Harnessing Structures for Value-Based Planning and Reinforcement Learning

Yuzhe Yang

Guo Zhang, Zhi Xu, Dina Katabi

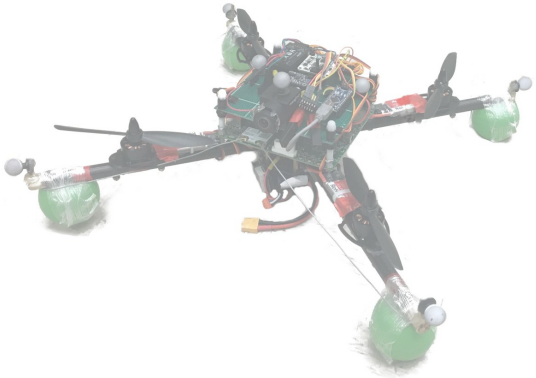
New Planning and Deep RL Framework

that exploits the “*global structure*” in tasks

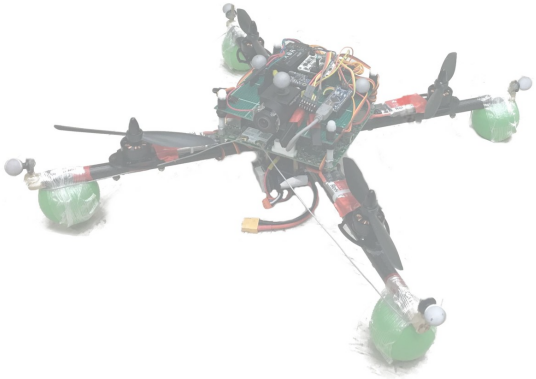
Motivation

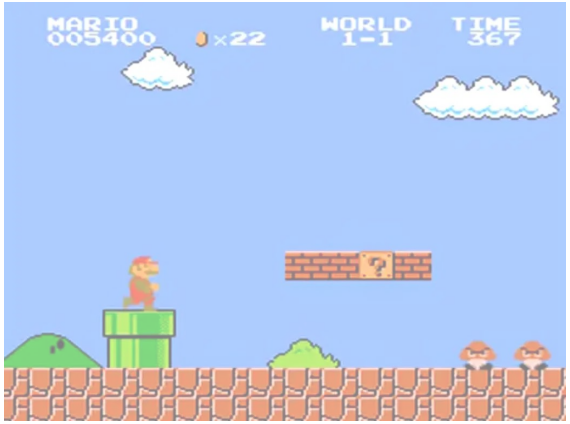
Motivation

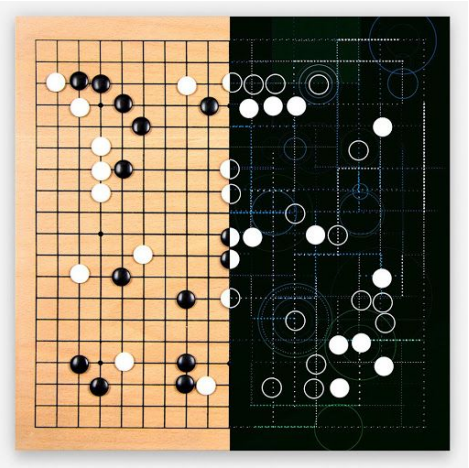
Motivation



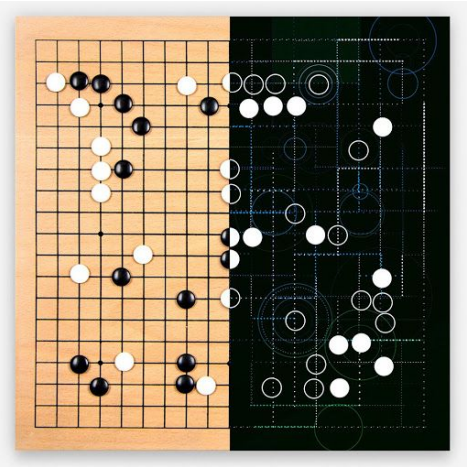
Motivation







Motivation



Can structure help?

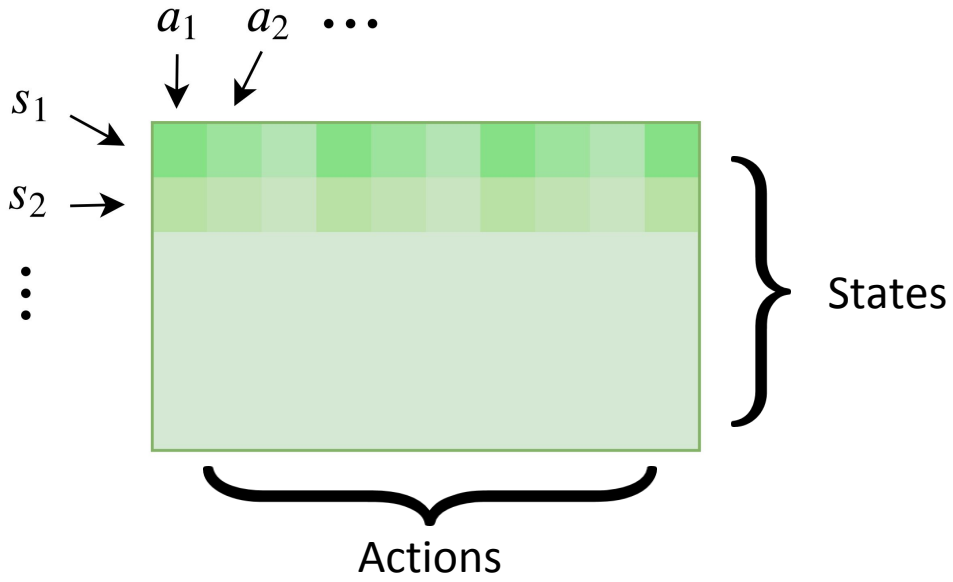
What Structure?

What Structure?

- Focus on *Q-value*

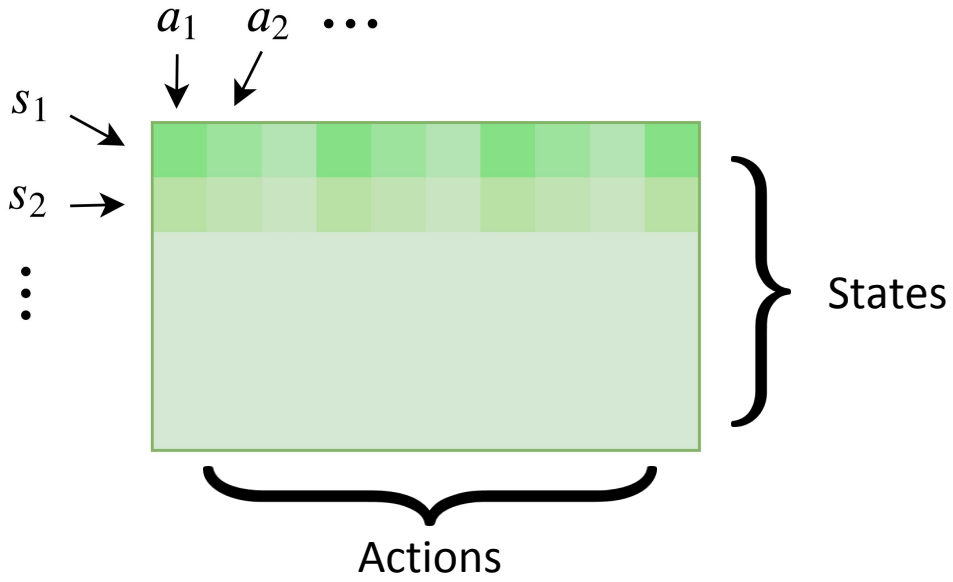
What Structure?

- Focus on *Q-value*



What Structure?

- Focus on *Q-value*



Global stucture: low-rank

Warm-up: Toy Example

- Randomly sampled deterministic MDP and Q-value iteration

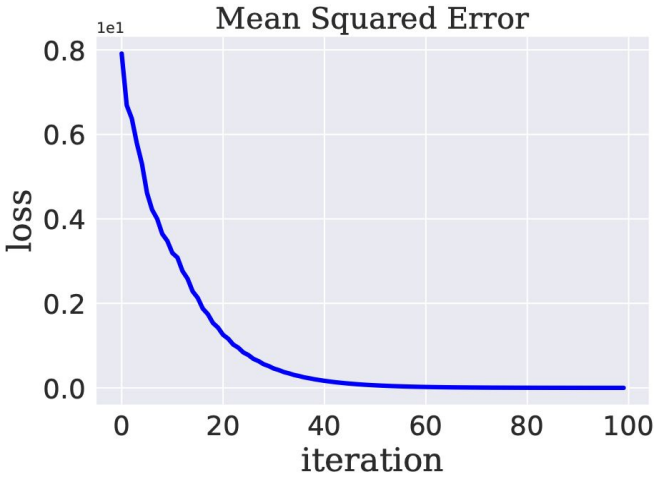
Warm-up: Toy Example

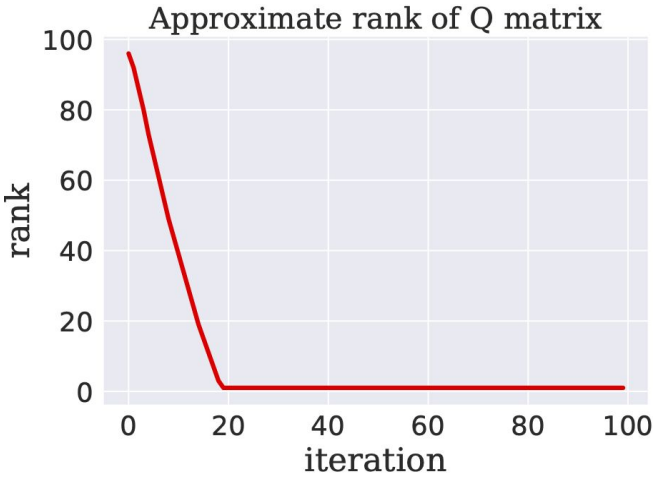
- Randomly sampled deterministic MDP and Q-value iteration

$$Q^{(t+1)}(s,a) = \sum_{s' \in \mathcal{S}} P(s'|s,a)\big[r(s,a) + \gamma \max_{a' \in \mathcal{A}} Q^{(t)}(s',a')\big], \quad \forall\,(s,a) \in \mathcal{S} \times \mathcal{A},$$

Warm-up: Toy Example

- Randomly sampled deterministic MDP and Q-value iteration

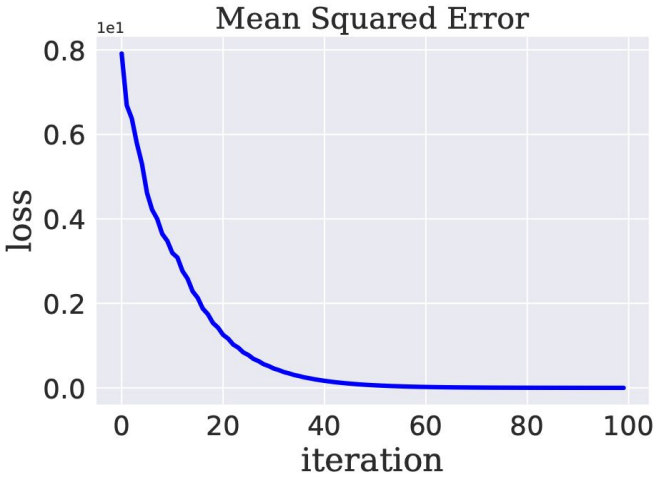


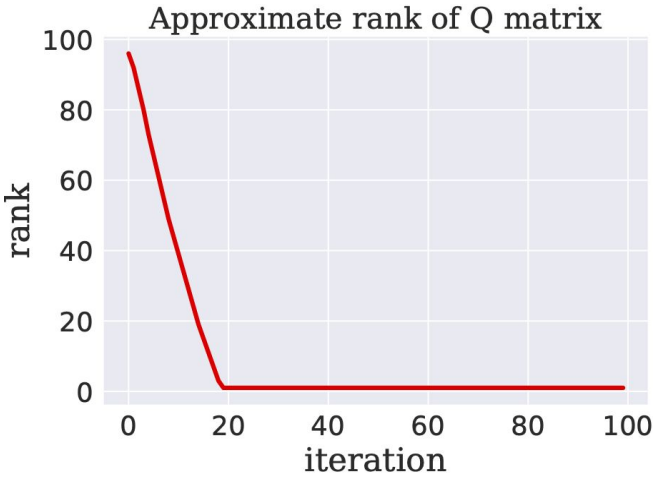


(Approx. rank: first *k* SVs capture > 99% variance, i.e., $\sum_{i=1}^{k} \sigma_i^2 / \sum_j \sigma_j^2 \geq 0.99$)

Warm-up: Toy Example

- Randomly sampled deterministic MDP and Q-value iteration





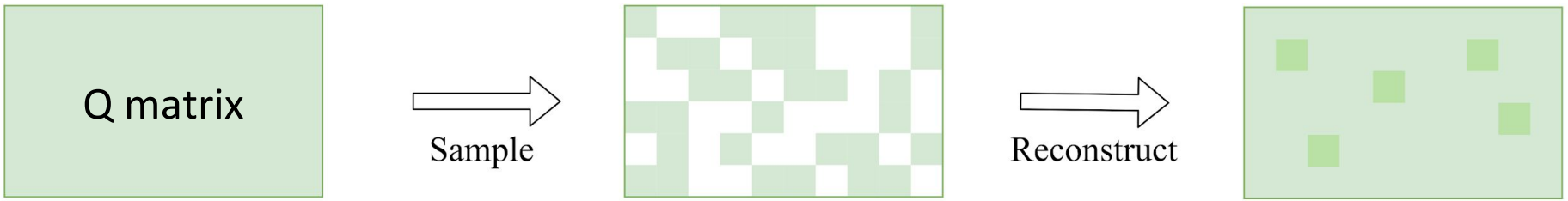
Exploit the structure during the learning process?
Enforce/regularize such a structure throughout the iterations?

How Do We Exploit the Structure?

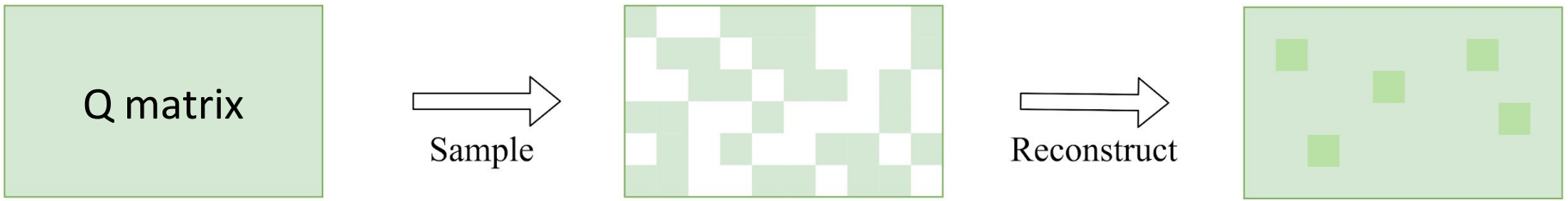
Idea?

Q matrix

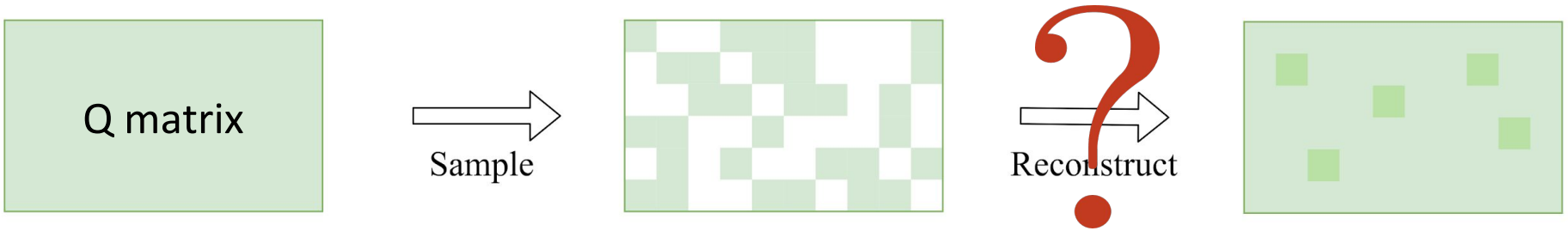
Idea?



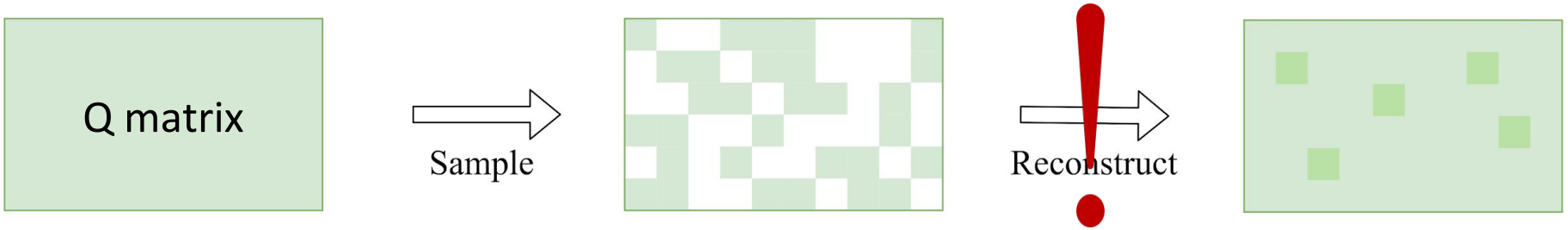
Idea: Compute few and reconstruct the rest



Idea: Compute few and reconstruct the rest

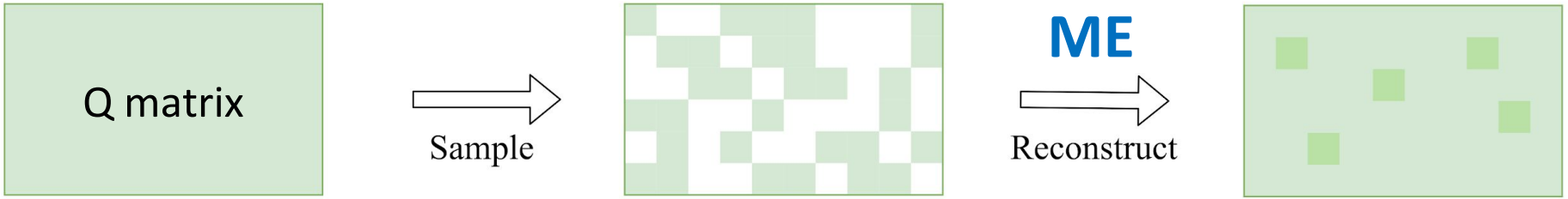


Idea: Compute few and reconstruct the rest



Low-rank Matrix Estimation (ME)

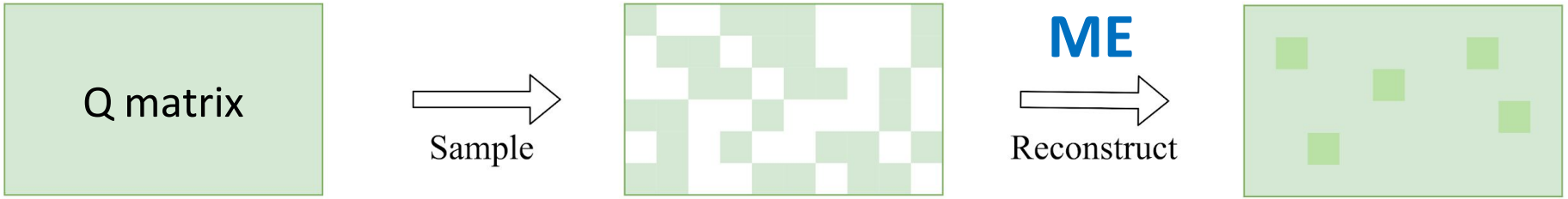
Idea: Compute few and reconstruct the rest



Low-rank Matrix Estimation (ME)

$$\min_{\hat{M}\in\mathbb{R}^{n\times m}} \frac{1}{2}\sum_{(i,j)\in\Omega}\left(\hat{M}_{ij}-X_{ij}\right)^2+\lambda||\hat{M}||_*$$

Idea: Compute few and reconstruct the rest



Low-rank Matrix Estimation (ME)

$$\min_{\hat{M}\in\mathbb{R}^{n\times m}} \frac{1}{2}\sum_{(i,j)\in\Omega}\left(\hat{M}_{ij}-X_{ij}\right)^2+\lambda||\hat{M}||_*$$

ME as a principled reconstruction oracle to exploit the low-rank structure

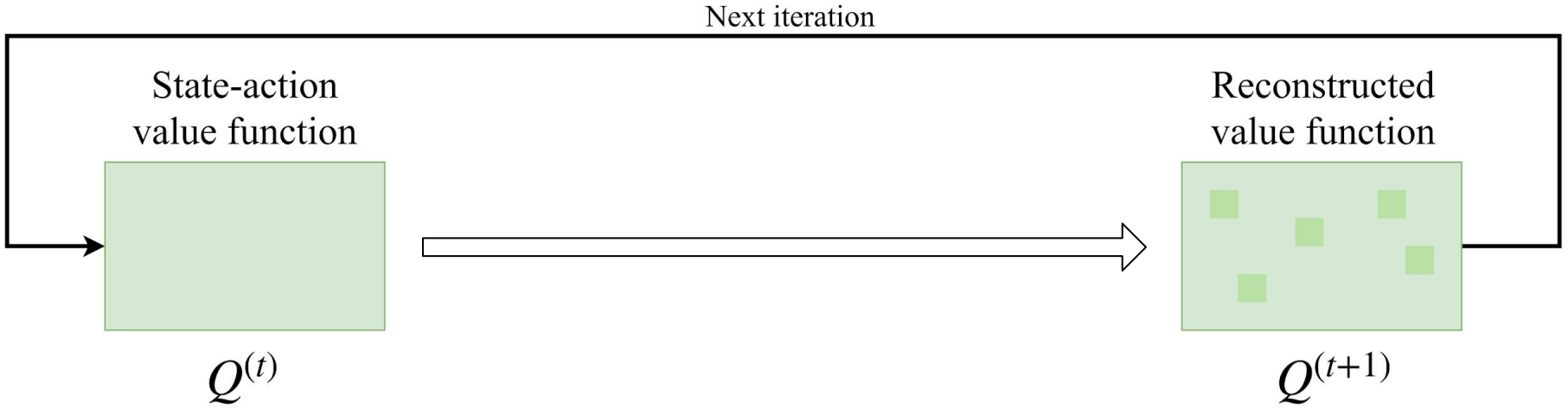
1. Structured Value-based Planning (SVP)

2. Structured Value-based Deep RL (SV-RL)

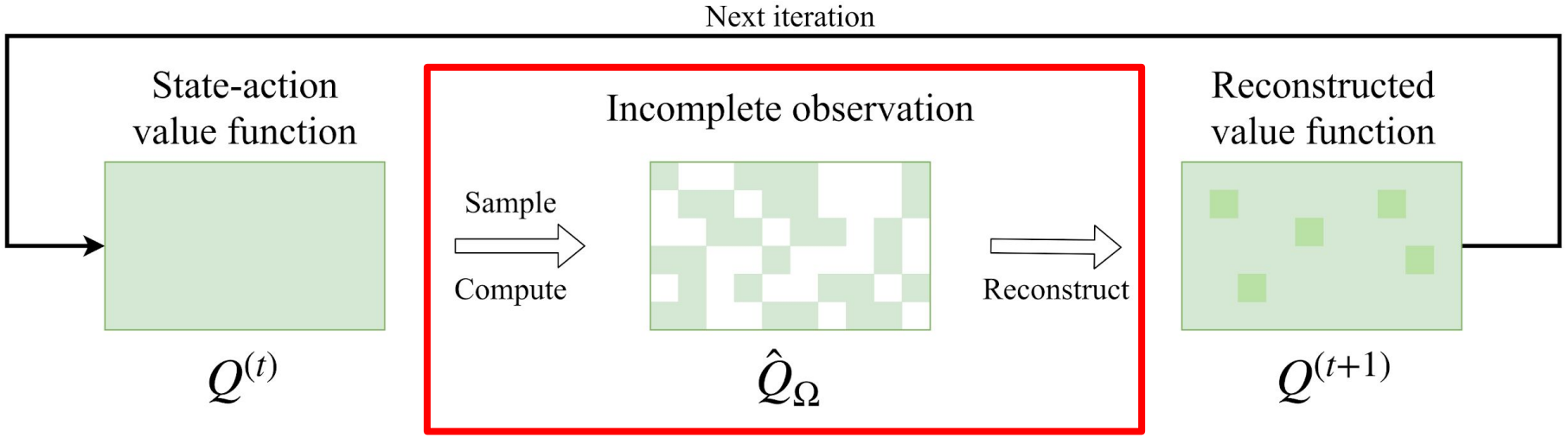
1. Structured Value-based Planning (SVP)

2. Structured Value-based Deep RL (SV-RL)

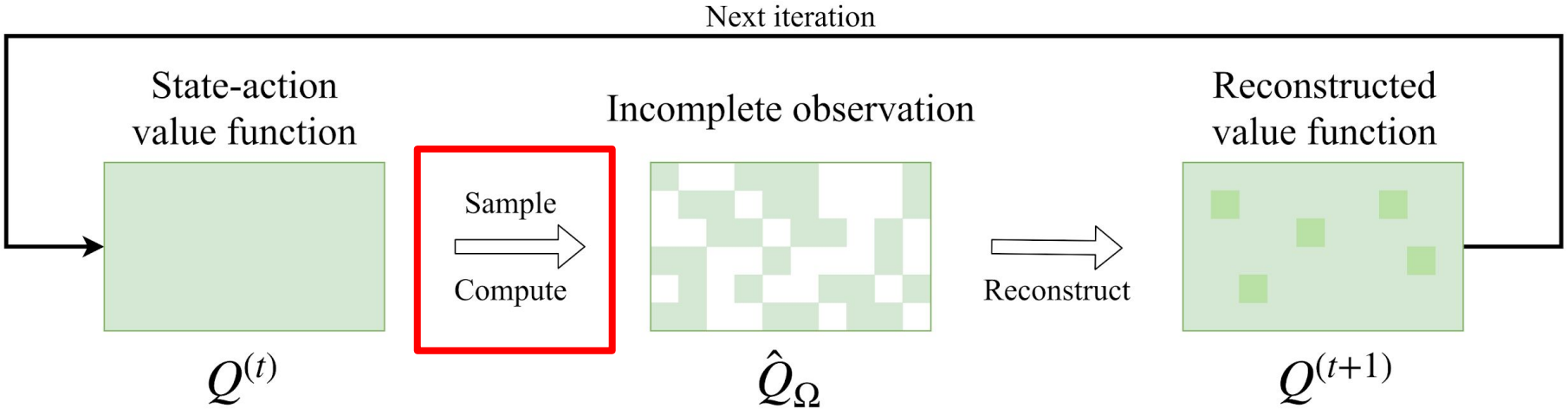
SVP: Structured Value-based Planning



SVP: Structured Value-based Planning

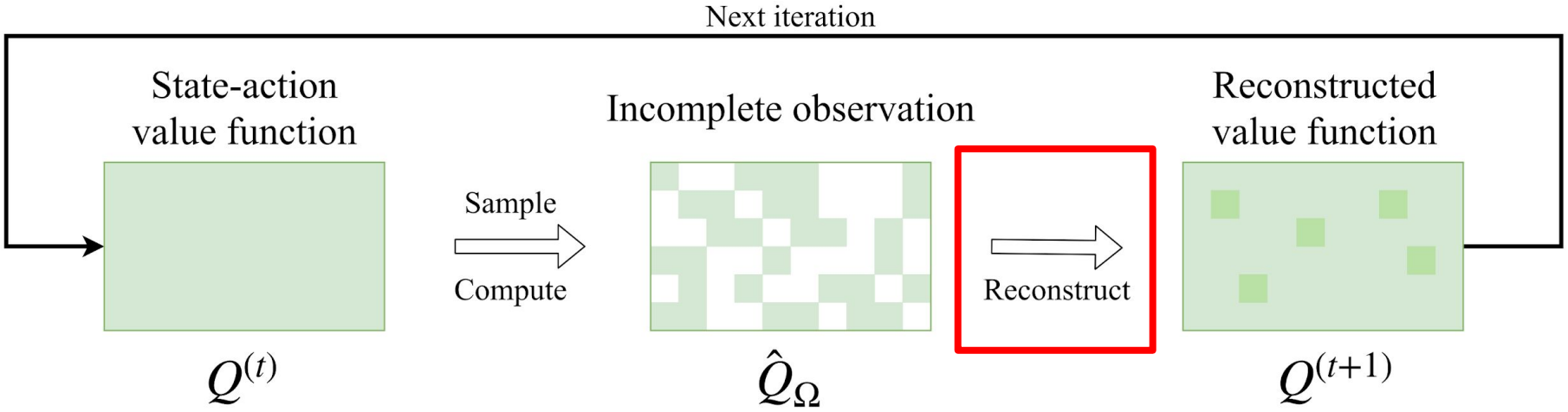


SVP: Structured Value-based Planning



$$\hat{Q}(s,a) \leftarrow \sum_{s'} P(s'|s,a)\left(r(s,a) + \gamma \max_{a'} Q^{(t)}(s',a')\right), \quad \forall\,(s,a) \in \Omega.$$

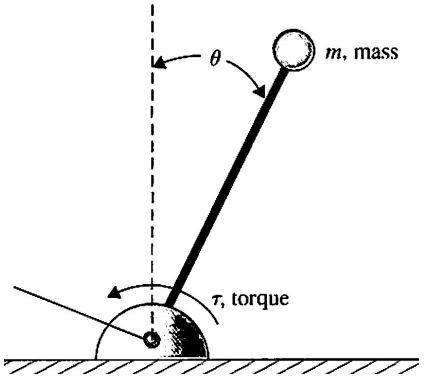
SVP: Structured Value-based Planning



$$Q^{(t+1)} = \mathrm{ME}\left(\{\hat{Q}(s,a)\}_{(s,a)\in\Omega}\right)$$

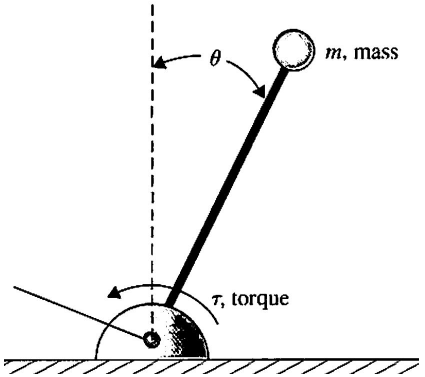
Stochastic Control: Inverted Pendulum

- Discretization: Q matrix = 2500 * 1000



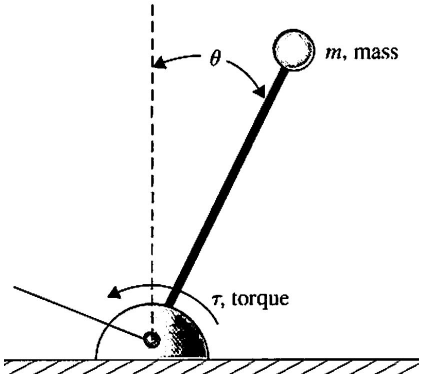
Stochastic Control: Inverted Pendulum

- Discretization: Q matrix = 2500 * 1000
- Verify low-rank structure:



Stochastic Control: Inverted Pendulum

- Discretization: Q matrix = 2500 * 1000
- Verify low-rank structure:

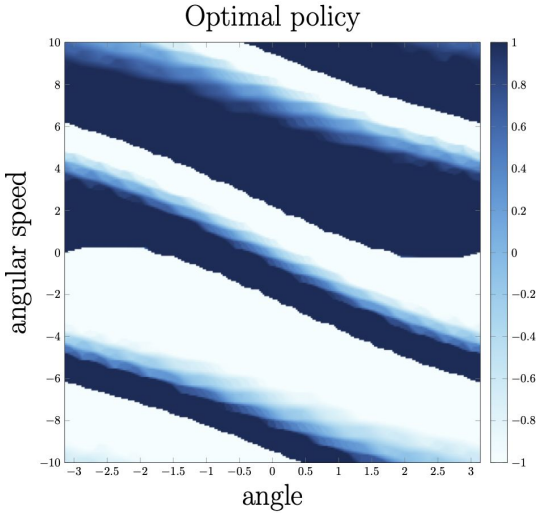


Approximate rank of Q* = 7

Desired low-rank property for SVP

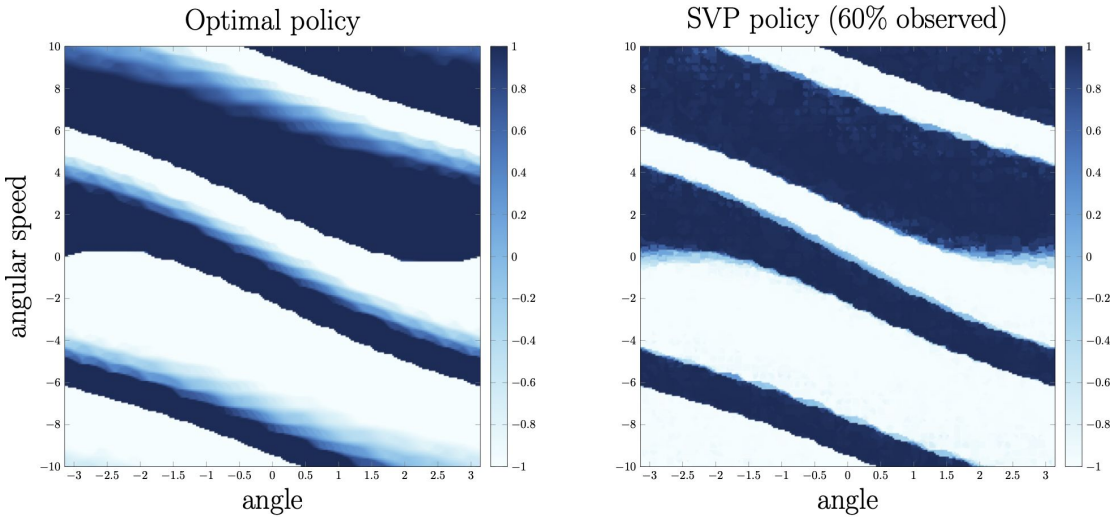
Stochastic Control: Inverted Pendulum

- Policy visualization:



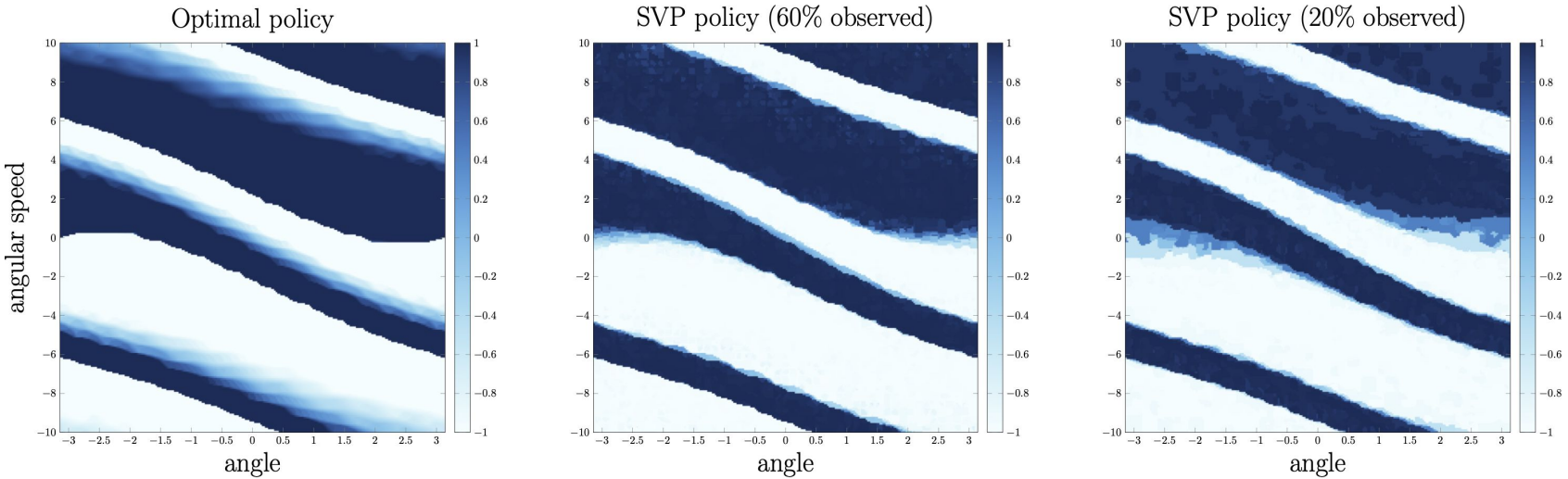
Stochastic Control: Inverted Pendulum

- Policy visualization:



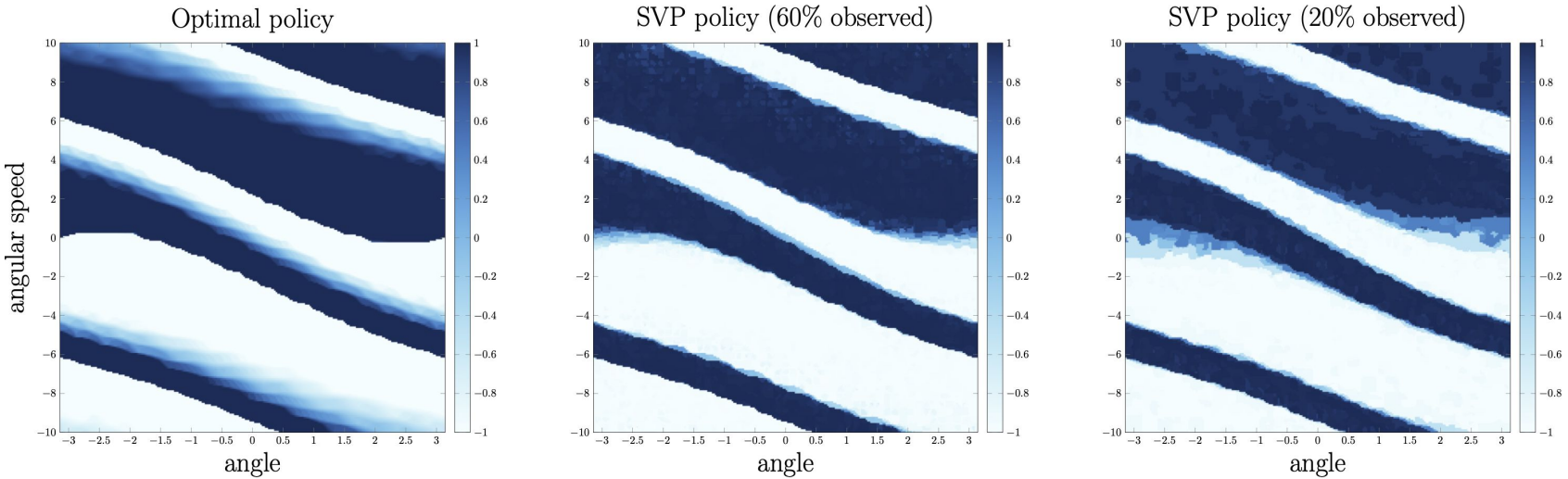
Stochastic Control: Inverted Pendulum

- Policy visualization:



Stochastic Control: Inverted Pendulum

- Policy visualization:



Success of SVP: a small amount of observations is sufficient!

1. Structured Value-based Planning (SVP)

2. Structured Value-based Deep RL (SV-RL)

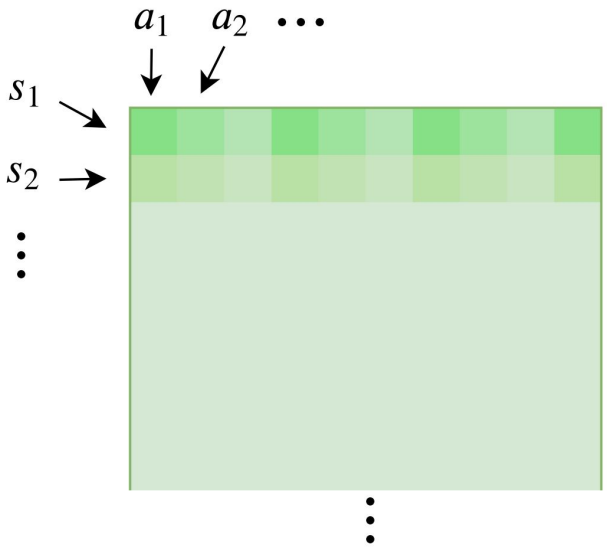
Extend to Deep RL?

- Intuition and development of SVP

Extend to Deep RL?

- Intuition and development of SVP
- Naive extension? Issues?

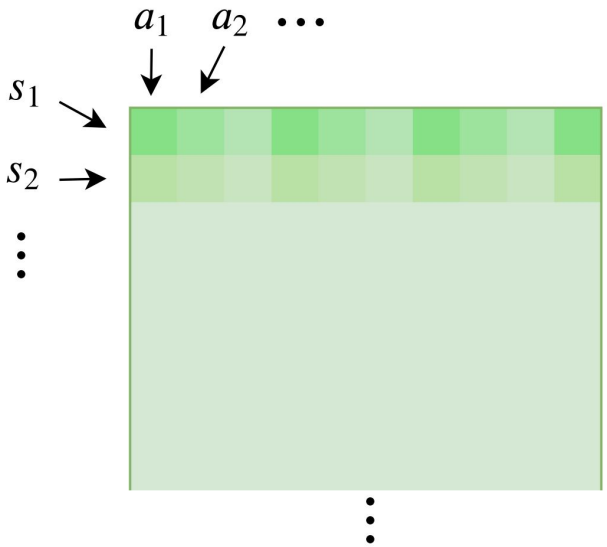
With images as states...



Idea: Batch of States as Proxy

- Intuition and development of SVP
- Naive extension? Issues?

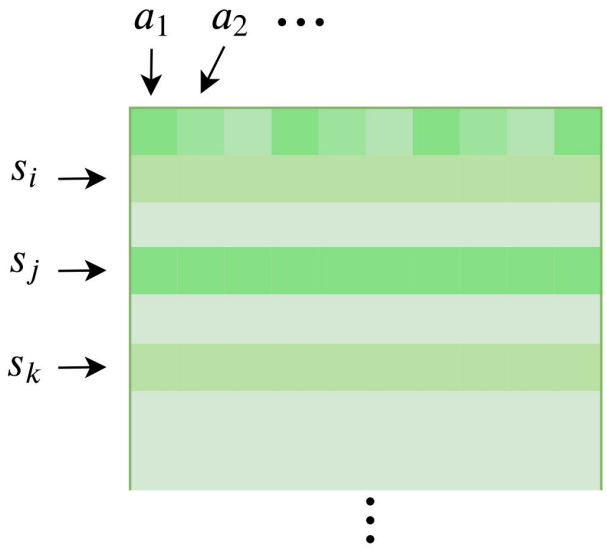
With images as states...



Idea: Batch of States as Proxy

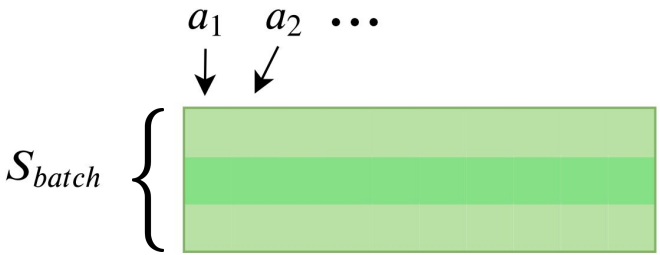
- Intuition and development of SVP
- Naive extension? Issues?

With images as states...



Idea: Batch of States as Proxy

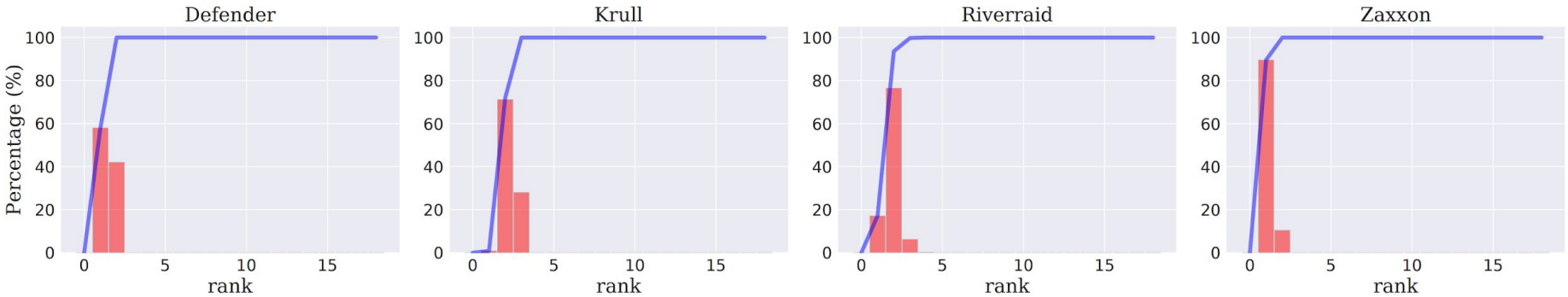
- Intuition and development of SVP
- Naive extension? Issues?



Natural to understand the rank of batches of states for the learned Q value

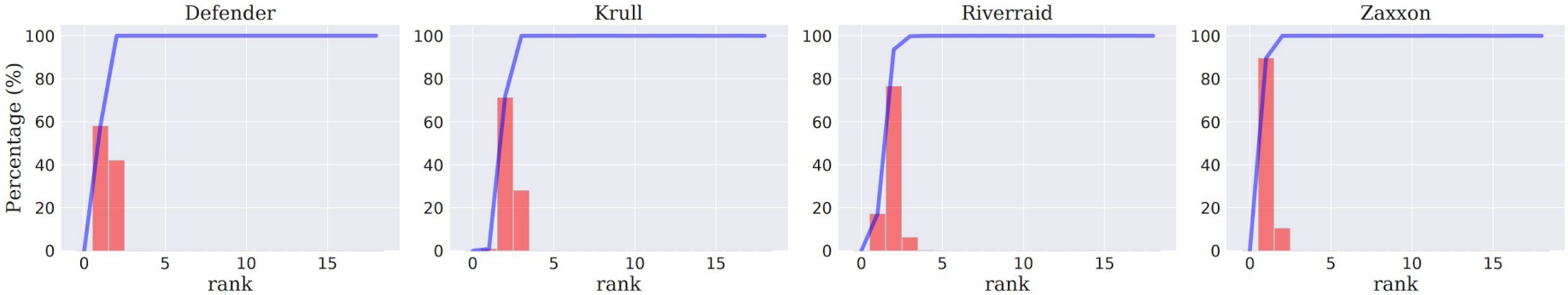
Evidence of low-rank structures

- Batch size = 32; Sample 10,000 sub-matrices from DQN



Evidence of low-rank structures

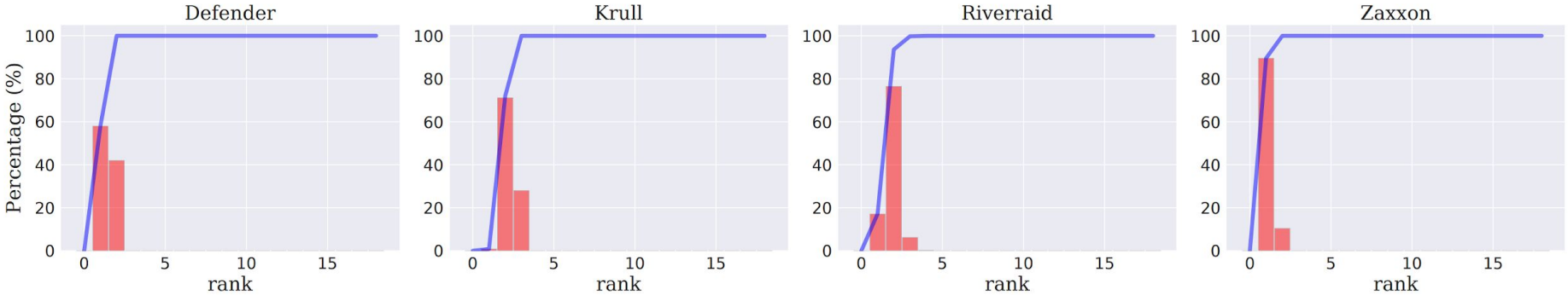
- Batch size = 32; Sample 10,000 sub-matrices from DQN



Structure widely exists: Majority of games (> 40)!

Evidence of low-rank structures

- Batch size = 32; Sample 10,000 sub-matrices from DQN



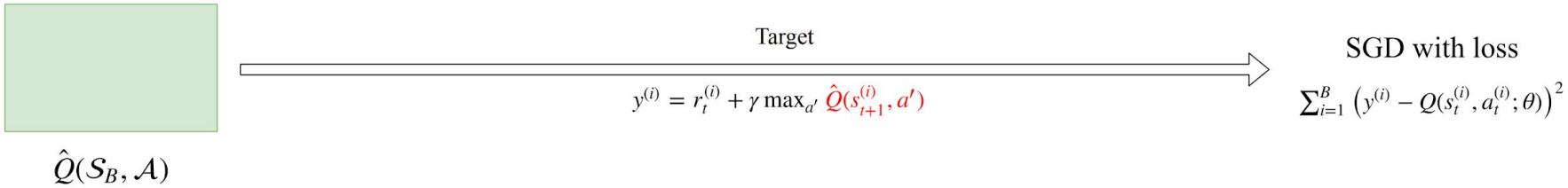
Structure widely exists: Majority of games (> 40)!

Harness the structure within the batch of states during the learning process

SV-RL: Structured Value-based Deep RL

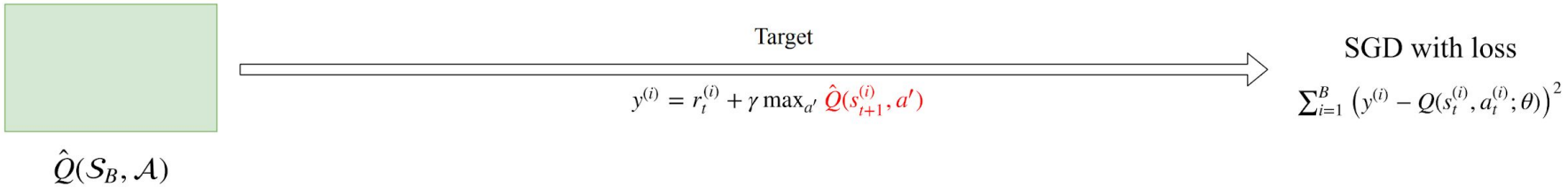
SV-RL: Structured Value-based Deep RL

- Original value-based RL

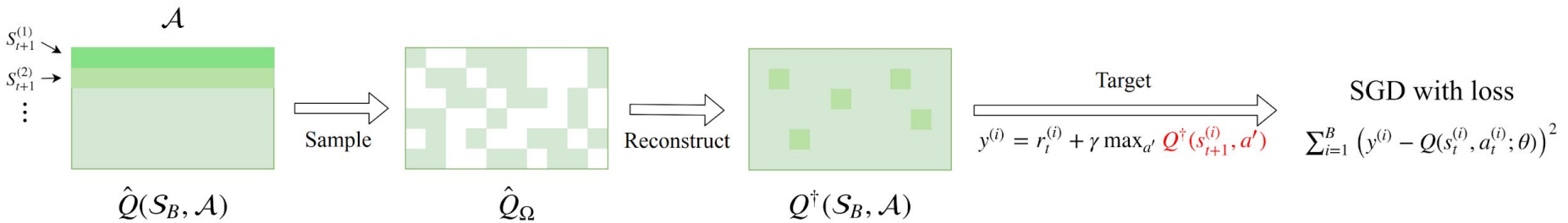


SV-RL: Structured Value-based Deep RL

- Original value-based RL

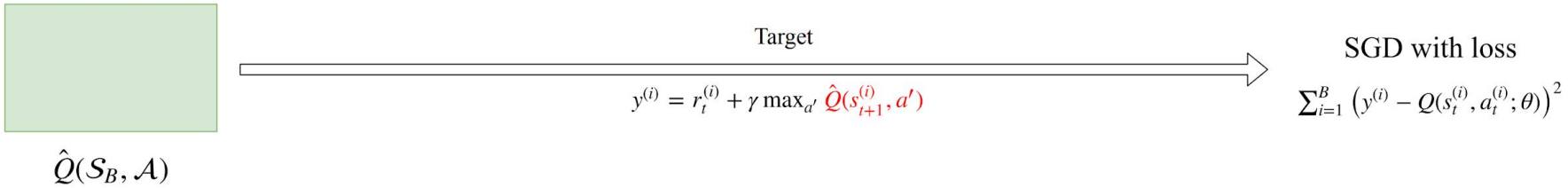


- SV-RL

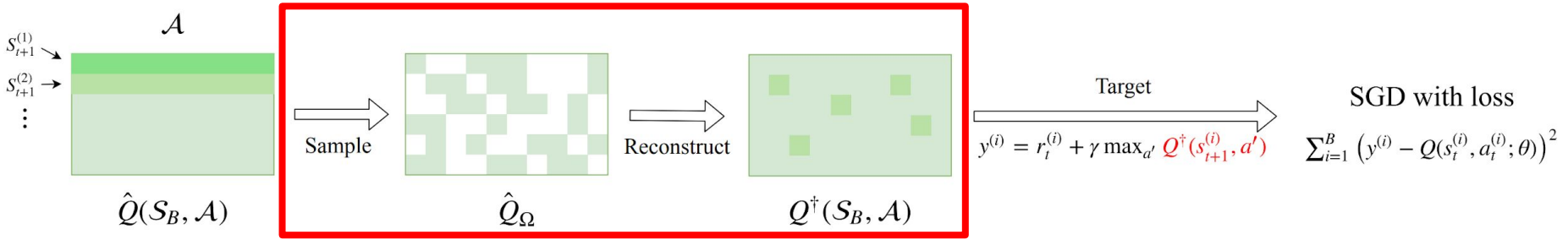


SV-RL: Structured Value-based Deep RL

- Original value-based RL



- SV-RL

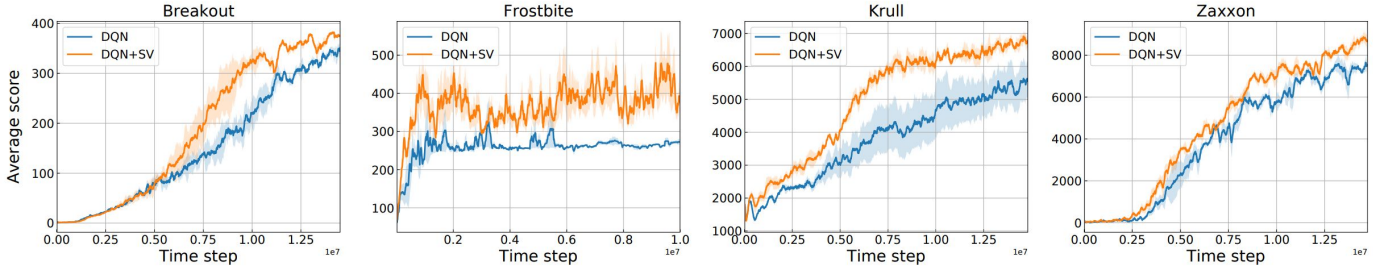


Empirical Evaluation: Atari

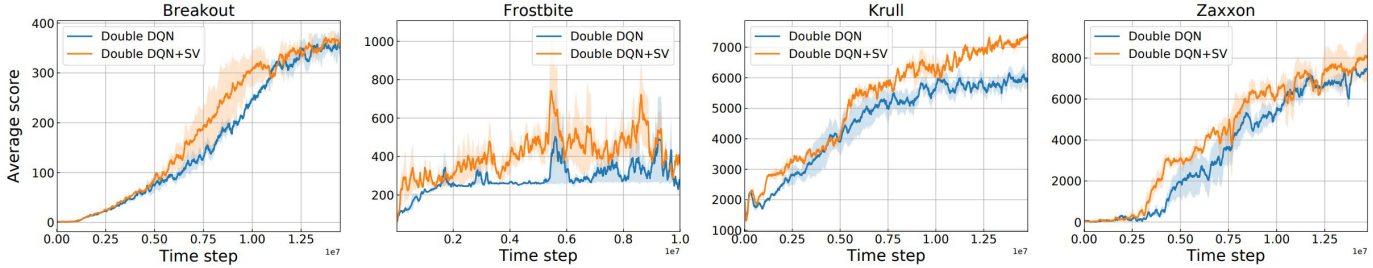
- Apply SV-RL on three representative value-base deep RL

Consistent Benefits for “Structured” Games

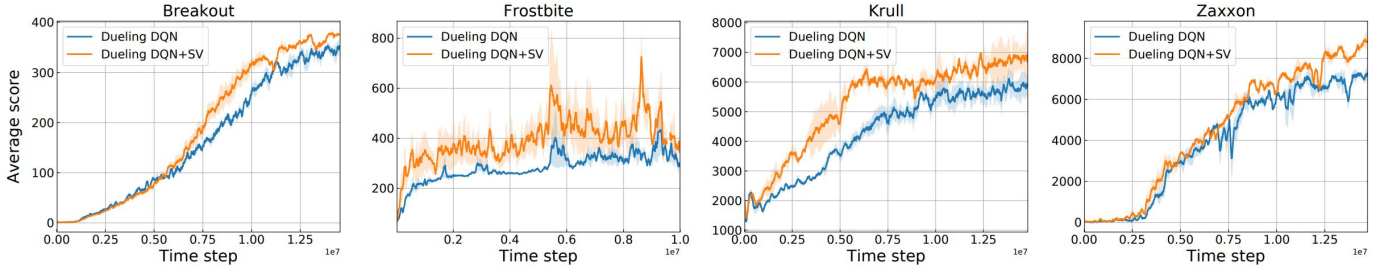
DQN



Double DQN



Dueling DQN

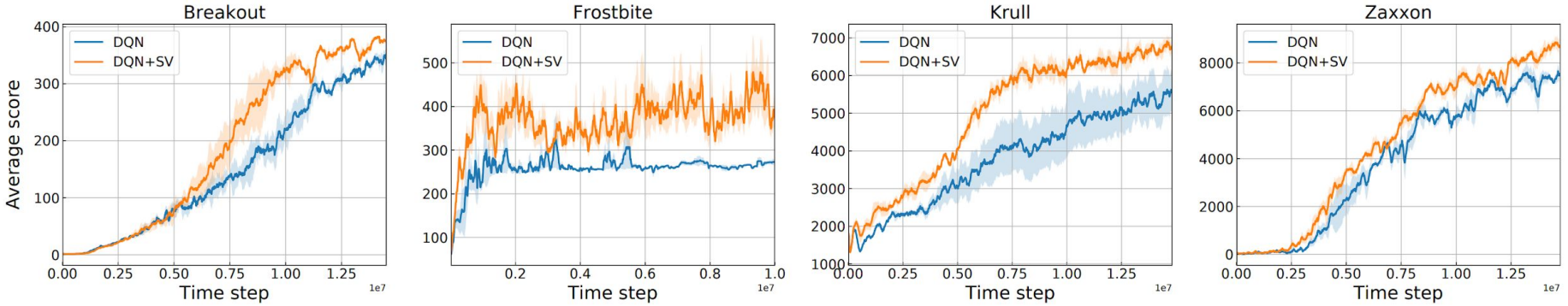


Empirical Evaluation: Atari

- Apply SV-RL on three representative value-base deep RL

- Consistent benefits for “structured” games:
  1. games that possess low-rank structure benefit from SV-RL
  2. consistent improvements across different RL techniques
  3. more games - see paper

Empirical Evaluation: Atari

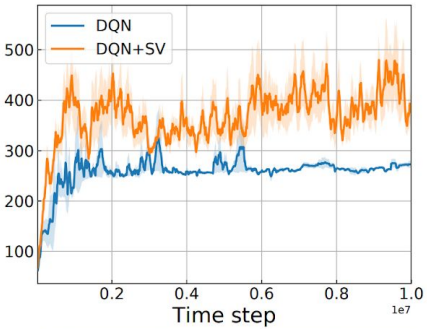


- Further observations? Performance gains vary.

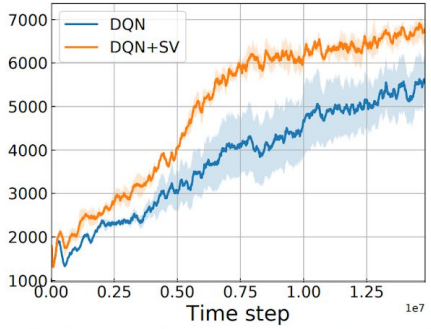
Diagnose & Interpret Performance

	Frostbite	Krull	Alien	Seaquest
SV-RL	Better	Better	Slightly Better	Worse

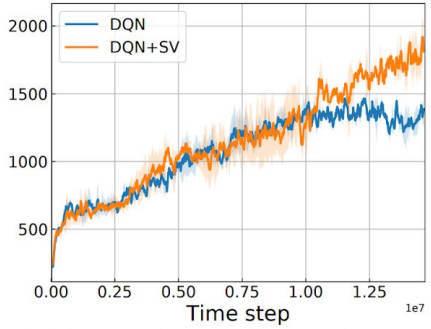
Diagnose & Interpret Performance



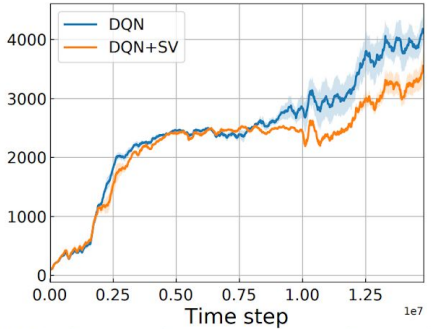
(a) Frostbite (better)



(b) Krull (better)

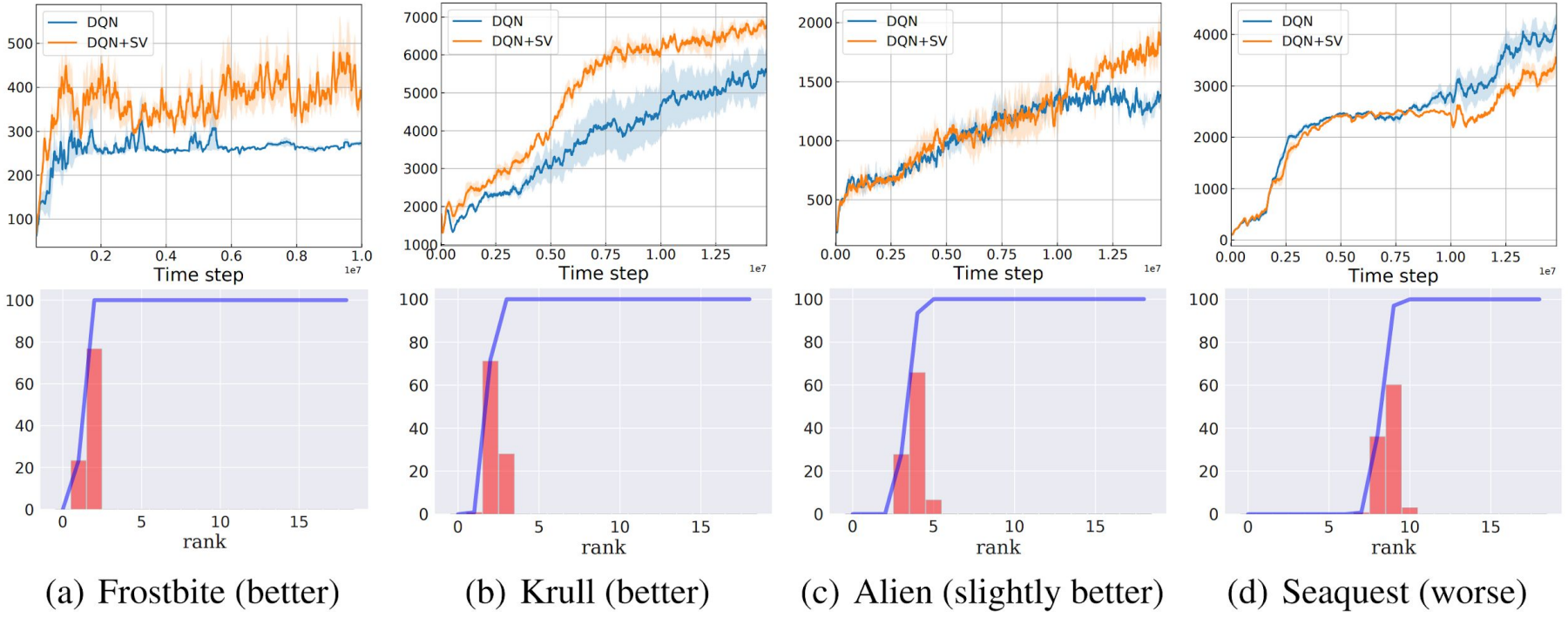


(c) Alien (slightly better)



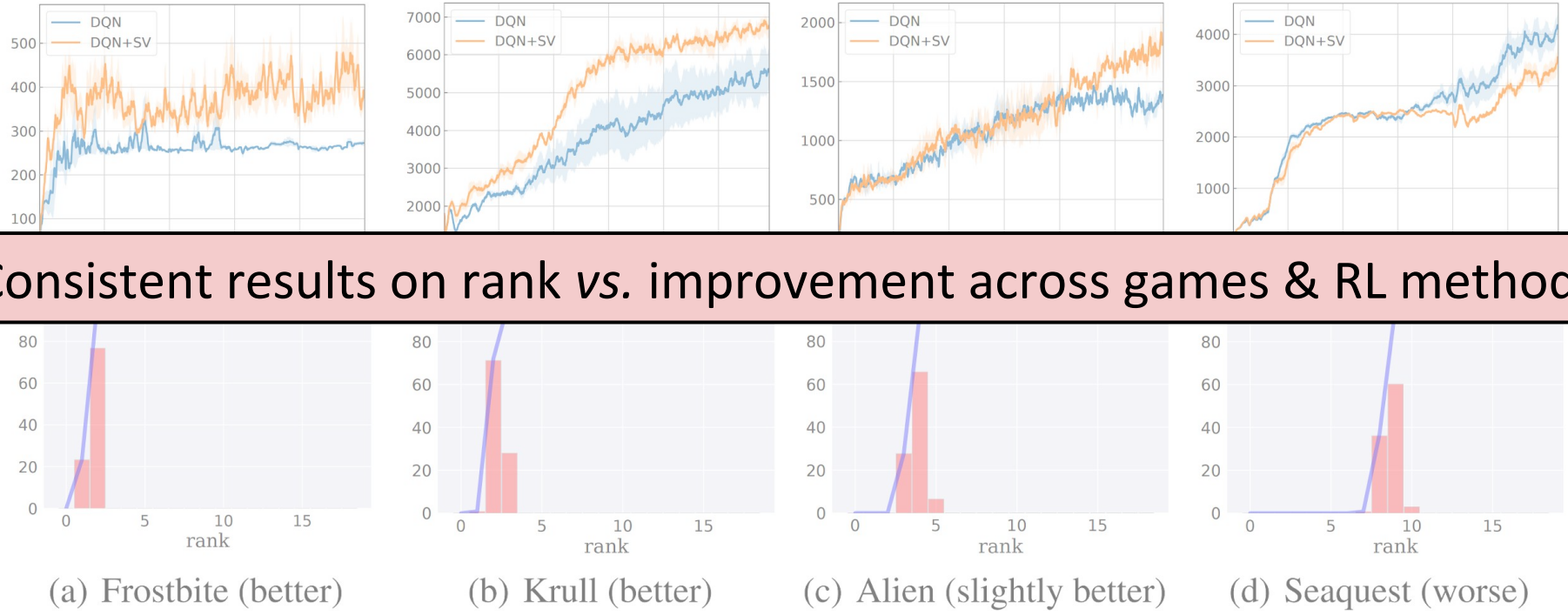
(d) Seaquest (worse)

Diagnose & Interpret Performance



(a) Frostbite (better) (b) Krull (better) (c) Alien (slightly better) (d) Seaquest (worse)

Diagnose & Interpret Performance



Consistent results on rank *vs.* improvement across games & RL methods

Diagnose & Interpret Performance

	Frostbite	Krull	Alien	Seaquest
SV-RL	Better	Better	Slightly Better	Worse
Rank	~2	~2	~5	~10

- Consistent interpretations:

Diagnose & Interpret Performance

	Frostbite	Krull	Alien	Seaquest
SV-RL	Better	Better	Slightly Better	Worse
Rank	~2	~2	~5	~10

- Consistent interpretations:

  If the learned Q function contains low-rank structure

SV-RL is able to exploit the structure!

Summary of Contributions

- Propose a generic framework that exploits the low-rank structures, for planning and deep reinforcement learning

Summary of Contributions

- Propose a generic framework that exploits the low-rank structures, for planning and deep reinforcement learning

- Demonstrate the effectiveness of our approach on classical stochastic control tasks

Summary of Contributions

- Propose a generic framework that exploits the low-rank structures, for planning and deep reinforcement learning
- Demonstrate the effectiveness of our approach on classical stochastic control tasks
- Extend our scheme to deep RL, which is naturally applicable for value-based techniques, and obtain consistent improvements across a variety of methods

Poster Sessions (New York time):

Apr. 28th: 12 AM - 2 AM
Apr. 29th: 12 PM - 2 PM

Source Code

https://github.com/YyzHarry/SV-RL

Project Page

http://svrl.csail.mit.edu