Fast-slow MDPs: What they are and how to solve them

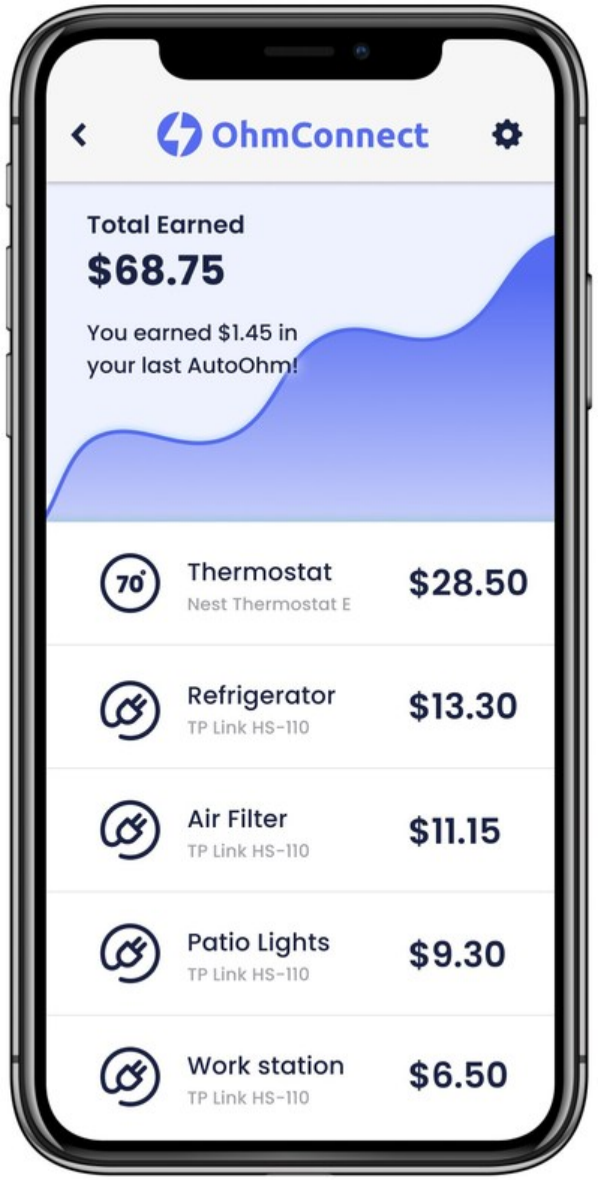
Daniel Jiang
joint work with Yijia Wang
University of Pittsburgh

Decision Sciences Seminar, April 13, 2022
Fuqua School of Business, Duke University

1. *Motivation via three example applications*

Demand response provider

- Energy demand response is the practice of paying energy consumers to reduce usage at certain times
- An energy aggregator / demand response provider
  - Bids an amount of demand reduction into the market, given **day-ahead price**
  - Offers a compensation to residential customers to reduce consumption
  - Potentially penalized by (*the more volatile*) **real-time price** if shortage between promised and realized demand reduction
- Profit = revenue from market - compensation



K. Khezeli, W. Lin, E. Bitar. Learning to buy (and sell) demand response. *Proceedings of the International Federation of Automatic Control (IFAC) World Congress*, 2017.

(Energy/carbon-aware) job scheduling in data centers

- Dynamic service allocation with multi-class queues
- Multiple queues of different job types (e.g., training different models) to be served by a single node
  - At each period, choose one type of job to serve
  - Cost = the *holding* costs endured by the jobs
- Energy/carbon-aware: Holding costs depend on:
  - Electricity prices, generation sources, etc. and **might *vary slowly* throughout the day**

P. Ansell, K. D. Glazebrook, J. Nino-Mora, and M. O'Keeffe. Whittle's index policy for a multi-class queueing system with convex holding costs. *Mathematical Methods of Operations Research*, 2003.

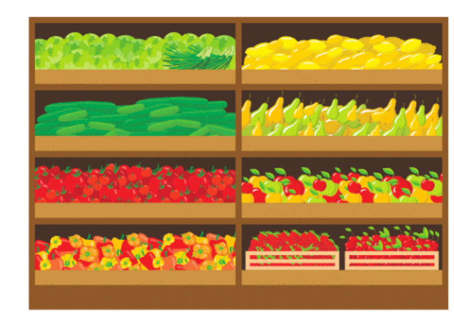
D. B. Brown and M. B. Haugh. Information relaxation bounds for infinite horizon MDPs. *Operations Research,* 2017.

https://blog.google/inside-google/infrastructure/data-centers-work-harder-sun-shines-wind-blows/

D. Lee and M. Vojnovic. Scheduling jobs with stochastic holding costs. *NeurIPS,* 2021.

Restless multi-armed bandit with environmental states

- A decision-maker faces:
    - A set of "arms," each associated with an evolving internal state
    - Global environmental states that affect the dynamics of each arm
- Which arms to **intervene** (at a cost) in each period?
- Applications:
    - Machine maintenance (environmental factors affect the likelihood of each machine failing)
    - Public health intervention decisions
    - Dynamic assortment planning
    - Preventative healthcare (limited screening resources for a set of patients)



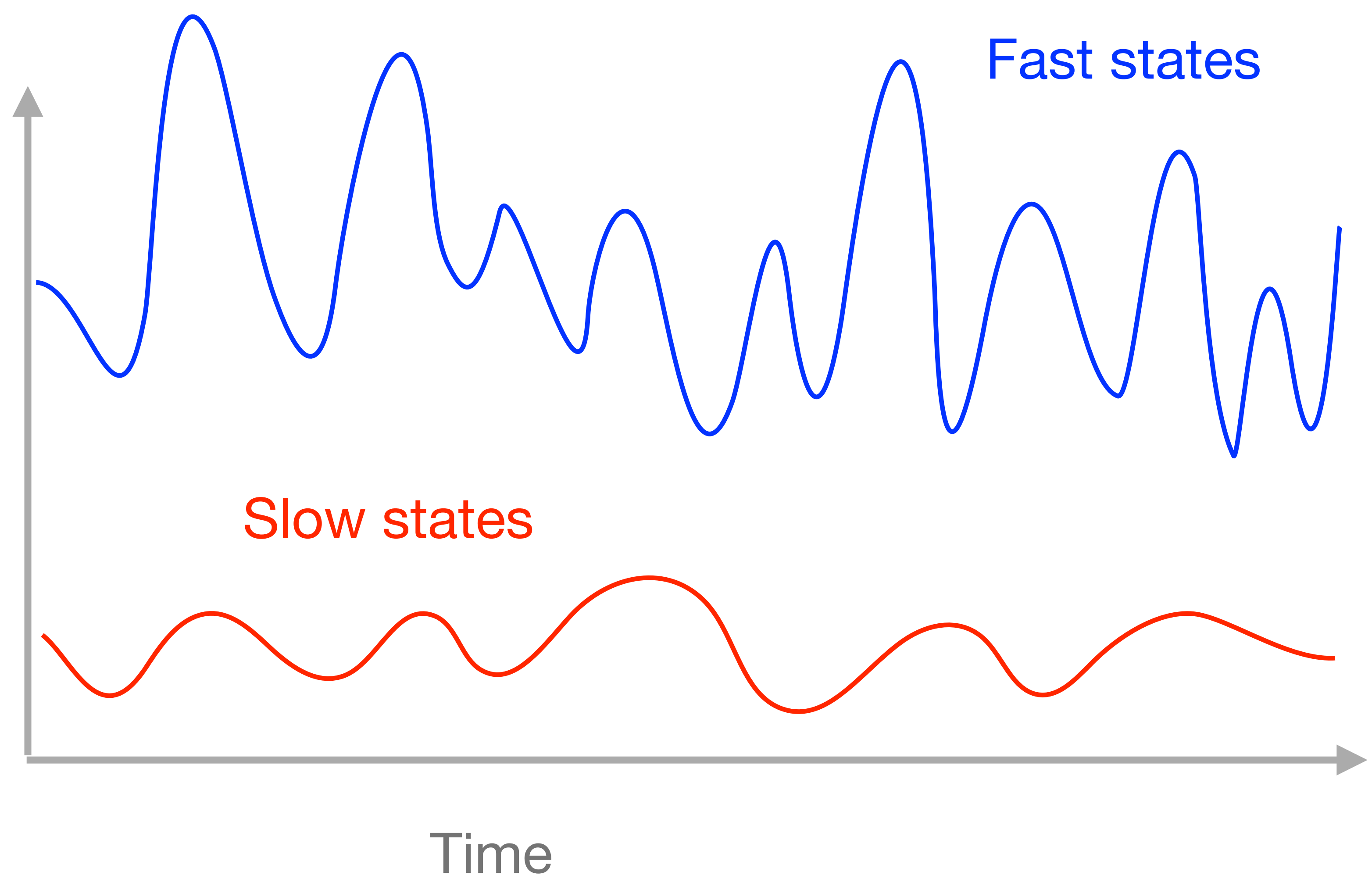
R. D. Smallwood and E. J. Sondik. The optimal control of partially observable Markov processes over a finite horizon. *Operations Research,* 1973.

B. Bhattacharya. Restless bandits visiting villages: A preliminary study on distributing public health services. In *Proceedings of the 1st ACM SIGCAS Conference on Computing and Sustainable Societies*, 2018.

D. B. Brown and J. E. Smith. Index policies and performance bounds for dynamic selection problems. *Management Science*, 2020.

E. Lee, M. S. Lavieri, and M. Volk. Optimal screening for hepatocellular carcinoma: A restless bandit model. *Manufacturing & Service Operations Management*, 2019.

What do they have in common?



Fast states from examples:

- Real-time prices
- Queue lengths
- Machine statuses

Shorter timescales

Slow states from examples:

- Day-ahead prices
- Holding cost of queue
- Environmental factors

Longer timescales

Current practice

- Additional state variables in a DP is expensive:
  - Each iteration of value iteration $\mathcal{O}(S^2 A)$
- **What do practitioners do (anecdotally)?**
  - From the beginning, *ignore/omit* slow states (contexts, environmental variables, etc) in the modeling
    - e.g. assume costs are deterministic, demand is stationary, weather doesn't change
- **This work:** a *compromise* between computational tractability and fully ignoring the slow state
  - We propose: an approach that periodically ignores slow states
  - We give evidence and argue that completely omitting slow states from the decision model is often not a viable heuristic

Outline

- **Fast-slow MDP**
  - Propose the concept of an MDP where some states move fast and others relatively slowly
- **Frozen-state approximation (another MDP)**
  - What if we “freeze” the slow state for a few periods at a time?
- **Algorithms**
  - Frozen-state value iteration / approximate value iteration
  - Regret analysis
- **Numerical experiments on motivating examples**

2. *Fast-slow Markov decision processes*

Fast-slow Markov decision processes

- A γ-discounted, infinite horizon MDP:
  - States $s \in \mathcal{S}$
  - Actions $a \in \mathcal{A}$
  - Rewards $r(s, a) \in [0, r_{\max}]$
  - Transition function
    - $s_{t+1} = f(s_t, a_t, w_{t+1}),\ w_{t+1} \in \mathcal{W}$

- Fast-slow MDP: (Slow: x, Fast: y)
  - States $s = (x, y) \in \mathcal{S} = (\mathcal{X} \times \mathcal{Y})$
  - Actions $a \in \mathcal{A}$
  - Rewards $r(s, a) \in [0, r_{\max}]$
  - Transition function
    - $x_{t+1} = f_{\mathcal{X}}(s_t, a_t, w_{t+1})$
    - $y_{t+1} = f_{\mathcal{Y}}(s_t, a_t, w_{t+1})$

Main assumption ("fast-slow property"):

$$\|y - f_{\mathcal{Y}}(x, y, a, w)\|_2 \leq d_{\mathcal{Y}} \quad \text{and} \quad \|x - f_{\mathcal{X}}(x, w)\|_2 \leq \alpha d_{\mathcal{Y}}\,.$$

Lipschitz assumptions (let $U^\star(s)$ be the optimal value function):

$$r(s, a) - r(s', a') \leq L_r \|(s, a) - (s', a')\|_2,$$

$$\|f(s, a, w) - f(s', a', w)\|_2 \leq L_f \|(s, a) - (s', a')\|_2,$$

$$\|U^\star(s) - U^\star(s')\|_2 \leq L_U \|s - s'\|_2.$$

← Can be removed, included for clarity

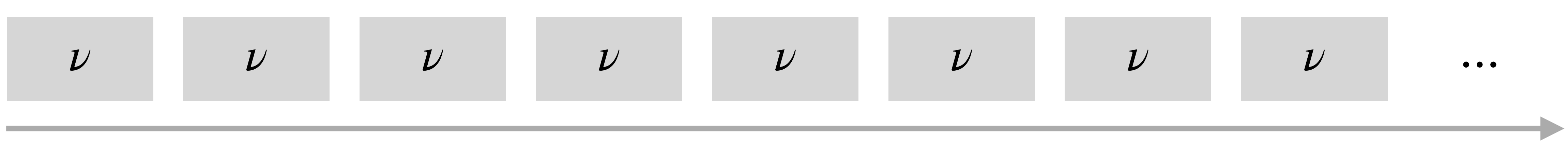
3. *Hierarchical reformulation*

Hierarchical reformulation (of any MDP)

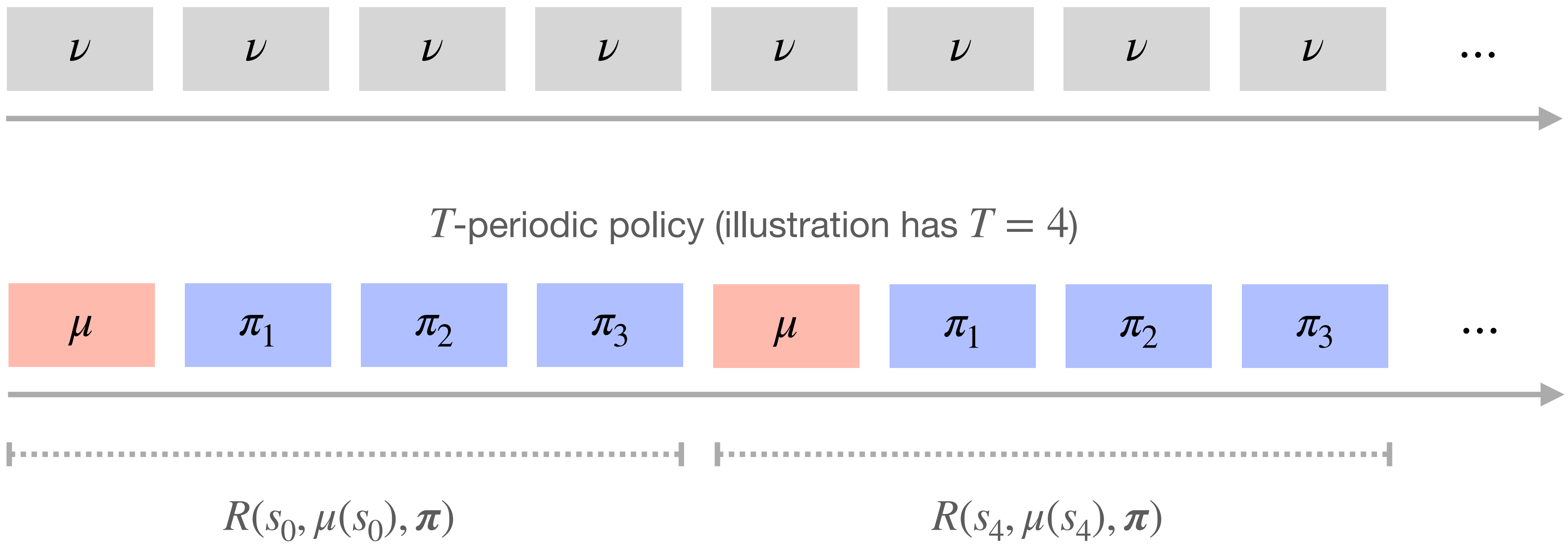
- A hierarchical reformulation is at the basis of our proposed approach
- Consider an MDP $\langle \mathcal{S}, \mathcal{A}, \mathcal{W}, f, r, \gamma \rangle$
- Let $\nu : \mathcal{S} \to \mathcal{A}$ be a stationary policy
- The value function is

$$U^{\nu}(s) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r\big(s_t, \nu\big) \,\middle|\, s_0 = s \right] = r\big(s, \nu\big) + \gamma\, \mathbb{E}\big[U^{\nu}(s')\big]$$

- The policy can be thought of as $(\nu, \nu, \ldots)$:



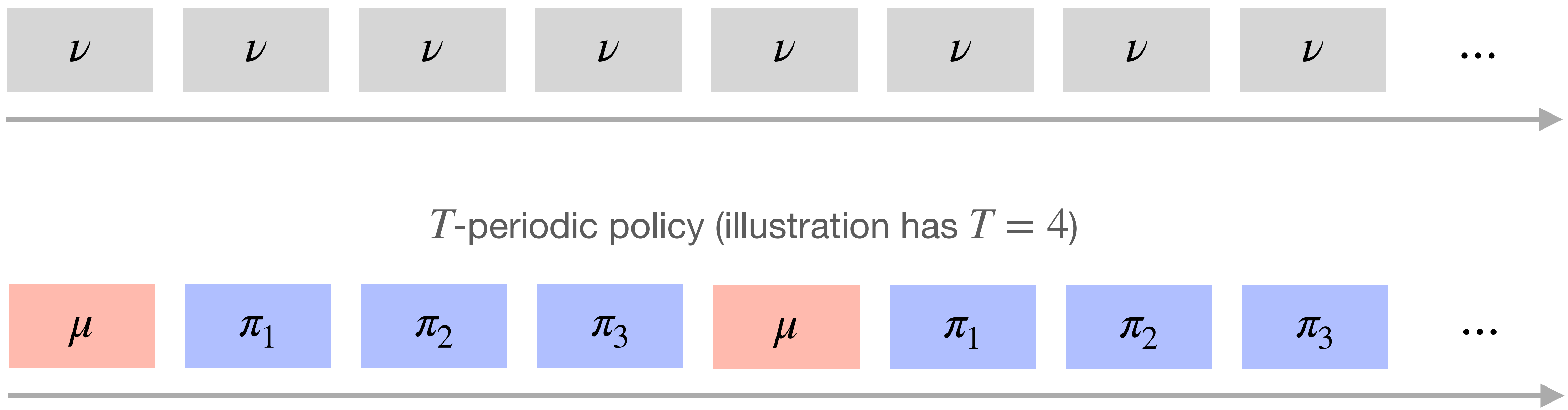
Hierarchical reformulation (of any MDP)



- Given a T-periodic policy $(\mu, \boldsymbol{\pi}) = (\mu, \pi_1, \ldots, \pi_{T-1})$, T-horizon reward is

$$R(s_0, \mu(s_0), \boldsymbol{\pi}) = r(s_0, \mu) + \sum_{t=1}^{T-1} \gamma^t \, r(s_t, \pi_t)$$

Hierarchical reformulation (of any MDP)



- Bellman equations of the base model and its hierarchical reformulation are:

$$U^\star(s_0) = \max_{a} \; r(s,a) + \gamma \, \mathbb{E}\left[U^\star(s_1)\right]$$

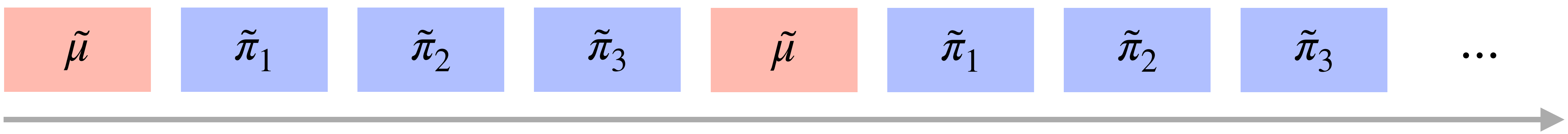
$$\bar{U}^\star(s_0) = \max_{(\mu,\boldsymbol{\pi})} \; \mathbb{E}\left[R(s_0, \mu(s_0), \boldsymbol{\pi}) + \gamma^T \, \bar{U}^\star(s_T)\right]$$

How can we take advantage of this?

Proposition. The optimal values are equal: $U^\star(s) = \bar{U}^\star(s)$. Therefore, we can use the hierarchical reformulation as a basis for our approximation.

4. *Frozen-state approximation and its regret*

Frozen-state approximation



What we hope for…

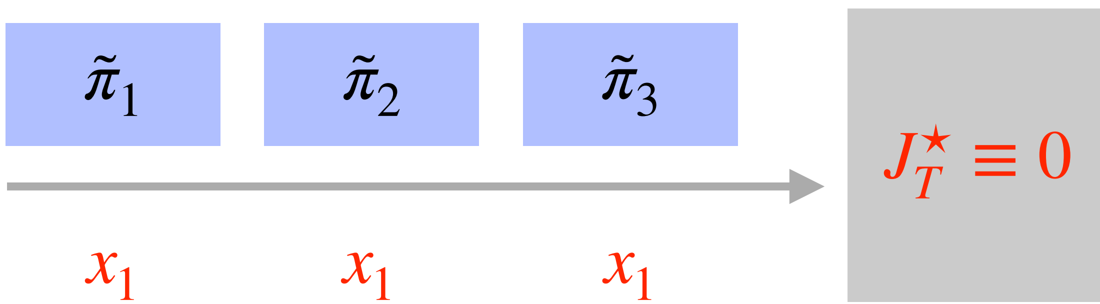
Implementation

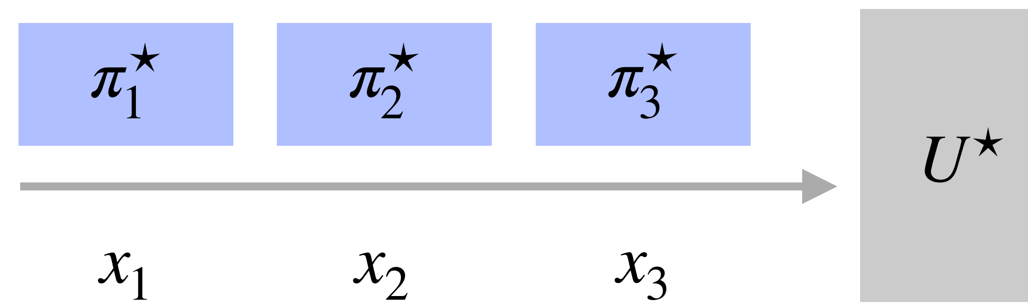
1. At $t = 0$, take a "upper-level" action (using $\tilde{\mu}$), i.e., an action that considers the γ^T timescale
2. At $t = 1$, observe slow state and pretend it is frozen until $t = T$ and that $t = T$ is the end of the horizon
3. Solve this *easier* lower-level finite horizon problem.
4. Execute this T-period lower-level policy $(\tilde{\pi}_1, \tilde{\pi}_2, \ldots, \tilde{\pi}_{T-1})$ in the real system
5. Repeat

Computation

- **Pre-compute** finite-horizon lower-level policy with frozen slow states
- Re-use pre-computed lower-level policy to solve infinite-horizon upper-level problem, which **takes advantage of** γ^T

Frozen-state, lower-level problem





Frozen-state lower-level MDP

$$J_1^\star(x,y) = \max_{\tilde{\pi}} \mathbb{E}\left[\sum_{t=1}^{T-1} \gamma^{t-1} r(x_1, y_t, \tilde{\pi}_t) \,\middle|\, (x_1, y_1) = (x, y)\right]$$

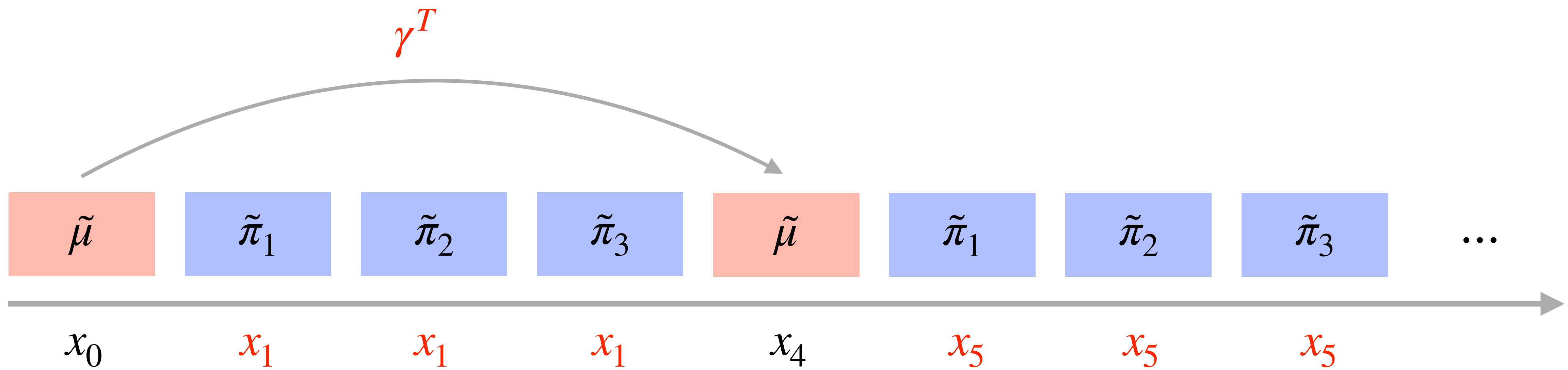
$$J_t^\star(x,y) = \max_a r(x,y,a) + \gamma\,\mathbb{E}\left[J_{t+1}^\star(x,y')\right], \quad J_T^\star \equiv 0$$

$$\tilde{\pi}_t^\star(x,y) = \mathrm{argmax}_a\, r(x,y,a) + \gamma\,\mathbb{E}\left[J_{t+1}^\star(x,y')\right].$$

Computational benefits

- Small number of successor states (since slow state is frozen)
  - $\mathcal{O}(S^2A) \to \mathcal{O}(XY^2A)$
- Independent across x
- Independent from upper-level problem (replaced $U^\star$ by 0)

Frozen-state, upper-level problem



Frozen-state upper-level MDP

Let $(\tilde{\boldsymbol{\pi}}^\star, J_1^\star)$ be the optimal policy/value of the lower-level problem.

$$\tilde{R}(s_0, a, J_1^\star) = r(s_0, a) + \gamma\, J_1^\star\big(f(s_0, a, w)\big)$$

$$V^\star(s_0, J_1^\star, \tilde{\boldsymbol{\pi}}^\star) = \max_a \ \mathbb{E}\Big[\tilde{R}(s_0, a, J_1^\star) + \gamma^T V^\star(s_T, J_1^\star, \tilde{\boldsymbol{\pi}}^\star)\Big]$$ [transitions according to $\tilde{\boldsymbol{\pi}}^\star$]

After solving both levels, let $(\tilde{\mu}^\star, \tilde{\pi}^\star)$ be the solution of the frozen-state approximation.

In the exact reformulation, we were maximizing over policies, now it is just a single action.

Per-cycle reward approximation error

Proposition. The difference between true and approximate T-horizon rewards:

$$\Big|\underbrace{\mathbb{E}[R(s_0, a, \boldsymbol{\pi}^\star)]}_{\text{True}} - \underbrace{\mathbb{E}[\tilde{R}(s_0, a, J_1^\star)]}_{\text{Frozen}}\Big|$$

$$\leq \underbrace{\alpha d_{\mathcal{Y}} \left(L_r \sum_{i=1}^{T-2} \gamma^i \sum_{j=0}^{i-1} L_f^j \right)}_{\text{error from freezing}} + \underbrace{\gamma^{T-1} L_U \left[\alpha d_{\mathcal{Y}} \sum_{j=0}^{T-2} L_f^j + \gamma d_{\mathcal{Y}} (\alpha + 2)(T-1) \right]}_{\text{end of horizon error}}$$

Main ideas.

1. $\mathbb{E}\big[R(x_0, y_0, a, \boldsymbol{\pi}^*)\big] = \mathbb{E}\Big[r(x_0, y_0, a) + \gamma \left(H^{T-1} U^\star\right)(x_1, y_1) - \gamma^T U^\star(x_T, y_T)\Big]$

   where $(HU)(x, y) = \max_a \ r(x, y, a) + \gamma \, \mathbb{E}\big[f(s, a, w))\big]$ [true Bellman operator]

2. $\mathbb{E}\big[\tilde{R}(x_0, y_0, a, J_1^\star)\big] = r(x_0, y_0, a) + \gamma\big(\tilde{H}^{T-1} \mathbf{0}\big)(x_1, y_1)$

   where $\left(\tilde{H} J_{t+1}\right)(x, y) = \max_a \ r(x, y, a) + \gamma \, \mathbb{E}\big[J_{t+1}(x, f_{\mathcal{Y}}(x, y, a, w))\big]$ [frozen Bellman operator]

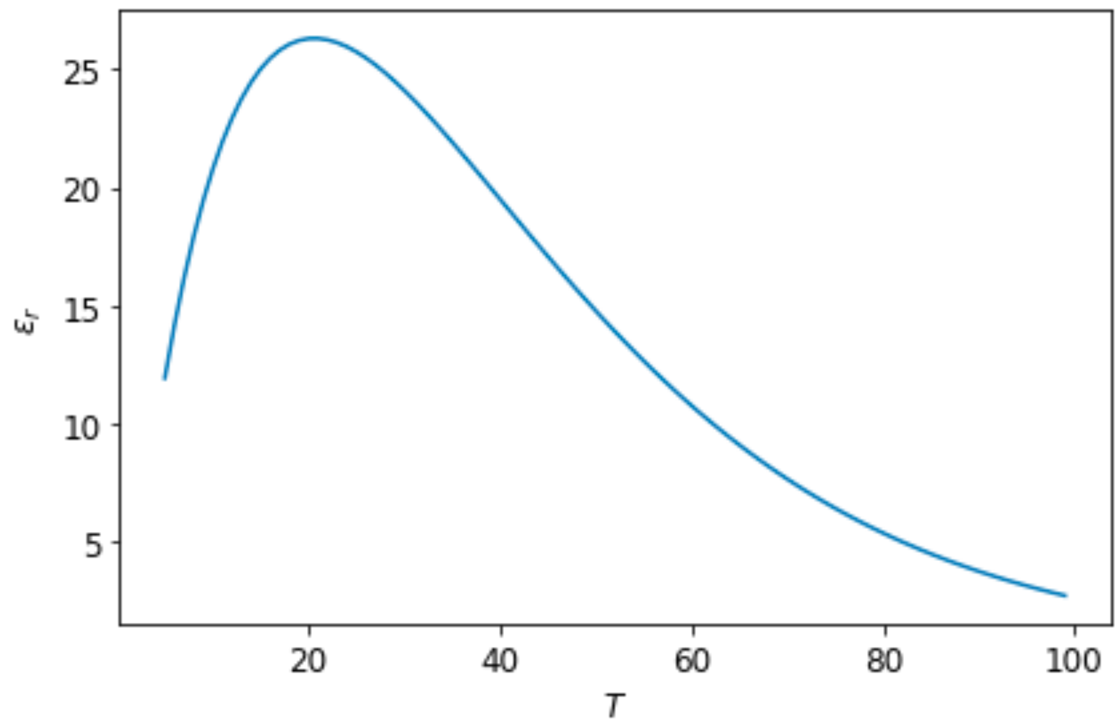
Per-cycle reward approximation error

Proposition. The difference between true and approximate T-horizon rewards:

$$\Big| \underbrace{\mathbb{E}[R(s_0, a, \boldsymbol{\pi}^\star)]}_{\text{True}} - \underbrace{\mathbb{E}[\tilde{R}(s_0, a, J_1^\star)]}_{\text{Frozen}} \Big|$$

$$\leq \underbrace{\alpha d_{\mathcal{Y}} \left(L_r \sum_{i=1}^{T-2} \gamma^i \sum_{j=0}^{i-1} L_f^j \right)}_{\text{error from freezing}} + \underbrace{\gamma^{T-1} L_U \left[\alpha d_{\mathcal{Y}} \sum_{j=0}^{T-2} L_f^j + \gamma d_{\mathcal{Y}} (\alpha + 2)(T-1) \right]}_{\text{end of horizon error}}$$

Initial increase due to error from freezing states



Eventual decrease due terminal value error being discounted more and more

5. Frozen-state value iteration

Standard value iteration on the base model

Recall: Given an MDP and Bellman operator H, where $(HU)(s) = \max_a r(s,a) + \gamma \mathbb{E} U(f(s,a,w))$, the *value iteration* algorithm is $U^k = H^k U^0$

- Convergence to optimal value function: $\lim_{t\to\infty} H^t U = U^\star$ for any initial estimate V
- $\|U^{\nu_k} - U^*\|_\infty \leq \dfrac{2 r_{\max} \gamma^{k+1}}{(1-\gamma)^2}$, where $\nu^k(s) = \text{argmax}_a \; r(s,a) + \gamma \, \mathbb{E}\big[U^k(f(s,a,w))\big]$

Depends on

- $\|U^k - U^\star\|_\infty \leq \gamma^k \|U^0 - U^\star\|_\infty$
- $\|U^0 - U^\star\|_\infty \leq \dfrac{r_{\max}}{1-\gamma}$
- $\|U^{\nu^k} - U^\star\|_\infty \leq \dfrac{2\|U^k - U^\star\|_\infty}{1-\gamma}$

Algorithm 1: Exact VI for the Base Model

Input: Initial values U_0, number of iterations k.

Output: Approximation to the optimal policy ν^k.

```
for i = 1, 2, ..., k do
    for s in the state space S do
        U^i(s) = max_a r(s,a) + γ E[U^{i-1}(f(s,a,w))].
    end
end
for s in the state space S do
    ν^k(s) = arg max_a r(s,a) + γ E[U^k(f(s,a,w))].
end
```

Frozen-state value iteration (FSVI)

Algorithm 2: Frozen-State Value Iteration (FSVI)

Input: Initial values $J_T^* \equiv 0$ and V^0, number of iterations k.

Output: Approximation of the T-periodic frozen-state policy $(\tilde{\mu}^k, \tilde{\boldsymbol{\pi}}^*)$ and J_1^*.

for $t = T-1, T-2, \ldots, 1$ **do**
  for *each slow state* $x \in \mathcal{X}$ **do**
    for *each fast state* $y \in \mathcal{Y}$ **do**
      $J_t^*(x, y) = \max_a r(x, y, a) + \gamma \mathbb{E}\big[J_{t+1}^*(x, f_{\mathcal{Y}}(x, y, a, w))\big]$.
      $\tilde{\pi}_t^*(x, y) = \arg\max_a r(x, y, a) + \gamma \mathbb{E}\big[J_{t+1}^*(x, f_{\mathcal{Y}}(x, y, a, w))\big]$.
    end
  end
end

Note: Freezing the state only happens "within" the algorithm to more efficiently compute $J_1^\star$

for $i = 1, 2, \ldots, k$ **do**
  for $s_0 = (x_0, y_0)$ *in the state space* $\mathcal{X} \times \mathcal{Y}$ **do**
    $V^i(x_0, y_0, J_1^*, \tilde{\boldsymbol{\pi}}^*) = \max_a \mathbb{E}\big[\tilde{R}(s_0, a, J_1^*) + \gamma^T V^{i-1}(x_T, y_T, J_1^*, \tilde{\boldsymbol{\pi}}^*)\big]$.
  end
end

for $s_0 = (x_0, y_0)$ *in the state space* $\mathcal{X} \times \mathcal{Y}$ **do**
  $\tilde{\mu}^k(x_0, y_0) = \arg\max_a \mathbb{E}\big[\tilde{R}(s_0, a, J_1^*) + \gamma^T V^k(x_T, y_T, J_1^*, \tilde{\boldsymbol{\pi}}^*)\big]$.
end

Solving the lower level incurs a one time fixed cost

Pre-compute lower-level problem, a finite-horizon DP:

- To solve lower-level DP: $\mathcal{O}(XY^2AT)$
- To compute multi-step transition: $\mathcal{O}(S^2T)$

Upper-level problem (infinite-horizon VI on slow-timescale MDP with γ^T discounting):

- Per upper-level VI iteration: $\mathcal{O}(S^2A)$

$\mathcal{O}(S^2A)$ per iteration is the same as Base VI...but keep in mind that here the discount factor is γ^T instead of γ!

Regret of a periodic policy $(\mu, \boldsymbol{\pi})$

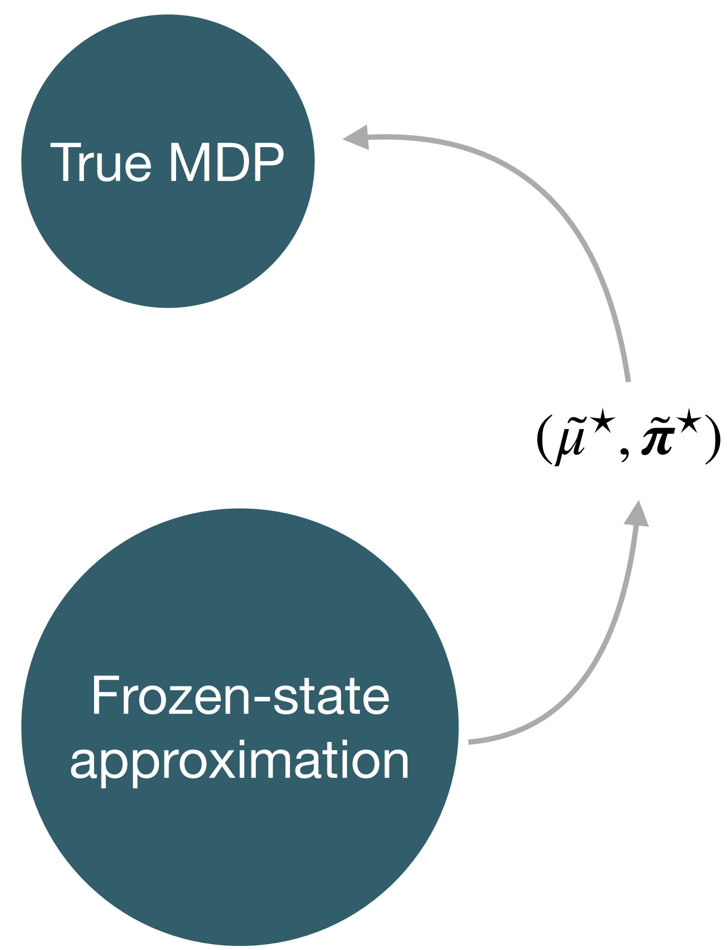
Definition. Suppose the optimal policy is $\nu^\star$. The regret is

$$\mathscr{R}(s, \mu, \boldsymbol{\pi}) = U^{\nu^\star}(s) - \bar{U}^{\mu}(s, \boldsymbol{\pi}) = \bar{U}^{\star}(s) - \bar{U}^{\mu}(s, \boldsymbol{\pi}) \quad \text{and} \quad \mathscr{R}(\mu, \boldsymbol{\pi}) = \max_{s} \mathscr{R}(s, \mu, \boldsymbol{\pi}),$$

where we have used the equivalence between the base and hierarchical formulations.

Remarks:

- We always measure regret with respect to the *true* MDP.
  - Although $(\mu, \boldsymbol{\pi})$ is computed *with the help of frozen states*, it is evaluated in the original MDP with true dynamics.
- Consider $\mathscr{R}(\tilde{\mu}^\star, \tilde{\boldsymbol{\pi}}^\star)$, notice that $V^\star(J_1^\star, \tilde{\boldsymbol{\pi}}^\star)$ does not directly enter the regret definition.
  - It is the optimal value of the approximation, but doesn't reflect the performance of $(\tilde{\mu}^\star, \tilde{\boldsymbol{\pi}}^\star)$ in the true model.



Main idea behind regret analysis

Lemma (Approximation to FSVI).

- Suppose we *approximately* solve the lower-level problem and obtain $\boldsymbol{\pi}, J_1$, instead of the optimal solutions $\boldsymbol{\pi}^\star, U^\star$.
- Suppose we approximately solve the upper-level problem and obtain V instead of $V^\star(J_1, \boldsymbol{\pi})$, as we expected.
- Let μ be greedy with respect to both J_1 and V:
  - $\mu(s_0) = \operatorname{argmax}_{a \in \mathscr{A}} \mathbb{E}\left[\tilde{R}(s_0, a, J_1) + \gamma^T V(s_T(a, \boldsymbol{\pi}))\right].$
- Then,

$$\mathscr{R}(\mu, \boldsymbol{\pi}) \leq \left(\frac{2\gamma^T}{(1-\gamma^T)^2} + \frac{2}{1-\gamma^T}\right)\epsilon_r(\boldsymbol{\pi}^\star, J_1) + \left(\frac{2\gamma^{2T}}{(1-\gamma^T)^2} + \frac{2\gamma^T}{1-\gamma^T}\right) L_U\, d(\alpha, d_{\mathscr{Y}}, T) + \frac{2\gamma^T}{1-\gamma^T}\|V^\star(J_1, \boldsymbol{\pi}) - V\|_\infty.$$

Reward error

End of horizon error

V-approximation error

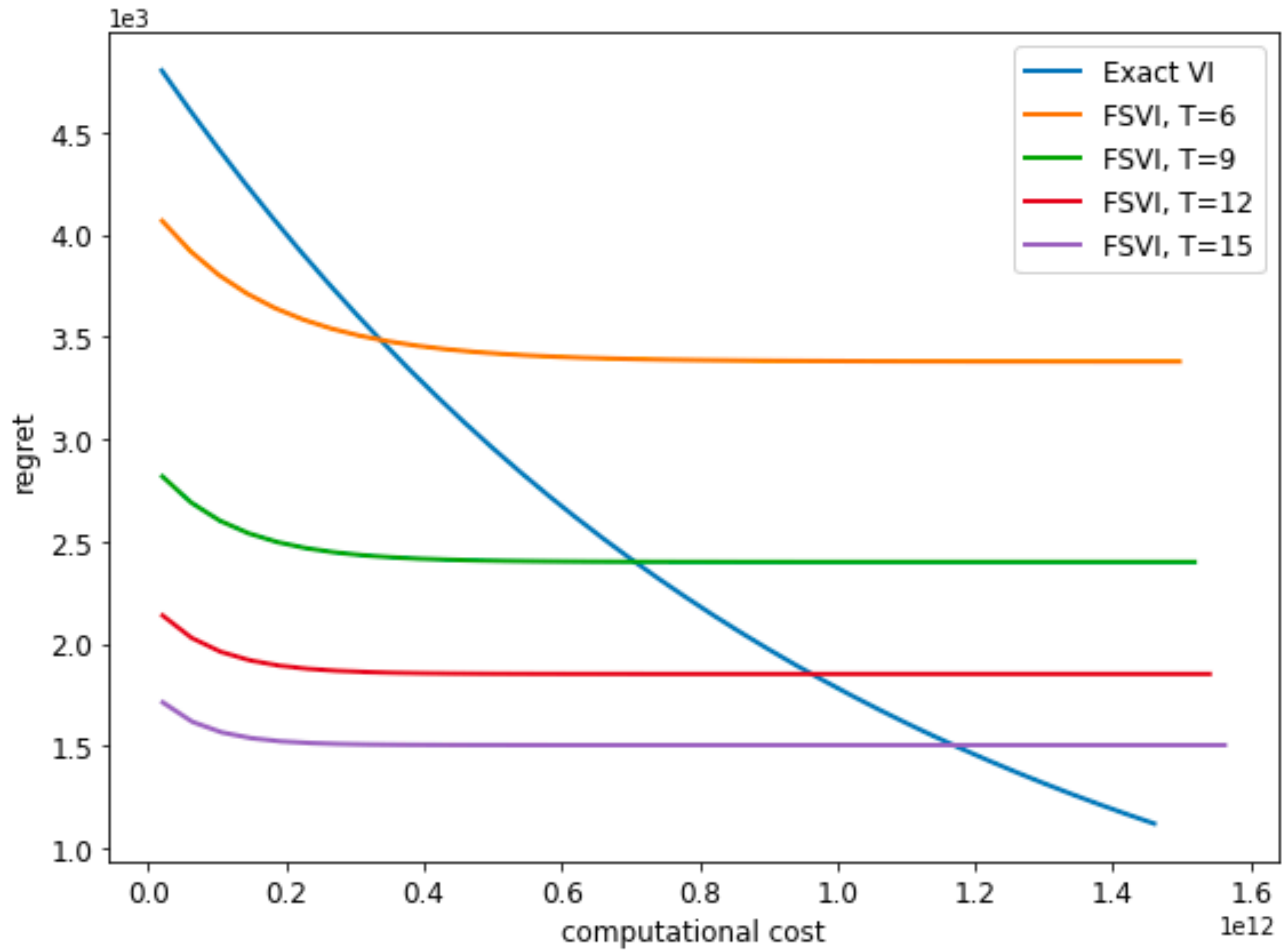
Regret of FSVI

Theorem. The regret of FSVI after k upper-level iterations is:

$$\mathcal{R}(\mu, \boldsymbol{\pi}) \leq \left(\frac{2\gamma^T}{(1-\gamma^T)^2} + \frac{2}{1-\gamma^T}\right)\epsilon_r(\boldsymbol{\pi}^\star, J_1) + \left(\frac{2\gamma^{2T}}{(1-\gamma^T)^2} + \frac{2\gamma^T}{1-\gamma^T}\right)L_U\, d(\alpha, d_{\mathcal{Y}}, T) + \frac{2r_{\max}\gamma^{(k+1)T}}{(1-\gamma)(1-\gamma^T)},$$

which replaces the V-approximation error term with the VI error.

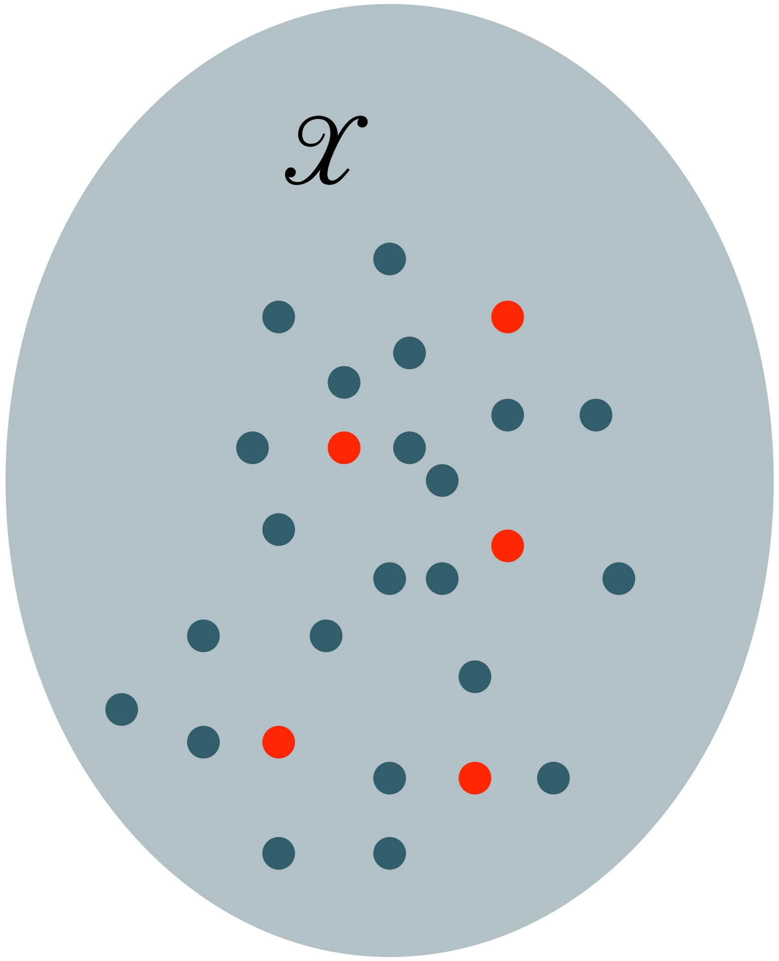
Comparison of FSVI versus Base VI sub-optimality



6. *Nominal-state approximation for the lower level*

Nominal state version of FSVI for nearly factored MDPs

- In FSVI, one still has to solve the lower-level MDP for each x.
- What if we solve it for a few slow states only?
  - $\mathcal{O}(S^2A) \to \mathcal{O}(XY^2A) \to \mathcal{O}(X_{\text{nom}}Y^2A)$
- Nominal FSVI:
  - Reward function *nearly* factored:
    - $g(x) + h(y,a) - r(x,y,a) \quad \leq \zeta$
  - Solve lower level for a few nominal states:
    - $J_{t,\text{nom}}(x^\star, y) = \max_a \; g(x^\star) + h(y,a) + \gamma\, \mathbb{E}\big[J_{t+1,\text{nom}}(x^\star, y')\big]$
  - Extrapolate to nearby states:
    - $J_{t,\text{nom}}(x,y) = \sum_{i=0}^{T-t-1} \gamma^i \big(g(x) - g(x^\star)\big) + J_{t,\text{nom}}(x^\star, y)\,.$
  - Theoretical analysis requires analyzing the new reward error:
    - $\Big|\mathbb{E}\big[\tilde{R}(s_0, a, J_1^\star)\big] - \mathbb{E}\big[\tilde{R}(s_0, a, J_{1,\text{nom}}\big]\Big|$



7. *Feature-based approximate value iteration*

Scaling to larger state spaces using feature-based approximate value iteration

Architecture:

- Consider M pre-selected states $\tilde{\mathcal{S}} = \{s_1, s_2, \ldots, s_M\}$.
- Consider an M-dimensional feature vector $\boldsymbol{\phi}(s)$, where $\boldsymbol{\phi}(s_m)$ are linearly independent.
- Assume there exists $\gamma' \in [\gamma, 1)$ s.t. for any s, there exists $\theta_m(s)$, where
  - $\sum_m \theta_m(s) \leq 1$ and $\boldsymbol{\phi}(s) = \frac{\gamma'}{\gamma} \sum_{m=1}^{M} \theta_m(s)\, \boldsymbol{\phi}(s_m)$.
- Lower level: $\hat{J}(s, \boldsymbol{\omega}_t) = \boldsymbol{\phi}^\intercal(s)\boldsymbol{\omega}_t$.
- Upper level: $\hat{V}(s, \boldsymbol{\beta}^k) = \boldsymbol{\phi}^\intercal(s)\, \boldsymbol{\beta}^k$.
- Update procedure:
  1. Compute Bellman update at pre-selected states only: $y(s_m)$.
  2. Compute next parameter vector ($\boldsymbol{\omega}_{t-1}$ or $\boldsymbol{\beta}^{k+1}$) such that the updated value function evaluated at the pre-selected states is equal: e.g., $\hat{J}(s_m, \boldsymbol{\omega}_{\boldsymbol{t-1}}) = y(s_m)$.

Limited "expansion" after going to parameter space and back:

$$\|(\Phi\Phi^\dagger)(J) - (\Phi\Phi^\dagger)(J')\|_\infty \leq \kappa\, \|J - J'\|_\infty \quad (\kappa = \gamma'/\gamma)$$

J. N. Tsitsiklis and B. Van Roy. Feature-based methods for large scale dynamic programming. *Machine Learning*, 1996.

Scaling to larger state spaces using feature-based approximate value iteration

Algorithm 4: Frozen-State Approximate Value Iteration (FSAVI)

Input: $\tilde{\mathcal{S}} = \{s_1, s_2, \ldots, s_M\}$, $\boldsymbol{\phi}$, initial weights $\boldsymbol{\omega}_T = \boldsymbol{\beta}_0 = \mathbf{0}$, number of iterations k.

Output: Approximation of the T-periodic frozen-state policy $\left(\hat{\mu}_{(\boldsymbol{\beta}^k, \boldsymbol{\omega}^*)}, \hat{\boldsymbol{\pi}}_{\boldsymbol{\omega}^*}\right)$ and $\hat{J}_1(\boldsymbol{\omega}^*)$

for $t = T-1, T-2, \ldots, 1$ **do**
  for *each pre-selected state* $s = (x, y) \in \tilde{\mathcal{S}}$ **do**
    $J_t(x, y) = \max_a r(x, y, a) + \gamma\,\mathbb{E}\big[\hat{J}_{t+1}(x, f_{\mathcal{Y}}(x, y, a, w), \boldsymbol{\omega}_{t+1})\big]$.
  end
  Set remaining entries of J_t to zero. Update parameter vector: $\boldsymbol{\omega}_t^* = \Phi^\dagger J_t$.
end
Let $\hat{\boldsymbol{\pi}}_{\boldsymbol{\omega}^*}$ be greedy with respect to $\hat{J}_t(\boldsymbol{\omega}_t^*) = \Phi\boldsymbol{\omega}_t^*$, similar to (23).
for $i = 1, 2, \ldots, k$ **do**
  for *each pre-selected state* $s_0 \in \tilde{\mathcal{S}}$ **do**
    $V^i(s_0) = \max_a \mathbb{E}\big[\tilde{R}(s, a, \hat{J}_1(\boldsymbol{\omega}_1^*)) + \gamma^T\,\hat{V}(s_T(a, \tilde{\boldsymbol{\pi}}_{\text{avi}}), \boldsymbol{\beta}_{i-1})\big]$.
    Set remaining entries of V^i to zero. Update parameter vector: $\boldsymbol{\beta}_i = \Phi^\dagger V^i$.
  end
end
for s_0 *in the state space* $\mathcal{S}$ **do**
  $\hat{\mu}_{(\boldsymbol{\beta}^k, \boldsymbol{\omega}^*)}(s_0) = \arg\max_a \mathbb{E}\big[\tilde{R}(s_0, a, \hat{J}_1(\boldsymbol{\omega}_1^*)) + \gamma^T\hat{V}(s_T(a, \tilde{\boldsymbol{\pi}}_{\boldsymbol{\omega}^*}), \boldsymbol{\beta}_k)\big]$.
end

Regret of FSAVI

Theorem. The regret of FSAVI after k upper-level iterations is:

$$\mathscr{R}(\mu, \boldsymbol{\pi}) \leq \left(\frac{2\gamma^T}{(1-\gamma^T)^2} + \frac{2}{1-\gamma^T}\right)\epsilon_r(\boldsymbol{\pi}^\star, \hat{J}_1(\boldsymbol{\omega}_1^\star))$$

$$+\left(\frac{2\gamma^{2T}}{(1-\gamma^T)^2} + \frac{2\gamma^T}{1-\gamma^T}\right)L_U\, d(\alpha, d_{\mathscr{Y}}, T) + \underbrace{\left(\frac{1+\kappa}{1-\kappa\gamma^T}\right)\varepsilon_{\text{up}}}_{\|V^\star_{\omega^\star} - \hat{V}(\beta^\star)\|_\infty} + \underbrace{(\kappa\gamma^T)^k\left(\frac{\kappa^2 - \kappa^2(\kappa\gamma)^{T+1}}{(1-\kappa\gamma^T)(1-\kappa\gamma)}\right)r_{\max}}_{\|\hat{V}(\beta^*) - \hat{V}(\beta_k)\|_\infty},$$

which $\epsilon_r(\boldsymbol{\pi}^\star, \hat{J}_1(\boldsymbol{\omega}_1^\star)) = \epsilon_r(\boldsymbol{\pi}^\star, J_1^\star) + \underbrace{\left(\frac{1+\kappa}{1-\kappa\gamma} - \frac{(\kappa\gamma)^T(1+\gamma)}{\gamma - \kappa\gamma^2}\right)\varepsilon_{\text{low}}}_{\|J_t^\star - \hat{J}_t(\omega_t^\star)\|_\infty}$.

Upper-level feature approximation error

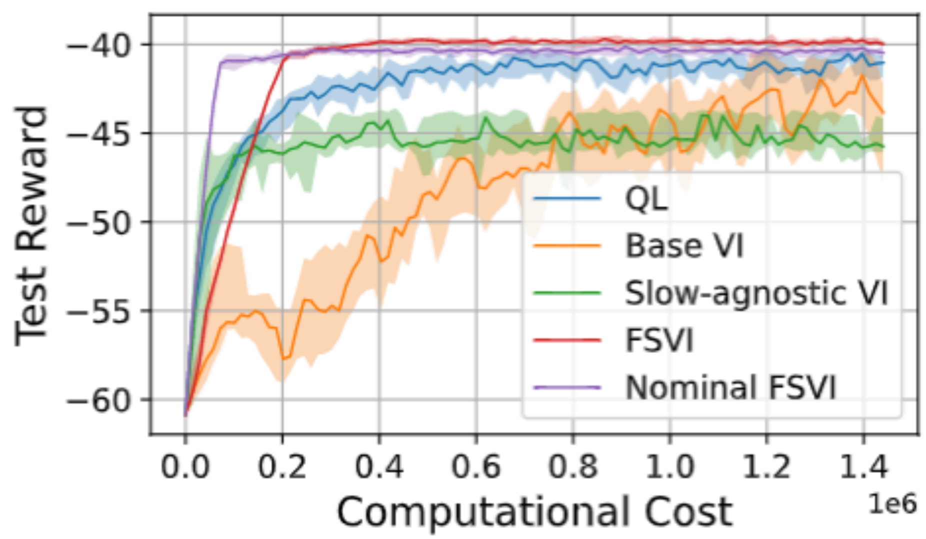
Lower-level feature approximation error

8. *Numerical results*

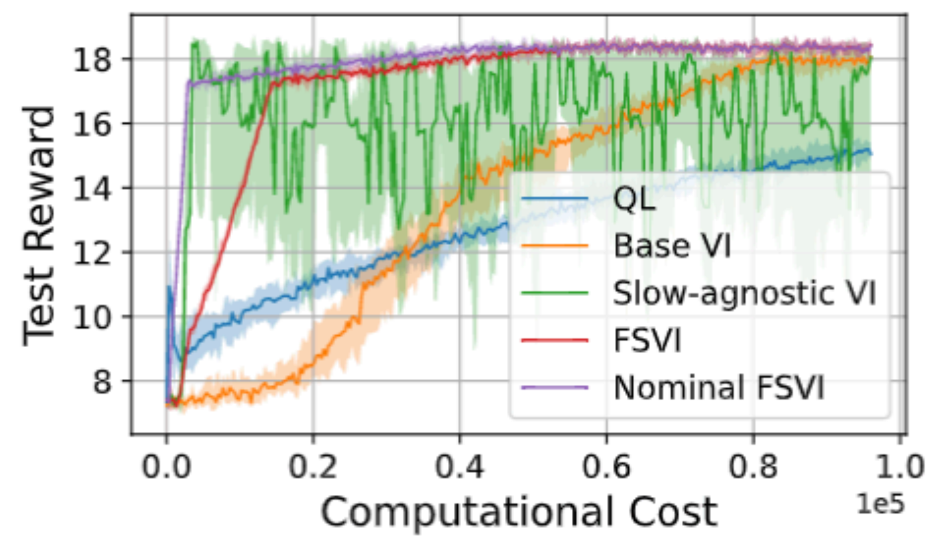
Baseline algorithms

- Base model + VI / AVI
- Slow-agnostic VI / AVI
- Q-learning (QL)
- Deep Q-networks (DQN)
- **Ours: FSVI / Nominal FSVI**
- **Ours: FSAVI / Nominal FSAVI**

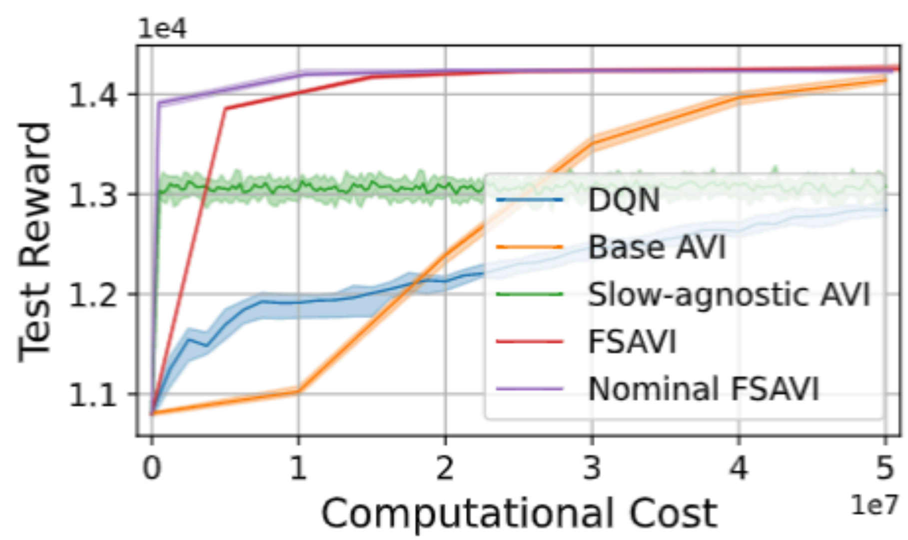
Overall performance comparison



(a) Multi-class service allocation



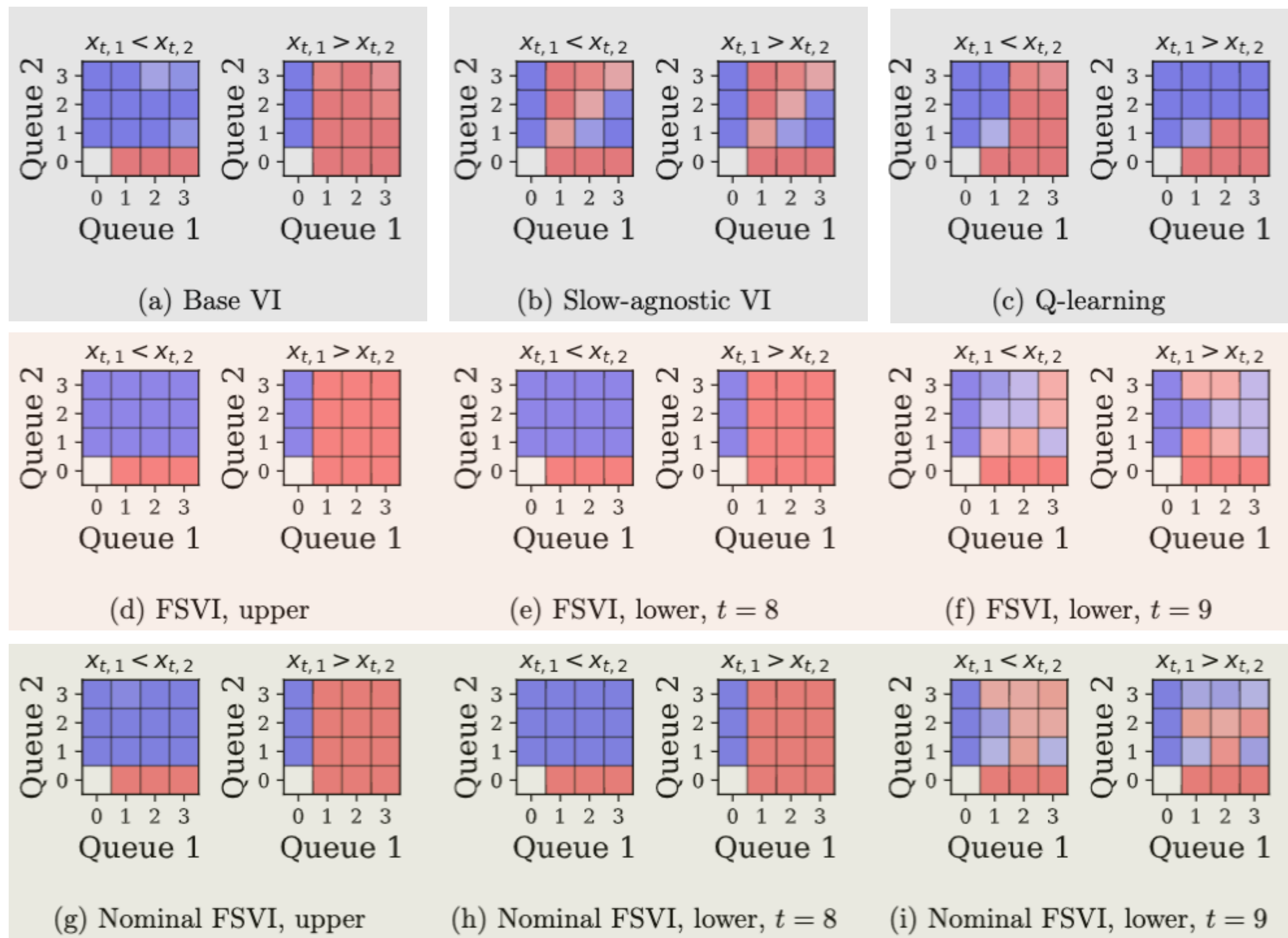
(b) Restless two-armed bandit



(c) Energy demand response

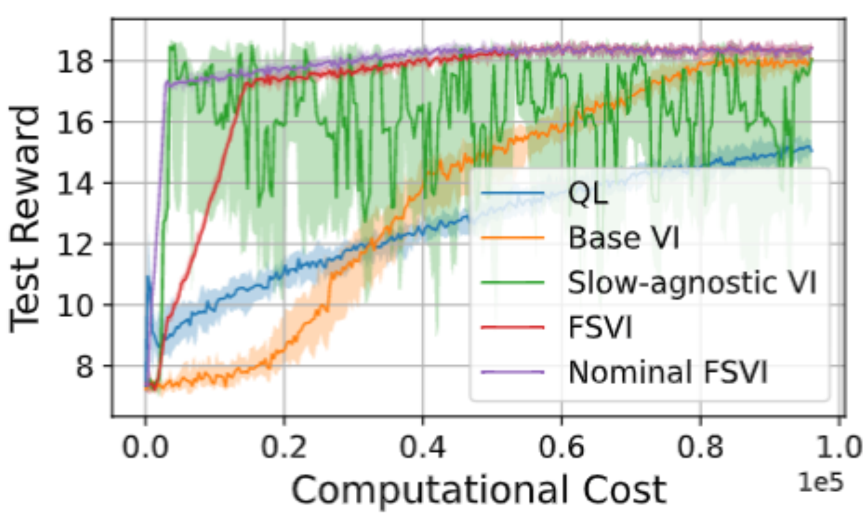
Service allocation in multi-class queues

- 2 queues, 1 server
- Stochastic holding cost (linear in queue length)
- Actions: serve 1 or serve 2
- Slow state: holding cost
- Fast state: queue lengths

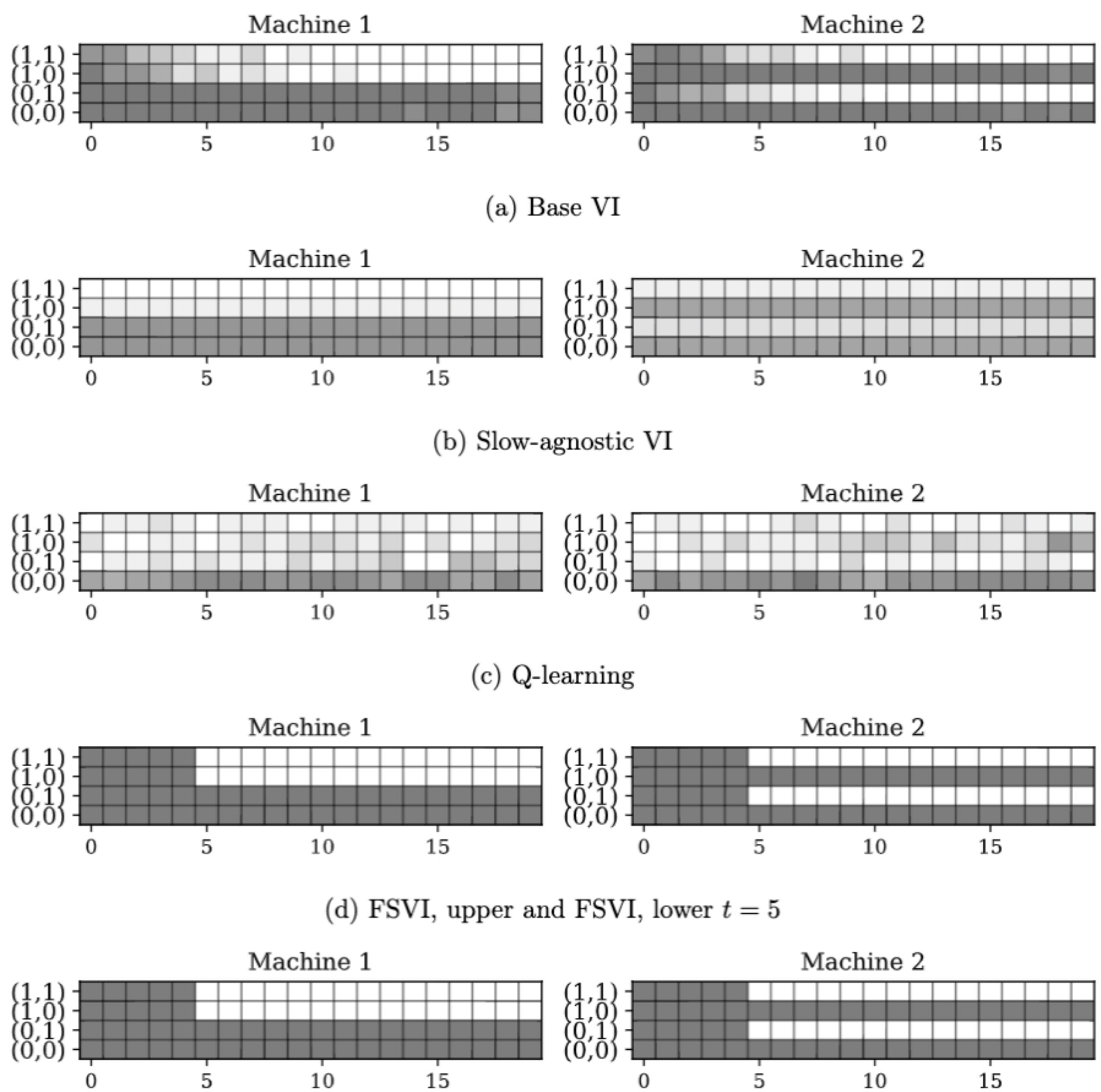


Restless bandits for machine maintenance

- 2 machines, either operating or not ($y_{t,i} \in \{0,1\}$)
- Actions: maintain or not maintain ($a_{t,i} \in \{0,1\}$)
- State of machine i influenced by current state, whether it is maintained, and overall condition of the system x_t
- Slow state: system condition
- Fast state: operating status of each machine



(b) Restless two-armed bandit



(a) Base VI

(b) Slow-agnostic VI

(c) Q-learning

(d) FSVI, upper and FSVI, lower $t = 5$

(e) Nominal FSVI, upper and Nominal FSVI, lower $t = 5$

Energy demand response (AVI)

- Energy aggregator bids a quantity a_t
- Also, sets a compensation $\alpha_t = (\alpha_{t,1}, \alpha_{t,2})$ for each of 2 large customers
- Slow state: day-ahead price x_t
- Fast state: real-time price y_t^-, y_t^+

Payment from forward contract (day-ahead price)

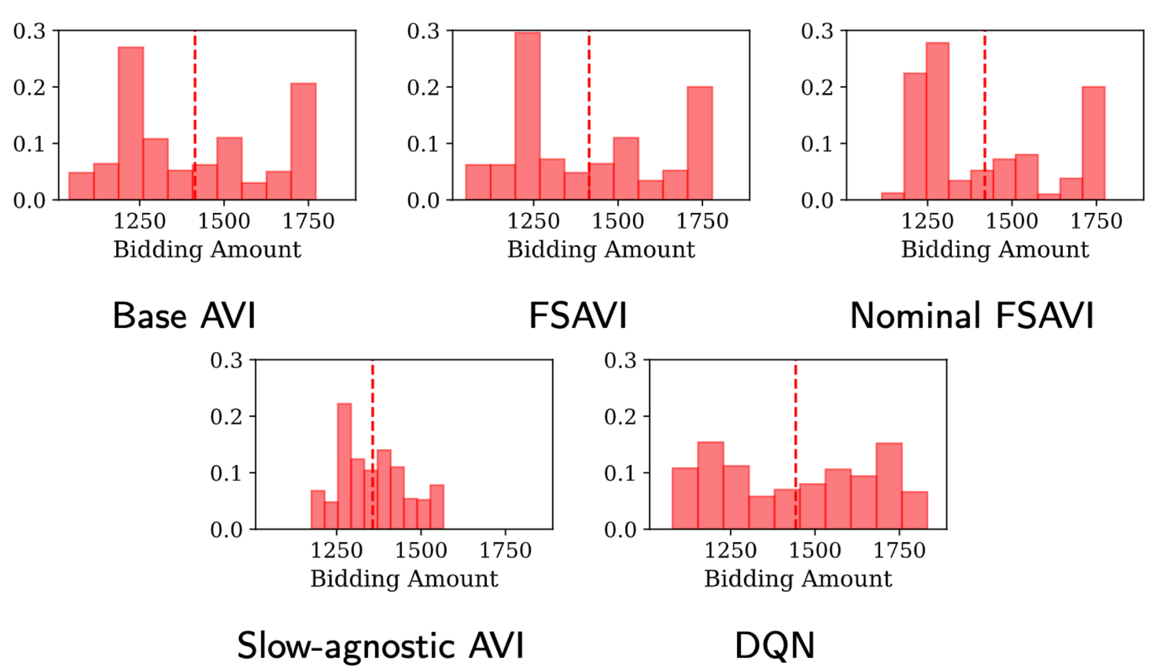
Compensation paid to customers

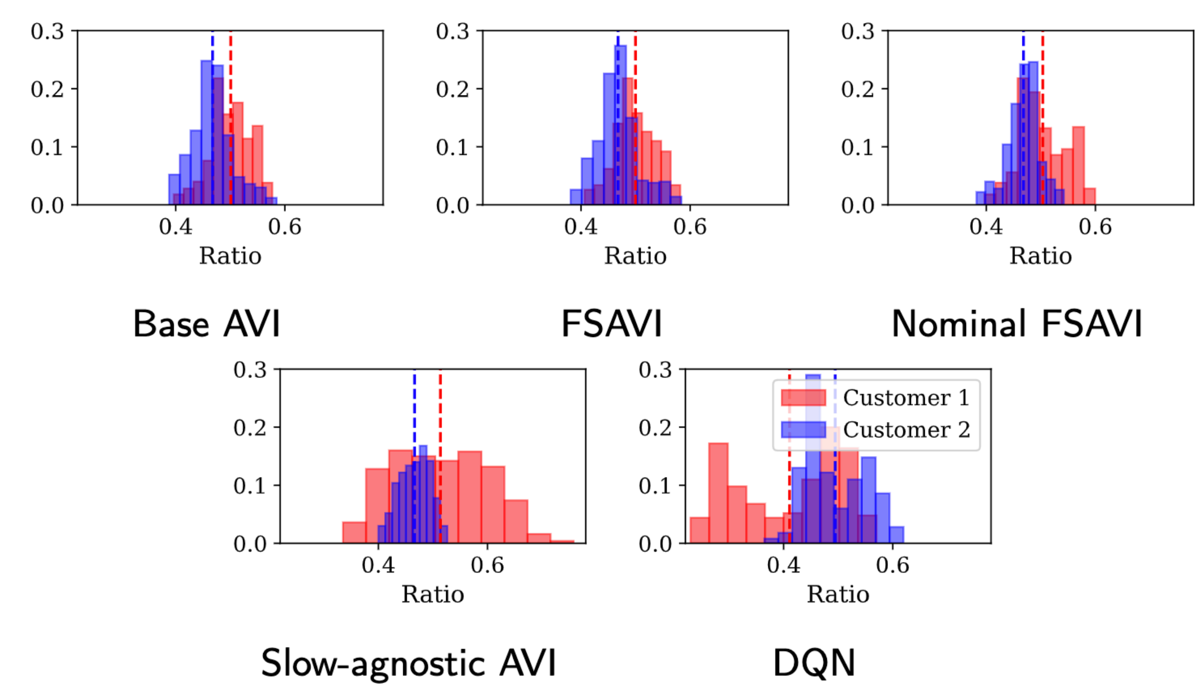
$$r(x_t, y_t^+, y_t^-, a_t, \alpha_t) = x_t a_t - \sum_{m=1}^{2} q_{t,m}\, \mathbb{E}\left[d_m(x_t, \alpha_{t,m})\right]$$

$$+\mathbb{E}\left[x_t y_t^+ \left(\sum_{m=1}^{2} d_m(x_t, \alpha_{t,m}) - a_t\right)^+ - x_t y_t^- \left(a_t - \sum_{m=1}^{2} d_m(x_t, \alpha_{t,m})\right)^+\right].$$

Overage penalty

Shortage penalty





Conclusion

Thank you!

Please feel free to email me at drjiang@pitt.edu for additional comments and discussion.