu. “Adaptive Discretization for Online Reinforcement Learning.” *Operations Research,* 2022.

Sean R. Sinclair, Tianyu Wang, Gauri Jain, Siddhartha Banerjee, Christina Lee Yu. “Adaptive Discretization for Model-Based Reinforcement Learning.” *Advances in Neural Information Processing Systems,* 2020.

Sean R. Sinclair, Siddhartha Banerjee, Christina Lee Yu. “Adaptive Discretization for Episodic Reinforcement Learning in Metric Spaces.” *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, 2019.







Part II: Exploiting latent low rank structure in action-value function using matrix completion

Joint work with Tyler Sam and Yudong Chen

Sample Complexity with Generative Model

- Policy π and $Q = \{Q_h\}_{\{h\in[H]\}}$ are ϵ-optimal if for all x, a, h,

$$|V_h^{\pi^*}(x) - V_h^{\pi}(x)| \leq \epsilon \quad \text{and} \quad |Q_h^{\pi^*}(x,a) - Q_h(x,a)| \leq \epsilon$$

- Optimal sample complexity to find an ϵ-optimal policy is

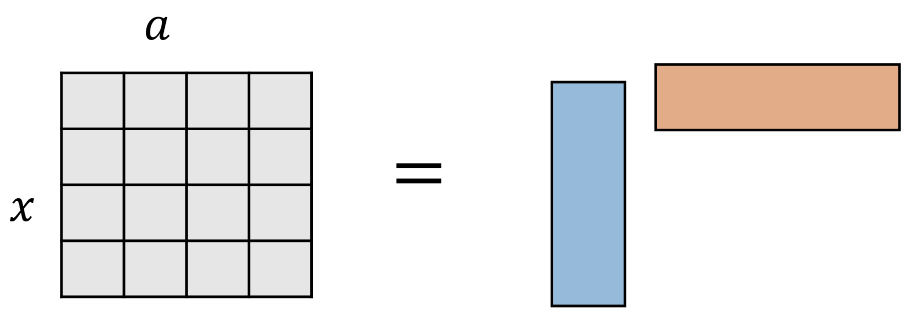
$$\tilde{\Theta}\left(\frac{|S||A|H^3}{\epsilon^2}\right)$$

[Azar, Munos, Kappen, 2012] [Sidford, Wang, Wu, Yang, Ye, 2018]

- Need to sample from each $(x, a) \in S\times A$ to construct estimate $\hat{Q}_h$
- Q: can we reduce sample complexity if $\hat{Q}_h$ low rank?

Motivating low rank structure

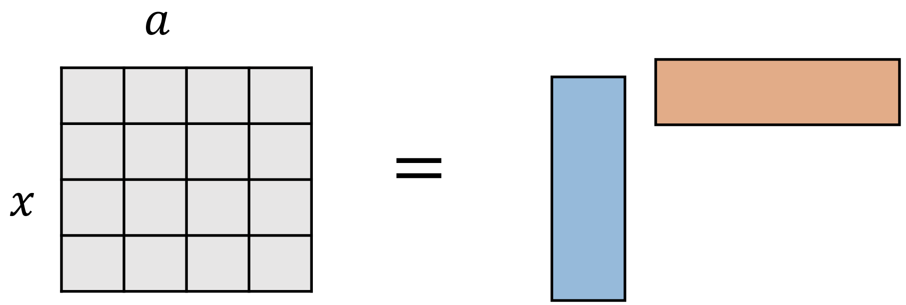
$Q_h^{\pi^*} =$



- Large discrete state/action space with latent low dimension structure
- If Q function is approximated by smooth continuous function, then it is also approximately low rank [Udell Townsend 2017]
- E.g. recommendation systems where states are related to customers and actions are related to products

Reducing Sample Complexity

$Q_h^{\pi^*} =$

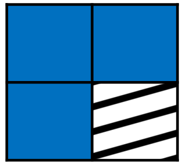


- If $Q_h^{\pi^*}$ were low rank, could we sample from only $O(S + A)$ state-action pairs and use matrix estimation to construct $\hat{Q}_h$?
- [Shah-Song-Xu-Yang, 2020] show sample complexity of $\tilde{O}\left(\frac{|S|+|A|}{\epsilon^2}\right)$
  … but requires bounded horizon, e.g. $H < 20$; is this fundamental?

Information Theoretic Lower Bound [Sam, Chen, Yu, 2023]

Setup: S = A = {1,2} and assume $Q_h^{\pi^*}$ is rank 1 for all $h \in [H]$

Samples from MDP are constrained to $(x, a) \in \{(1,1), (1,2), (2,1)\}$ s.t. algorithm needs to use low rank structure to estimate $\hat{Q}_h^{\pi}(2,2)$



Result: There exists instances for which learning a 1/8-optimal policy with probability at least 0.9 requires $\Omega(4^H)$ samples

- Only Q^* low rank is too weak, as $Q^{\hat{\pi}}$ may not be low rank
- Estimation error in last step is amplified exponentially over horizon
- Need stronger low rank conditions on MDP

Summary of Results

	MDP Setting	Sample Complexity
this work	Low-rank Q_h^* & suboptimality gap $\Delta_{\min} > 0$	$\tilde{O}\left(\frac{d^5(\|S\|+\|A\|)H^4}{\Delta_{\min}^2}\right)^{\dagger}$
this work	ϵ-optimal policies have low-rank Q_h^π	$\tilde{O}\left(\frac{d^5(\|S\|+\|A\|)H^6}{\epsilon^2}\right)^{\dagger}$
this work	Transition kernels and rewards are low-rank	$\tilde{O}\left(\frac{d^5(\|S\|+\|A\|)H^5}{\epsilon^2}\right)^{\ddagger}$
	Low-rank Q_h^* & constant horizon [Shah et al, 2020]	$\tilde{O}\left(\frac{\|S\|+\|A\|}{\epsilon^2}\right)^{\ddagger}$
	Tabular MDP with homogeneous rewards [Sidford et al, 2018]	$\tilde{\Theta}\left(\frac{\|S\|\|A\|H^3}{\epsilon^2}\right)$

† Achieved by Low Rank Monte Carlo Policy Iteration (LR-MCPI)

‡ Achieved by Low Rank Empirical Value Iteration (LR-EVI)

Empirical Dynamic Programming [Haskell et al. 2016]

- Compute via backwards recursion starting with $\hat{V}_{H+1}(x) = 0 \; \forall \; x \in \mathcal{S}$
- Given $\hat{V}_{h+1}$ or $\hat{\pi}_{h+1}, \hat{\pi}_{h+2} \ldots \hat{\pi}_H$, compute $\hat{Q}_h$ via Bellman update,

$$\hat{Q}_h(x,a) = r_h(x,a) + \hat{\mathbb{E}}[\hat{V}_{h+1}(x_{h+1}) \mid x_h = x, a_h = a]$$

$$\hat{Q}_h(x,a) = r_h(x,a) + \hat{\mathbb{E}}\Big[\sum_{\ell=h+1}^{H} r_\ell(x_\ell, \hat{\pi}_\ell(x_\ell)) \,\Big|\, x_h = x, a_h = a\Big]$$

Approximate expectations with empirical samples

- Compute $\hat{V}_h(x) = \max_{a \in \mathcal{A}} \hat{Q}_h(x,a)$ and $\hat{\pi}_h(x) = \arg\max_{a \in \mathcal{A}} \hat{Q}_h(x,a)$

Low Rank + Empirical Dynamic Programming

- Compute via backwards recursion starting with $\hat{V}_{H+1}(x) = 0 \; \forall \; x \in \mathcal{S}$
- Given $\hat{V}_{h+1}$ or $\hat{\pi}_{h+1}, \hat{\pi}_{h+2} \ldots \hat{\pi}_H$, compute $\hat{Q}_h$ for $(x, a) \in \Omega$ via empirical Bellman update, replacing expectations with samples
- Use matrix completion to estimate Q function for all (x, a)

$$\{\hat{Q}_h(x,a)\}_{(x,a)\in\Omega} \longrightarrow \boxed{\text{matrix estimation}} \longrightarrow \bar{Q}_h(x,a) \;\; \forall \; (x,a)$$

- Compute $\hat{V}_h(x) = \max_{a \in \mathcal{A}} \bar{Q}_h(x,a)$ and $\hat{\pi}_h(x) = \arg\max_{a \in \mathcal{A}} \bar{Q}_h(x,a)$

Need low rank assumptions that give guarantees on relationship of $\hat{Q}$ relative to a meaningful low rank matrix

Low Rank Monte Carlo Policy Iteration (LR-MCPI)

[Sam, Chen, Y., 2022]

- Compute via backwards recursion starting with $\hat{V}_{H+1}(x) = 0 \; \forall \; x \in \mathcal{S}$
- Given $\hat{\pi}_{h+1}, \hat{\pi}_{h+2} \dots \hat{\pi}_H$, compute $\hat{Q}_h$ for $(x, a) \in \Omega$ via

$$\hat{Q}_h(x,a) = r_h(x,a) + \hat{\mathbb{E}}\left[\sum_{\ell=h+1}^{H} r_\ell(x_\ell, \hat{\pi}_\ell(x_\ell)) \,\middle|\, x_h = x, a_h = a\right]$$

Monte Carlo policy evaluation – N_h full trajectory rollouts for each $(x, a) \in \Omega$

- Use matrix completion to estimate Q fn for all (x, a)
- Compute $\hat{V}_h(x) = \max_{a \in \mathcal{A}} \bar{Q}_h(x,a)$ and $\hat{\pi}_h(x) = \arg\max_{a \in \mathcal{A}} \bar{Q}_h(x,a)$

Need low rank assumptions that give guarantees on relationship of $\hat{Q}$ relative to a meaningful low rank matrix

Low Rank Empirical Value Iteration (LR-EVI)

[Shah et al. 2020] [Yang et al. 2020]

- Compute via backwards recursion starting with $\hat{V}_{H+1}(x) = 0 \; \forall \; x \in \mathcal{S}$
- Given $\hat{V}_{h+1}$, compute $\hat{Q}_h$ for $(x, a) \in \Omega$ via

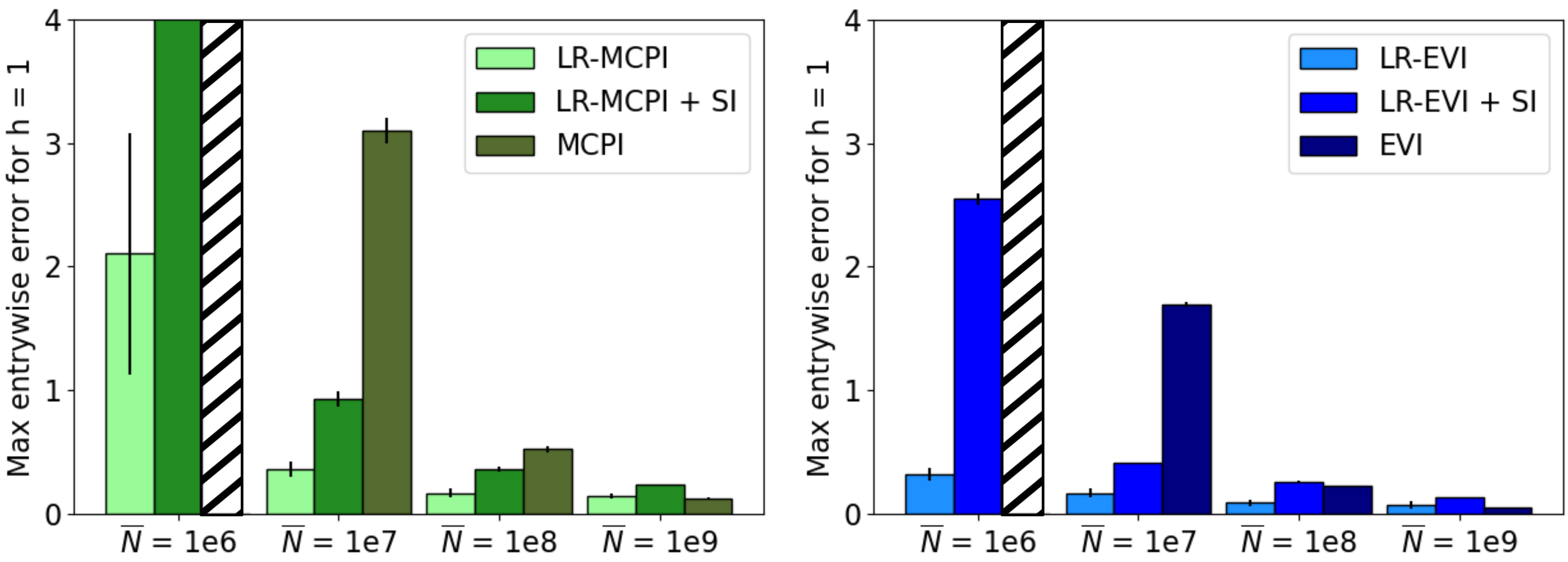
$$\hat{Q}_h(x,a) = r_h(x,a) + \hat{\mathbb{E}}\left[\hat{V}_{h+1}(x_{h+1}) \;\middle|\; x_h = x, a_h = a \right]$$

empirical value iteration – N_h samples from $T_h(\cdot \mid x, a)$ for each $(x, a) \in \Omega$

- Use matrix completion to estimate Q fn for all (x, a)
- Compute $\hat{V}_h(x) = \max_{a \in \mathcal{A}} \bar{Q}_h(x,a)$ and $\hat{\pi}_h(x) = \arg\max_{a \in \mathcal{A}} \bar{Q}_h(x,a)$

Need low rank assumptions that give guarantees on relationship of $\hat{Q}$ relative to a meaningful low rank matrix

Empirical Results – Oil Discovery



How to design RL algorithms that **provably** and **efficiently** exploit structure arising in real-world systems?

1. What types of structure are reasonable and common?

   E.g. smoothness, low rank, exogenous input-driven dynamics, weakly coupled states, …

2. What type of information is commonly available?

   E.g. historical traces of auxiliary variables or historical trajectories, …

3. How to exploit it to lead to efficient learning?

RL simulators (… beyond AIGym …)

- Park (computer systems) – https://github.com/park-project/park [Mao et al 2019]
- ORGym (operations) – https://github.com/hubbs5/or-gym [Hubbs et al 2020]
- MARO (operations) – https://github.com/microsoft/maro [Jiang et al 2020]
- ORSuite (operations) – https://github.com/cornell-orie/ORSuite [Archer et al 2022]
- SustainGym (sustainability) – https://chrisyeh96.github.io/sustaingym/ [Yeh et al 2023]